OTOBO Administration Manual

Release 11.0

Rother OSS GmbH

Sep 20, 2024

Contents

Contents

Administration Area

Any system requires configuration. Configuring a system should be an easy task and the tools for configuration fit-for-purpose.

OTOBO offers several administration tools to configure, monitor, control and extend OTOBO.

1.1 Ticket Settings

A ticket is similar to a medical report created for a hospital patient. When a patient first visits the hospital, a medical report is created to hold all necessary personal and medical information on him. Over multiple visits, as he is attended to by the same or additional doctors, the attending doctor updates the report by adding new information on the patient's health and the ongoing treatment. This allows any other doctors or the nursing staff to get a complete picture on the case at hand. When the patient recovers and leaves the hospital, all information from the medical report is archived and the report is closed.

Ticket systems such as OTOBO handle tickets like normal emails. The messages are saved in the system. When a customer sends a request, a new ticket is generated by the system which is comparable to a new medical report being created. The response to this new ticket is comparable to a doctor's entry in the medical report. A ticket is closed if an answer is sent back to the customer, or if the ticket is separately closed by the system. If a customer responds again on an already closed ticket, the ticket is reopened with the new information added.

Every ticket is stored and archived with complete information. Since tickets are handled like normal emails, attachments and contextual annotations will also be stored with each email. In addition, information on relevant dates, employees involved, working time needed for ticket resolution, etc. are also saved. At any later stage, tickets can be sorted, and it is possible to search through and analyze all information using different filtering mechanisms.

1.1.1 Attachments

For any size of organization it is often required to send a service agreement, the terms of service or a privacy statement out when a customer signs a contract.

Fig. 1.1: Ticket Zoom Screen

OTOBO can handle an infinite number of attachments (PDF, image, etc.) and can bundle them into templates. Your agents don't need to maintain the attachments on their own, nor don't they need to upload them again and again - they can just use the predefined templates.

Use this screen to add attachments for use in templates. A fresh OTOBO installation doesn't contain any attachments by default. The attachment management screen is available in the Attachments module of the Ticket Settings group.
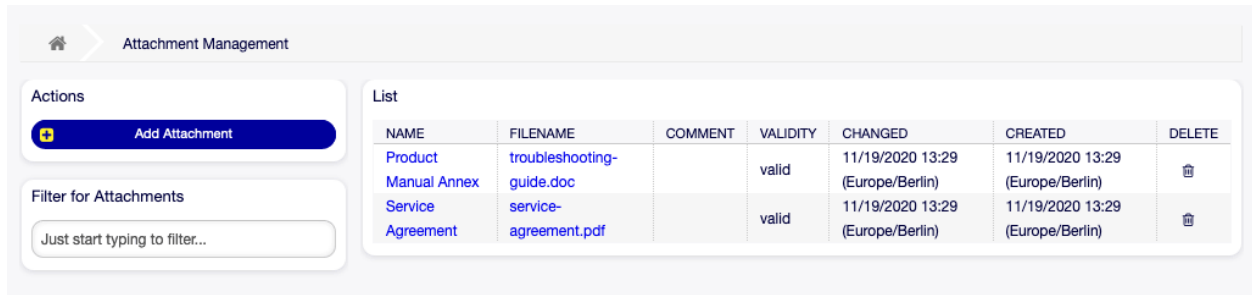
Fig. 1.2: Attachment Management Screen

Manage Attachments

To add an attachment:

1. Click on the Add Attachment button in the left sidebar.
2. Fill in the required fields.
3. Click on the Save button.
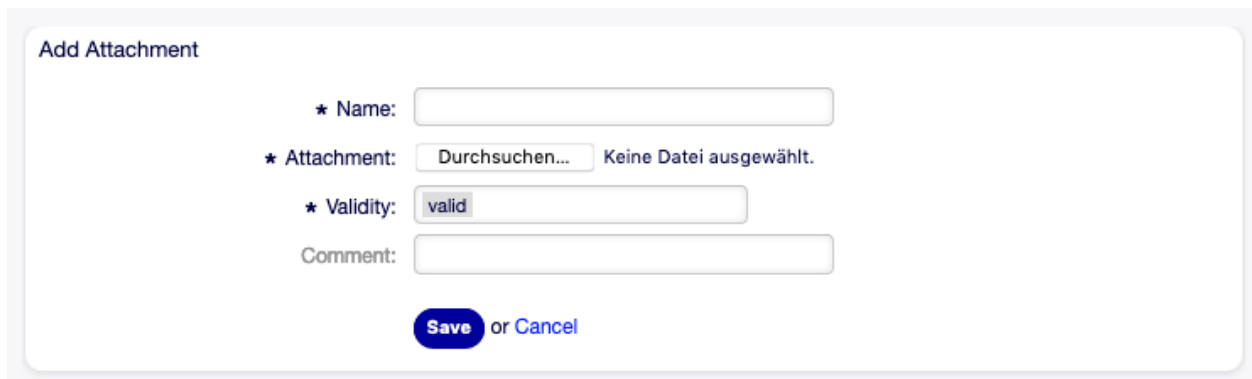
Fig. 1.3: Add Attachment Screen

To edit an attachment:

1. Click on an attachment in the list of attachments.
2. Modify the fields.
3. Click on the Save or Save and finish button.

To delete an attachment:

1. Click on the trash icon in the last column of the overview table.
2. Click on the Confirm button.

Fig. 1.4: Edit Attachment Screen

Fig. 1.5: Delete Attachment Screen

Note: If several attachments are added to the system, use the filter box to find a particular attachment by just typing the name to filter.

Attachment Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Attachment * Open the file dialog to add a file from the file system. This field is mandatory in the attachment add screen, but optional in the attachment edit screen. Adding a new file in the edit screen will overwrite the existing attachment.

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

1.1.2 Auto Responses

Quick and transparent service is vital to maintaining a good working relationship with your customer. Email, fax, social media and other non-real-time communications are patient, but you want to engage your customer immediately upon receipt of a request.

OTOBO allows you to respond to a customer immediately upon receipt of a request giving the customers instantaneous feedback assuring them that their request is in processing, establishing expectation.

Automatic responses can be sent to customers based on the occurrence of certain events, such as the creation of a ticket in a specific queue, the receipt of a follow-up message in regards to a closed or rejected ticket, etc.

Use this screen to add automatic responses for use in queues. A fresh OTOBO installation contains some automatic responses by default. The automatic response management screen is available in the Auto Responses module of the Ticket Settings group.
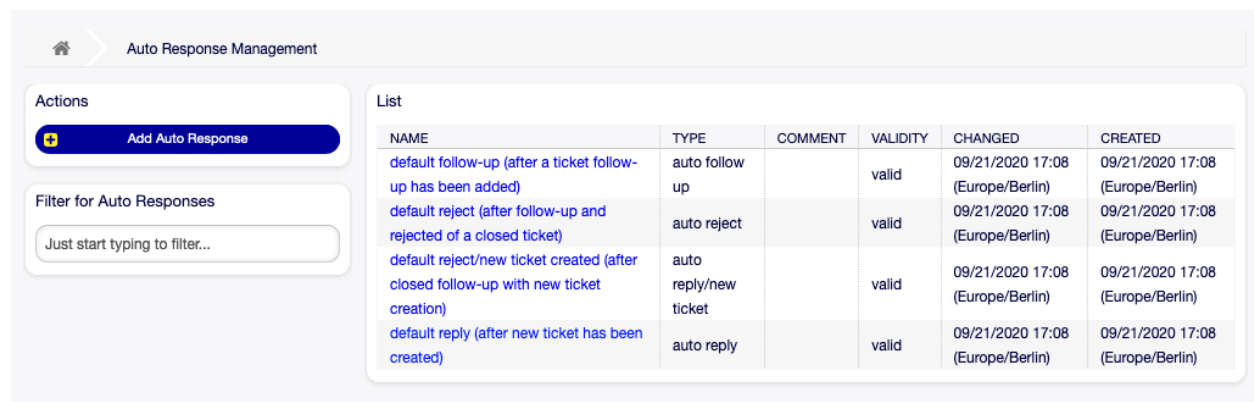
Fig. 1.6: Auto Response Management Screen

Manage Auto Responses

Note: Adding automatic responses requires a valid system address. Create system addresses in the Email Addresses screen.

To add an automatic response:

1. Click on the Add Auto Response button in the left sidebar.
2. Fill in the required fields.
3. Click on the Save button.

Warning: Auto responses can not be deleted from the system. They can only be deactivated by setting the Validity option to invalid or invalid-temporarily.

To edit an automatic response:

1. Click on an automatic response in the list of automatic responses.

Fig. 1.7: Add Auto Response Screen

2. Modify the fields.

3. Click on the Save or Save and finish button.

Fig. 1.8: Edit Auto Response Screen

Note: If several automatic responses are added to the system, use the filter box to find a particular automatic response by just typing the name to filter.

Auto Response Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Subject * The subject of the email sent to the users.

Response The body of the email sent to the users.

Type * The event type that triggers the sending of this automatic response. Only one automatic response can be sent automatically per event. The following event types are available:

> **auto follow up** Is triggered for any follow-up message on an existing ticket. This includes also article notes which are visible for the customer. If the System Configuration option 'AutoResponseForWebTickets' is activated the automatic response is also sent for follow up articles created through the customer web interface or through a telephone call otherwise only for follow up articles created from a customer email. This automatic repsonse does not apply for new tickets, see the 'auto reply' event type.

> **auto reject** If the Queues setting 'follow up option' is set to 'reject' this event will be triggered if a customer answers in a ticket, that is already closed.

> **auto remove** If a ticket is deleted by the system, the customer gets this automatic response.

> **auto reply** If a new ticket is generated the customer will get this auto response. If the System Configuration option 'AutoResponseForWebTickets' is activated the automatic response is also sent for new tickets created through the customer web interface or through a telephone call (telephone ticket) otherwise only for new tickets created from a customer email. This automatic repsonse does not apply if the 'auto reply/new ticket' event type occours.

> **auto reply/new ticket** If the Queues setting 'follow up option' is set to 'new ticket' this event will be triggered if a customer answers in a ticket, that is already closed. You can use this message to inform the customer of the new ticket number.

Auto response from * The sender email address, from which the automatic response will be sent.

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

Auto Response Variables

Using variables in the text makes it possible to personalize messages. Variables, known as OTOBO tags, are replaced by OTOBO when generating the message. Find a list of available tags stems for this resource at the bottom of both add and edit screens.

For example, the variable `<OTOBO_TICKET_TicketNumber>` expands to the ticket number allowing a template to include something like the following.

```
Ticket#<OTOBO_TICKET_TicketNumber>
```

This tag expands, for example to:

```
Ticket#2018101042000012
```

Reference

You can use the following tags:

<OTOBO_CUSTOMER_SUBJECT[20]>
To get the first 20 character of the subject.

<OTOBO_CUSTOMER_EMAIL[5]>
To get the first 5 lines of the email.

<OTOBO_CUSTOMER_REALNAME>
To get the name of the ticket's customer user (if given).

<OTOBO_CUSTOMER_*>
To get the article attribute (e. g. <OTOBO_CUSTOMER_From>, <OTOBO_CUSTOMER_To>, <OTOBO_CUSTOMER_Cc>, <OTOBO_CUSTOMER_Subject>, <OTOBO_CUSTOMER_Body>).

<OTOBO_CUSTOMER_DATA_*>
Options of the current customer user data (e. g. <OTOBO_CUSTOMER_DATA_UserFirstname>).

<OTOBO_OWNER_*>
Ticket owner options (e. g. <OTOBO_OWNER_UserFirstname>).

<OTOBO_RESPONSIBLE_*>
Ticket responsible options (e. g. <OTOBO_RESPONSIBLE_UserFirstname>).

<OTOBO_CURRENT_*>
Options of the current user who requested this action (e. g. <OTOBO_CURRENT_UserFirstname>).

<OTOBO_TICKET_*>
Options of the ticket data (e. g. <OTOBO_TICKET_TicketNumber>, <OTOBO_TICKET_TicketID>, <OTOBO_TICKET_Queue>, <OTOBO_TICKET_State>).

<OTOBO_TICKET_DynamicField_*>
Options of ticket dynamic fields internal key values (e. g. <OTOBO_TICKET_DynamicField_TestField>, <OTOBO_TICKET_DynamicField_TicketFreeText1>).

<OTOBO_TICKET_DynamicField_*_Value>
Options of ticket dynamic fields display values, useful for Dropdown and Multiselect fields (e. g. <OTOBO_TICKET_DynamicField_TestField_Value>, <OTOBO_TICKET_DynamicField_TicketFreeText1_Value>).

<OTOBO_CONFIG_*>
Config options (e. g. <OTOBO_CONFIG_HttpType>).

Example response:

```
Thanks for your email.

You wrote:
<snip>
<OTOBO_CUSTOMER_EMAIL[6]>
```

Fig. 1.9: Auto Response Variables

inction

soiré

Fig. 1.11: Add Priority Screen

Note: It's recommended to limit your system to 5 priorities or less and reuse the current 5 to keep the use of the traffic light system.

To edit a priority:

1. Click on a priority in the list of priorities.
2. Modify the fields.
3. Click on the Save or Save and finish button.

Fig. 1.12: Edit Priority Screen

Note: If several priorities are added to the system, use the filter box to find a particular priority by just typing the name to filter.

If you change the name of a priority which is used in the system configuration, a validation check will warn you and give you the following options:

Save and update automatically Apply the change and also update the affected settings.

Don't save, update manually Apply the change, but don't update the affected settings. The updates need to be done manually.

Cancel Cancel the action.

> **Warning:** Changing the name of this object should be done with care, the check only provides verification for certain settings and ignores things where the name can't be verified. Some examples are dashboard filters, action control lists (ACLs), and processes (sequence flow actions) to name a few. Documentation of your setup is key to surviving a name change.

Fig. 1.13: Priority Validation Dialog

Priority Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

1.1.4 Queues

Teams need a workspace and the ability to dispatch work based on skill level, security level, department or responsibility just to name a few. Other teams may also need to view or react to these requests as well.

OTOBO uses queues to provide your teams with structure. Queues provide a powerful way to divide and disperse the work to the responsible group of people.

Use this screen to add queues to the system. In a fresh OTOBO installation there are 4 default queues: Raw, Junk, Misc and Postmaster. All incoming messages will be stored in the Raw queue if no filter rules are defined. The Junk queue can be used to store spam messages. The queue management screen is available in the Queues module of the Ticket Settings group.

Manage Queues

To add a queue:

1. Click on the Add Queue button in the left sidebar.

2. Fill in the required fields.
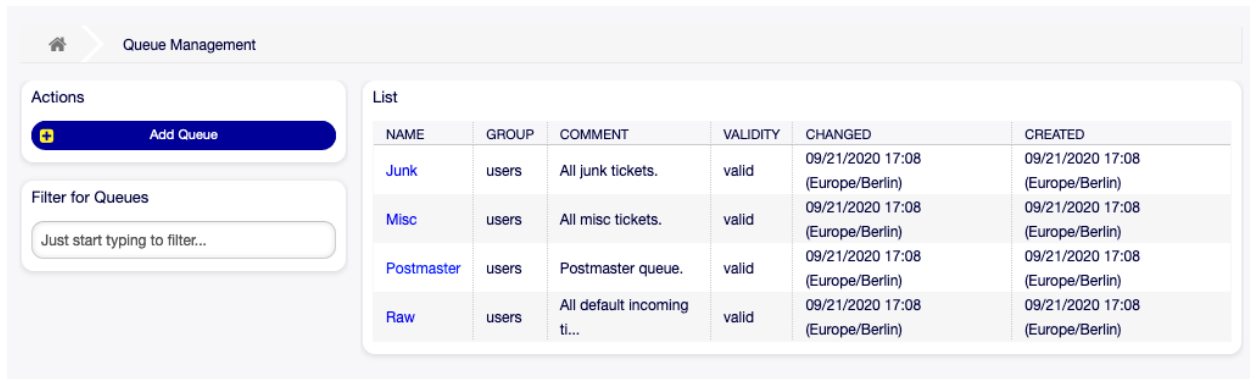
Fig. 1.14: Queue Management Screen

3. Click on the Save button.

Warning: Queues can not be deleted from the system. They can only be deactivated by setting the Validity option to invalid or invalid-temporarily.

To edit a queue:

1. Click on a queue in the list of queues.

2. Modify the fields.

3. Click on the Save or Save and finish button.

Note: If several queues are added to the system, use the filter box to find a particular queue by just typing the name to filter.

If you change the name of a queue which is used in the system configuration, a validation check will warn you and give you the following options:

Save and update automatically Apply the change and also update the affected settings.

Don't save, update manually Apply the change, but don't update the affected settings. The updates need to be done manually.

Cancel Cancel the action.

Warning: Changing the name of this object should be done with care, the check only provides verification for certain settings and ignores things where the name can't be verified. Some examples are dashboard filters, action control lists (ACLs), and processes (sequence flow actions) to name a few. Documentation of your setup is key to surviving a name change.
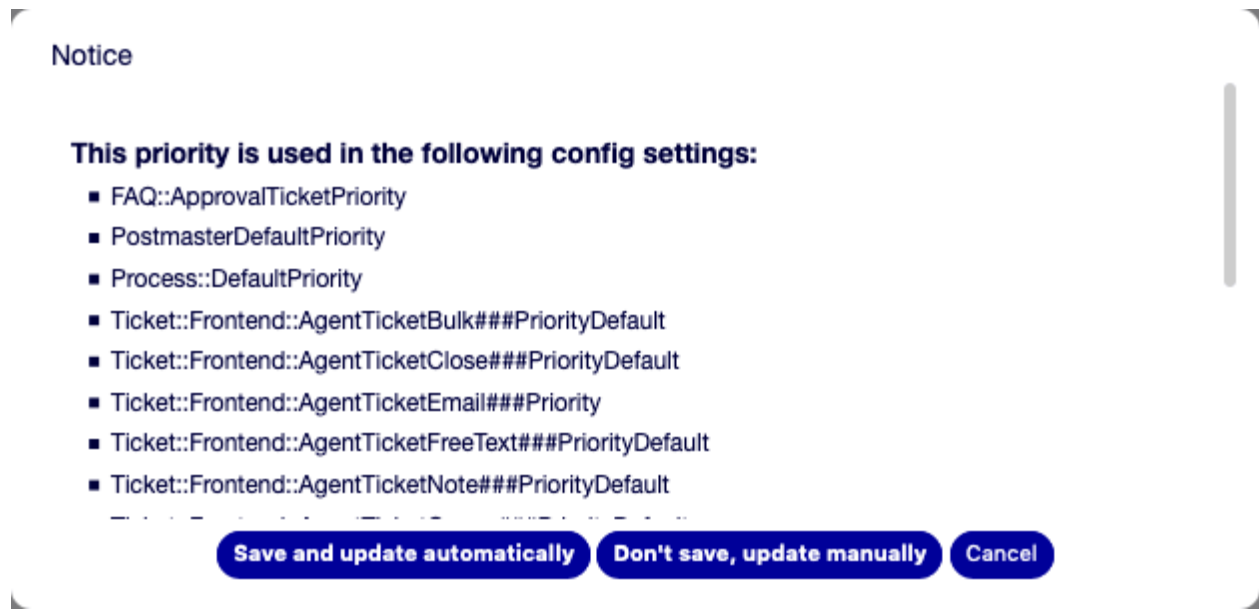

Queue Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Add Queue

* Name:

Sub-queue of:

* Group: admin

Unlock timeout minutes:

0 = no unlock - 24 hours = 1440 minutes - Only business hours are counted.

If an agent locks a ticket and does not close it before the unlock timeout has passed, the ticket will unlock and will become available for other agents.

Escalation - first response time (minutes): (Notify by)

0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted.

If there is not added a customer contact, either email-external or phone, to a new ticket before the time defined here expires, the ticket is escalated.

Escalation - update time (minutes): (Notify by)

0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted.

If there is an article added, such as a follow-up via email or the customer portal, the escalation update time is reset. If there is no customer contact, either email-external or phone, added to a ticket before the time defined here expires, the ticket is escalated.

Escalation - solution time (minutes): (Notify by)

0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted.

If the ticket is not set to closed before the time defined here expires, the ticket is escalated.

* Follow up Option: possible

Specifies if follow up to closed tickets would re-open the ticket, be rejected or lead to a new ticket.

* Ticket lock after a follow up: No

If a ticket is closed and the customer sends a follow up the ticket will be locked to the old owner.

* System address:

Will be the sender address of this queue for email answers.

Default sign key:

To use a sign key, PGP keys or S/MIME certificates need to be added with identifiers for selected queue system address.

* Salutation: system standard salutation (en)

The salutation for email answers.

* Signature: system standard signature (en)

The signature for email answers.

Calendar:

* Validity: valid

Comment:

Save or Cancel

Fig. 1.15: Add Queue Screen

Chapter 1. Administration Area

Edit Queue

* Name: Misc

Sub-queue of:

* Group: users

Unlock timeout minutes: 0

0 = no unlock - 24 hours = 1440 minutes - Only business hours are counted.

If an agent locks a ticket and does not close it before the unlock timeout has passed, the ticket will unlock and will become available for other agents.

Escalation - first response time (minutes): (Notify by)

0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted.

If there is not added a customer contact, either email-external or phone, to a new ticket before the time defined here expires, the ticket is escalated.

Escalation - update time (minutes): (Notify by)

0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted.

If there is an article added, such as a follow-up via email or the customer portal, the escalation update time is reset. If there is no customer contact, either email-external or phone, added to a ticket before the time defined here expires, the ticket is escalated.

Escalation - solution time (minutes): (Notify by)

0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted.

If the ticket is not set to closed before the time defined here expires, the ticket is escalated.

* Follow up Option: possible

Specifies if follow up to closed tickets would re-open the ticket, be rejected or lead to a new ticket.

* Ticket lock after a follow up: No

If a ticket is closed and the customer sends a follow up the ticket will be locked to the old owner.

* System address: otobo@localhost x

Will be the sender address of this queue for email answers.

Default sign key (otobo@localhost):

To use a sign key, PGP keys or S/MIME certificates need to be added with identifiers for selected queue system address.

* Salutation: system standard salutation (en)

The salutation for email answers.

* Signature: system standard signature (en)

The signature for email answers.

Calendar:

* Validity: valid

Comment: All misc tickets.

Save or Save and finish or Cancel

Fig. 1.16: Edit Queue Screen

Notice

This queue is used in the following config settings:

- Ticket::Frontend::CustomerTicketMessage###QueueDefault
- Ticket::Frontend::UserDefaultQueue

You can either have the affected settings updated automatically to reflect the changes you just made or do it on your own by pressing 'update manually'.

Save and update automatically   **Don't save, update manually**   **Cancel**

Fig. 1.17: Queue Validation Dialog

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Sub-queue of It is possible to add the new queue under an existing one as sub-queue. This will be displayed as Parent Queue::Child Queue.

Group * It is possible to limit access to the selected group. The group creates a permission link between the queue and an agent or a customer user.

Unlock timeout minutes Any ticket on open, which is locked, in this queue will automatically unlock after the set amount of minutes. The value 0 (default) means tickets in this queue remain locked.

Escalation - first response time (minutes) The maximum amount of working time allowed before the agent contacts with the customer.

Note: First response time will not trigger for an email ticket or telephone ticket created by an agent.

Notify by Notify before time limit is reached (in percent).

Escalation - update time (minutes) The maximum amount of working time allowed between the agent contacts with the customer.

Notify by Notify before time limit is reached (in percent).

Escalation - solution time (minutes) The maximum amount of working time allowed until the ticket is marked as solved.

Note: Solution time will not reset if the ticket is reopened.

Notify by Notify before time limit is reached (in percent).

Follow up Option * Specify the handling of a follow up on closed tickets. Possible values:

new ticket The follow up will create a new ticket.

possible The follow up will reopen the already closed ticket.

reject The follow up will be rejected.

See also:

See Auto Responses chapter for more information.

Ticket lock after a follow up * Only applicable if the Follow up Option is set to possible. Locks the previously closed ticket, upon reopening, to the last owner. This ensures that a follow up for a ticket is processed by the agent that has previously handled that ticket.

> **Warning:** This does not take out-of-office into account. Use this setting with care to ensure or in combination with Unlock timeout minutes.

System address * Select one of the Email Addresses as the sender identity for this queue.

> **Note:** This is an ID in the database. Making changes to the Email Addresses can have adverse effects here.

Default sign key This is only active if PGP Keys or S/MIME Certificates is enabled in the System Configuration. Choose the key to sign emails per default.

Salutation * Select one of the defined Salutations.

Signature * Select one of the defined Signatures.

Calendar Select the calendar which defines working hours for this queue. Calendars are defined in the System Configuration.

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

Chat Channel Chat channel that will be used for communication related to the tickets in this queue.

1.1.5 Queues ⮞ Auto Responses

Communicating a change in, for example, service times, service levels or other information which would be good for a customer to know when opening a ticket can be a tedious and error-ridden task. Attempting to make sure all departments have the correct information and transmitting this to their customers poses a challenge.

OTOBO gives you the power to quickly assign the appropriate automatic responses to any queue, containing pertinent service information, ensuring this information reaches your customers before expectations aren't reached.

Use this screen to add one or more automatic responses to one or more queues. The management screen is available in the Queues ⮞ Auto Responses module of the Ticket Settings group.

Manage Queues ⮞ Auto Responses Relations

To assign an automatic response to a queue:

1. Click on a queue in the Queues column.
2. Select the automatic responses you would like to add the queue to.
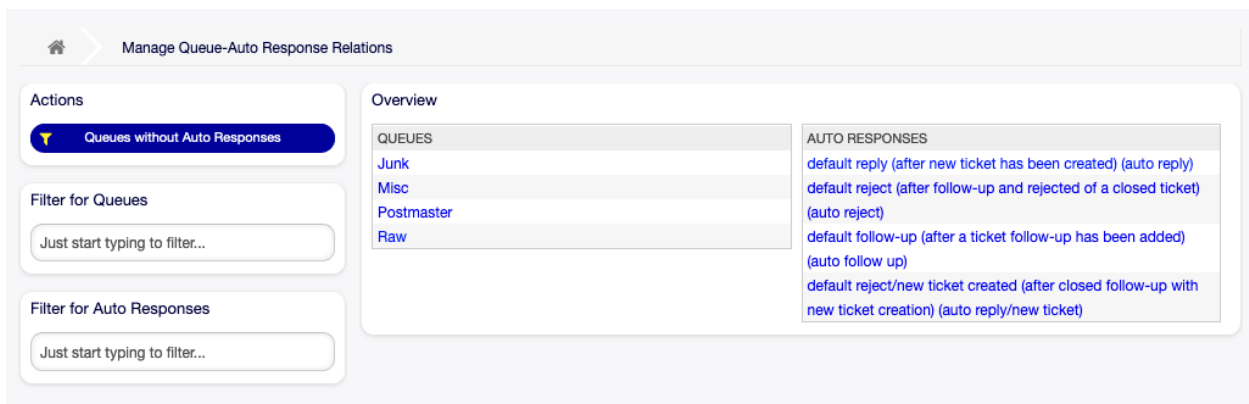
Fig. 1.18: Manage Queue-Auto Response Relations

3. Click on the Save or Save and finish button.
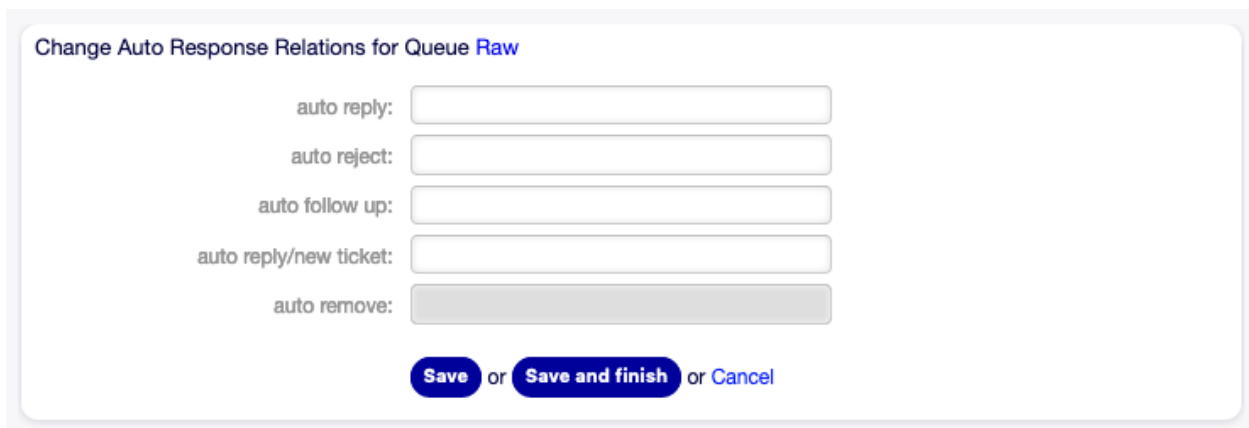
Fig. 1.19: Change Auto Response Relations for Queue

Note: It is not possible to assign multiple queues to an automatic response by clicking on the automatic response name. A click on the automatic response will open to the Auto Responses screen.

Note: If several automatic responses or queues are added to the system, use the filter box to find a particular automatic response or queue by just typing the name to filter.

Queues ⟷ Auto Responses Settings

The following settings are available when assigning some automatic responses to a queue.

auto reply This automatic response will be sent to users, when they send a message to the queue, and the message is not a follow-up message of a ticket.

auto reject This automatic response will be sent to users, when they send a message to the queue and reject is set as Follow up Option in the queue settings.

auto follow up This automatic response will be sent to users, when they send a message to the queue and possible is set as Follow up Option in the queue settings.

auto reply/new ticket This automatic response will be sent to users, when they send a message to the queue and new ticket is set as Follow up Option in the queue settings.

auto remove This automatic response will be sent to users, when the remove option is enabled.

Note: Auto reply, auto reject and auto reply/new ticket mutually cancel each other based on the Queues settings. Only one will take effect per queue.

1.1.6 Salutations

Addressing customers must be done in a standardized way. Your customers may not always be external customers requiring a less formal greeting.

OTOBO provides you with the tools needed to create a standardized communication form for any one of your queues. As defined in the Queue Settings: Salutations, Templates, and Signatures are combined to ensure a well formed standardized email communication.

Salutations can be linked to one or more Queues. A salutation is used only in email answers to tickets.

Use this screen to add salutations to the system. A fresh OTOBO installation already contains a standard salutation. The salutation management screen is available in the Salutations module of the Ticket Settings group.
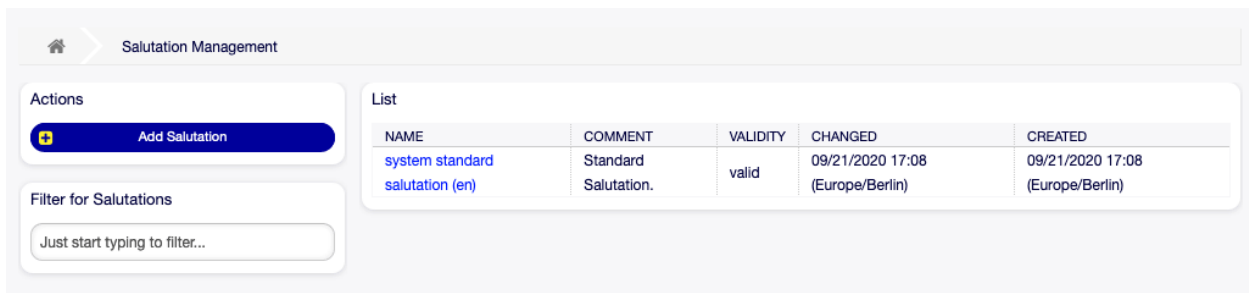
Fig. 1.20: Salutation Management Screen

Manage Salutations

To add a salutation:

1. Click on the Add Salutation button in the left sidebar.
2. Fill in the required fields.
3. Click on the Save button.

Warning: Salutations can not be deleted from the system. They can only be deactivated by setting the Validity option to invalid or invalid-temporarily.

To edit a salutation:

Fig. 1.21: Add Salutation Screen

1. Click on a salutation in the list of salutations.

2. Modify the fields.

3. Click on the Save or Save and finish button.

Fig. 1.22: Edit Salutation Screen

Note: If several salutations are added to the system, use the filter box to find a particular salutation by just typing the name to filter.

Warning: Before invalidating this object, please go to the Queues module of the Ticket Settings group and make sure all queues using this setting are using a valid object.

Salutation Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Salutation * The text that will be placed to the beginning of new emails.

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

Salutation Variables

Using variables in the text makes it possible to personalize messages. Variables, known as OTOBO tags, are replaced by OTOBO when generating the message. Find a list of available tags stems for this resource at the bottom of both add and edit screens.

Fig. 1.23: Salutation Variables

For example, the variable `<OTOBO_CUSTOMER_DATA_UserLastname>` expands to the customer's last name to be included in something like the following.

```
Dear <OTOBO_CUSTOMER_DATA_UserFirstname> <OTOBO_CUSTOMER_DATA_UserLastname>,
```

This tag expands, for example to:

```
Dear Lisa Wagner,
```

1.1.7 Service Level Agreements

Your organization must meet the time demands of your customers. Timely service matters. Response to questions, updates on issues, and solutions must be provided in an agreed amount of time. The agent must receive notification of possible breaches, to prevent ticket escalation.

OTOBO scales well with your needs and offers management of service level agreements (SLAs). OTOBO provides you with the possibility to create numerous service level agreements covering all of your service and customer need. Each SLA can cover multiple services and define the availability of service and escalation periods.

Use this screen to add service level agreements to the system. A fresh OTOBO installation doesn't contain any service level agreements by default. The service level agreement management screen is available in the Service Level Agreements module of the Ticket Settings group.
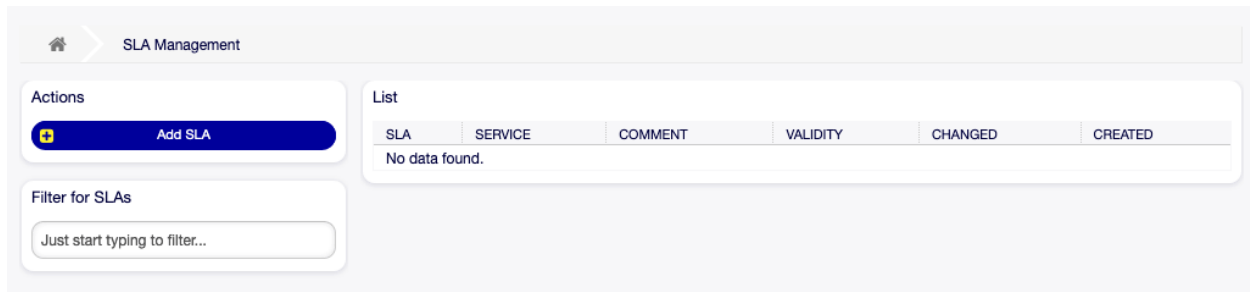
Fig. 1.24: Service Level Agreement Management Screen

See also:

To use this feature, `Ticket::Service` must be activated in the System Configuration under the Administration group to be selectable in the ticket screens. You may click on the link in the warning message of the notification bar to directly jump to the configuration setting.

Manage Service Level Agreements

Note: Adding service level agreements requires, that at least one service is added to the system. Create services in the Services screen.

To add a service level agreement:

1. Click on the Add SLA button in the left sidebar.
2. Fill in the required fields.
3. Click on the Save button.

Warning: Service level agreements can not be deleted from the system. They can only be deactivated by setting the Validity option to invalid or invalid-temporarily.

To edit a service level agreement:

1. Click on a service level agreement in the list of service level agreements.
2. Modify the fields.

Fig. 1.25: Add Service Level Agreement Screen

3. Click on the Save or Save and finish button.

Note: If several service level agreements are added to the system, use the filter box to find a particular service level agreement by just typing the name to filter.

Service Level Agreement Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

SLA * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Service Select one or more of the Services to be assigned to this SLA.

Calendar Select the calendar which defines working hours for this queue. Calendars are defined in the System Configuration.

Escalation - first response time (minutes) The maximum amount of working time allowed before the agent contacts with the customer.

Note: First response time will not trigger for an email ticket or telephone ticket created by an agent.

Notify by Notify before time limit is reached (in percent).

Fig. 1.26: Edit Service Level Agreement Screen

Escalation - update time (minutes) The maximum amount of working time allowed between the agent contacts with the customer.

Notify by Notify before time limit is reached (in percent).

Escalation - solution time (minutes) The maximum amount of working time allowed until the ticket is marked as solved.

Note: Solution time will not reset if the ticket is reopened.

Notify by Notify before time limit is reached (in percent).

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

Dialog message Is being displayed if a customer chooses this SLA on ticket creation (only in the external interface).

1.1.8 Services

Requests to your service desk should be categorized based on affected service to correctly reach a team of experts to deal with the issue. Because not all your agents can deal with specific problems. Lack of experience or access to resources to fix an issue requires defining the service affected in a ticket helps to categorize the issue and target the correct teams.

OTOBO allows adding all services offered to your customers. These services may be later bound to Service Level Agreements to ensure a timely solution based on customer-specific agreements.

Use this screen to add services to the system. A fresh OTOBO installation doesn't contain any services by default. The service management screen is available in the Services module of the Ticket Settings group.
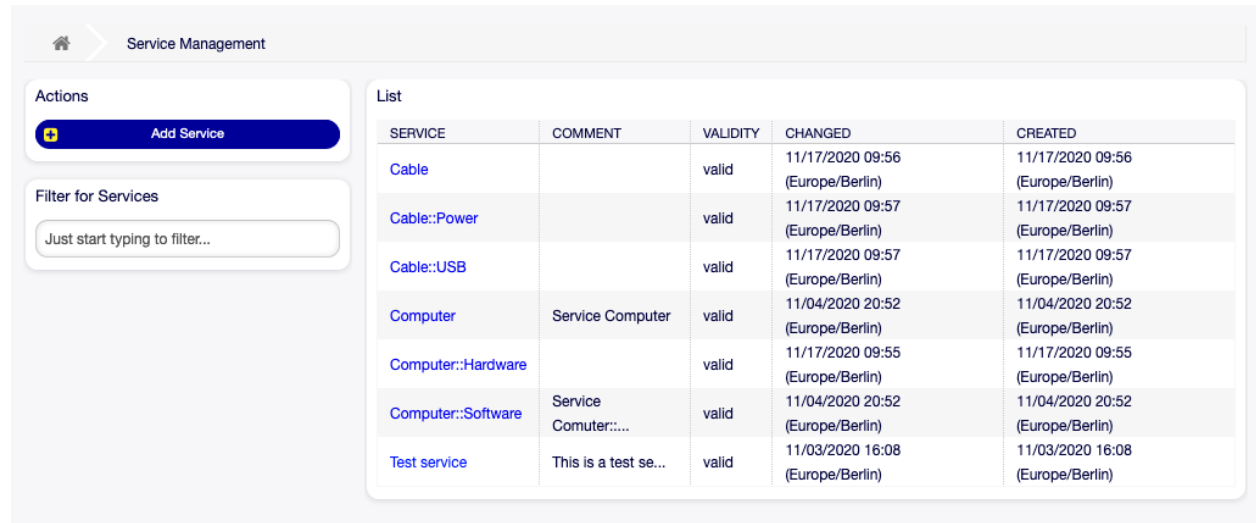
Fig. 1.27: Service Management Screen

See also:

To use this feature, `Ticket::Service` must be activated in the System Configuration under the Administration group to be selectable in the ticket screens. You may click on the link in the warning message of the notification bar to directly jump to the configuration setting.

Manage Services

To add a service:

1. Click on the Add Service button in the left sidebar.

2. Fill in the required fields.

3. Click on the Save button.

Warning: Services can not be deleted from the system. They can only be deactivated by setting the Validity option to invalid or invalid-temporarily.

To edit a service:

1. Click on a service in the list of services.

2. Modify the fields.

3. Click on the Save or Save and finish button.

Fig. 1.28: Add Service Screen

Fig. 1.29: Edit Service Screen

Note: If several services are added to the system, use the filter box to find a particular service by just typing the name to filter.

Warning: Changing the name of this object should be done with care, the check only provides verification for certain settings and ignores things where the name can't be verified. Some examples are dashboard filters, action control lists (ACLs), and processes (sequence flow actions) to name a few. Documentation of your setup is key to surviving a name change.

Service Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Service * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Sub-service of It is possible to add the new service under an existing one as sub-service. This will be displayed as Parent Service::Child Service.

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

Note: A service can only be selectable in the ticket screens, if the service is assigned to the customer user of the ticket or it is set as default service for all customer users in the Customer Users ⮕ Services screen.

1.1.9 Signatures

Corporate identity and team information are essential in each communication. The name of the employee writing and other vital details like disclaimers are some examples of necessary information to include in the communication with the customer.

OTOBO provides you with the same tools here, as with Salutations, to create a standardized communication form for any one of your queues. As defined in the Queue Settings: Salutations, Templates, and Signatures are combined to ensure a well formed standardized email communication.

Signatures can be linked to one or more Queues. A signature is used only in email answers to tickets.

Use this screen to add signatures to the system. A fresh OTOBO installation already contains a standard signature. The signature management screen is available in the Signatures module of the Ticket Settings group.

Manage Signatures

To add a signature:
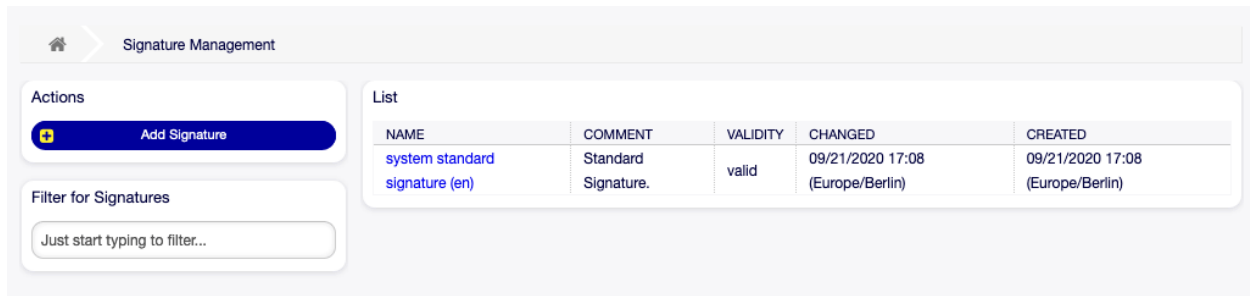
1. Click on the Add Signature button in the left sidebar.

Fig. 1.30: Signature Management Screen

2. Fill in the required fields.

3. Click on the Save button.

> **Warning:** Signatures can not be deleted from the system. They can only be deactivated by setting the Validity option to invalid or invalid-temporarily.

To edit a signature:

1. Click on a signature in the list of signatures.

2. Modify the fields.

3. Click on the Save or Save and finish button.

> **Note:** If several signatures are added to the system, use the filter box to find a particular signature by just typing the name to filter.

> **Warning:** Before invalidating this object, please go to the Queues module of the Ticket Settings group and make sure all queues using this setting are using a valid object.

Signature Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Signature * The text that will be placed to the end of new emails.

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

Fig. 1.31: Add Signature Screen
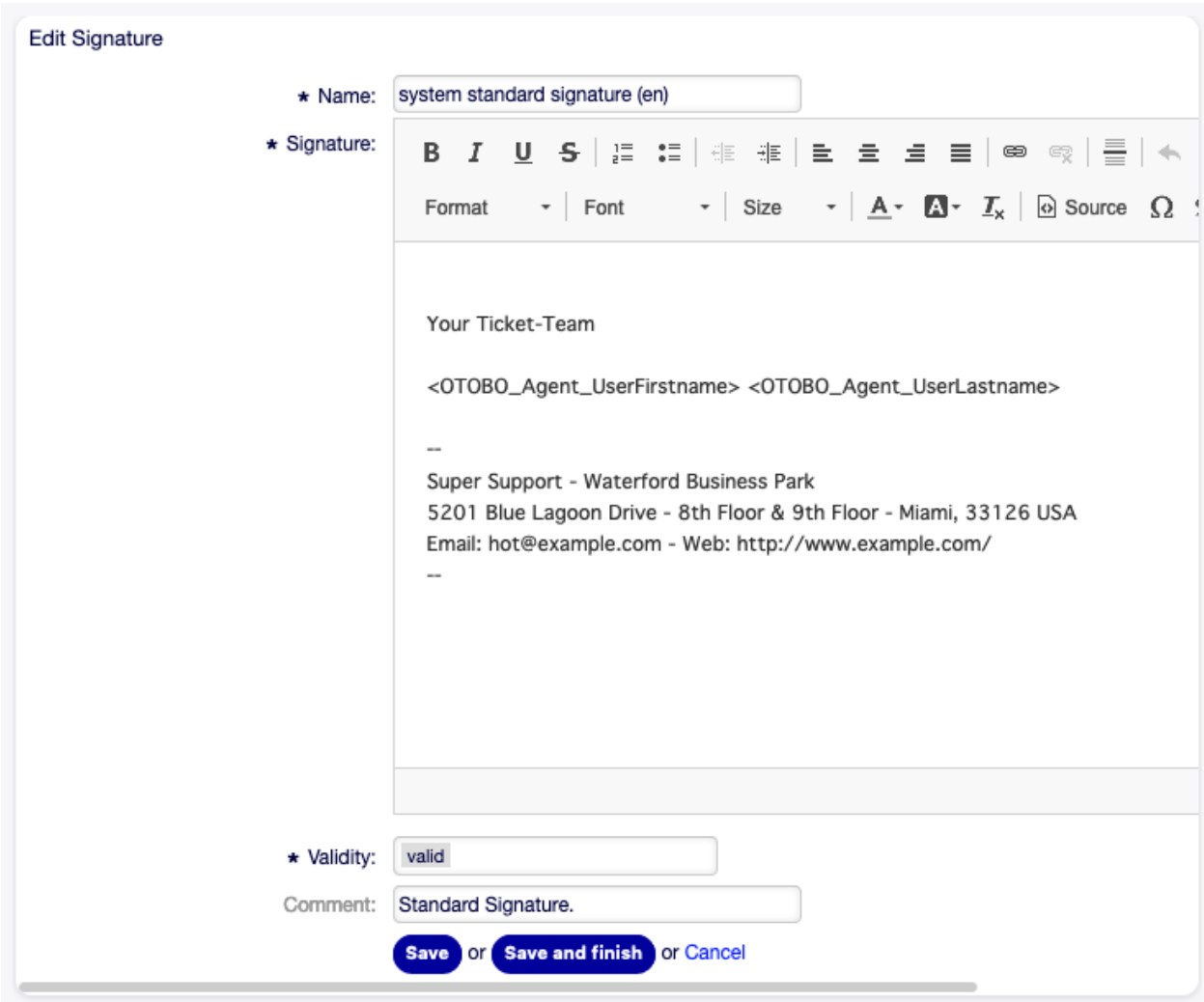
Fig. 1.32: Edit Signature Screen

Signature Variables

Using variables in the text makes it possible to personalize messages. Variables, known as OTOBO tags, are replaced by OTOBO when generating the message. Find a list of available tags stems for this resource at the bottom of both add and edit screens.

Fig. 1.33: Signature Variables

For example, the variable `<OTOBO_CURRENT_UserFirstname> <OTOBO_CURRENT_UserLastname>` expands to the agent's first and last name allowing a template to include something like the following.

```
Best regards,

<OTOBO_CURRENT_UserFirstname> <OTOBO_CURRENT_UserLastname>
```

This tag expands, for example to:

```
Best regards,

Steven Weber
```

1.1.10 State pre-selection for Templates

1.1.11 States

Active tracking of tickets leads to a better sense of workload and provides metrics as a key performance indicator. Sorting tasks and setting appointments can help to level-off the workload and keep your service desk running.

OTOBO uses ticket states to ensure that your agents always know which tickets are being attended to and which not. Additionally, detailed reports on the states of your tickets can be provided by ticket search or reports and personalized sorting is possible using dashboards and queue and service overviews.

Nine states are pre-defined. More states can be added, but the default states are enough to get you going and mostly enough for any situation.

The default states are the following:

closed successful A ticket is complete. The customer received a solution which worked.

closed unsuccessful A ticket is complete. The customer received no solution or the solution was not appropriate.

merged The ticket content is found in a different ticket.

new The ticket is created by the customer without contact with an agent.

open The ticket is currently in progress. Customer and agent are in contact with each another.

pending auto close+ The ticket will be set as closed successful upon reaching the pending time.

pending auto close- The ticket will be set as closed unsuccessful upon reaching the pending time.

pending reminder The ticket should be worked on again upon reaching the pending time.

removed The ticket has been removed from the system.

Note: Pending jobs are checked per default every two hours and forty-five minutes. This time is a static time, which means the times are 02:45, 04:45, 06:45 and so on. The job can be run more often or seldom and is configured in the System Configuration module of the Administration group.

Use this screen to add states to the system. A fresh OTOBO installation contains several states by default. The state management screen is available in the States module of the Ticket Settings group.
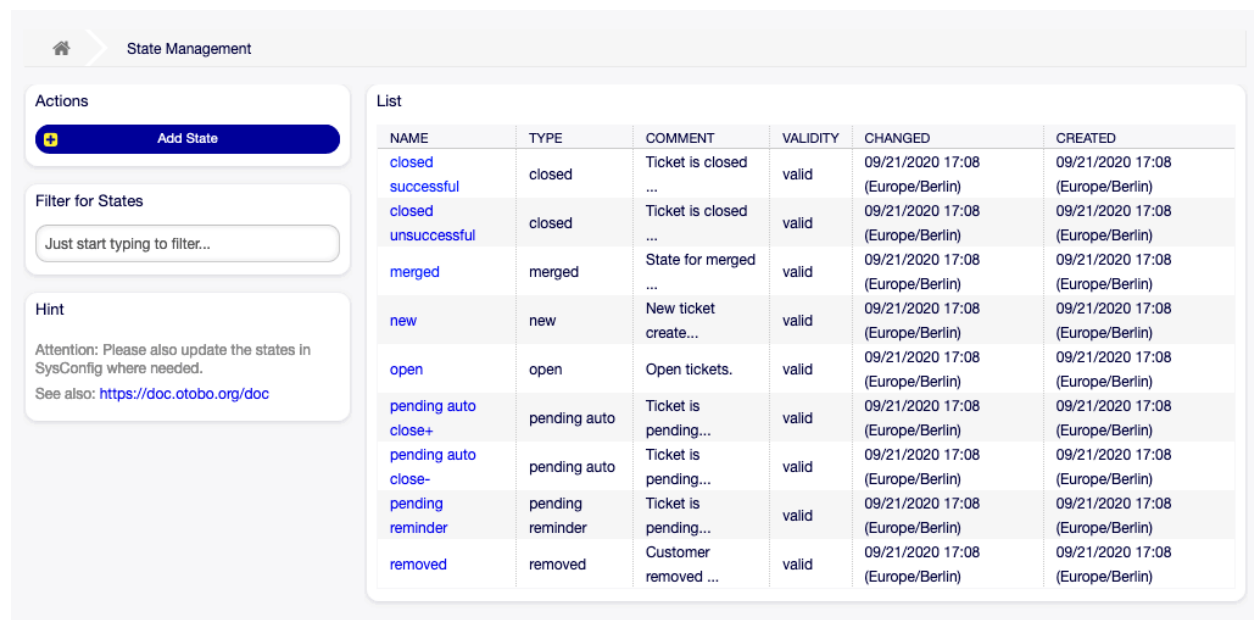
Fig. 1.34: State Management Screen

Manage States

To add a state:

1. Click on the Add State button in the left sidebar.

2. Fill in the required fields.

3. Click on the Save button.

Fig. 1.35: Add State Screen

Warning: States can not be deleted from the system. They can only be deactivated by setting the Validity option to invalid or invalid-temporarily.

To edit a state:

1. Click on a state in the list of states.

2. Modify the fields.

3. Click on the Save or Save and finish button.

Fig. 1.36: Edit State Screen

Note: If several states are added to the system, use the filter box to find a particular state by just typing the name to filter.

Notice

This state is used in the following config settings:

- PostmasterFollowUpState
- PostmasterFollowUpStateClosed
- Ticket::Frontend::AgentTicketBulk###StateDefault
- Ticket::Frontend::AgentTicketCompose###StateDefault
- Ticket::Frontend::AgentTicketEmail###StateDefault
- Ticket::Frontend::AgentTicketEmailOutbound###StateDefault
- Ticket::Frontend::AgentTicketFreeText###StateDefault
- Ticket::Frontend::AgentTicketNote###StateDefault

Save and update automatically   **Don't save, update manually**   **Cancel**

Fig. 1.37: State Validation Dialog

If you change the name of a state which is used in the system configuration, a validation check will warn you and give you the following options:

Save and update automatically Apply the change and also update the affected settings.
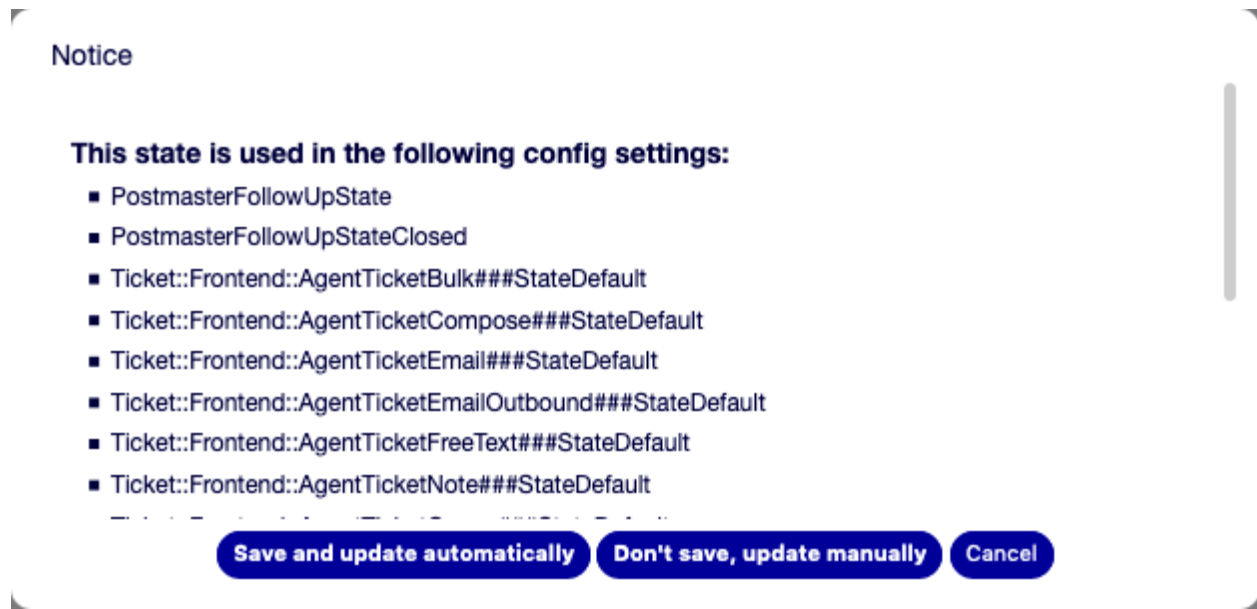
Don't save, update manually Apply the change, but don't update the affected settings. The updates need to be done manually.

Cancel Cancel the action.

> **Warning:** Changing the name of this object should be done with care, the check only provides verification for certain settings and ignores things where the name can't be verified. Some examples are dashboard filters, action control lists (ACLs), and processes (sequence flow actions) to name a few. Documentation of your setup is key to surviving a name change.

State Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

State type * Every state is linked to a type, which needs to be specified if a new state is created or an existing one is edited. The following types are available:

- closed
- merged
- new
- open

- pending auto

- pending reminder

- removed

Note: State types are predefined and cannot be changed in the software due to their special mechanics. When adding new states for pending auto and pending reminder state types you must make further configurations in the System Configuration module of the Administration group.

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

State Configuration Options

The following options are relevant and noteworthy. Please review these when managing states:

- `Daemon::SchedulerCronTaskManager::Task###TicketPendingCheck`

- `Ticket::StateAfterPending`

1.1.12 Templates

Providing the correct and consistent answer all the time regardless of employee or knowledge-level is important to maintain a professional appearance to your customers. Additionally, speed in sending standard answers is key to wading through the masses of requests in growing service desks.

OTOBO templates offer you a variety of ways to deal with standardizing communications and help to pre-define texts so that the customer always receives the same level and quality of service from all agents.

Use this screen to add templates for use in communications. A fresh OTOBO installation already contains a template by default. The template management screen is available in the Templates module of the Ticket Settings group.

Manage Templates

Note: To add attachments to a template, it needs to create the attachment first in the Attachments screen.

To add a template:

1. Click on the Add Template button in the left sidebar.

2. Fill in the required fields.

3. Click on the Save button.

To edit a template:

1. Click on a template in the list of templates.

Fig. 1.38: Template Management Screen

2. Modify the fields.

3. Click on the Save or Save and finish button.

To delete a template:

1. Click on the trash icon in the list of templates.

2. Click on the Confirm button.

Note: If several templates are added to the system, a filter box is useful to find a particular template by just typing to filter.

Template Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Type * There are different kind of templates that are used for different purposes. A template can be:

    Answer To be used as a ticket response or reply.

    Create To be used for new phone or email ticket.

    Email To be used for writing an email to a customer user or to someone else.

    Forward To be used to forward an article to someone else.

    Note To be used to create notes.

    Phone call To be used for inbound and outbound phone calls.

    Process dialog To be used in process management to pre-fill text of articles in user task activity dialogs.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Template The body of the article added by the template.

Fig. 1.39: Add Template Screen

Fig. 1.40: Edit Template Screen

Fig. 1.41: Delete Template Screen

Attachments It is possible to add one or more attachments to this template. Attachments can be added in the Attachments screen.

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

Template Variables

Using variables in the text makes it possible to personalize messages. Variables, known as OTOBO tags, are replaced by OTOBO when generating the message. Find a list of available tags stems for this resource at the bottom of both add and edit screens.

For example, the variable `<OTOBO_TICKET_TicketNumber>` expands to the ticket number allowing a template to include something like the following.

```
Ticket#<OTOBO_TICKET_TicketNumber>
```

This tag expands, for example to:

```
Ticket#2018101042000012
```

1.1.13 Templates ⧉ Attachments

Making changes to standard attachments is sometimes overwhelming. The question is, "Where were the attachments used?" or "How to quickly update them across the board?". It's also important to know who are using which attachments, before updating them. A new attachment can have multiple uses within your organization.

OTOBO empowers you to manage this providing you an overview to manage the 1:n relationships and quickly identify the templates using your attachments.

Use this screen to add one or more attachments to one or more templates. To use this function, at least one attachment and one template need to have been added to the system. The management screen is available in the Templates ⧉ Attachments module of the Ticket Settings group.

Manage Templates ⧉ Attachments Relations

To add some attachments to a template:

1. Click on a template in the Templates column.
2. Select the attachments you would like to add to the template.
3. Click on the Save or Save and finish button.

To assign an attachment to some templates:

1. Click on an attachment in the Attachments column.
2. Select the templates you would like to assign the attachment to.
3. Click on the Save or Save and finish button.

Reference

You can use the following tags:

`<OTOBO_AGENT_SUBJECT[20]>`
To get the first 20 characters of the subject of the current/latest agent article (current for Answer and Forward, latest for Note template type). This tag is not supported for other template types.

`<OTOBO_AGENT_BODY[5]>`
To get the first 5 lines of the body of the current/latest agent article (current for Answer and Forward, latest for Note template type). This tag is not supported for other template types.

`<OTOBO_CUSTOMER_SUBJECT[20]>`
To get the first 20 characters of the subject of the current/latest article (current for Answer and Forward, latest for Note template type). This tag is not supported for other template types.

`<OTOBO_CUSTOMER_BODY[5]>`
To get the first 5 lines of the body of the current/latest article (current for Answer and Forward, latest for Note template type). This tag is not supported for other template types.

`<OTOBO_OWNER_*>`
Ticket owner options (e. g. `<OTOBO_OWNER_UserFirstname>`).

`<OTOBO_RESPONSIBLE_*>`
Ticket responsible options (e. g. `<OTOBO_RESPONSIBLE_UserFirstname>`).

`<OTOBO_CURRENT_*>`
Options of the current user who requested this action (e. g. `<OTOBO_CURRENT_UserFirstname>`).

`<OTOBO_TICKET_*>`
Options of the ticket data (e. g. `<OTOBO_TICKET_TicketNumber>`, `<OTOBO_TICKET_TicketID>`, `<OTOBO_TICKET_Queue>`, `<OTOBO_TICKET_State>`).

`<OTOBO_TICKET_DynamicField_*>`
Options of ticket dynamic fields internal key values (e. g. `<OTOBO_TICKET_DynamicField_TestField>`, `<OTOBO_TICKET_DynamicField_TicketFreeText1>`).

`<OTOBO_TICKET_DynamicField_*_Value>`
Options of ticket dynamic fields display values, useful for Dropdown and Multiselect fields (e. g. `<OTOBO_TICKET_DynamicField_TestField_Value>`, `<OTOBO_TICKET_DynamicField_TicketFreeText1_Value>`).

`<OTOBO_CUSTOMER_DATA_*>`
Options of the current customer user data (e. g. `<OTOBO_CUSTOMER_DATA_UserFirstname>`).

`<OTOBO_CONFIG_*>`
Config options (e. g. `<OTOBO_CONFIG_HttpType>`).

Note: Create type templates only supports this smart tags: **`<OTOBO_CURRENT_*>`** and **`<OTOBO_CONFIG_*>`**

Example template:

```
The current ticket state is: "<OTOBO_TICKET_State>"

Your email address is: "<OTOBO_CUSTOMER_UserEmail>"
```
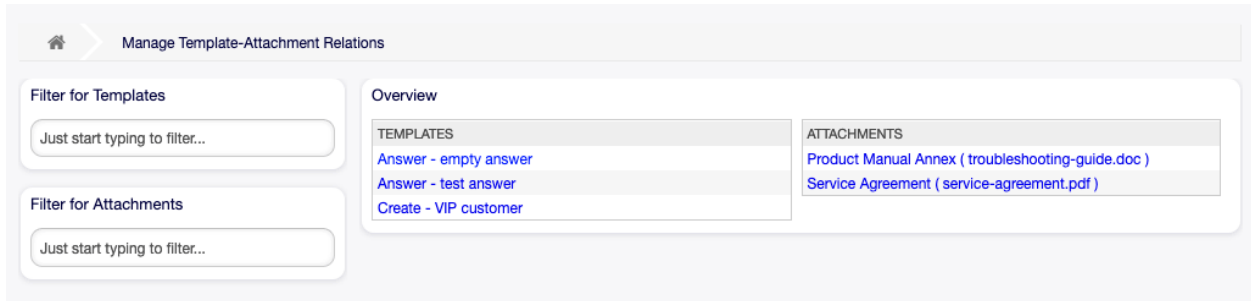
Fig. 1.42: Template Variables

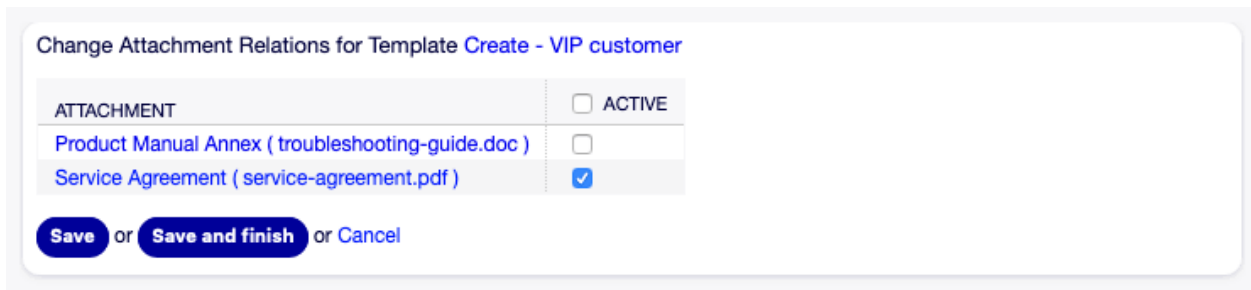Fig. 1.43: Manage Template-Attachment Relations

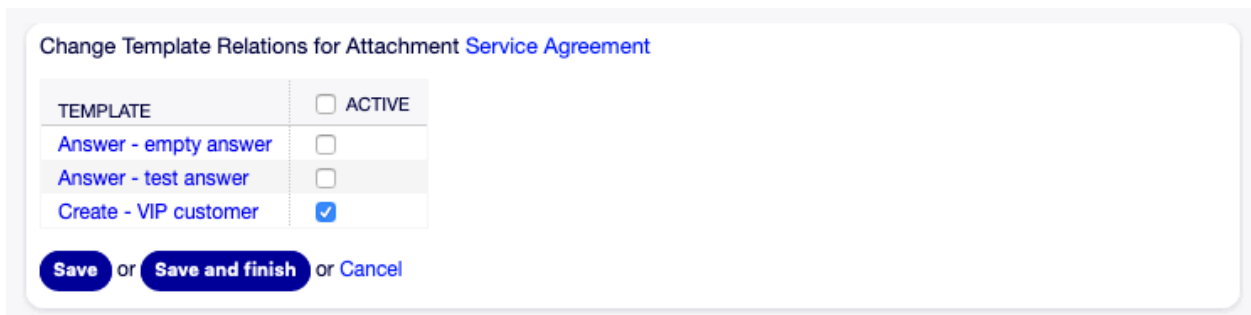Fig. 1.44: Change Attachment Relations for Template

Fig. 1.45: Change Template Relations for Attachment

Note: If several templates or attachments are added to the system, use the filter box to find a particular template or attachment by just typing the name to filter.

Multiple templates or attachments can be assigned in both screens at the same time. Additionally clicking on a template or clicking on an attachment in the relations screen will open the Edit Template screen or the Edit Attachment screen accordingly.

Warning: Accessing a template or an attachment provides no back link to the relations screen.

1.1.14 Templates ⮂ Queues

Sharing, distributing and making changes to standard text modules and information for sharing with customers or communication between teams can be an impossible mission.

OTOBO can quickly aid you in ensuring that all of your teams have the correct templates available for use by assignment based on the queue.

Use this screen to add one or more templates to one or more queues. To use this function, at least one template and one queue need to have been added to the system. The management screen is available in the Templates ⮂ Queues module of the Ticket Settings group.
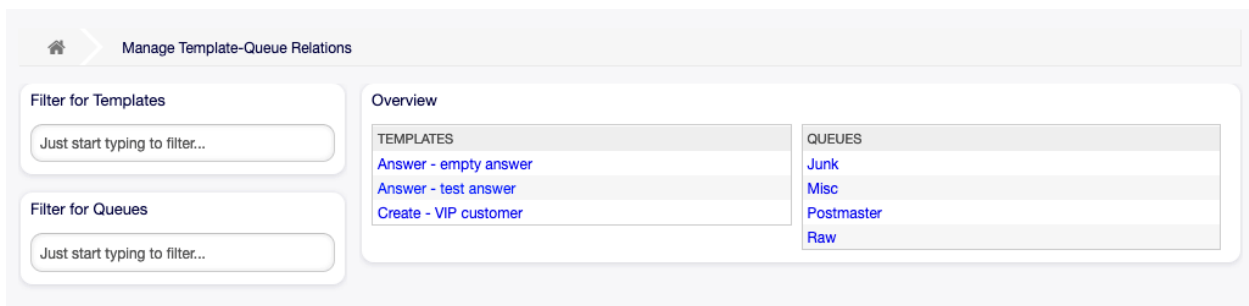

Fig. 1.46: Manage Template-Queue Relations Screen

Manage Templates ⮂ Queues Relations

To assign a template to a queue:

1. Click on a template in the Templates column.

2. Select the queues you would like to add the template to.

3. Click on the Save or Save and finish button.

To assign a queue to a template:

1. Click on a queue in the Queues column.

2. Select the templates you would like to assign the queue to.

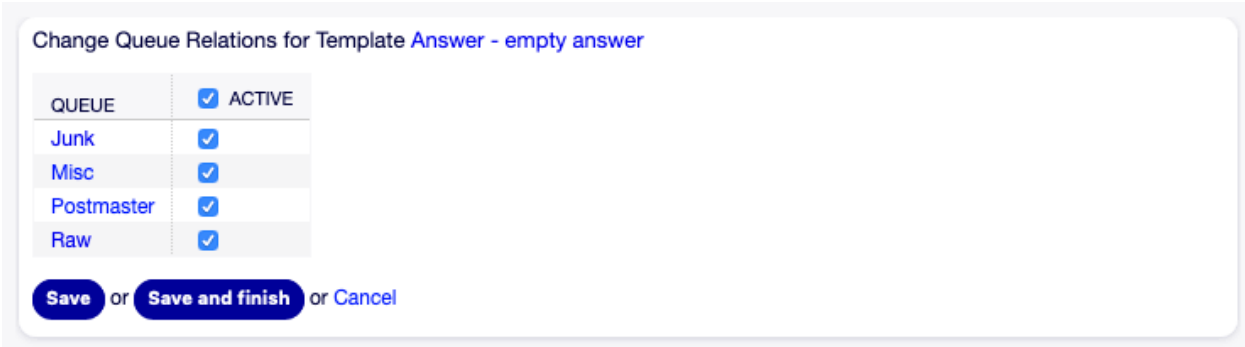3. Click on the Save or Save and finish button.
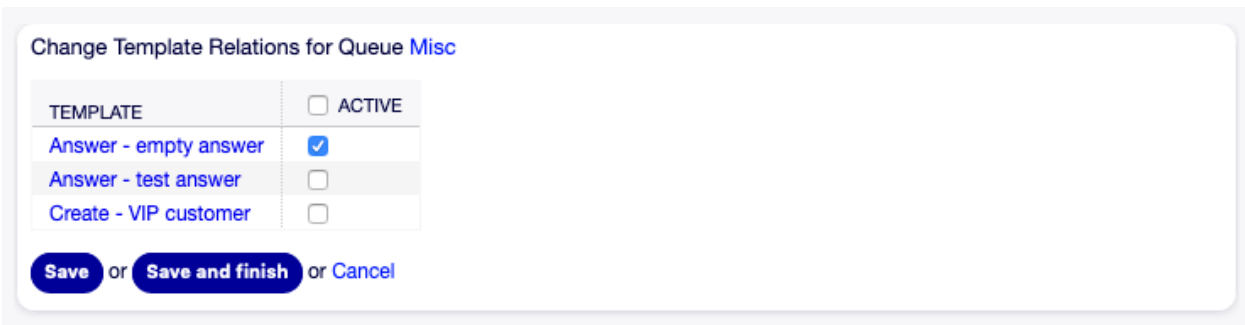

Fig. 1.47: Change Queue Relations for Template

Fig. 1.48: Change Template Relations for Queue

Note: If several templates or queues are added to the system, use the filter box to find a particular template or queue by just typing the name to filter.

Multiple templates or queues can be assigned in both screens at the same time. Additionally clicking on a template or clicking on a queue in the relations screen will open the Edit Template screen or the Edit Queue screen accordingly.

Warning: Accessing a queue or a template provides no back link to the relations screen.

1.1.15 Types

Good KPIs (Key Performance Indicators) require knowing the type of work your organization performs. Not all tasks take the same amount of effort even when performed by the same team. Creating a queue structure for this purpose can be overpowered due to the amount of configuration required to create and manage a queue.

OTOBO provisions for KPIs with minimal overhead using ticket types. Typical types used in IT service desks are unclassified, incident and problem. You can quickly define new types with ease.

Use this screen to add types to the system. A fresh OTOBO installation contains an unclassified type by default. The type management screen is available in the Types module of the Ticket Settings group.
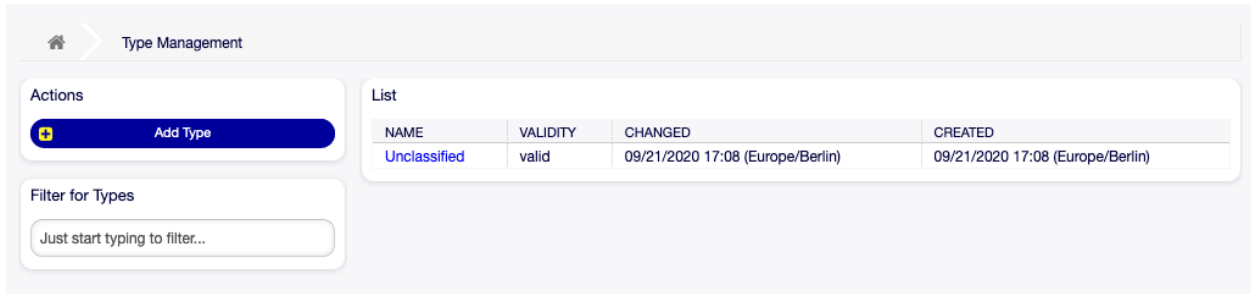
Fig. 1.49: Type Management Screen

Warning: `Ticket::Type` must first be activated via System Configuration under the Administration group to be selectable in the ticket screens. You may click on the link in the warning message to directly jump to the configuration setting.

Fig. 1.50: Type Activation Warning

Manage Types

To add a type:

1. Click on the Add Ticket Type button in the left sidebar.

2. Fill in the required fields.

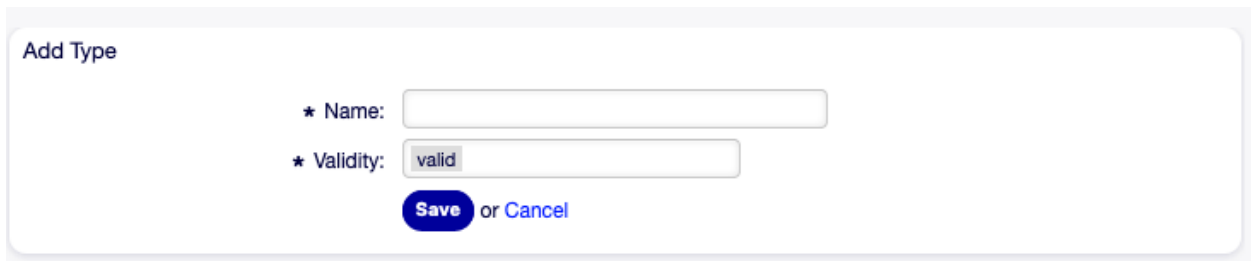3. Click on the Save button.

Fig. 1.51: Add Type Screen

Warning: Types can not be deleted from the system. They can only be deactivated by setting the Validity option to invalid or invalid-temporarily.

To edit a type:

1. Click on a type in the list of types.

2. Modify the fields.

3. Click on the Save or Save and finish button.

Fig. 1.52: Edit Type Screen

Note: If several types are added to the system, use the filter box to find a particular type by just typing the name to filter.

Type Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

1.2 Communication & Notifications

Clear, complete, and transparent communication is key to ensuring that your organization offers outstanding service to your customers. Your organization would like to give your customers the most consistent service and ensure that your service is highly recognizable, every time. Your teams want to have an identity which instills a specific sense of security and familiarity.

OTOBO provides you with the tools to build teams, trust and security to better and uniformly serve your customer. With system addresses you can assign your inbound mails to certain teams and allow them all to use the same address without causing confusion by using team mailboxes via a mail client. Templates, automatic answers, and attachment management let you leverage central management of the most important communications. Many other tools are also available and covered in the sections below.

1.2.1 Admin Notification

Corporations may need to make general announcements to everyone or publish news to several groups of agents or individuals. Also, OTOBO administrator may need to contact with specific agents regarding an event.

OTOBO gives the administration the tool Admin Notification making sending announcements and news to the masses of users accurate and timely, to target group of people, simple. Administrators can send notifications based on a specific list of recipients, or a group of users inside OTOBO, with powerful text editor enhanced content.

Use this screen to send administrative messages to specific agents, group or role members. The message compose screen is available in the Admin Notification module of the Communication & Notifications group.

The following settings are available when composing an administrative message. The fields marked with an asterisk are mandatory.

From * This email address will be added into the From field of the message.

Send message to users One or more Agents can be selected in this field, to whom the message will be sent.

Send message to group members One or more Groups can be selected in this field, to whose members the message will be sent.

Group members need to have permission With these radio buttons can be selected, if a group member needs read-only or read-write permissions to receive the message.

Send message to role members One or more Roles can be selected in this field, to whose members the message will be sent.

Also send to customers in groups Select this checkbox to send the message also for customers in groups.

Note: This option is available only, if `CustomerGroupSupport` setting is enabled.

Subject * The subject of the messages.

Body * The body text of the message.

1.2.2 Appointment Notifications

Missing appointments can damage your image with a customer. Once there is an appointment assigned in the calendar, it's normal to receive notification:

- Upon a new or changed event
- Upon cancellation of an event
- Before the event, as a reminder

Notification relieves the agent the stress of mentally tracking appointments.

OTOBO appointment notifications satisfies this need. Here an administrator can easily set notifications with general rules, including trigger events and filters. Afterward, appointments fitting the bill notify the correct users at the correct time.

Use this screen to add appointment notifications to the system. In a fresh OTOBO installation an appointment reminder notification is already added by default. The appointment notification management screen is available in the Appointment Notifications module of the Communication & Notifications group.
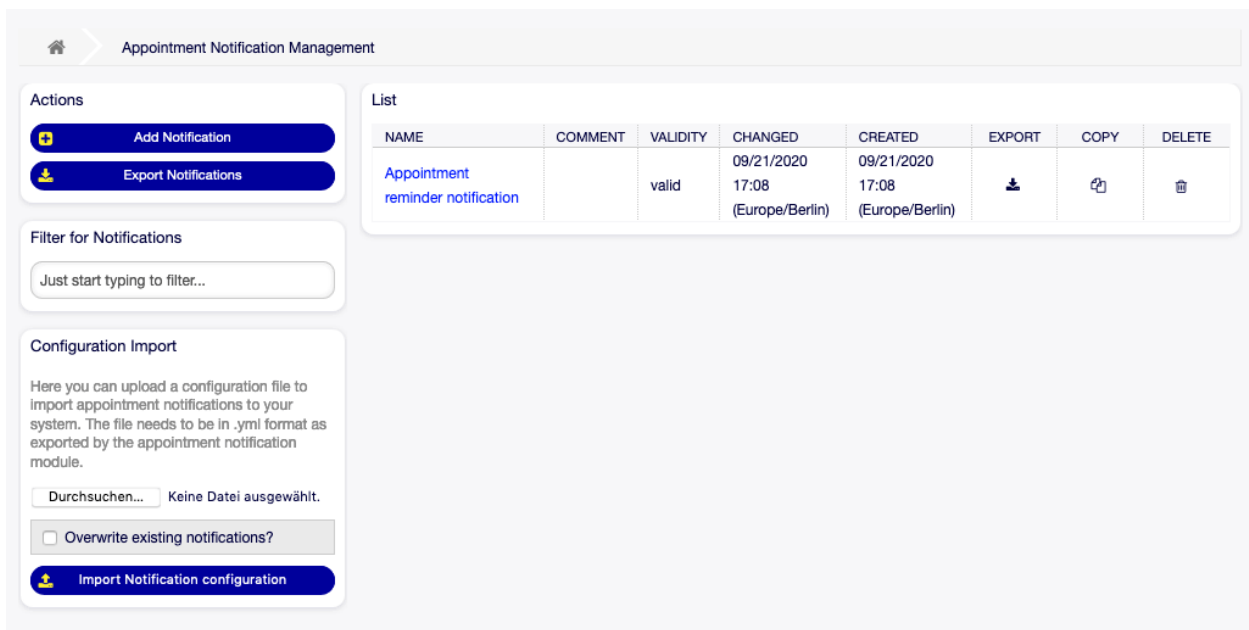
Fig. 1.53: Create Administrative Message Screen

Fig. 1.54: Appointment Notification Management Screen

Manage Appointment Notifications

To add an appointment notification:

1. Click on the Add Notification button in the left sidebar.

2. Fill in the required fields as explained in Appointment Notification Settings.

3. Click on the Save button.

To edit an appointment notification:

1. Click on an appointment notification in the list of appointment notifications.

2. Modify the fields as explained in Appointment Notification Settings.

3. Click on the Save or Save and finish button.

To delete an appointment notification:

1. Click on the trash icon in the list of appointment notifications.
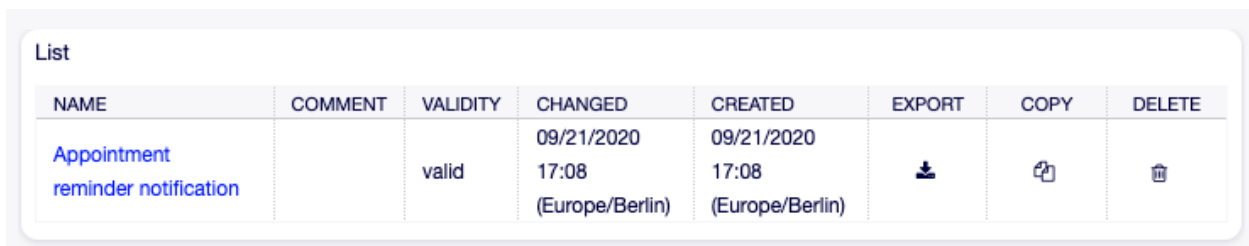
2. Click on the Confirm button.

Fig. 1.55: Delete Appointment Notification Screen

To export all appointment notifications:

1. Click on the Export Notifications button in the left sidebar.

2. Choose a location in your computer to save the `Export_Notification.yml` file.

To import appointment notifications:

1. Click on the Browse… button in the left sidebar.

2. Select a previously exported `.yml` file.

3. Click on the Overwrite existing notifications? checkbox, if you would like to overwrite the existing notifications.

4. Click on the Import Notification configuration button.

Appointment Notification Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

See also:

For an example, see the default appointment reminder notification which is included in a fresh OTOBO installation.

Basic Appointment Notification Settings

Fig. 1.56: Appointment Notification Settings - Basic

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

Show in agent preferences Define how the notification should be displayed in agent preferences. The following options are available:

    No The notification won't be displayed in agent preferences. The notification is sent to all appropriate agents by the defined method.

    Yes The notification will be displayed in agent preferences for selection. The agents may opt-in or opt-out.

Yes, but require at least one active notification method. The notification will be displayed in agent preferences, but require at least one active notification method. This is annotated by an asterisk next to the name.
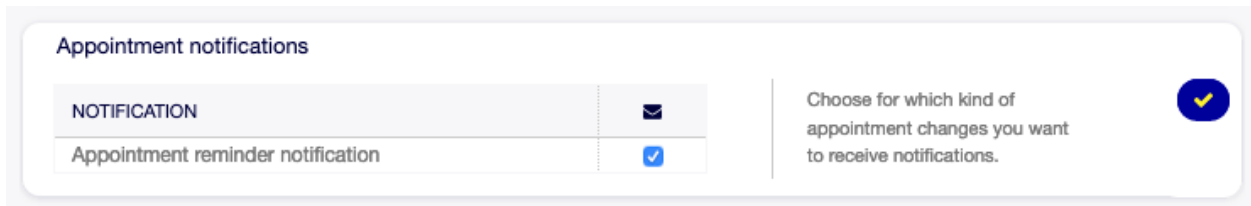

Fig. 1.57: Personal Appointment Notification Settings

Agent preferences tooltip This message will be shown on the agent preferences screen as a tooltip for this notification.

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.
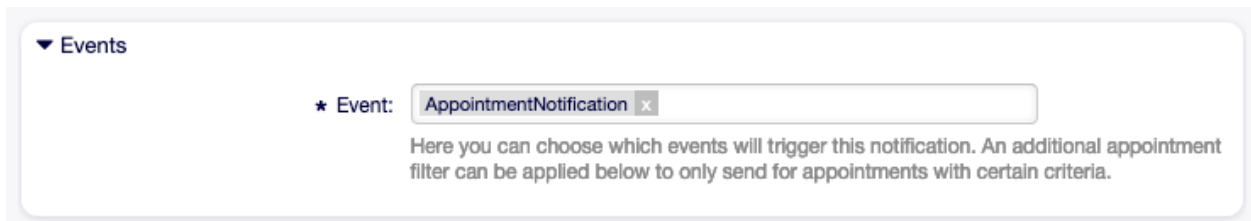
Appointment Events

Fig. 1.58: Appointment Notification Settings - Events

Event * Here you can choose which events will trigger this notification. An additional appointment filter can be applied below to only send for appointments with certain criteria.

Possible events are:

AppointmentCreate Executed after an appointment has been created.

AppointmentUpdate Executed after an appointment has been updated.

AppointmentDelete Executed after an appointment has been deleted.

AppointmentNotification This is a special appointment event that will be executed by the OTOBO daemon in time. If an appointment contains a date/time value for notifications, as already described in this documentation, and such a notification date is reached, the OTOBO daemon will execute this kind of event for every related appointment separately.

CalendarCreate Executed after a calendar has been created.

CalendarUpdate Executed after a calendar has been updated.

Appointment Filter

This widget can optionally be used to narrow the list of appointments by matching configured values:

Calendar Select which calendar the related appointment needs to be part of.

Fig. 1.59: Appointment Notification Settings - Appointment Filter

Title Filter for a part or complete title of the appointment.

Location Filter for a part or complete location of the appointment.

Resource Choose from a list of teams or resources assigned to the appointments.

Appointment Notification Recipients

Fig. 1.60: Appointment Notification Settings - Recipients

Send to Select which agents should receive the notifications. Possible values are:

- Agent (resources), who are selected within the appointment
- All agents with (at least) read permission for the appointment (calendar)
- All agents with write permission for the appointment (calendar)

Send to these agents One or more agents can be selected who should receive the notifications.

Send to all group members (agents only) One or more groups can be selected whom agents should receive the notifications.

Send to all role members One or more roles can be selected whom agents should receive the notifications.

Send on out of office If this option is checked, the notification will be sent even if the agent is currently out of office.

Once per day Notify users just once per day about a single appointment using a selected transport. If this is the first notification about an appointment, then the notification will be sent. If a notification was already sent before and this option is checked, the OTOBO daemon will check the time the last notification was sent. If there was no notification sent in the last 24 hours, the notification will be sent again.

Appointment Notification Methods

Fig. 1.61: Appointment Notification Settings - Notification Methods

Enable this notification method Enable or disable this notification method. A notification method can be email, web view or SMS.

Active by default in agent preferences This is the default value for assigned recipient agents who didn't make a choice for this notification in their preferences yet. If the box is enabled, the notification will be sent to such agents.

Additional recipient email addresses Additional recipients can be added here. Use comma or semi-

---

colon to separate the email addresses.

Article visible for customer An article will be created if the notification is sent to the customer or an additional email address.

Email template Select which email template should be used for the notification.

Note: Additional email templates can be added by placing a `.tt` file into the folder `<OTOBO_Home>/Kernel/Output/HTML/Templates/Standard/NotificationEvent/Email/` . See the existing email templates for an example.

Enable email security Checking this option will encrypt the notification email.

Note: To use this feature, PGP Keys or S/MIME Certificates need to be enabled.

Email security level If Enable email security is checked, then this setting is activated. The following options are available:

PGP sign only Sign only the notification email with PGP key. If no PGP keys have been added to the system, this option is not visible.

PGP encrypt only Encrypt only the notification email with PGP key. If no PGP keys have been added to the system, this option is not visible.

PGP sign and encrypt Sign and encrypt the notification email with PGP key. If no PGP keys have been added to the system, this option is not visible.

SMIME sign only Sign only the notification email with S/MIME certificate. If no S/MIME certificates have been added to the system, this option is not visible.

SMIME encrypt only Encrypt only the notification email with S/MIME certificate. If no S/MIME certificates have been added to the system, this option is not visible.

SMIME sign and encrypt Sign and encrypt the notification email with S/MIME certificate. If no S/MIME certificates have been added to the system, this option is not visible.

Note: To use this feature, PGP Keys or S/MIME Certificates need to be enabled.

If signing key/certificate is missing Select the method, that should be used if signing key or certificate is missing.

If encryption key/certificate is missing: Select the method, that should be used if encryption key or certificate is missing.

Appointment Notification Text

The main content of a notification can be added for each languages with localized subject and body text. It is also possible to define static text content mixed with OTOBO smart tags.

Subject * The localized subject for a specific language.

Text * The localized body text for a specific language.

Add new notification language Select which languages should be added to create localized notifications. The language of the customer or agent will be used as found in the customer and agent
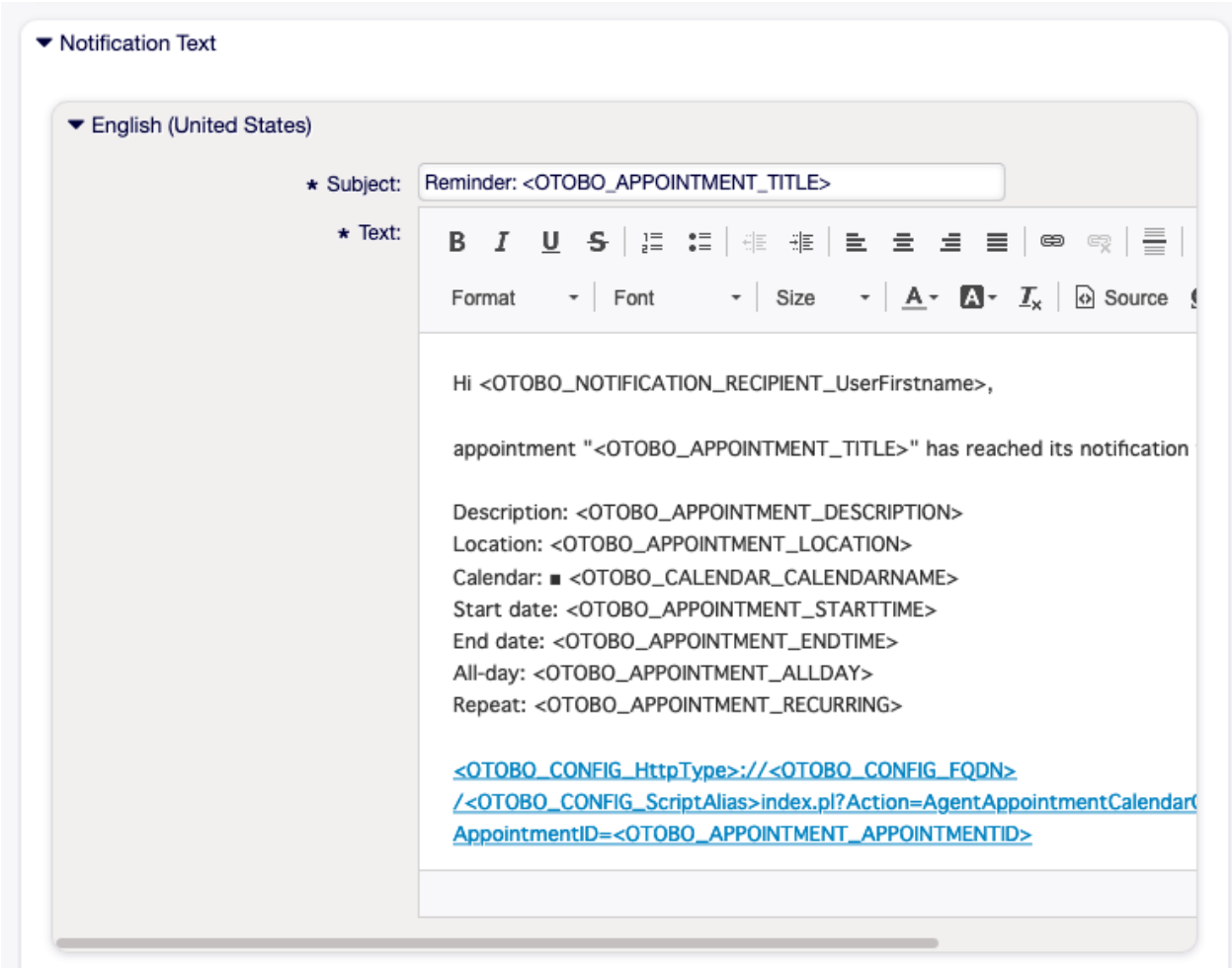
Fig. 1.62: Appointment Notification Settings - Notification Text

preferences. Secondarily, the system default language will be chosen. The fall back will always be English.

> **Warning:** Deleting a language in `DefaultUsedLanguages` setting that already has a notification text here will make the notification text unusable. If a language is not present or enabled on the system, the corresponding notification text could be deleted if it is not needed anymore.

Appointment Notification Variables

Using variables in the text makes it possible to personalize messages. Variables, known as OTOBO tags, are replaced by OTOBO when generating the message. Find a list of available tags stems for this resource at the bottom of both add and edit screens.

Fig. 1.63: Appointment Notification Variables

For example, the variable `<OTOBO_APPOINTMENT_TITLE[20]>` expands to the first 20 characters of the title allowing a template to include something like the following.

```
Title: <OTOBO_APPOINTMENT_TITLE[20]>
```

This tag expands, for example to:

```
Title: Daily meeting in the…
```

1.2.3 Communication Log

Managers, leaders, team leads, and system administrators may need to track past communication to follow up specific messages. In some cases, issues arise, and a target recipient did not receive a message. Without access to the mail server logs, tracking the communication is difficult.

OTOBO introduces the Communication Log module. It's designed to track the communication: building and spooling the mail and the connection between client and server.

Use this screen to inspect the internal logs about communication handling. The communication log overview screen is available in the Communication Log module of the Communication & Notifications group.
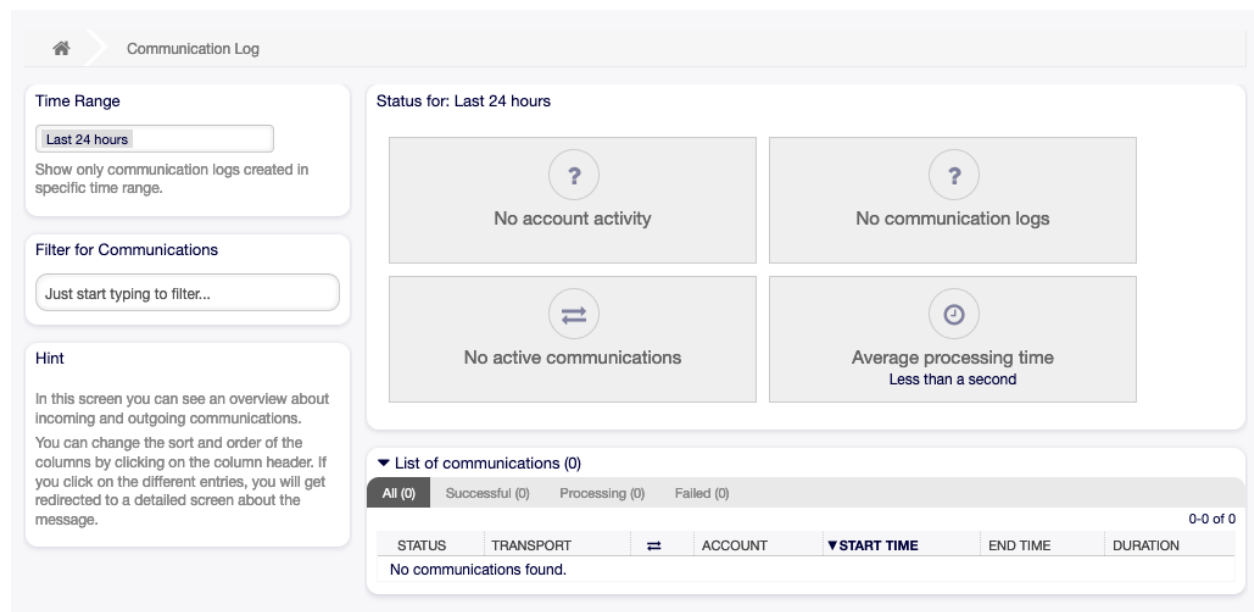

Fig. 1.64: Communication Log Overview Screen

Communication Log Overview

The communication log overview page is a dashboard-like screen with several metrics indicating the overall health of the system, depending on filtered communications.

Account status This widget will signal if you have any issues with configured accounts used for fetching or sending messages.

Communication status This widget will notify you if there are any errors with either account connections or message processing.

Communication state This widget will display if there are any active communications currently in the system.

Average processing time This is a cumulative time statistic that is needed to complete a communication.

STATUS	TRANSPORT	⇄	ACCOUNT	▼START TIME	END TIME
✖	Email	←	mail.example.com / test (IMAP)	11/24/2020 14:20:16 (Europe/Berlin)	11/24/2020 14:20:21 (Europe/
✖	Email	←	mail.example.com / test (IMAP)	11/24/2020 14:10:06 (Europe/Berlin)	11/24/2020 14:10:12 (Europe/
✖	Email	←	mail.example.com / test (IMAP)	11/24/2020 14:00:17 (Europe/Berlin)	11/24/2020 14:00:23 (Europe/
✖	Email	←	mail.example.com / test (IMAP)	11/24/2020 13:50:02 (Europe/Berlin)	11/24/2020 13:50:08 (Europe/
✖	Email	←	mail.example.com / test (IMAP)	11/24/2020 13:40:12 (Europe/Berlin)	11/24/2020 13:40:18 (Europe/
✖	Email	←	mail.example.com / test (IMAP)	11/24/2020 13:30:03 (Europe/Berlin)	11/24/2020 13:30:08 (Europe/

Status for: Last 1 hour

Failing accounts
Number of accounts with problems: 2

Failing communications
Number of reported problems: 6

No active communications

Average processing time
4 seconds

▼ List of communications (12)

All (12)   Successful (6)   Processing (0)   Failed (6)
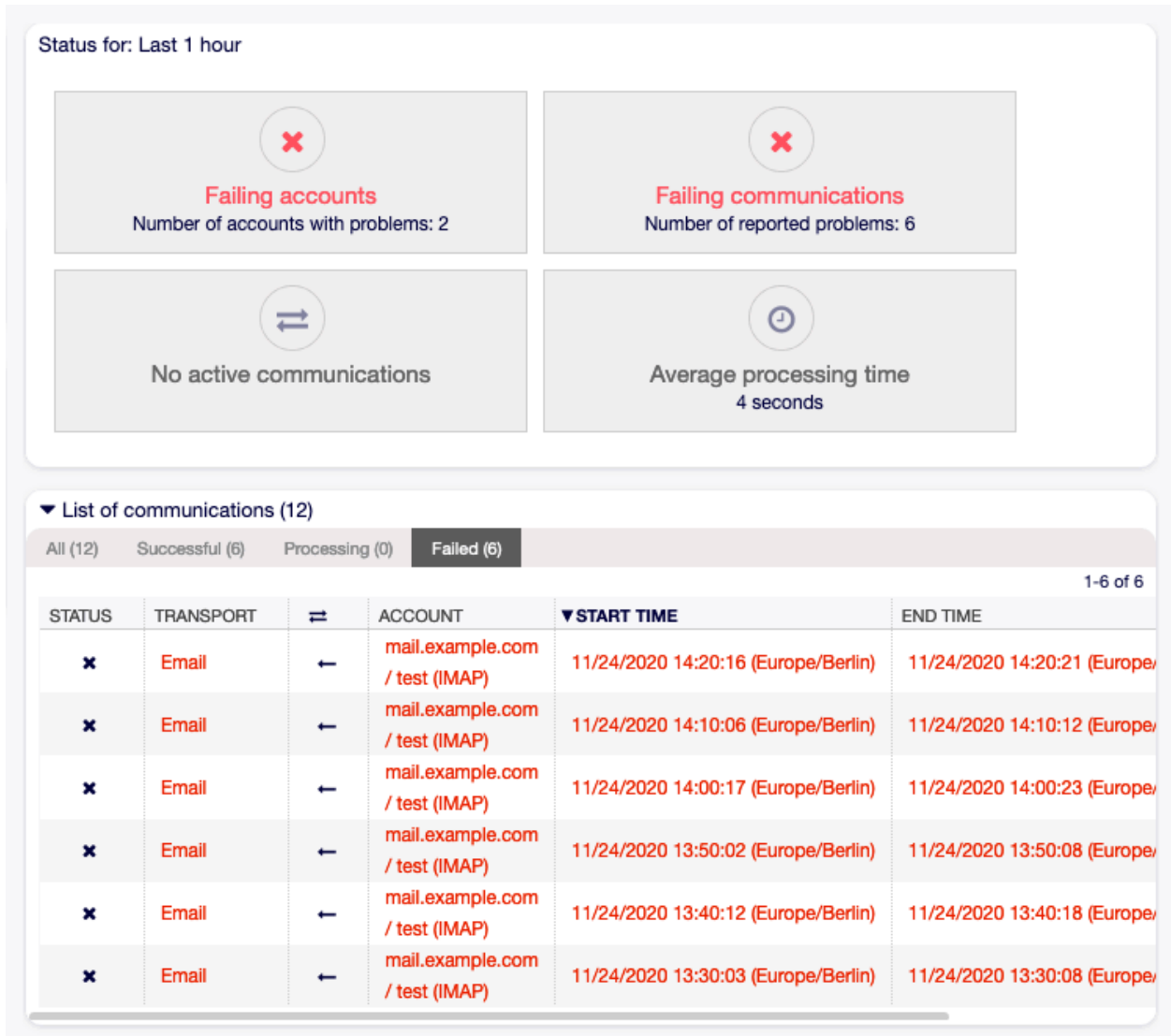
1-6 of 6

Fig. 1.65: Account Status Screen

You can select the time range in the left sidebar in order to filter communications depending on their creation time. In addition to this, you can also dynamically filter for any keywords, state of the communication, and you can sort the overview table by all columns.

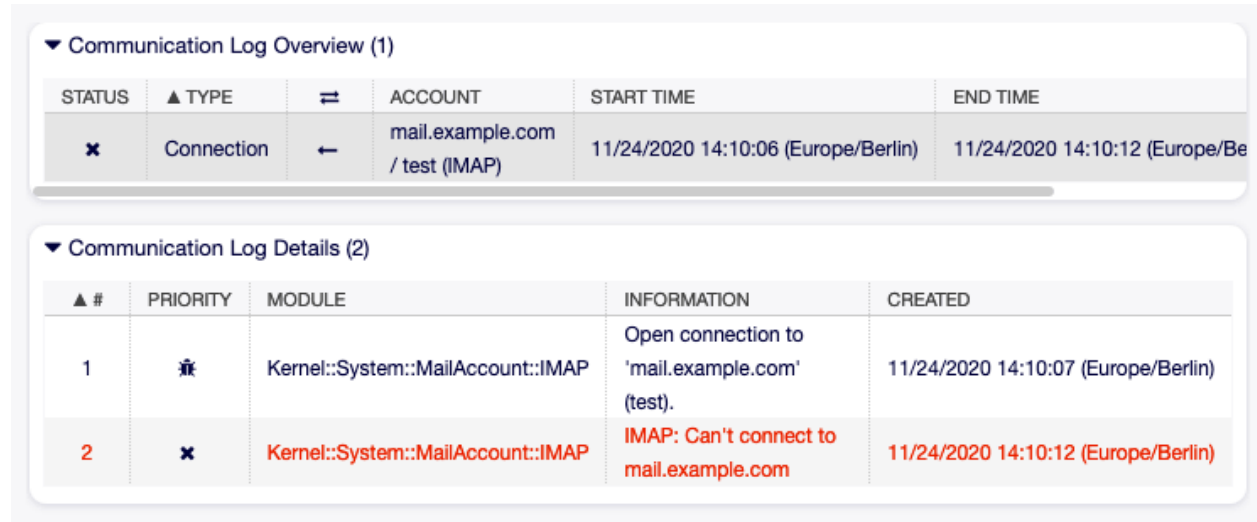If you click on a communication row in any table, you will be presented with a detailed view screen.

Fig. 1.66: Communication Log Detailed View Screen

Every communication can contain one or more logs, which can be of Connection or Message type.

Connection This type of logs will contain any log messages coming from the modules responsible for connecting to your accounts and fetching/receiving messages.

Message This type of logs will contain any log messages related to specific message processing. Any module working on message themselves can log their actions in this log, giving you a clear overview of what's going on.

You can filter log entries based on their priority, by choosing desired priority in the left sidebar. Log level rules apply: by selecting a specific priority, you will get log entries that have that priority set and higher, with Error being the highest.

1.2.4 Email Addresses

The main channel of communication with the customers is often email. An organization consists of multiple departments or teams. Email addresses differ for each group which is servicing your customers. You may have the following:

```
support@example.org
hr@exapmle.org
sales@example.org
```

These addresses are just some examples, and you may have many more. Use these channels to receive and send messages, and in mail clients, one can often send with the wrong address.

OTOBO manages as many email addresses for your teams as needed. All your email addresses, whether for sending or receiving, are kept and configured nicely in one place. In the Queue Settings,

the correct address is always chosen preventing that someone working in multiple roles sends an email out with the wrong account.

To enable OTOBO to send emails, you need a valid email address to be used by the system. OTOBO is capable of working with multiple email addresses, since many support installations need to use more than one. A queue can be linked to many email addresses, and vice versa. The address used for outgoing messages from a queue can be set when the queue is created.

Use this screen to add system email addresses to the system. An email address is already added to the system at installation time of OTOBO. The email address management screen is available in the Email Addresses module of the Communication & Notifications group.
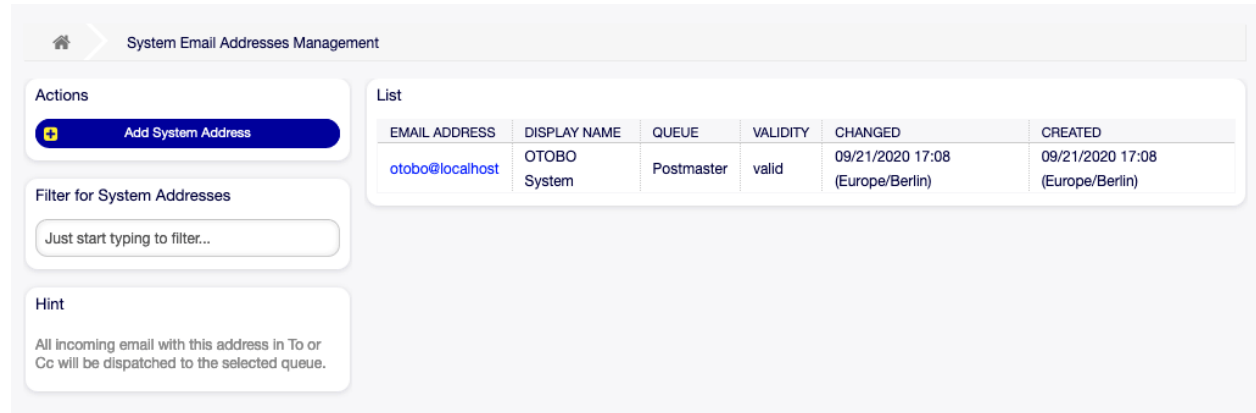

Fig. 1.67: Email Address Management Screen

Manage Email Addresses

To add an email address:

1. Click on the Add System Address button in the left sidebar.
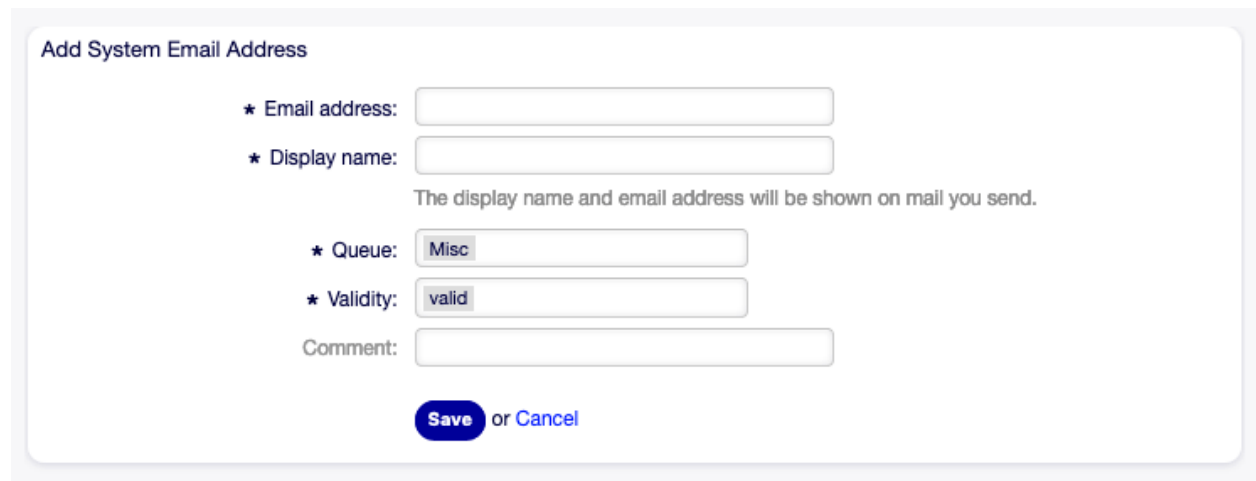2. Fill in the required fields.
3. Click on the Save button.

Fig. 1.68: Email Address Add Screen

> **Warning:** Email addresses can not be deleted from the system. They can only be deactivated by setting the Validity option to invalid or invalid-temporarily.

Note: Once an email address is added and set to valid, OTOBO cannot send an email to this address. This prevents loopbacks which could crash your system. If you need to transfer information between departments please use the ticket split option in the article menu. This will allow you to create a new ticket to another team for assigning a task, for example.

Fig. 1.69: Article Menu

To edit an email address:

1. Click on an email address in the list of email addresses.

2. Modify the fields.

3. Click on the Save or Save and finish button.

Fig. 1.70: Email Address Edit Screen

Note: If several email addresses are added to the system, use the filter box to find a particular email address by just typing the name to filter.

Email Address Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Email address * The email address to be added.

Display name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the sender information of the article.

From:	OTOBO Feedback <marketing@otobo.com>
To:	Your OTOBO System <otobo@localhost>
Subject:	Welcome to OTOBO!

Fig. 1.71: Sender Information

Queue * The queue, to which the email address will be added as default email address.

> **Note:** This setting will apply if the email is distributed via the recipient address. This setting can be overridden by Postmaster Filters or in the Mail Account Settings when Dispatching by selected Queue is chosen.

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

> **Note:** An email address can only be set to invalid or invalid-temporarily, if it is not assigned to any queue.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

1.2.5 PGP Keys

Secure communications protect your customers and you. In the GDPR encryption is explicitly mentioned as one of the security and personal data protection measures in a few articles. Although under the GDPR encryption is not mandatory, it is indeed essential in some areas.

OTOBO empowers you to encrypt communications where needed by means of S/MIME Certificates or PGP Keys.

> **Note:** Setup of services and software required for encryption are not covered here because of independence to this software.

Use this screen to add PGP keys to the system. The PGP management screen is available in the PGP Keys module of the Communication & Notifications group.

Manage PGP Keys

> **Note:** To be able to use PGP keys in OTOBO, you have to activate its setting first.
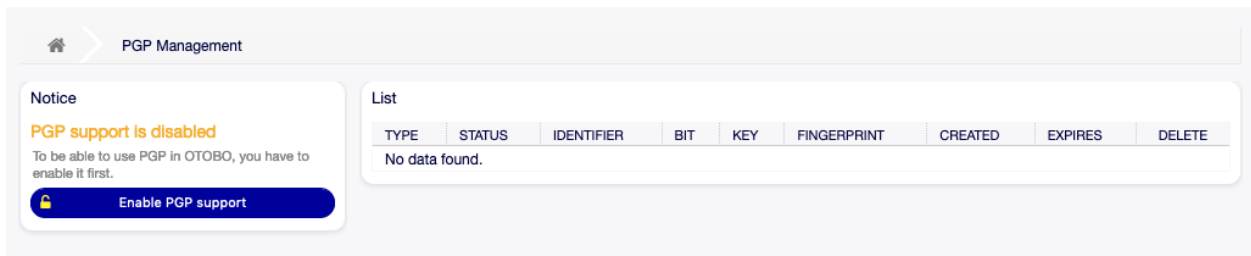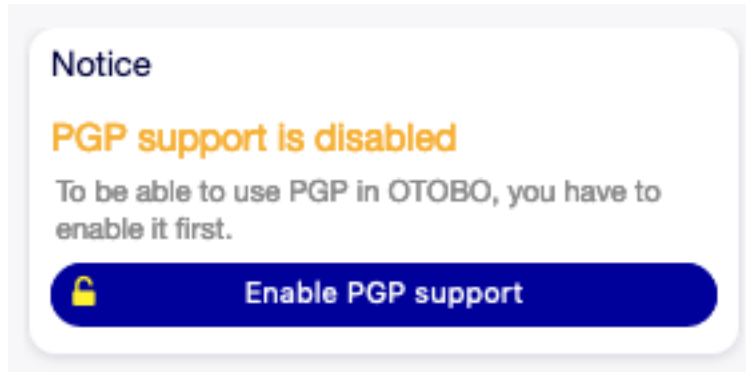
Fig. 1.72: PGP Management Screen

Fig. 1.73: Enable PGP Support

To add a PGP key:

1. Click on the Add PGP Key button in the left sidebar.
2. Click on Browse… button to open the file dialog.
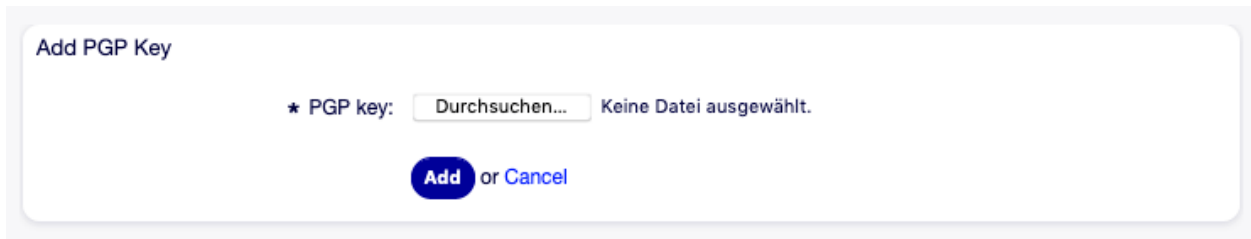3. Select a PGP key from the file system.
4. Click on the Add button.

Fig. 1.74: Add PGP Key Screen

To delete a PGP key:

1. Click on the trash icon in the list of PGP keys.
2. Click on the Confirm button.

Note: If several PGP keys are added to the system, use the search box to find a particular PGP key.
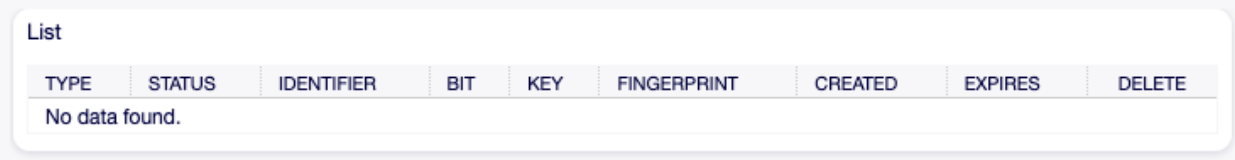
Fig. 1.75: Delete PGP Key Screen

PGP Configuration Options

`Core → Crypt → PGP`

1.2.6 Postmaster Filters

Pre-sorting standard mail done in a mail room takes care that not every piece of mail sent to the office goes to the same group of people. After a second look at the envelope, rerouting occurs where needed.

OTOBO uses so-called postmaster filters to read the emails envelope and take further action. Depending upon, for example, a subject or sender, an email bound for the service desk could land in a sub-queue or be redirected to a completely different team to create transparency and give your customer the fastest service possible.

Use this screen to add postmaster filters to the system. The postmaster filter management screen is available in the PostMaster Filters module of the Communication & Notifications group.
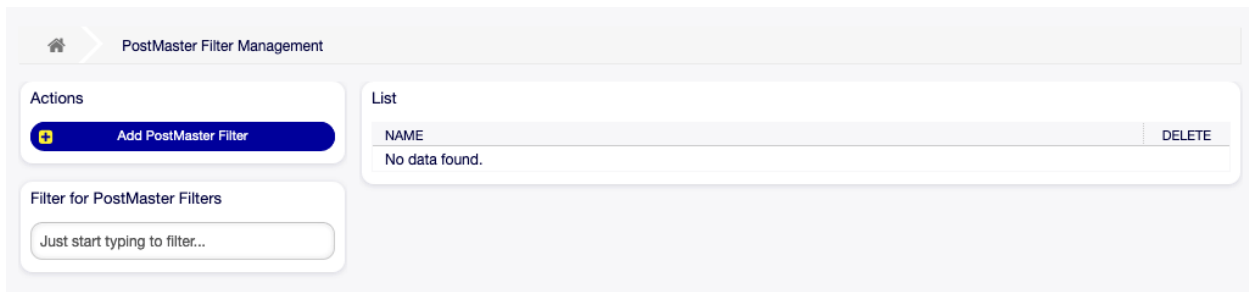

Fig. 1.76: Postmaster Filter Management Screen

Manage Postmaster Filters

Note: When adding or editing a postmaster filter, please keep in mind that they are evaluated in ASCIIbetical order by name.

To add a postmaster filter:

1. Click on the Add PostMaster Filter button in the left sidebar.
2. Fill in the required fields.
3. Click on the Save button.

To edit a postmaster filter:

1. Click on a postmaster filter in the list of postmaster filters.

---

2. Modify the fields.

3. Click on the Save or Save and finish button.

To delete a postmaster filter:

1. Click on the trash icon in the list of postmaster filters.

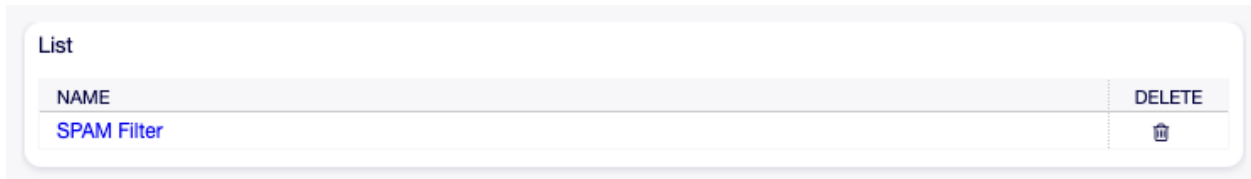2. Click on the Confirm button.

Fig. 1.77: Delete Postmaster Filter Screen

Note: If several postmaster filters are added to the system, a filter box is useful to find a particular postmaster filter by just typing to filter.

Postmaster Filter Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.
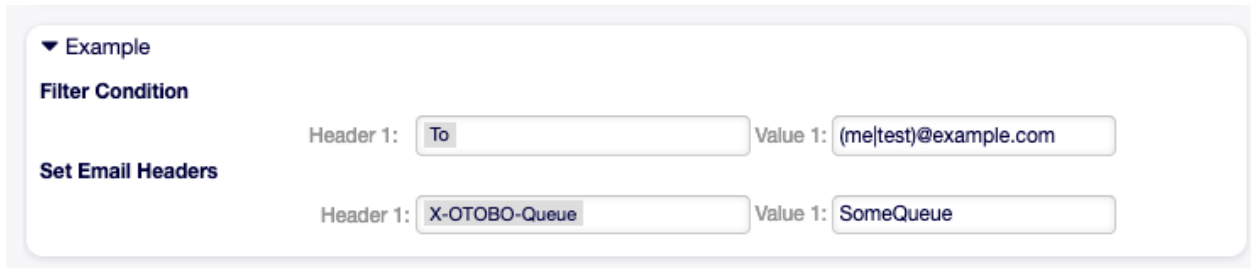

Fig. 1.78: Postmaster Filter Settings Example

Basic Postmaster Filter Settings

Fig. 1.79: Postmaster Filter Settings - Basic

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Note: When adding or editing one of the postmaster filters, remember multiple filters may apply to a single mail. Rules are executed and sorted by the ASCII value of the names. Based on the sorted order in the overview, they are applied from top to bottom. Look at the ASCII table to see how to sort your names based on the ASCIIbetical order.

Stop after match * Postmaster filters are evaluated in ASCIIbetical order. This setting defines the evaluation of the subsequent postmaster filters.

No All postmaster filters are executed.

Yes The current postmaster filter is still evaluated, but evaluation of the remaining filters is canceled.

Filter Condition

Fig. 1.80: Postmaster Filter Settings - Filter Condition

A postmaster filter consists of one or more conditions that must be met in order for the defined actions to be executed on the email. Filter conditions can be defined for specific mail header entries or for strings in the mail body.

Search header field … for value Select a mail header or an `X-OTOBO` header from the first drop-down list, and enter a value as search term for the selected mail header to the second field. Even regular expressions can be used for extended pattern matching.

A list of mail header entries can be found in RFC5322. It is also possible to define `X-OTOBO` headers as filter condition. The different `X-OTOBO` headers and their meaning are the following:

X-OTOBO-AttachmentCount This contains as value the number of attachments which are contained in the email (e.g. 0 for mails without attachments).

X-OTOBO-AttachmentExists Depending on whether attachments are included in the email this `X-OTOBO` header is set to `yes`, or it has a `no` value if no attachments are included.

X-OTOBO-BodyDecrypted If the incoming mail was encrypted, it is possible to add a search term to look for the body of the incoming encrypted mail.

X-OTOBO-CustomerNo Set the customer ID for the ticket.

X-OTOBO-CustomerUser Set the customer user for the ticket.

X-OTOBO-DynamicField-<DynamicFieldName> Saves an additional information value for the ticket on <DynamicFieldName> dynamic field. The possible values depend on dynamic field configuration (e.g. text: Notebook, date: 2010-11-20 00:00:00, integer: 1).

X-OTOBO-FollowUp-* These headers are the same as the ones without the `FollowUp` prefix, but these headers are applied only for follow-up mails.

X-OTOBO-FollowUp-State-Keep If set to 1, the incoming follow-up message will not change the ticket state. For this purpose the header can be customized in the system configuration using option `KeepStateHeader`.

X-OTOBO-Ignore If set to `Yes` or `True`, the incoming message will completely be ignored and never delivered to the system.

X-OTOBO-IsVisibleForCustomer Controls if the article is shown to customer users. Possible values are 0 or 1.

X-OTOBO-Lock Set the lock state of a ticket. Possible values are `locked` or `unlocked`.

X-OTOBO-Loop If set to `Yes` or `True`, no auto answer is delivered to the sender of the message (mail loop protection).

X-OTOBO-Owner Set the agent as owner for the ticket.

X-OTOBO-OwnerID Set the agent ID as owner for the ticket.

X-OTOBO-Priority Set the priority for the ticket.

X-OTOBO-Queue Set the queue where the ticket shall be sorted. If this is set, all other filter rules that try to sort a ticket into a specific queue are ignored. If you use a sub-queue, specify it as Parent::Sub.

X-OTOBO-Responsible Set the agent as responsible for the ticket.

X-OTOBO-ResponsibleID Set the agent ID as responsible for the ticket.

X-OTOBO-SenderType Set the sender type for the ticket. Possible values are `agent`, `system` or `customer`.

X-OTOBO-Service Set the service for the ticket. If you use a sub-service, specify it as Parent::Sub.

X-OTOBO-SLA Set the service level agreement for the ticket.

X-OTOBO-State Set the state for the ticket.

X-OTOBO-State-PendingTime Set the pending time for the ticket (you should sent a pending state via `X-OTOBO-State`). You can specify absolute dates like 2010-11-20 00:00:00 or relative dates, based on the arrival time of the email. Use the form `+ $Number $Unit`, where `$Unit` can be s (seconds), m (minutes), h (hours) or d (days). Only one unit can be specified. Examples of valid settings: +50s (pending in 50 seconds), +30m (30 minutes), +12d (12 days).

> **Note:** Settings like +1d 12h are not possible. You can specify +36h instead.

X-OTOBO-Title Set the title for the ticket.

X-OTOBO-Type Set the type for the ticket.

> **Warning:** These headers must be manually injected into the mail by means not provided for by OTOBO. OTOBO only accepts `X-OTOBO` headers from trusted sources.

See also:

The Mail Account Settings defines the trust level.

Negate If checked, the condition will use the negate search term.

Set Email Headers

Fig. 1.81: Postmaster Filter Settings - Set Email Headers

In this section you can choose the actions that are triggered if the filter rules match.

Set email header … with value Select an `X-OTOBO` header from the first drop-down list, and add a value to the second field that should be set as value of the selected `X-OTOBO` header.

> **See also:**
>
> The `X-OTOBO` headers are already described above.

1.2.7 PostMaster Mail Accounts

Just as a company doesn't just have one department which receives traditional mail, your service desk will also serve multiple teams. Each team can have its physical email mailbox.

OTOBO eases setup for email mailboxes. OTOBO manages polling one or multiple email mailboxes of any internet standard type.

Use this screen to add mail accounts to the system. The mail account management screen is available in the PostMaster Mail Accounts module of the Communication & Notifications group.

Fig. 1.82: Mail Account Management Screen

> **Warning:** When fetching mail, OTOBO deletes the mail from the POP or IMAP server. There is no option to also keep a copy on the server. If you want to retain a copy on the server, you should create forwarding rules on your mail server. Please consult your mail server documentation for details.

Note: If you choose IMAP, you can specify a folder for collection. Selective dispatching of mails is then possible.

All data for the mail accounts are saved in the OTOBO database. The `bin/otobo.Console.pl Maint::PostMaster::MailAccountFetch` command uses the settings in the database and fetches the mail. You can execute it manually to check if all your mail settings are working properly.

On a default installation, the mail is fetched every 10 minutes when the OTOBO daemon is running.

Manage Mail Accounts

To add a mail account:

1. Click on the Add Mail Account button in the left sidebar.

2. Fill in the required fields.

3. Click on the Save button.

Fig. 1.83: Add Mail Account Screen

To edit a mail account:

1. Click on a mail account in the list of mail accounts.

---

2. Modify the fields.

3. Click on the Save or Save and finish button.

Fig. 1.84: Edit Mail Account Screen

To delete a mail account:

1. Click on the trash icon in the list of mail accounts.

2. Click on the Confirm button.

Fig. 1.85: Delete Mail Account Screen

Note: If several mail accounts are added to the system, a filter box is useful to find a particular mail account by just typing to filter.

Mail Account Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Type * There are different kind of protocols that are used for fetching mail. The following protocols are supported:

- IMAP
- IMAPS
- IMAPTLS
- POP3
- POP3S
- POP3TLS

Username * The username of the mail account.

Password * The password of the mail account.

Host * The host name of the mail account. Specify how the current article subject should be dealt with. The following methods are available:

IMAP Folder The folder in the mail account to be fetched. Other folders remain untouched.

Trusted * If Yes is selected, any `X-OTOBO` headers attached to an incoming message are evaluated and executed. Because the `X-OTOBO` header can execute some actions in the ticket system, you should set this option to Yes only for known senders.

> **See also:**
>
> The `X-OTOBO` headers are explained in the filter conditions of Postmaster Filters.

Dispatching * The distribution of incoming messages can be controlled if they need to be sorted by queue or by the content of the To: field.

> **Dispatching by email To: field** The system checks if a queue is linked with the address in the To: field of the incoming mail. You can link an address to a queue in the Email Addresses screen. If the address in the To: field is linked with a queue, the new message will be sorted into the linked queue. If no link is found between the address in the To: field and any queue, then the message flows into the Raw queue in the system, which is the postmaster default queue after a default installation.
>
> > **See also:**
> >
> > The postmaster default queue can be changed in system configuration setting `PostmasterDefaultQueue`.
>
> **Dispatching by selected Queue** All incoming messages will be sorted into the specified queue. The address where the mail was sent to is disregarded in this case.

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

POP3 and IMAP OAuth2 Authentification

Authentication via OAuth2 / OpenID Connect is possible by using an additional package in OTOBO.

Please install the package **MailAccount-OAuth2** in the OTOBO package management.

Note: After installing the package sometimes a restart of the OTOBO webserver or container is needed.

Azure Configuration

Go to https://portal.azure.com

In the next step switch to "Azure Active Directory" and add a new "Enterprise Application":
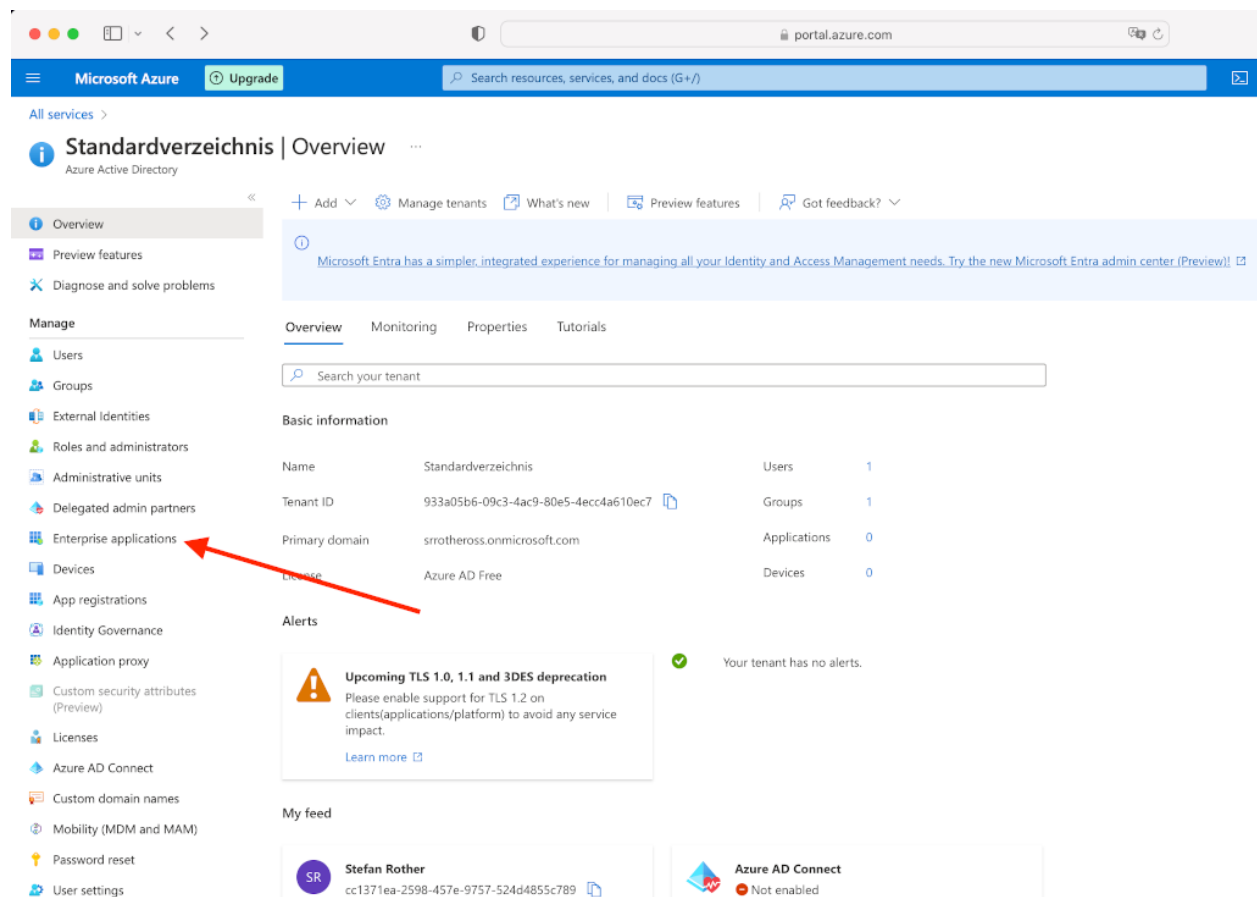

Create your own application

Assign a name to the app

The mailbox user must be assigned to the application. You will need the Application ID lateron in OTOBO (Attention, the application ID of the "Enterprise APP" may differ from that of the "Application Registration". In this case, please use the Application/Client ID of the registration.).

You will also need the domain's tenant ID*
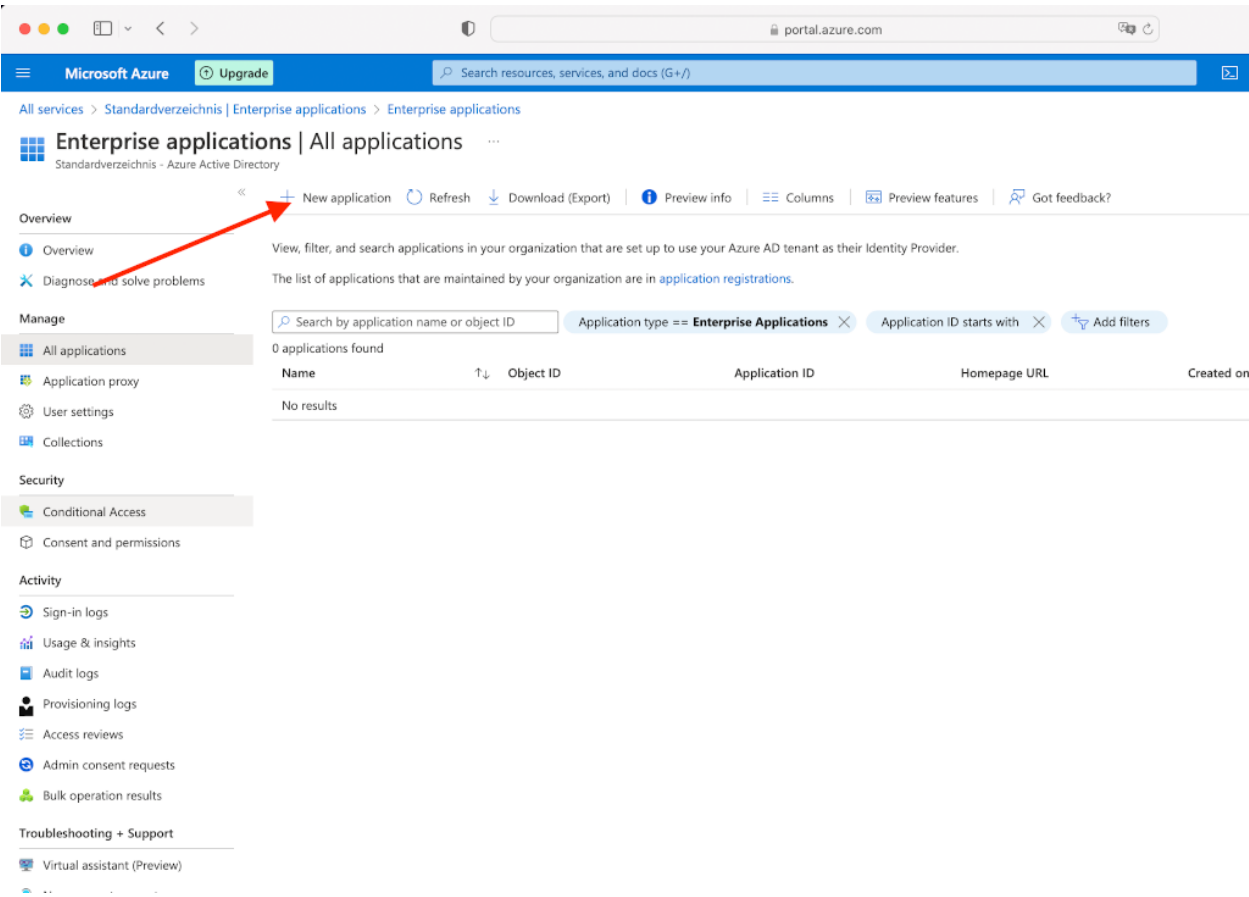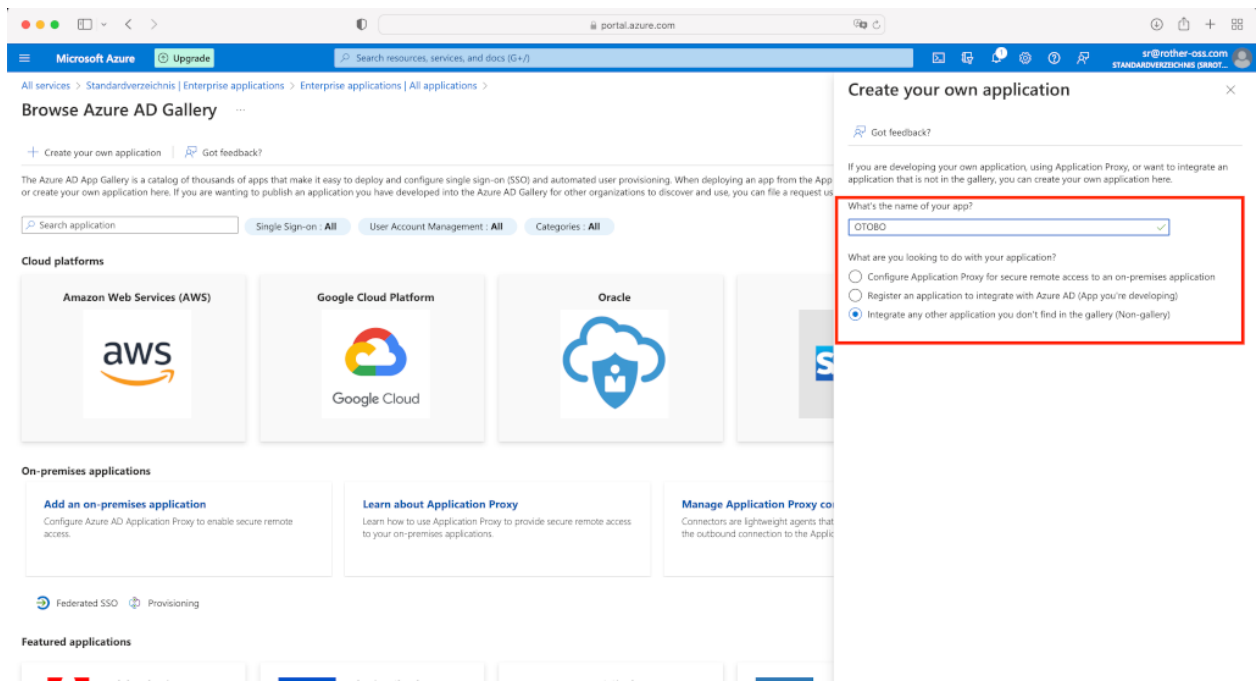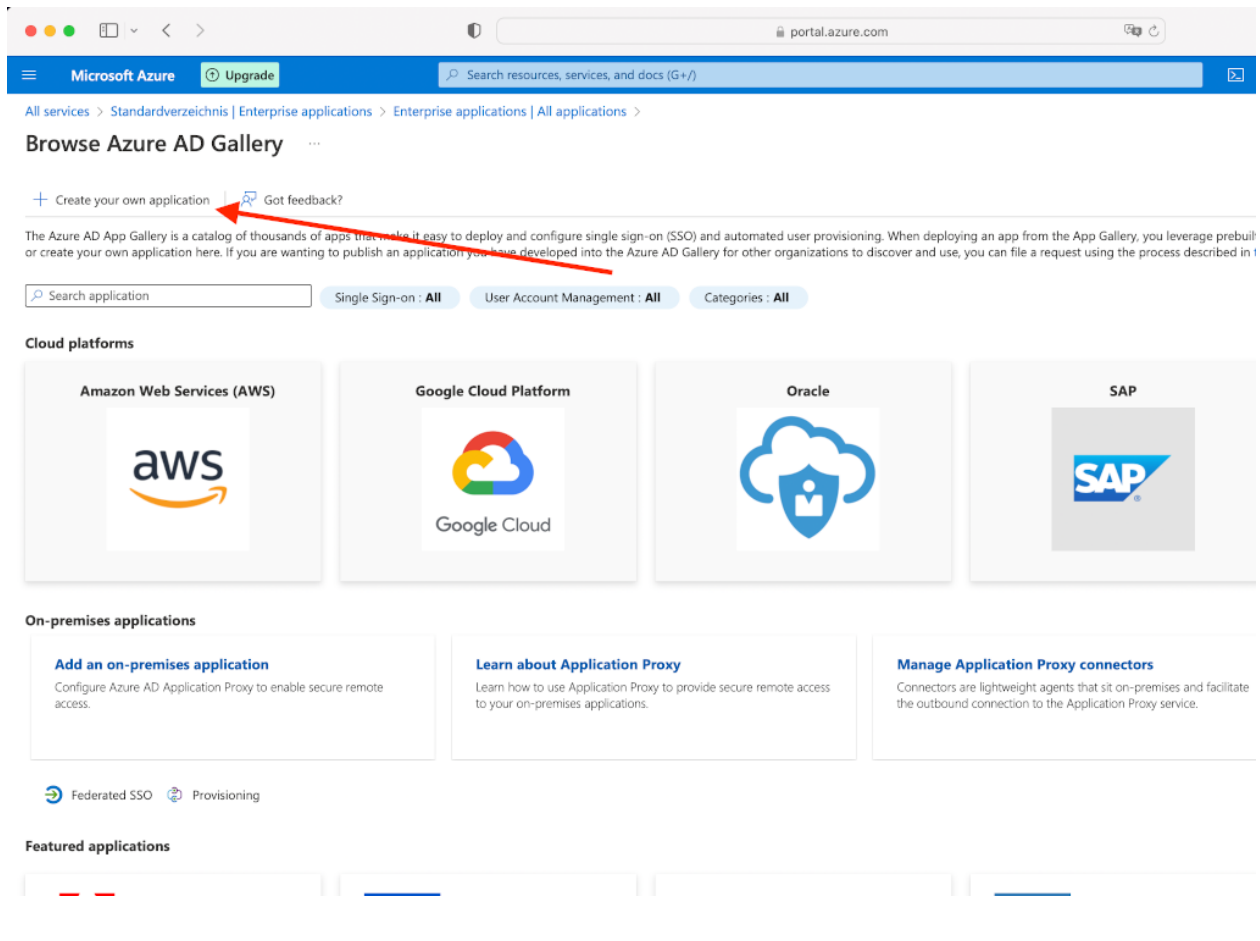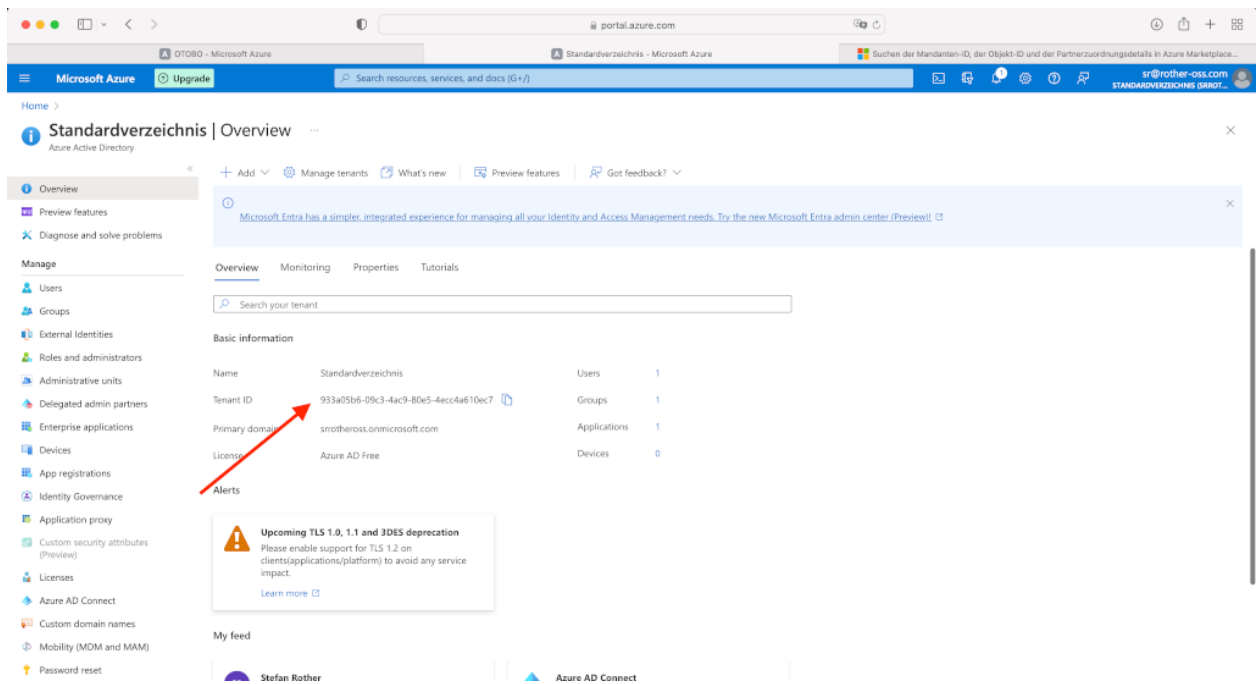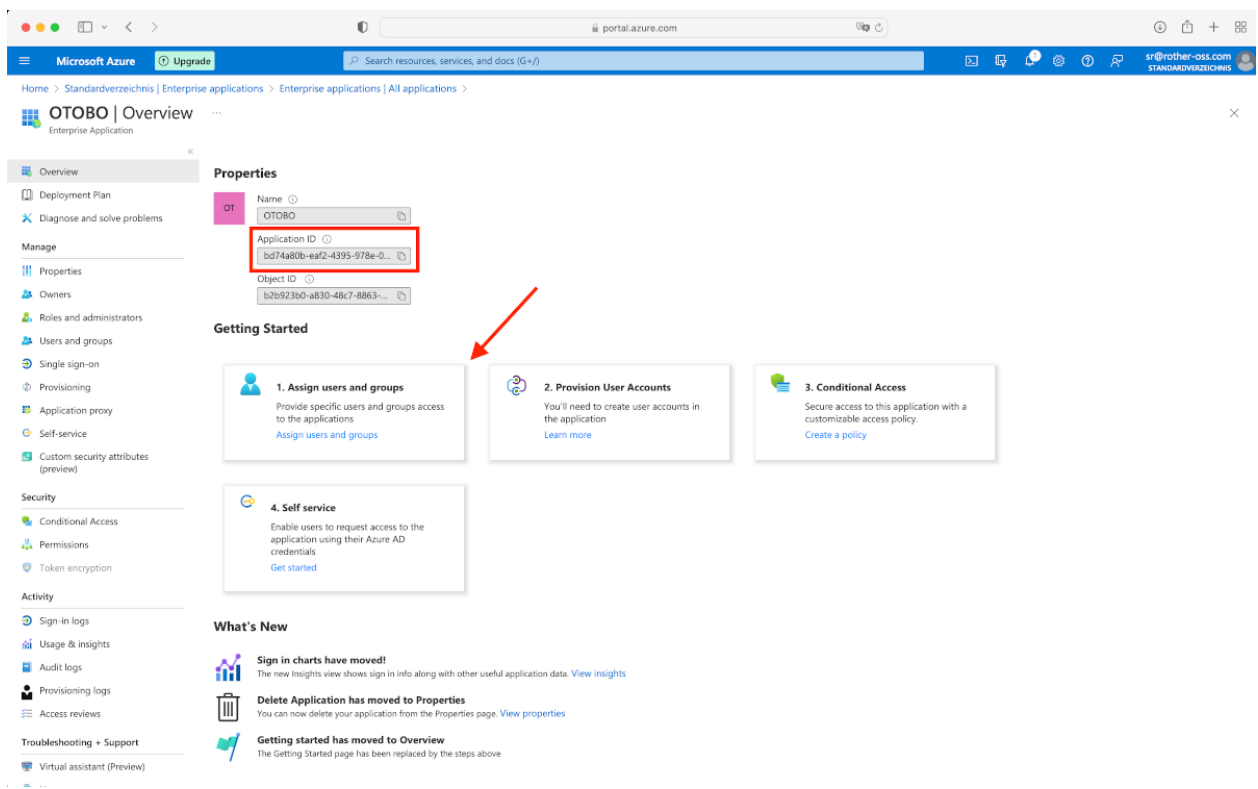
In the next step you have to add a new app in App registration.
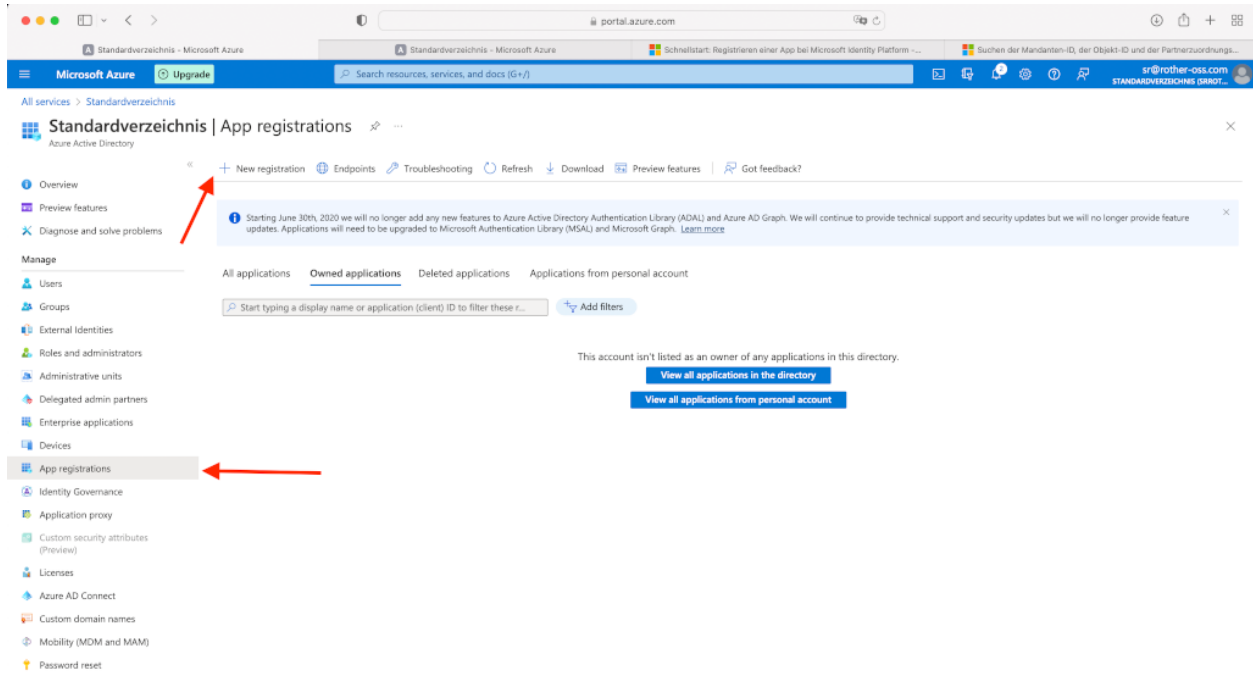
Create a Redirect URL of type Web and a secret client key.

Redirect URL = https://<OTOBO address>/otobo/index.pl?Action=AdminMailAccount

Switch to "API permissions" and add "IMAP.AccessAsUser.All" and "POP.AccessAsUser.All"

The Azure configuration is now complete. Please check whether port 143 and 993 are enabled.

OTOBO OAuth2 Configuration

Please switch to the `OTOBO Admin Interface -> System Configuration` now and enable the profile / option `OAuth2::MailAccount::Profiles###Custom1`.

In the next step, add the ApplicationID/ClientID and the Client secret (You need the value, not the secret id):

In the option `OAuth2::MailAccount::Providers###MicrosoftAzure`, please adjust the provider. For `AuthURL` and `TokenURL`, the path **"common"** must be replaced by the TenantID.

Now the desired profile can be selected under Admin -> PostMaster Mail Account.

When you save your settings, you will be redirected to the Active Directory login. If everything works as it is supposed to, you are redirected to the mail account overview after logging in and there is the corresponding mailbox. Of course, it says IMAPOAuth2 and not IMAPS.

Note: it is important that you log in with the account that owns the mailbox. Otherwise errors can occur. Sometimes it is helpful to use a private tab for this.
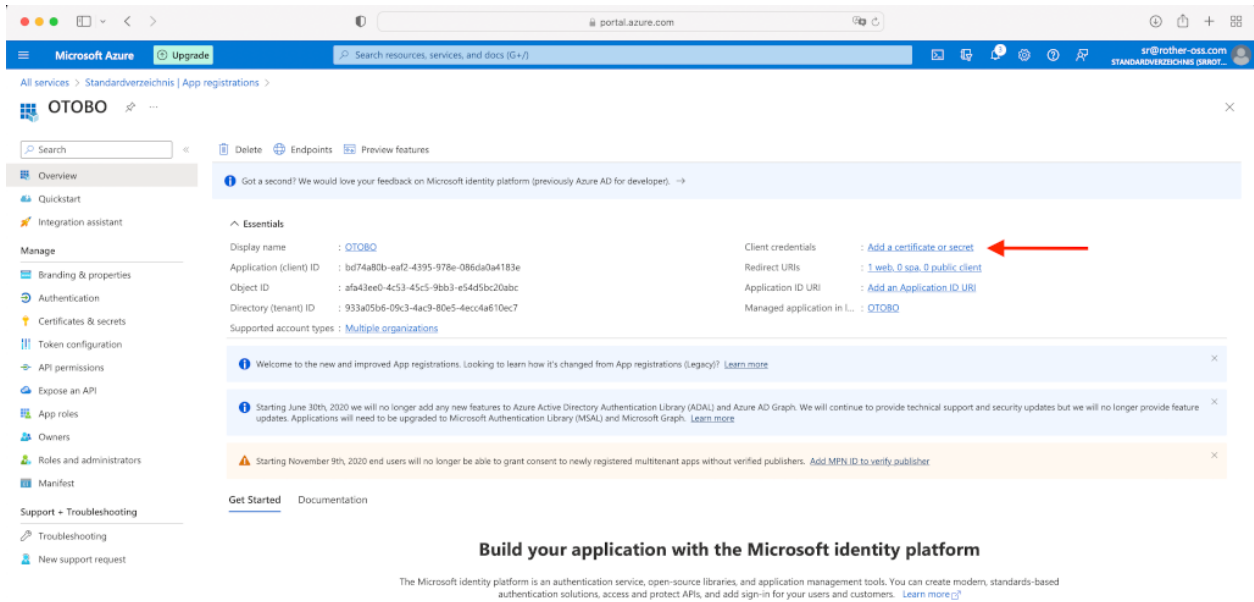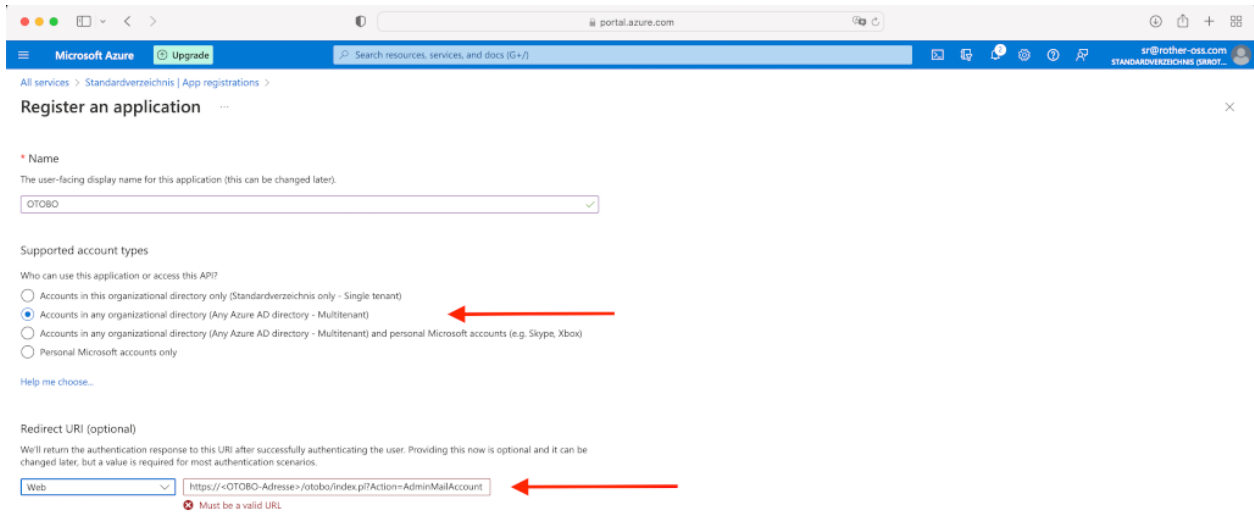
Fig. 1.86: Please add a new client secret and note the value (not the secret id) as we need it later. It will only appear during the creation and you will not be able to see it afterwards anymore. Apparently Microsoft only allows a time of validity for two years max.
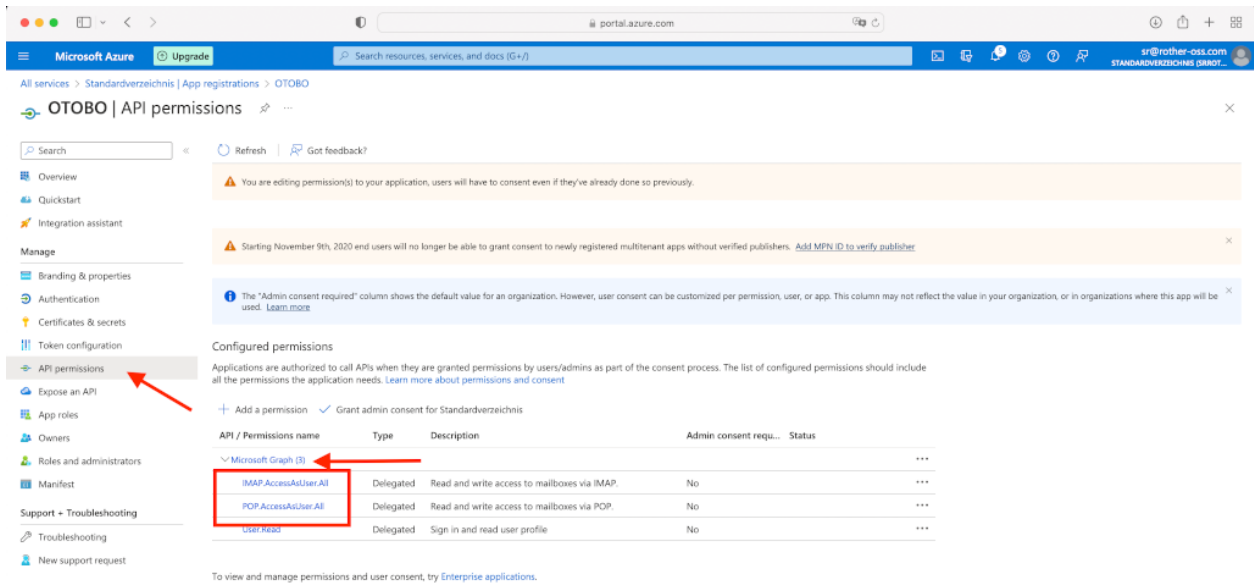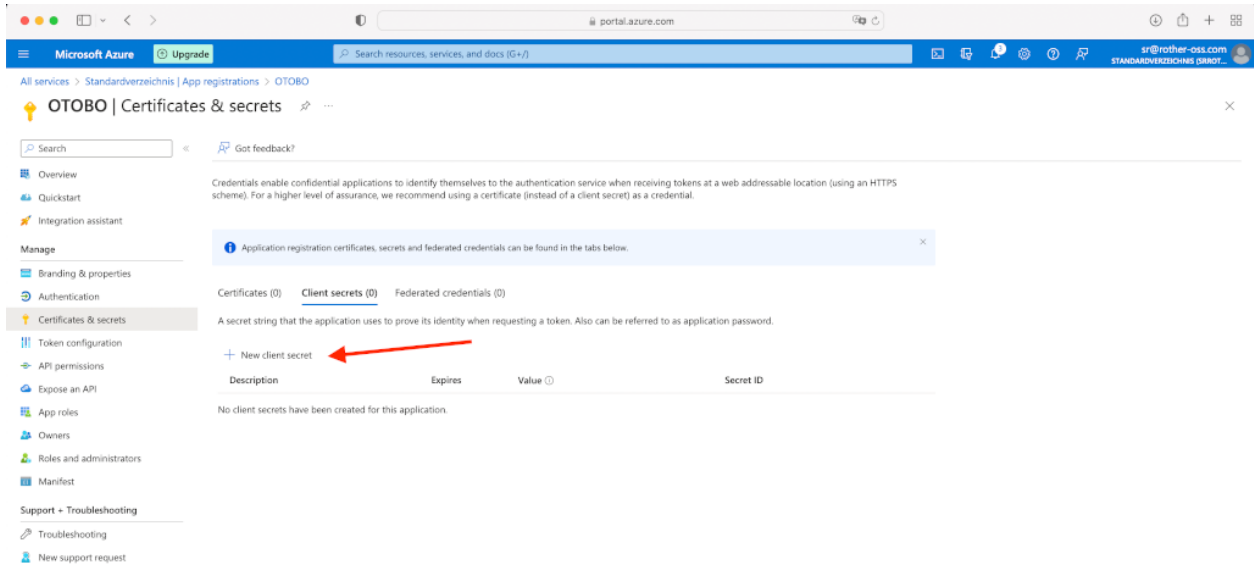
Fig. 1.87: Please click on "Add permission" and choose Microsoft Graph, then new delegated permissions in the bar on the right. If Microsoft Graph is no show up as like in the screenshot.

You can check on the console if fetching mails from the newly created account is successful:

```
otobo> bin/otobo.Console.pl Maint::PostMaster::MailAccountFetch
```

Or for a Docker installation:

```
docker_admin> docker exec -t -u otobo otobo_web_1 bin/otobo.Console.pl␣
→Maint::PostMaster::MailAccountFetch
```

Note: Add `--debug` for more verbosity

1.2.8 S/MIME Certificates

Faculty and staff have key roles safeguarding critical information by implementing information security policies, standards, and controls. Safe email communication is a vital part of protecting this communication.

OTOBO empowers you to encrypt communications where needed by means of S/MIME Certificates or PGP Keys.

Note: Setup of services and software required for encryption are not covered here because of independence to this software.

Use this screen to add S/MIME certificates to the system. The S/MIME management screen is available in the S/MIME Certificates module of the Communication & Notifications group.
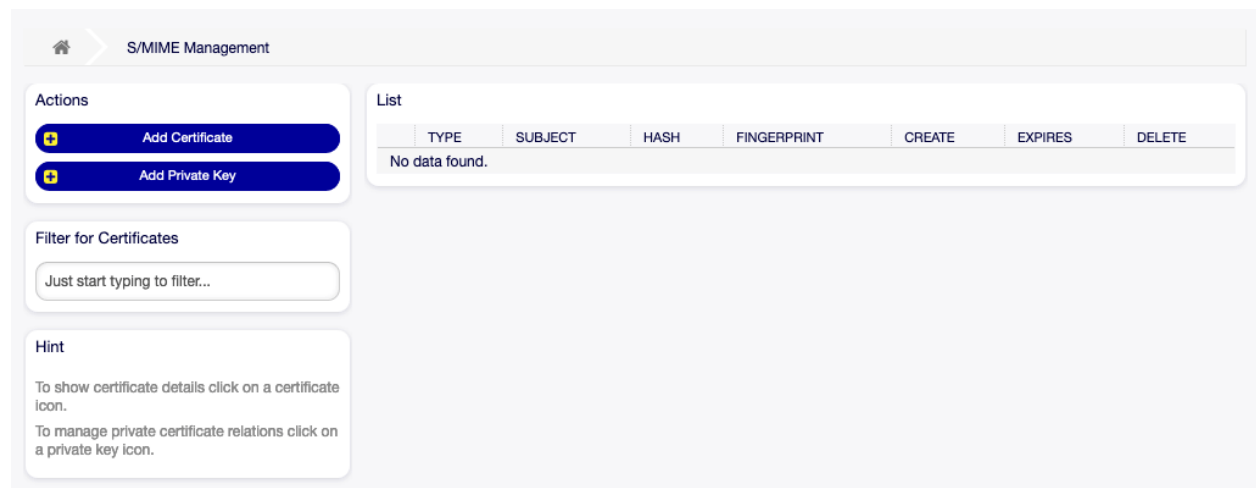

Fig. 1.88: S/MIME Management Screen

Manage S/MIME Certificates

Note: To be able to use S/MIME certificates in OTOBO, you have to activate its setting first.
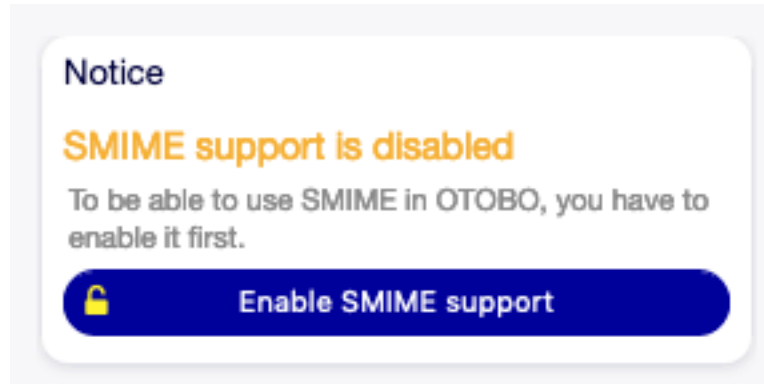
To add an S/MIME certificate:

Fig. 1.89: Enable S/MIME Support

1. Click on the Add Certificate button in the left sidebar.
2. Click on Browse… button to open the file dialog.
3. Select an S/MIME certificate from the file system.
4. Click on the Add button.
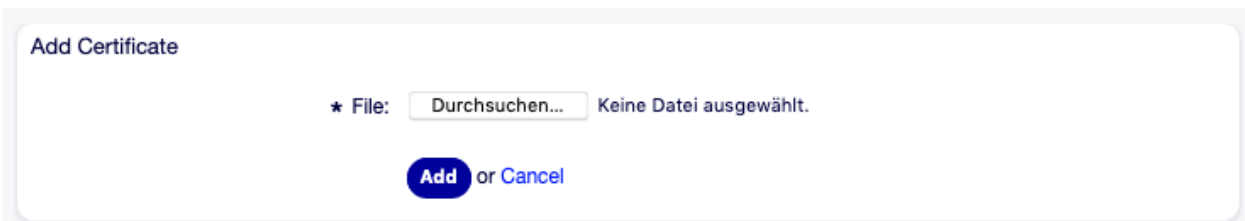
Fig. 1.90: Add S/MIME Certificate Screen

Note: Only non binary keys contain ASCII (Base64) armored data started with a ----- BEGIN CERTIFICATE ----- line can be uploaded which are most commonly `key.pem` or `root.crt`. Conversion of other formats like `cert.p7b` must be done using OpenSSL.

To add a private key:

1. Click on the Add Private Key button in the left sidebar.
2. Click on Browse… button to open the file dialog.
3. Select a private key from the file system.
4. Click on the Submit button.

To delete an S/MIME certificate:

1. Click on the trash icon in the list of S/MIME certificates.
2. Click on the Confirm button.

Note: If several S/MIME certificates are added to the system, use the filter box to find a particular S/MIME certificate by just typing the name to filter.
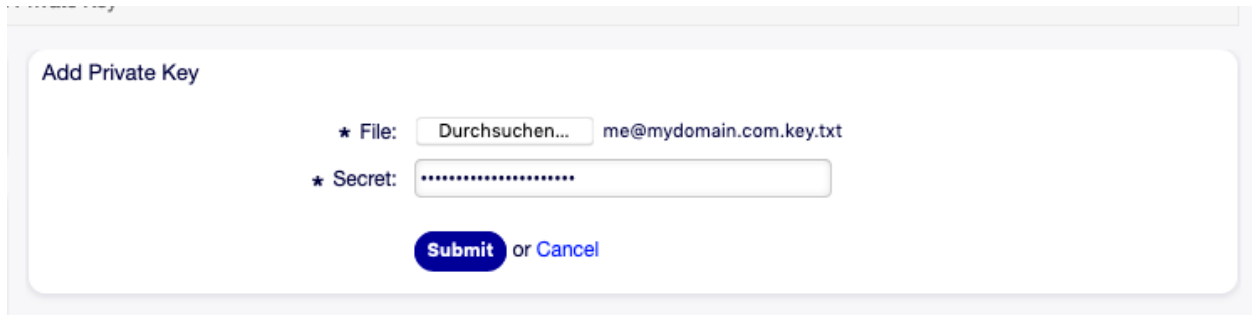
Fig. 1.91: Add S/MIME Private Key Screen

Fig. 1.92: Delete S/MIME Certificate Screen

S/MIME Configuration Options

`Core → Crypt → SMIME`

1.2.9 Ticket Notifications

Streamlining communication can save hours of labor and prevent mistakes. Sending certain messages at pre-defined stages of communication not only keeps the customer and agents informed about specific events, but it can also aid your agents by programmatically doing automated updates to the customer.

The flexible OTOBO is an industry leader in email communication and offers you complete control of notifications based on any event in your system.

Use this screen to add ticket notifications to the system. In a fresh OTOBO installation several ticket notifications are already added by default. The ticket notification management screen is available in the Ticket Notifications module of the Communication & Notifications group.

Manage Ticket Notifications

To add a ticket notification:

1. Click on the Add Notification button in the left sidebar.

2. Fill in the required fields as explained in Ticket Notification Settings.

3. Click on the Save button.

To edit a ticket notification:

1. Click on a ticket notification in the list of ticket notifications.

Fig. 1.93: Ticket Notification Management Screen

2. Modify the fields as explained in Ticket Notification Settings.

3. Click on the Save or Save and finish button.

To delete a ticket notification:

1. Click on the trash icon in the list of ticket notifications.

2. Click on the Confirm button.

NAME	COMMENT	VALIDITY	CHANGED	CREATED	EXPORT	COPY	DELETE
Ticket create notification		valid	09/21/2020 17:08 (Europe/Berlin)	09/21/2020 17:08 (Europe/Berlin)	⬇	⧉	🗑
Ticket email delivery failure notification		valid	09/21/2020 17:08 (Europe/Berlin)	09/21/2020 17:08 (Europe/Berlin)	⬇	⧉	🗑

Fig. 1.94: Delete Ticket Notification Screen

To export all ticket notifications:

1. Click on the Export Notifications button in the left sidebar.

2. Choose a location in your computer to save the `Export_Notification.yml` file.

> **Warning:** Certain settings are exported as numeric IDs and will break when importing to a system where these settings do not appear or reference other named items.

To import ticket notifications:

1. Click on the Browse… button in the left sidebar.

2. Select a previously exported `.yml` file.

3. Click on the Overwrite existing notifications? checkbox, if you would like to overwrite the existing notifications.

4. Click on the Import Notification configuration button.

Ticket Notification Settings
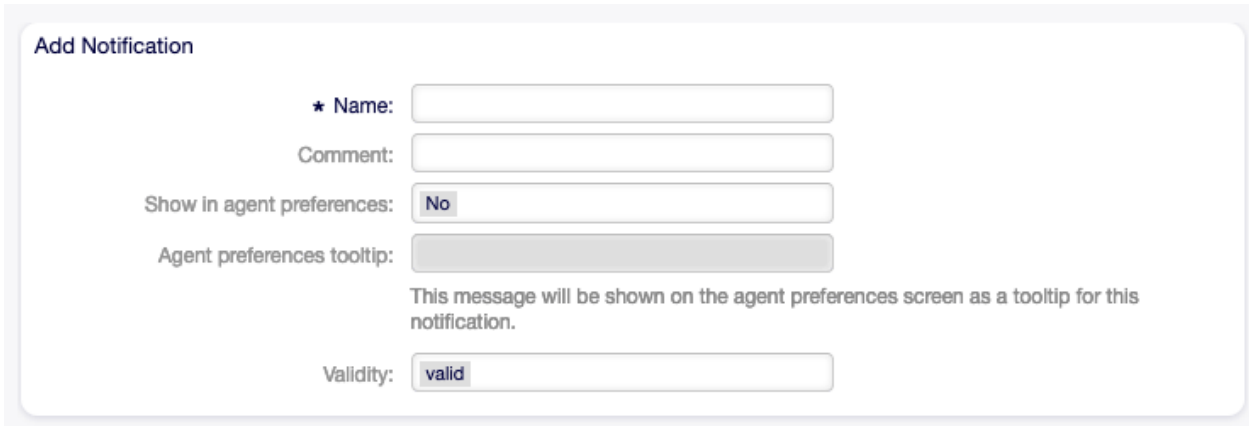
The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

See also:

For an example, see a default ticket notification which is included in a fresh OTOBO installation.

Basic Ticket Notification Settings

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Fig. 1.95: Ticket Notification Settings - Basic

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

Show in agent preferences Define how the notification should be displayed in agent preferences. The following options are available:

No The notification won't be displayed in agent preferences. The notification is sent to all appropriate agents by the defined method.

Yes The notification will be displayed in agent preferences for selection. The agents may opt-in or opt-out.

Yes, but require at least one active notification method. The notification will be displayed in agent preferences, but require at least one active notification method. This is annotated by an asterisk next to the name.

Fig. 1.96: Personal Ticket Notification Settings

Agent preferences tooltip This message will be shown on the agent preferences screen as a tooltip for this notification.

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

Events

Fig. 1.97: Ticket Notification Settings - Events
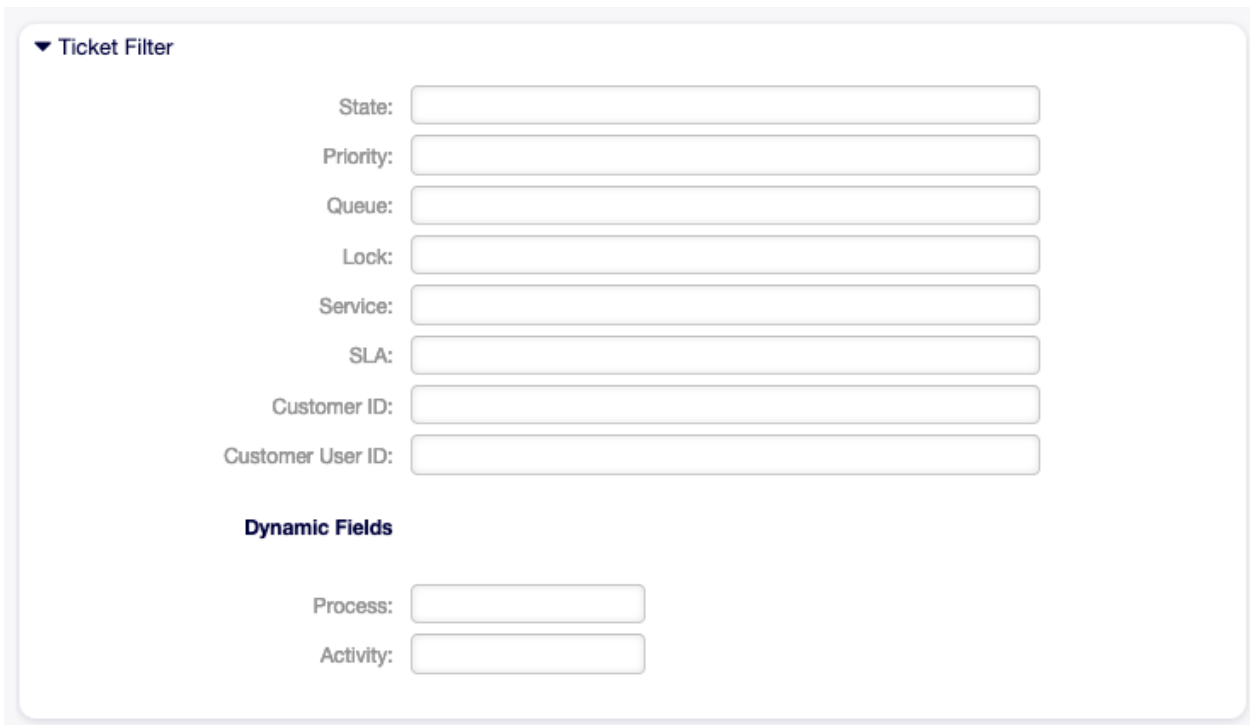
Event Here you can choose which events will trigger this notification. An additional ticket filter can be applied below to only send for tickets with certain criteria.

Ticket Filter[1]

Fig. 1.98: Ticket Notification Settings - Ticket Filter

This widget can optionally be used to narrow the list of tickets by matching configured values:

Note: The values in this list can grow as your system grows. The more Dynamic Fields are and features you have in your system, the longer the list will be.

State Filter for a state of the ticket.

---

[1] Use of regular expressions as a filter do not work here.

Priority Filter for a priority of the ticket.

Queue Filter for a queue in which the ticket is located.

Lock Filter for a lock state of the ticket.

Customer ID Filter for a customer ID of the ticket.

Customer User ID Filter for a customer user ID of the ticket.

Dynamic Fields Filter for some dynamic fields added to the system. For the complete list of dynamic fields see the Dynamic Fields chapter.

Article Filter[1]

Fig. 1.99: Ticket Notification Settings - Article Filter

Note: This widget works only if `ArticleCreate` or `ArticleSend` is selected in the Events widget.

Article sender type Filter for the sender type of the ticket. Possible values are agent, system or customer.

Customer visibility Filter for the customer visibility. Possible values are Invisible to customer or Visible to customer.

Communication channel Filter for the communication channel. Possible values are Chat, Email, OTOBO, Phone or SMS.

Include attachments to notification If Yes is selected, attachments will be included to notification. Selecting No will not use this feature.

Attachment Name Filter for attachment name.

Bcc Filter for blind carbon copy field.

Body Filter for body text.

Cc Filter for carbon copy field.

From Filter for the sender field.

Subject Filter for the subject field.

To Filter for the main recipients field.

SMS phone number Filter for an SMS phone number.

SMS text Filter for the SMS text.

SMS transaction number Filter for an SMS transaction number.

Ticket Notification Recipients

Fig. 1.100: Ticket Notification Settings - Recipients

Send to Select which agents should receive the notifications. Possible values are:

- Agent who created the ticket
- Agent who is responsible for the ticket
- Agent who owns the ticket
- All agents subscribed to both the ticket's queue and service
- All agents subscribed to the ticket's queue
- All agents subscribed to the ticket's service
- All agents watching the ticket
- All agents with write permission for the ticket
- All recipients of the first article
- All recipients of the last article
- Customer user of the ticket

Send to these agents One or more agents can be selected who should receive the notifications.

Send to all group members (agents only) One or more groups can be selected whom agents should receive the notifications.

Send to all role members One or more roles can be selected whom agents should receive the notifications.

Send on out of office If this option is checked, the notification will be sent even if the agent is currently out of office.

Once per day Notify users just once per day about a single ticket using a selected transport. If this is the first notification about a ticket, then the notification will be sent. If a notification was already sent before and this option is checked, the OTOBO daemon will check the time the last notification was sent. If there was no notification sent in the last 24 hours, the notification will be sent again.

Capability to Send Ticket Notifications Only During Working Hours

A new option has been added to the ticket notification settings. Now, within the recipient's section, an additional field named "Calendar" is displayed.

Fig. 1.101: Recipient block configuration

This field allows for the selection of whether notifications should be sent to recipients during working hours or outside of them:

- Selecting the option for working hours will result in notifications being sent only during the established work schedule.

Fig. 1.102: Option working hours

- Conversely, choosing the non-working hours option will result in notifications being sent outside of the defined working hours.

Fig. 1.103: Option outside working hours

Ticket Notification Methods

Enable this notification method Enable or disable this notification method. A notification method can be email, web view or SMS.

Additional recipient email addresses Additional recipients can be added here. Use comma or semi-colon to separate the email addresses.

Article visible for customer An article will be created if the notification is sent to the customer or an additional email address.

Email template Select which email template should be used for the notification.

Fig. 1.104: Ticket Notification Settings - Notification Methods

---

Note: Additional email templates can be added by placing a `.tt` file into the folder `<OTOBO_Home>/Kernel/Output/HTML/Templates/Standard/NotificationEvent/Email/`. See the existing email templates for an example.

Enable email security Checking this option will encrypt the notification email.

Note: To use this feature, PGP Keys or S/MIME Certificates need to be enabled.

Email security level If Enable email security is checked, then this setting is activated. The following options are available:

PGP sign only Sign only the notification email with PGP key. If no PGP keys have been added to the system, this option is not visible.

PGP encrypt only Encrypt only the notification email with PGP key. If no PGP keys have been added to the system, this option is not visible.

PGP sign and encrypt Sign and encrypt the notification email with PGP key. If no PGP keys have been added to the system, this option is not visible.

SMIME sign only Sign only the notification email with S/MIME certificate. If no S/MIME certificates have been added to the system, this option is not visible.

SMIME encrypt only Encrypt only the notification email with S/MIME certificate. If no S/MIME certificates have been added to the system, this option is not visible.

SMIME sign and encrypt Sign and encrypt the notification email with S/MIME certificate. If no S/MIME certificates have been added to the system, this option is not visible.

Note: To use this feature, PGP Keys or S/MIME Certificates need to be enabled.

If signing key/certificate is missing Select the method, that should be used if signing key or certificate is missing.

If encryption key/certificate is missing: Select the method, that should be used if encryption key or certificate is missing.

Notification Text

The main content of a notification can be added for each languages with localized subject and body text. It is also possible to define static text content mixed with OTOBO smart tags.

Subject * The localized subject for a specific language.

Text * The localized body text for a specific language.

Add new notification language Select which languages should be added to create localized notifications. The language of the customer or agent will be used as found in the customer and agent preferences. Secondarily, the system default language will be chosen. The fall back will always be English.
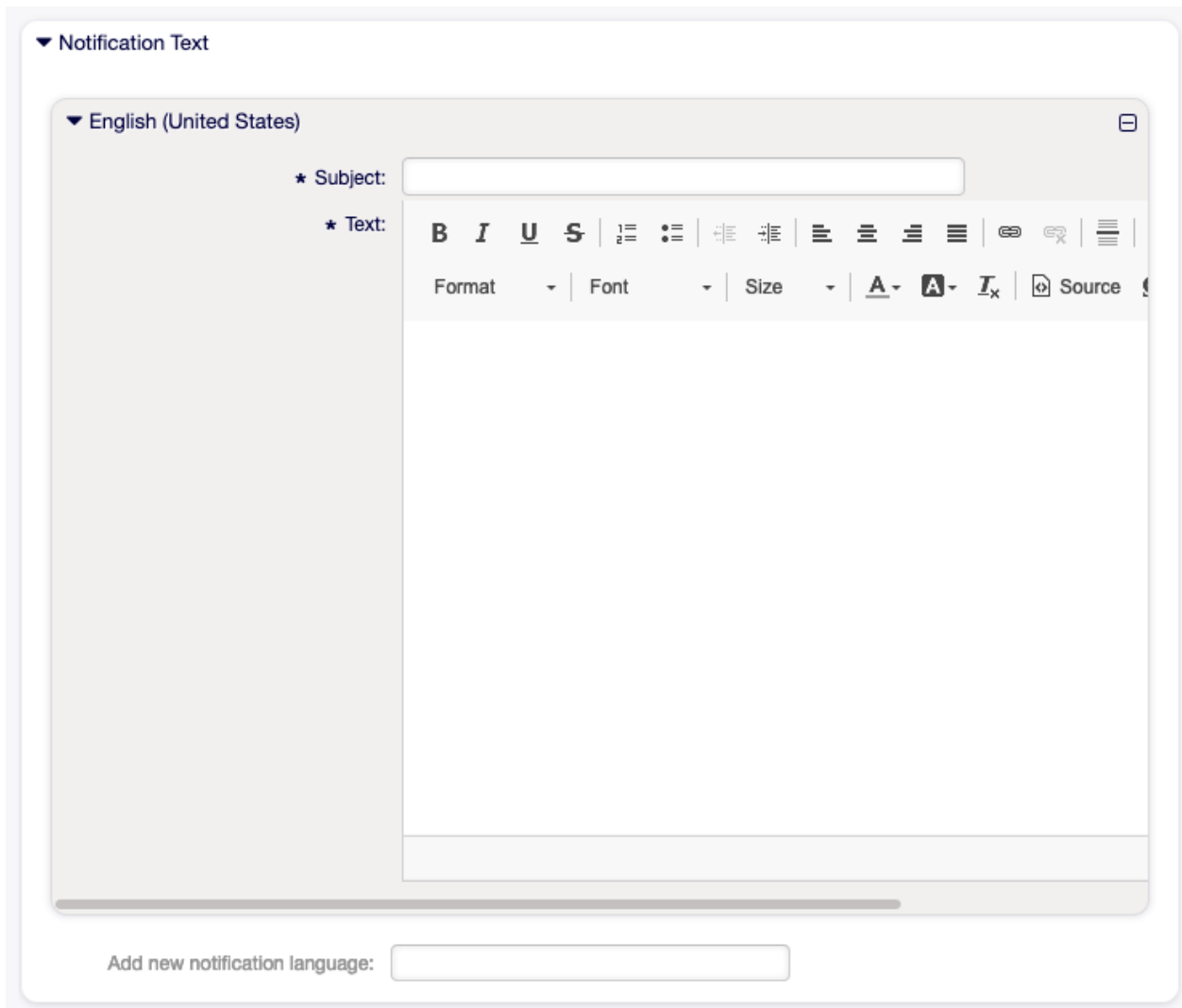
Fig. 1.105: Ticket Notification Settings - Notification Text

> **Warning:** Deleting a language in `DefaultUsedLanguages` setting that already has a notification text here will make the notification text unusable. If a language is not present or enabled on the system, the corresponding notification text could be deleted if it is not needed anymore.

Ticket Notification Variables

Using variables in the text makes it possible to personalize messages. Variables, known as OTOBO tags, are replaced by OTOBO when generating the message. Find a list of available tags stems for this resource at the bottom of both add and edit screens.

For example, the variable `<OTOBO_TICKET_TicketNumber>` expands to the ticket number allowing a template to include something like the following.

```
Ticket#<OTOBO_TICKET_TicketNumber>
```

This tag expands, for example to:

```
Ticket#2018101042000012
```

1.3 Users, Groups & Roles

Simple and complex organizations need a flexible way to control access to their service desk. Access, resources, and permissions must be orchestrated so that, the users and customers have access to the resources they need and their data is protected as needed by the system.

OTOBO provides powerful tools for this purpose and their use is described in the following chapter.

1.3.1 Agents

Agent and access management for your service desk should be easy. Flexibility in adding, editing, invalidating access and a quick overview of which permissions a user has will help you maintain a clean permissions system and record of your setup in OTOBO.

OTOBO aids you giving you the power to manage agents within OTOBO across multiple back ends. OTOBO can use up-to ten back end sources, even marking some as read-only. Managing user settings centrally, an administrator can quickly invalidate a compromised account or set an account to out-of-office in case of an unexpected illness.

Use this screen to add agents to the system. A fresh OTOBO installation contains an agent with administrator privileges by default. The agent management screen is available in the Agents module of the Users, Groups & Roles group.

> **Warning:** The superuser account username is root@localhost. Don't use the superuser account to work with OTOBO! Create new agents and work with these accounts instead. One of the adverse effects is that Access Control Lists (ACL) will not have an effect on this user.

Manage Agents

▼ Tag Reference

Notifications are sent to an agent or a customer.
You can use the following tags:

`<OTOBO_AGENT_SUBJECT[20]>`
To get the first 20 character of the subject (of the latest agent article).

`<OTOBO_AGENT_BODY[5]>`
To get the first 5 lines of the body (of the latest agent article).

`<OTOBO_AGENT_*>`
To get the article attribute (e. g. `<OTOBO_AGENT_From>`, `<OTOBO_AGENT_To>`, `<OTOBO_AGENT_Cc>`).

`<OTOBO_CUSTOMER_SUBJECT[20]>`
To get the first 20 character of the subject (of the latest customer article).

`<OTOBO_CUSTOMER_BODY[5]>`
To get the first 5 lines of the body (of the latest customer article).

`<OTOBO_CUSTOMER_REALNAME>`
To get the name of the ticket's customer user (if given).

`<OTOBO_CUSTOMER_*>`
To get the article attribute (e. g. `<OTOBO_CUSTOMER_From>`, `<OTOBO_CUSTOMER_To>`, `<OTOBO_CUSTOMER_Cc>`).

`<OTOBO_CUSTOMER_DATA_*>`
Attributes of the current customer user data (e. g. `<OTOBO_CUSTOMER_DATA_UserFirstname>`).

`<OTOBO_OWNER_*>` or `<OTOBO_TICKET_OWNER_*>`
Attributes of the current ticket owner user data (e. g. `<OTOBO_OWNER_UserFirstname>` or
`<OTOBO_TICKET_OWNER_UserFirstname>`).

`<OTOBO_RESPONSIBLE_*>` or `<OTOBO_TICKET_RESPONSIBLE_*>`
Attributes of the current ticket responsible user data (e. g. `<OTOBO_RESPONSIBLE_UserFirstname>` or
`<OTOBO_TICKET_RESPONSIBLE_UserFirstname>`).

`<OTOBO_CURRENT_*>`
Attributes of the current agent user who requested this action (e. g. `<OTOBO_CURRENT_UserFirstname>`).

`<OTOBO_*>` or `<OTOBO_NOTIFICATION_RECIPIENT_*>`
Attributes of the recipient user for the notification (e. g. `<OTOBO_UserFullname>` or
`<OTOBO_NOTIFICATION_RECIPIENT_UserFullname>`).

`<OTOBO_TICKET_*>`
Attributes of the ticket data (e. g. `<OTOBO_TICKET_TicketNumber>`, `<OTOBO_TICKET_TicketID>`,
`<OTOBO_TICKET_Queue>`, `<OTOBO_TICKET_State>`).

`<OTOBO_TICKET_DynamicField_*>`
Ticket dynamic fields internal key values (e. g. `<OTOBO_TICKET_DynamicField_TestField>`,
`<OTOBO_TICKET_DynamicField_TicketFreeText1>`).

`<OTOBO_TICKET_DynamicField_*_Value>`
Ticket dynamic fields display values, useful for Dropdown and Multiselect fields (e. g.
`<OTOBO_TICKET_DynamicField_TestField_Value>`, `<OTOBO_TICKET_DynamicField_TicketFreeText1_Value>`).

`<OTOBO_CONFIG_*>`
Config options (e. g. `<OTOBO_CONFIG_HttpType>`).

Example notification:

Subject: Ticket Created: <OTOBO_TICKET_Title>
Text:

```
Hi <OTOBO_NOTIFICATION_RECIPIENT_UserFirstname>,

ticket [<OTOBO_CONFIG_TicketHook><OTOBO_TICKET_TicketNumber>] has been
created in queue <OTOBO_TICKET_Queue>.

<OTOBO_CUSTOMER_REALNAME> wrote:
<OTOBO_CUSTOMER_Body[30]>

<OTOBO_CONFIG_HttpType>://<OTOBO_CONFIG_FQDN>
/<OTOBO_CONFIG_ScriptAlias>index.pl?Action=AgentTicketZoom;TicketID=
<OTOBO_TICKET_TicketID>
```

Fig. 1.106: Ticket Notification Variables
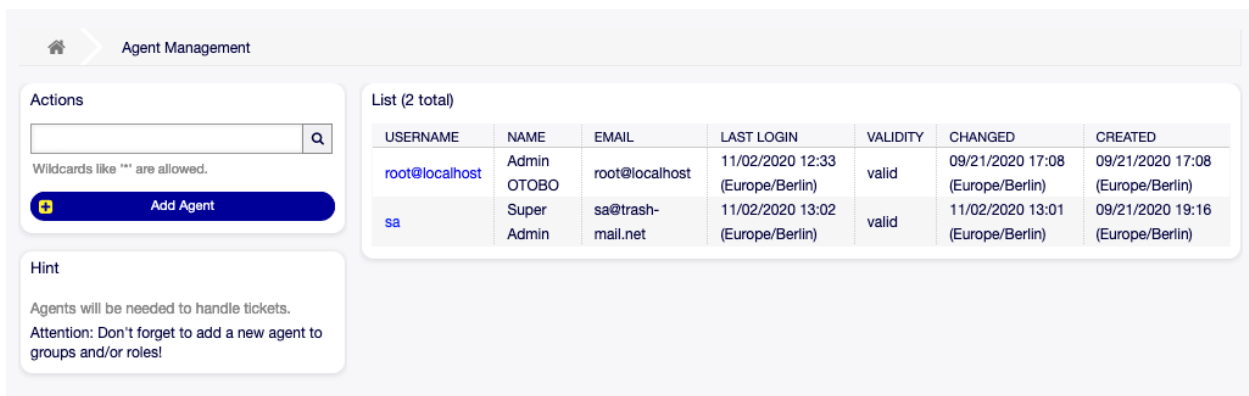
Fig. 1.107: Agent Management Screen

Note: Adding or editing an agent is possible only by using database back end. Using explicitly external directory services like LDAP and, based on configuration, some databases are read-only. Personal preferences like out-of-office can still be set.

To add an agent:

1. Click on the Add Agent button in the left sidebar.

2. Fill in the required fields.

3. Click on the Save button.
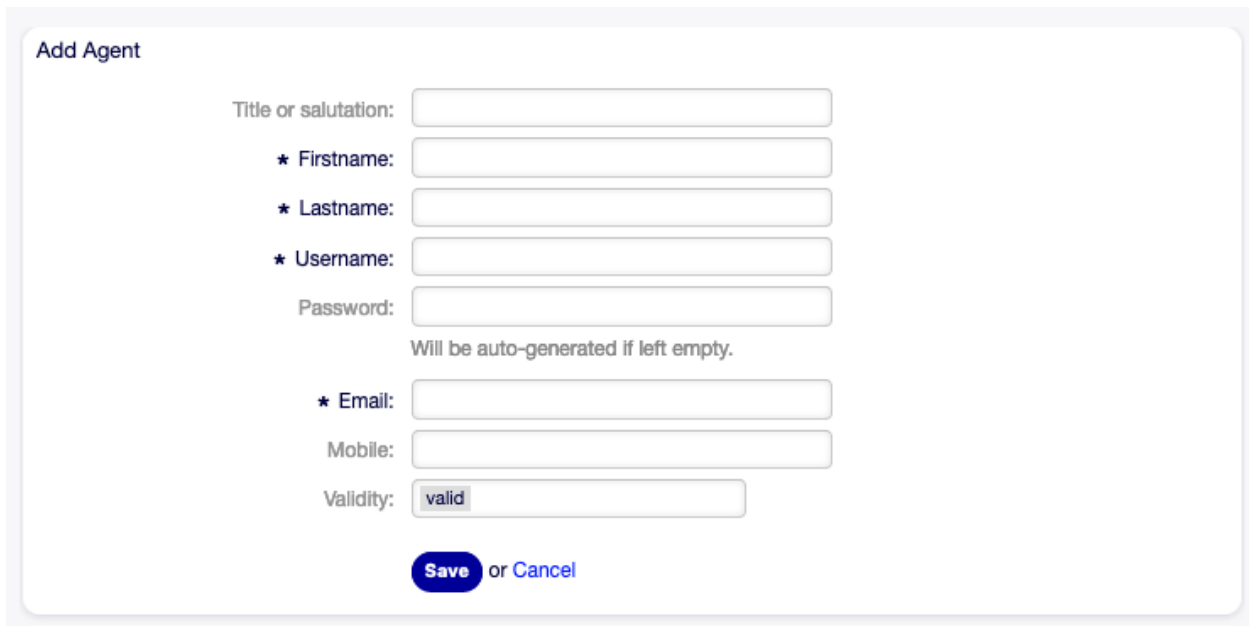
Fig. 1.108: Add Agent Screen

Warning: Agents can not be deleted from the system. They can only be deactivated by setting the Validity option to invalid or invalid-temporarily.

To edit an agent:

1. Click on an agent in the list of agents.

2. Modify the fields.

3. Click on the Save or Save and finish button.

Fig. 1.109: Edit Agent Screen

It is also possible to edit the agent personal preferences. To do this, click on the Edit personal preferences for this agent button in the left sidebar of the Edit Agent screen.

To find an agent:

1. Enter a search term to the search box in the left sidebar.

2. Click on the magnifying glass icon in the right part of the field or hit an `Enter`.

Note: If several agents are added to the system, use the search box to find a particular agent. Only the first 1000 agents are listed by default.

The agent permissions can be controlled by adding an agent to Groups or Roles. This can result a complex matrix of permissions. The effective permissions for an agent can be checked in the bottom of the Edit Agent screen. If roles (recommended) are used, this screen will reflect the combined permissions as dictated by the roles.

Agent Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Note: These are the default fields available for the internal database table.

Title or salutation Some name prefix can be added here like Mr., Dr., Jr., etc.

Fig. 1.110: Effective Permissions for Agent Widget

Firstname * The first name of the agent.

Lastname * The last name of the agent.

> **See also:**
>
> The agent display name can be set via the system configuration setting `FirstnameLastnameOrder`.

Username * The username of the agent to login to the system.

Password The password of the agent. This will be auto-generated if left empty.

Email * The email address of the agent.

Note: The email syntax and validity of an available MX record could prevent you from submitting this form. For some systems it may be acceptable to turn off these checks. See the Agent Configuration Options.

Mobile The mobile phone number of the agent.

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

Agent Back End Sources

Agents can be read and synchronized from an Active Directory® or LDAP server.

Note: The following configurations are taken care of by the Customer Solution Team in **OTOBO**. Please contact the Customer Solution Team via support@otrs.ch or in the OTOBO Portal.

Agent Authentication Back End

The users table houses your agent data per default. You may attach a directory server (LDAP/Active Directory) to your installation for authentication. To achieve this copy the following block from the `Defaults.pm` and paste it into the `Config.pm`.

```
# This is an example configuration for an LDAP auth. backend.
# (take care that Net::LDAP is installed!)
#     $Self->{AuthModule} = 'Kernel::System::Auth::LDAP';
#     $Self->{'AuthModule::LDAP::Host'} = 'ldap.example.com';
#     $Self->{'AuthModule::LDAP::BaseDN'} = 'dc=example,dc=com';
#     $Self->{'AuthModule::LDAP::UID'} = 'uid';

# Check if the user is allowed to auth in a posixGroup
# (e. g. user needs to be in a group xyz to use otobo)
#     $Self->{'AuthModule::LDAP::GroupDN'} = 'cn=otoboallow,ou=posixGroups,dc=example,
→dc=com';
#     $Self->{'AuthModule::LDAP::AccessAttr'} = 'memberUid';
# for ldap posixGroups objectclass (just uid)
#     $Self->{'AuthModule::LDAP::UserAttr'} = 'UID';
# for non ldap posixGroups objectclass (with full user dn)
#     $Self->{'AuthModule::LDAP::UserAttr'} = 'DN';

# The following is valid but would only be necessary if the
# anonymous user do NOT have permission to read from the LDAP tree
#     $Self->{'AuthModule::LDAP::SearchUserDN'} = '';
#     $Self->{'AuthModule::LDAP::SearchUserPw'} = '';

# in case you want to add always one filter to each ldap query, use
# this option. e. g. AlwaysFilter => '(mail=*)' or AlwaysFilter => '(objectclass=user)
→'
# or if you want to filter with a locigal OR-Expression, like AlwaysFilter =>
→'(|(mail=*abc.com)(mail=*xyz.com))'
#     $Self->{'AuthModule::LDAP::AlwaysFilter'} = '';

# in case you want to add a suffix to each login name, then
# you can use this option. e. g. user just want to use user but
# in your ldap directory exists user@domain.
#     $Self->{'AuthModule::LDAP::UserSuffix'} = '@domain.com';

# In case you want to convert all given usernames to lower letters you
# should activate this option. It might be helpful if databases are
# in use that do not distinguish selects for upper and lower case letters
# (Oracle, postgresql). User might be synched twice, if this option
# is not in use.
#     $Self->{'AuthModule::LDAP::UserLowerCase'} = 0;

# In case you need to use OTOBO in iso-charset, you can define this
# by using this option (converts utf-8 data from LDAP to iso).
#     $Self->{'AuthModule::LDAP::Charset'} = 'iso-8859-1';

# Net::LDAP new params (if needed - for more info see perldoc Net::LDAP)
#     $Self->{'AuthModule::LDAP::Params'} = {
#         port     => 389,
#         timeout  => 120,
#         async    => 0,
#         version  => 3,
#     };

# Die if backend can't work, e. g. can't connect to server.
#     $Self->{'AuthModule::LDAP::Die'} = 1;

You should uncomment the following settings as a minimum.
```

The minimum required to connect to a directory server is:

```
$Self->{AuthModule} = 'Kernel::System::Auth::LDAP';
$Self->{'AuthModule::LDAP::Host'} = 'ldap.example.com';
$Self->{'AuthModule::LDAP::BaseDN'} = 'dc=example,dc=com';
$Self->{'AuthModule::LDAP::UID'} = 'uid';
```

Host The DNS name or IP of your directory server.

BaseDN The starting point in your directory tree.

UID The attribute used for login and identification.

Note: This is `sAMAccountName` for an Active Directory.

To use multiple back ends, add an additional section of the example code to the `Config.pm`. Please make sure to add a numeric value [1-9] to all settings to indicate which settings belong to which back end.

```
### Backend One
$Self->{AuthModule} = 'Kernel::System::Auth::LDAP';
$Self->{'AuthModule::LDAP::Host'} = 'ldap.example.com';
$Self->{'AuthModule::LDAP::BaseDN'} = 'dc=example,dc=com';
$Self->{'AuthModule::LDAP::UID'} = 'uid';

### Backend Two
$Self->{AuthModule1} = 'Kernel::System::Auth::LDAP';
$Self->{'AuthModule::LDAP::Host1'} = 'ldap.example.com';
$Self->{'AuthModule::LDAP::BaseDN1'} = 'dc=example,dc=com';
$Self->{'AuthModule::LDAP::UID1'} = 'uid';
```

Warning: All back ends will are used in succession. The UID must be unique to all back ends, otherwise some side effects may occur.

To synchronize with a specific directory server (see Agent User Data below), you must add the appropriate setting to your Agent Authentication Back End. To achieve this copy the following block from the `Defaults.pm` and paste it into the `Config.pm`.

```
$Self->{'AuthModule::UseSyncBackend'} = 'AuthSyncBackend';
```

To use multiple back ends, add an additional section of the example code to the `Config.pm`. Please make sure to add a numeric value [1-9] to all settings to indicate which settings belong to which back end.

```
$Self->{'AuthModule::UseSyncBackend1'} = 'AuthSyncBackend1';
```

Reuse of an Agent Synchronization Back End is also possible.

```
$Self->{'AuthModule::UseSyncBackend1'} = 'AuthSyncBackend';
```

Agent Synchronization Back End

It is advisable to synchronize the agent data so that agents need not be manually added to the users table prior to authorization. Additionally, groups and roles can be added automatically using security objects of the directory server.

Agent User Data

Syncing user data upon login. To achieve this copy the following block from the `Defaults.pm` and paste it into the `Config.pm`.

```perl
# This is an example configuration for an LDAP auth sync. backend.
# (take care that Net::LDAP is installed!)
#     $Self->{AuthSyncModule} = 'Kernel::System::Auth::Sync::LDAP';
#     $Self->{'AuthSyncModule::LDAP::Host'} = 'ldap.example.com';
#     $Self->{'AuthSyncModule::LDAP::BaseDN'} = 'dc=example,dc=com';
#     $Self->{'AuthSyncModule::LDAP::UID'} = 'uid';

# The following is valid but would only be necessary if the
# anonymous user do NOT have permission to read from the LDAP tree
#     $Self->{'AuthSyncModule::LDAP::SearchUserDN'} = '';
#     $Self->{'AuthSyncModule::LDAP::SearchUserPw'} = '';

# in case you want to add always one filter to each ldap query, use
# this option. e. g. AlwaysFilter => '(mail=*)' or AlwaysFilter => '(objectclass=user)
↪'
# or if you want to filter with a logical OR-Expression, like AlwaysFilter =>
↪'(|(mail=*abc.com)(mail=*xyz.com))'
#     $Self->{'AuthSyncModule::LDAP::AlwaysFilter'} = '';

# AuthSyncModule::LDAP::UserSyncMap
# (map if agent should create/synced from LDAP to DB after successful login)
# you may specify LDAP-Fields as either
#  * list, which will check each field. first existing will be picked ( ["givenName","cn
↪","_empty"] )
#  * name of an LDAP-Field (may return empty strings) ("givenName")
#  * fixed strings, prefixed with an underscore: "_test", which will always return this␣
↪fixed string
#     $Self->{'AuthSyncModule::LDAP::UserSyncMap'} = {
#         # DB -> LDAP
#         UserFirstname => 'givenName',
#         UserLastname  => 'sn',
#         UserEmail     => 'mail',
#     };
```

The minimum required to connect to a directory server is:

```perl
$Self->{AuthSyncModule} = 'Kernel::System::Auth::Sync::LDAP';
$Self->{'AuthSyncModule::LDAP::Host'} = 'ldap.example.com';
$Self->{'AuthSyncModule::LDAP::BaseDN'} = 'dc=example,dc=com';
$Self->{'AuthSyncModule::LDAP::UID'} = 'uid';
```

Host The DNS name or IP of your directory server.

BaseDN The starting point in your directory tree.

UID The attribute used for login and identification.

Note: This is `sAMAccountName` for an Active Directory.

Note: Multiple Agent Synchronization Back End blocks can be used. Please make sure to add a numeric value [1-9] to all settings to indicate which settings belong to which back end. Each `AuthSyncModule` must be explicitly used in an Agent Authentication Back End.

Agent Group Data

It is possible to use security objects to synchronize users to OTOBO groups. To achieve this copy the following block from the `Defaults.pm` and paste it into the `Config.pm`.

Note: Groups must be available in OTOBO to use this feature.

Agent Role Data

It is possible to use security objects to synchronize users to OTOBO roles. To achieve this copy the following block from the `Defaults.pm` and paste it into the `Config.pm`.

Note: Roles must be available in OTOBO to use this feature.

Agent Configuration Options

1.3.2 Agents ⬚ Groups

Efficient and straightforward management of permissions is essential in a growing business. Easy assignment of a particular user to a group for quick access, or to remove access, to resources is a must in every case.

The OTOBO interface provides you both with the possibility to manage an agent's access to one or more particular groups. As well, you can change multiple users access to any one group, efficiently and elegantly.

Use this screen to add one or more agents to one or more groups. To use this function, at least one agent and one group need to have been added to the system. The management screen is available in the Agents ⬚ Groups module of the Users, Groups & Roles group.

Manage Agents ⬚ Groups Relations

To assign some groups to an agent:

1. Click on an agent in the Agents column.

2. Select the permissions you would like to connect the agent to groups with.

3. Click on the Save or Save and finish button.

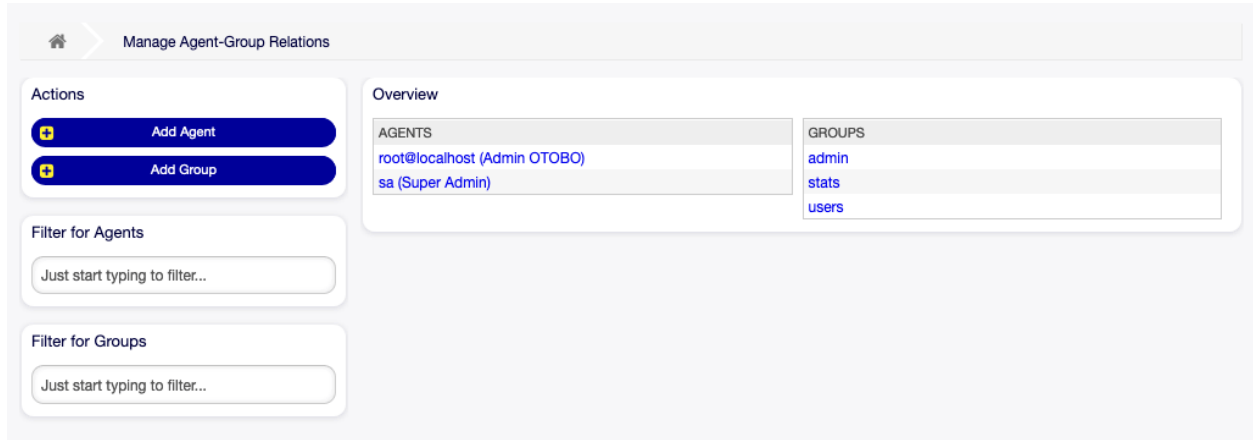To assign some agents to a group:
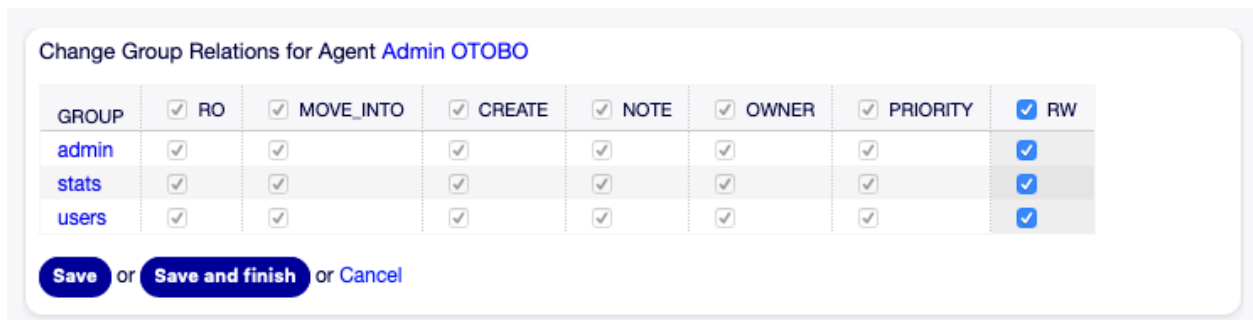

Fig. 1.111: Manage Agent-Group Relations

Fig. 1.112: Change Group Relations for Agent

1. Click on a group in the Groups column.
2. Select the permissions you would like to connect the group to agents with.
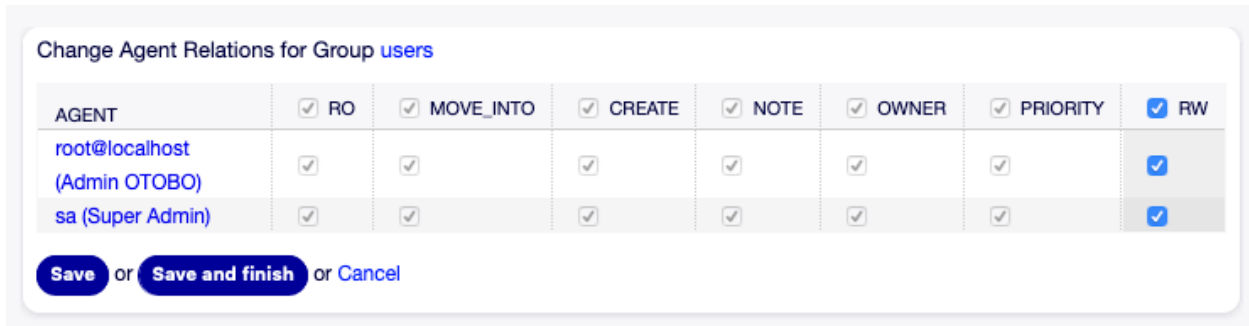3. Click on the Save or Save and finish button.

Fig. 1.113: Change Agent Relations for Group

Note: If several agents or groups are added to the system, use the filter box to find a particular agent or group by just typing the name to filter.

Multiple agents or groups can be assigned in both screens at the same time. Additionally clicking on an agent or clicking on a group in the relations screen will open the Edit Agent screen or the Edit Group screen accordingly.

Warning: Accessing an agent or a group provides no back link to the relations screen.

Agents ⧉ Groups Relations Reference

When assigning an agent to a group or vice versa, several permissions can be set as connection between an agent and a group. The following permissions are available by default:

ro Read only access to the ticket in this group/queue.

move_into Permissions to move tickets into this group/queue.

create Permissions to create tickets in this group/queue.

note Permissions to add notes to tickets in this group/queue.

owner Permissions to change the owner of tickets in this group/queue.

priority Permissions to change the ticket priority in this group/queue.

chat_owner Agents have full rights for a chat and can accept chat requests.

chat_participant Agents may normally participate in a chat.

chat_observer Agents may take part silently in a chat.

rw Full read and write access to the tickets in this group/queue.

See also:

Not all available permissions are shown by default. See `System::Permission` setting for permissions that can be added. These additional permissions can be added:

---

stats Gives access to the stats page.

bounce Permissions to bounce an email message (with bounce button in Ticket Zoom screen).

compose Permissions to compose an answer for a ticket.

customer Permissions to change the customer of a ticket.

forward Permissions to forward a message (with the forward button).

pending Permissions to set a ticket to pending.

phone Permissions to add a phone call to a ticket.

responsible Permissions to change the responsible agent for a ticket.

Note: By setting a checkbox in the header of a column will set all the checkboxes in the selected column. By setting the checkbox in the last rw column will set all the checkboxes in the selected row.

1.3.3 Agents ⬚ Roles

As an organization grows, groups cannot be the denominator used for processing access rights. Roles become more and more a need because a role has a special set of permissions assigned. One mustn't give individual permissions, but the role carries the permissions built-in.

OTOBO allows easy access to a predefined set of permissions via one or more roles defined. These roles are assigned easily to one or more agents, or one or more agent to a role.

Use this screen to add one or more agents to one or more roles. To use this function, at least one agent and one role need to have been added to the system. The management screen is available in the Agents ⬚ Roles module of the Users, Groups & Roles group.
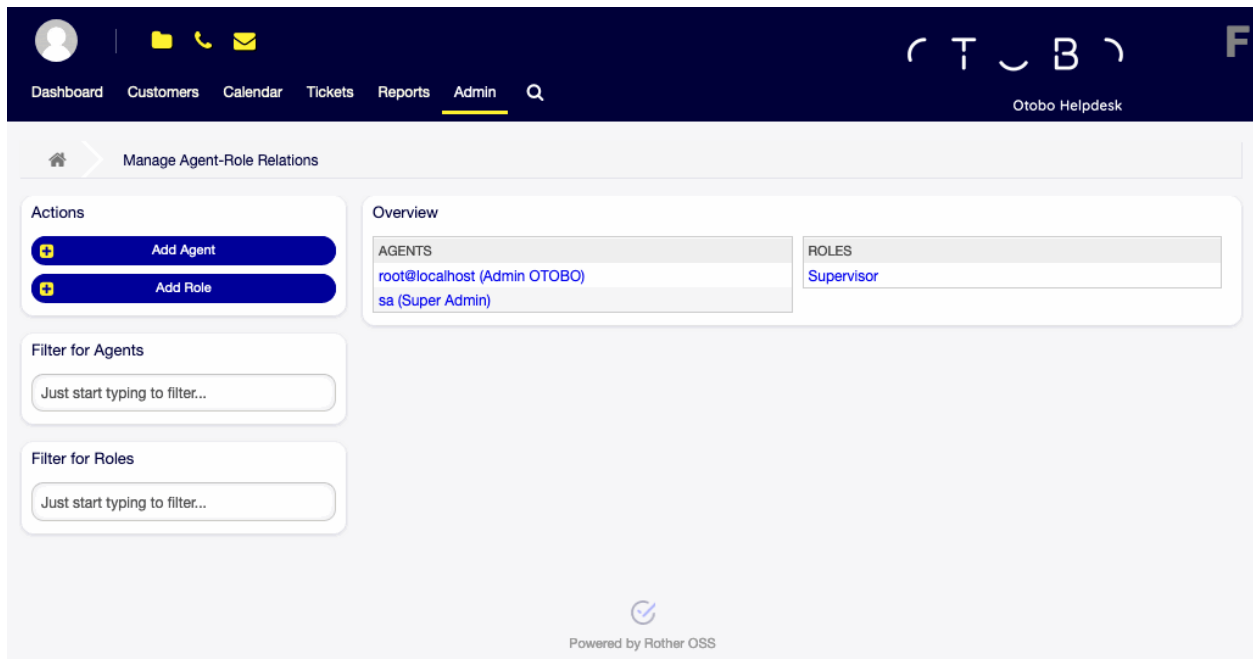

Fig. 1.114: Manage Agent-Role Relations

Manage Agents ⧄ Roles Relations

To assign some roles to an agent:

1. Click on an agent in the Agents column.

2. Select the roles you would like to the agent belongs to.

3. Click on the Save or Save and finish button.
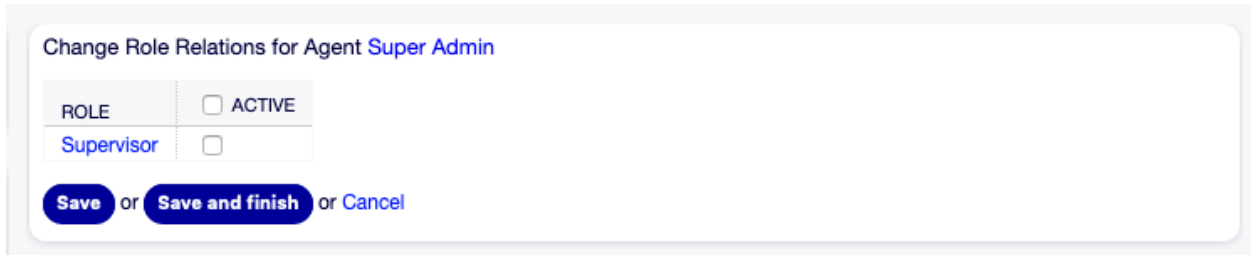

Fig. 1.115: Change Role Relations for Agent

To assign some agents to a role:

1. Click on a role in the Roles column.

2. Select the agents you would like to add to the role.

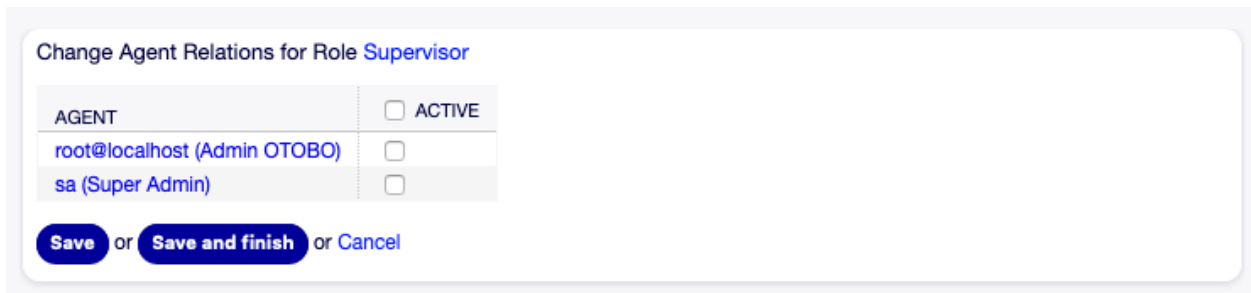3. Click on the Save or Save and finish button.

Fig. 1.116: Change Agent Relations for Role

Note: If several agents or roles are added to the system, use the filter box to find a particular agent or role by just typing the name to filter.

Multiple agents or roles can be assigned in both screens at the same time. Additionally clicking on an agent or clicking on a role in the relations screen will open the Edit Agent screen or the Edit Role screen accordingly.

Warning: Accessing an agent or a role provides no back link to the relations screen.

Note: By setting the checkbox in the header of the column Active will set all the checkboxes in the column.

1.3.4 Customers

Use this screen to add customer companies to the system. A fresh OTOBO installation contains no customers by default. The customer management screen is available in the Customers module of the Users, Groups & Roles group.
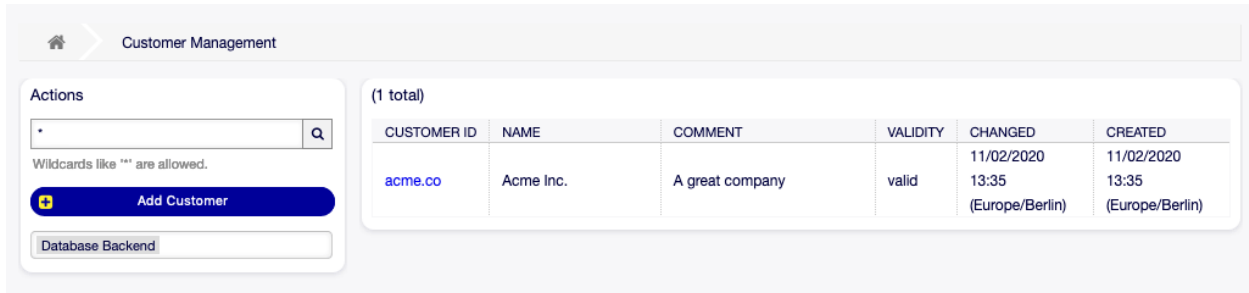

Fig. 1.117: Customer Management Screen

Manage Customers

Note: Adding or editing a customer is possible only by using database back end. Using external directory services like LDAP will disable the customer management functionality.

To add a customer:

1. Click on the Add Customer button in the left sidebar.

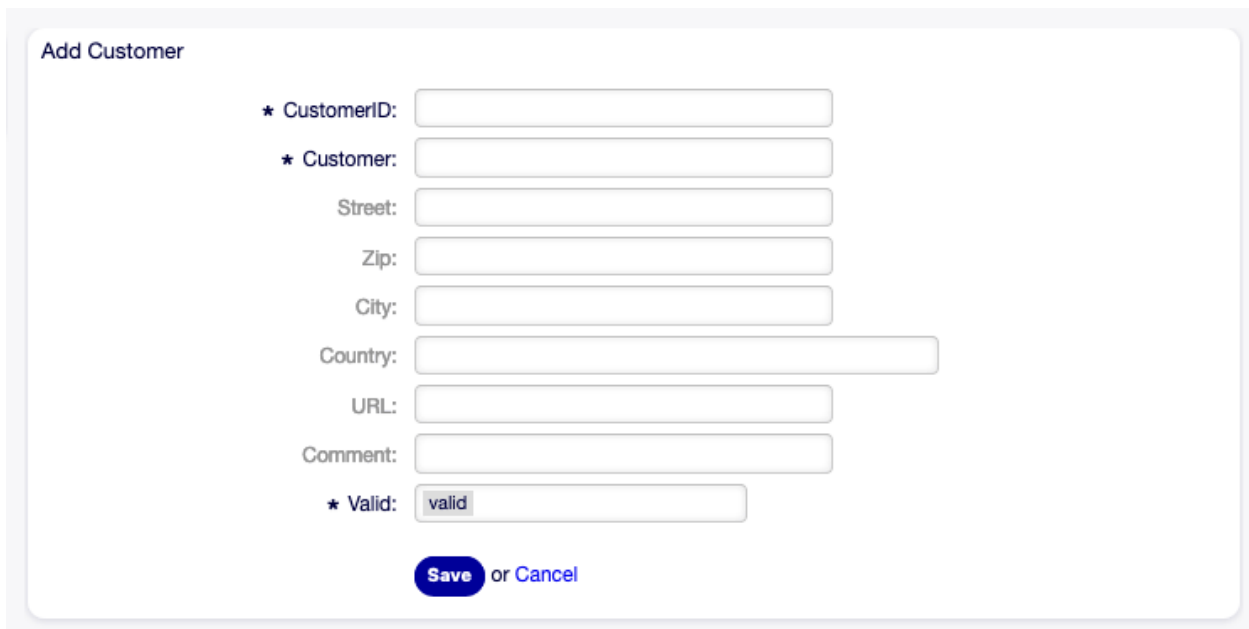2. Fill in the required fields.

3. Click on the Save button.

Fig. 1.118: Add Customer Screen

> **Warning:** Customers can not be deleted from the system. They can only be deactivated by setting the Validity option to invalid or invalid-temporarily.

To edit a customer:

1. Click on a customer in the list of customers.
2. Modify the fields.
3. Click on the Save or Save and finish button.

Fig. 1.119: Edit Customer Screen

To find a customer:

1. Enter a search term to the search box in the left sidebar.
2. Click on the magnifying glass icon in the right part of the field or hit an `Enter`.

> **Note:** If several customers are added to the system, use the search box to find a particular customer. Only the first 1000 customers are listed by default.

Customer Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Customer ID * The internal name of the customer. Should contain only letters, numbers and some special characters.

Customer * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Street The street name of the customer.

Zip The zip code of the customer.

City The headquarter city of the customer.

Country The country of the customer. Choose a country from the list.

URL The web page or other URL of the customer.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

Valid * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

1.3.5 Customers ⮂ Groups

Your organization grows, and it's not practical at some point to assign permissions to individual users, you need to assign the permissions to all customer users of a customer.

OTOBO allows you to assign group permissions to a customer. Access works just the same as for agents, preventing a customer from modifying and viewing a request. Thus allowing the customer to focus on the results of the original communication and funneling the discussion through one ticket.

See also:

Assign a single customer user to a group using Customer Users ⮂ Groups.

Use this screen to add one or more customers to one or more groups. To use this function, at least one customer and one group need to have been added to the system. The management screen is available in the Customers ⮂ Groups module of the Users, Groups & Roles group.
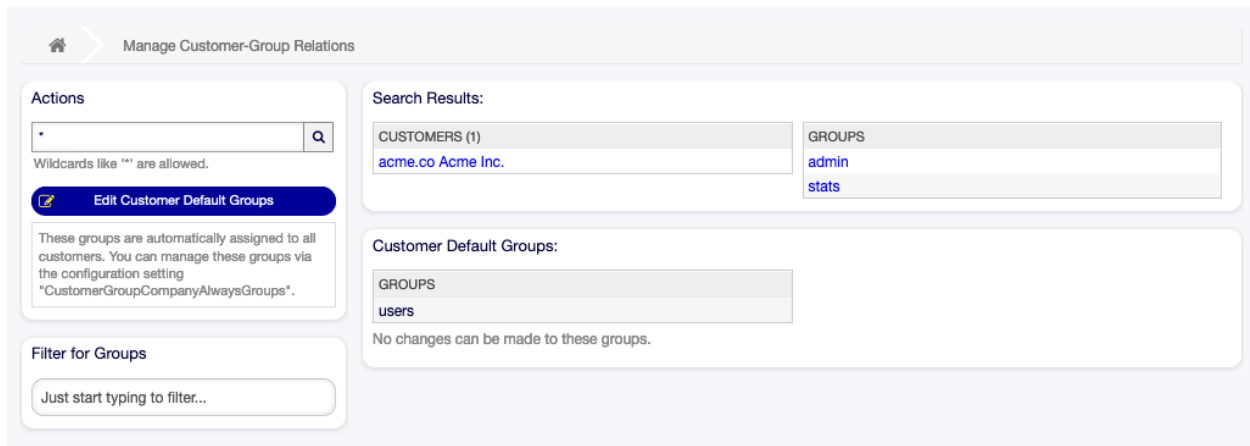

Fig. 1.120: Manage Customer-Group Relations

Customer group support needs to be enabled in at least one customer user back end to use this function. For the default OTOBO back end, this can be enabled in the system configuration by clicking on the Enable it here! button.

Note: To enable this feature in systems using a directory server or multiple non-default back ends, a custom configuration file needs to be placed in `Kernel/Config/Files` (for example named
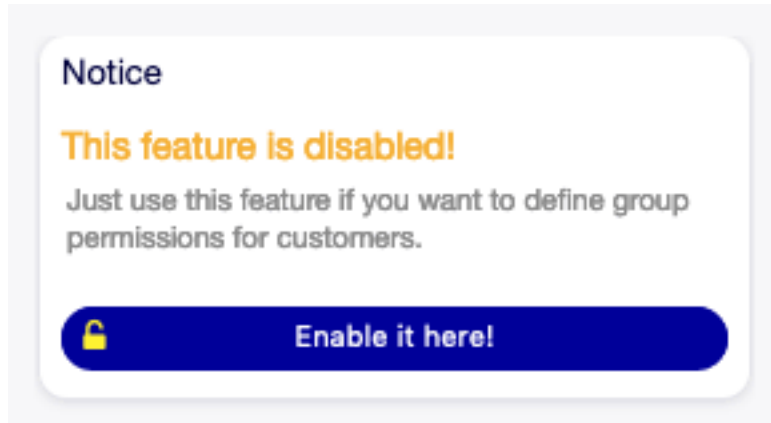
Fig. 1.121: Enable Customer Group Feature

`ZZZ_CustomerBackend.pm`). Once activated, all customer users from this back end will require group assignment.

> **Warning:** After making changes to the back end, the server cache will be deleted, which may cause a temporary drop in performance.

Manage Customers ⮂ Groups Relations

> **Note:** To be able to use this feature, you have to activate its setting first.
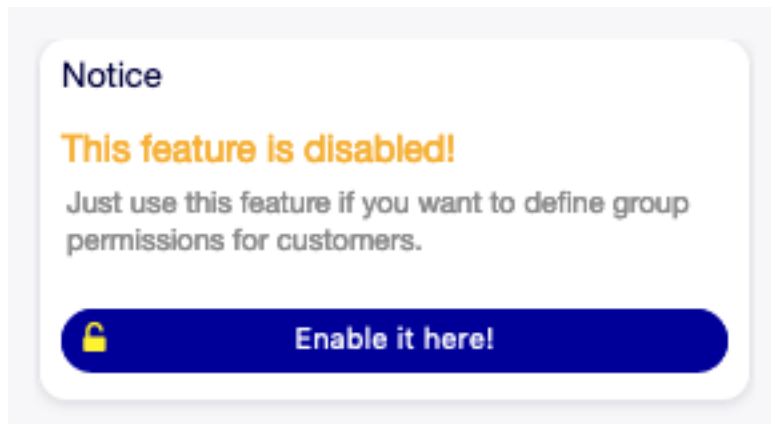

Fig. 1.122: Enable Customer-Group Support

To assign some groups to a customer:

1. Click on a customer in the Customers column.
2. Select the permissions you would like to connect the customer to groups with.
3. Click on the Save or Save and finish button.

Fig. 1.123: Change Group Relations for Customer

To assign some customers to a group:

1. Click on a group in the Groups column.

2. Select the permissions you would like to connect the group to customers with.

3. Click on the Save or Save and finish button.

Fig. 1.124: Change Customer Relations for Group

To change customer default groups:

1. Click on the Edit Customer Default Groups button in the left sidebar.

2. Add or modify groups in setting `CustomerGroupCompanyAlwaysGroups`.

3. Deploy the modified system configurations.

Fig. 1.125: `CustomerGroupCompanyAlwaysGroups` System Configuration Screen

These groups are automatically assigned to all customers.

Note: If several customers or groups are added to the system, use the search box to find a particular

customer or use the filter box to find a particular group by just typing the name to filter.

Multiple customers or groups can be assigned in both screens at the same time. Additionally clicking on a customer or clicking on a group in the relations will open the Edit Customer screen or the Edit Group screen accordingly.

> **Warning:** Accessing a customer or a group provides no back link to the relations screen.

Customers ⬚ Groups Relations Reference

When assigning a customer to a group or vice versa, several permissions can be set as connection between a customer and a group. The following permissions are available by default:

Same Customer Gives customer users group based access to tickets from customer users of the same customer (ticket `CustomerID` is a `CustomerID` of the customer user).

ro Read only access to the resource.

rw Full read and write access to the resource.

> **Note:** By setting a checkbox in the header of a column will set all the checkboxes in the selected column.

1.3.6 Customer Users

A record of who your company deals with requires more information about that individual: physical location for shipping and billing purposes, as well as contact information for email and phone contact.

OTOBO offers a great way to save individual information about contacts within organizations which your company serves. You can add as many personal connections into OTOBO as needed.

Use this screen to add a customer user to the system. A fresh OTOBO installation contains no customer users by default. The customer user management screen is available in the Customer Users module of the Users, Groups & Roles group.
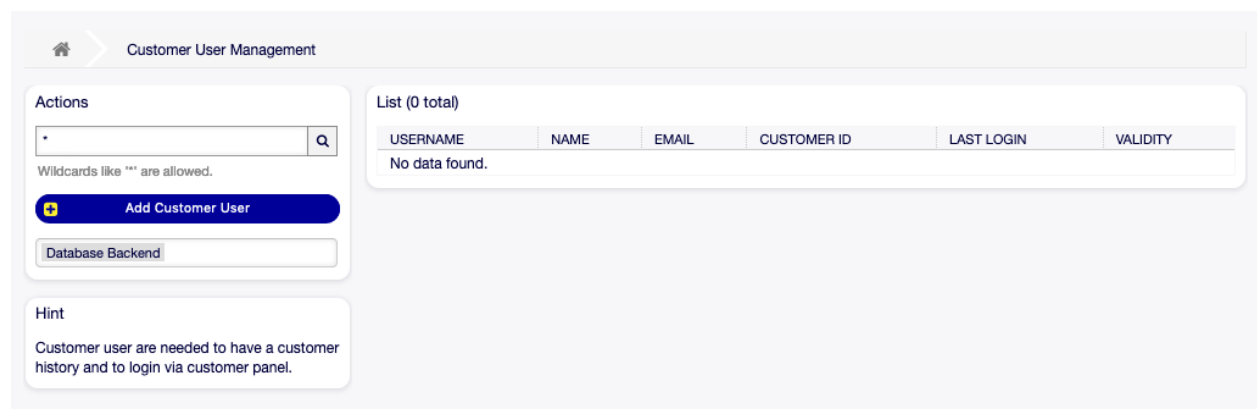

Fig. 1.126: Customer User Management Screen

Manage Customer Users

> **Warning:** A customer user can only be added to the system when at least one customer exists. Create one or more Customers first.

> **Note:** Adding or editing a customer user is possible only by using database back end. Using external directory services like LDAP will disable the customer user management functionality.

To add a customer user:

1. Click on the Add Customer User button in the left sidebar.
2. Fill in the required fields.
3. Click on the Save button.

> **Warning:** Customer users can not be deleted from the system. They can only be deactivated by setting the Validity option to invalid or invalid-temporarily.

To edit a customer user:

1. Click on a customer user in the list of customer users.
2. Modify the fields.
3. Click on the Save or Save and finish button.

To find a customer user:

1. Enter a search term to the search box in the left sidebar.
2. Click on the magnifying glass icon in the right part of the field or hit an `Enter`.

> **Note:** If several customer users are added to the system, use the search box to find a particular customer user. Only the first 1000 customer users are listed by default.

The customer user permissions can be controlled by adding a customer or a customer user to Groups. This can result a complex matrix of permissions. The effective permissions for a customer user can be checked in the bottom of the Edit Customer User screen.

See also:

Customer Users ⬚ Groups needs to be enabled to use this feature.

Customer User Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

> **Note:** These are the default fields available for the internal database table.

Title or salutation Some name prefix can be added here like Mr., Dr., Jr., etc.

Fig. 1.127: Add Customer User Screen

Edit Customer User

Title or salutation:

* Firstname: Wyle

* Lastname: Coyote

* Username: we

Password:

* Email: we@acme.example.com

* CustomerID: acme.co

Phone:

Fax:

Mobile:

Street:

Zip:

City:

Country:

Comment:

* Valid: valid

Google Authenticator:

Enter your shared secret to enable two factor authentication. WARNING: Make sure that you add the shared secret to your generator application and the application works well. Otherwise you will be not able to login anymore without the two factor token.Shared Secret

Interface language: English (United States)

Select the main interface language.Language

Time Zone: UTC

Select your personal time zone. All times will be displayed relative to this time zone.Time Zone

Ticket overview: off

Select after which period ticket overviews should refresh automatically.Refresh interval

Number of displayed tickets: 25

Select how many tickets should be shown in overviews by default.Tickets per page

Save or Save and finish or Cancel

Fig. 1.128: Edit Customer User Screen

Effective Permissions for Customer User

Group Permissions

GROUP	RO	RW
users	✔	✔

Table above shows effective group permissions for the customer user. The matrix takes into account all inherited permissions (e.g. via customer groups). Note: The table does not consider changes made to this form without submitting it.

Customer Access

CUSTOMER	DIRECT
acme.co Acme Inc.	✔

Table above shows granted customer access for the customer user by permission context. The matrix takes into account all inherited access (e.g. via customer groups). Note: The table does not consider changes made to this form without submitting it.

Fig. 1.129: Effective Permissions for Customer User Widget

Firstname * The first name of the customer user.

Lastname * The last name of the customer user.

Username * The username of the customer user to login to the system.

Password The password of the customer user. This will be auto-generated if left empty.

Email * The email address of the customer user.

CustomerID * The ID of the customer company the customer user belongs to. Select a customer from the list of Customers.

Phone The phone number of the customer user.

Fax The fax number of the customer user.

Mobile The cellphone number of the customer user.

Street The street name of the customer.

Zip The zip code of the customer.

City The headquarter city of the customer.

Country The country of the customer user.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

Valid * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

See also:

It is possible to assign multiple customers to customer users via the Customer Users ▯ Customers screen.

Customer User Back Ends

The system works with many customer user data attributes such as username, email address, phone number, etc. These attributes are displayed in both the agent and the external interface, and also used

---

for the authentication of customer users.

Customer data used or displayed within the system is highly customizable. The user login and the email address are always needed for customer authentication.

You can use two types of customer back end: database or LDAP. If you already have another customer back end (e.g. SAP), it is possible to write a module that uses it.

The administrator interface does not support the configuration of external back ends. Administrators need to edit the file `Kernel/Config.pm` by copying and pasting code snippets from `Kernel/Config/Defaults.pm` manually.

> **Warning:** Do not modify the file `Kernel/Config/Defaults.pm`, it will be overwritten after upgrading the system! Copy and paste the code snippets into `Kernel/Config.pm` instead.

Database

This is the default customer user back end for new installations. The example below shows the configuration of a database customer back end, which uses customer user data stored in the database table `customer_user`.

```perl
    # CustomerUser
    # (customer user database backend and settings)
    $Self->{CustomerUser} = {
        Name   => Translatable('Database Backend'),
        Module => 'Kernel::System::CustomerUser::DB',
        Params => {

            # if you want to use an external database, add the
            # required settings
#            DSN  => 'DBI:odbc:yourdsn',
#            Type => 'mssql', # only for ODBC connections
#            DSN => 'DBI:mysql:database=customerdb;host=customerdbhost',
#            User => '',
#            Password => '',
            Table => 'customer_user',
#            ForeignDB => 0,    # set this to 1 if your table does not have create_
→time, create_by, change_time and change_by fields

            # CaseSensitive defines if the data storage of your DBMS is case sensitive␣
→and will be
            # preconfigured within the database driver by default.
            # If the collation of your data storage differs from the default settings,
            # you can set the current behavior ( either 1 = CaseSensitive or 0 =␣
→CaseINSensitive )
            # to fit your environment.
            #
#            CaseSensitive => 0,

            # SearchCaseSensitive will control if the searches within the data␣
→storage are performed
            # case sensitively (if possible) or not. Change this option to 1, if you␣
→want to search case sensitive.
            # This can improve the performance dramatically on large databases.
            SearchCaseSensitive => 0,
        },
```

```
        # customer unique id
        CustomerKey => 'login',

        # customer #
        CustomerID    => 'customer_id',
        CustomerValid => 'valid_id',

        # The last field must always be the email address so that a valid
        #   email address like "John Doe" <john.doe@domain.com> can be constructed␣
→from the fields.
        CustomerUserListFields => [ 'first_name', 'last_name', 'email' ],

#        CustomerUserListFields => ['login', 'first_name', 'last_name', 'customer_id',␣
→'email'],
        CustomerUserSearchFields          => [ 'login', 'first_name', 'last_name',␣
→'customer_id' ],
        CustomerUserSearchPrefix          => '*',
        CustomerUserSearchSuffix          => '*',
        CustomerUserSearchListLimit       => 250,
        CustomerUserPostMasterSearchFields => ['email'],
        CustomerUserNameFields            => [ 'title', 'first_name', 'last_name' ],
        CustomerUserEmailUniqCheck        => 1,

#        # Configures the character for joining customer user name parts. Join single␣
→space if it is not defined.
#        # CustomerUserNameFieldsJoin => '',

#        # show now own tickets in customer panel, CompanyTickets
#        CustomerUserExcludePrimaryCustomerID => 0,
#        # generate auto logins
#        AutoLoginCreation => 0,
#        # generate auto login prefix
#        AutoLoginCreationPrefix => 'auto',
#        # admin can change customer preferences
#        AdminSetPreferences => 1,
        # use customer company support (reference to company, See CustomerCompany␣
→settings)
        CustomerCompanySupport => 1,
        # cache time to live in sec. - cache any database queries
        CacheTTL => 60 * 60 * 24,
#        # Consider this source read only.
#        ReadOnly => 1,
        Map => [

            # Info about dynamic fields:
            #
            # Dynamic Fields of type CustomerUser can be used within the mapping (see␣
→example below).
            # The given storage (third column) then can also be used within the␣
→following configurations (see above):
            # CustomerUserSearchFields, CustomerUserPostMasterSearchFields,␣
→CustomerUserListFields, CustomerUserNameFields
            #
            # Note that the columns 'frontend' and 'readonly' will be ignored for␣
→dynamic fields.

            # note: Login, Email and CustomerID needed!
```

```
            # var, frontend, storage, shown (1=always,2=lite), required, storage-type,
↪  http-link, readonly, http-link-target, link class(es)
            [ 'UserTitle',       Translatable('Title or salutation'), 'title',      ␣
↪    1, 0, 'var', '', 0, undef, undef ],
            [ 'UserFirstname',   Translatable('Firstname'),           'first_name', ␣
↪    1, 1, 'var', '', 0, undef, undef ],
            [ 'UserLastname',    Translatable('Lastname'),            'last_name',  ␣
↪    1, 1, 'var', '', 0, undef, undef ],
            [ 'UserLogin',       Translatable('Username'),            'login',      ␣
↪    1, 1, 'var', '', 0, undef, undef ],
            [ 'UserPassword',    Translatable('Password'),            'pw',         ␣
↪    0, 0, 'var', '', 0, undef, undef ],
            [ 'UserEmail',       Translatable('Email'),               'email',      ␣
↪    1, 1, 'var', '', 0, undef, undef ],
#           [ 'UserEmail',       Translatable('Email'),               'email',      ␣
↪    1, 1, 'var', '[% Env("CGIHandle") %]?Action=AgentTicketCompose;ResponseID=1;
↪TicketID=[% Data.TicketID | uri %];ArticleID=[% Data.ArticleID | uri %]', 0, '',
↪'AsPopup OTOBOPopup_TicketAction' ],
            [ 'UserCustomerID',  Translatable('CustomerID'),          'customer_id',␣
↪    0, 1, 'var', '', 0, undef, undef ],
#           [ 'UserCustomerIDs', Translatable('CustomerIDs'),         'customer_ids
↪',   1, 0, 'var', '', 0, undef, undef ],
            [ 'UserPhone',       Translatable('Phone'),               'phone',      ␣
↪    1, 0, 'var', '', 0, undef, undef ],
            [ 'UserFax',         Translatable('Fax'),                 'fax',        ␣
↪    1, 0, 'var', '', 0, undef, undef ],
            [ 'UserMobile',      Translatable('Mobile'),              'mobile',     ␣
↪    1, 0, 'var', '', 0, undef, undef ],
            [ 'UserStreet',      Translatable('Street'),              'street',     ␣
↪    1, 0, 'var', '', 0, undef, undef ],
            [ 'UserZip',         Translatable('Zip'),                 'zip',        ␣
↪    1, 0, 'var', '', 0, undef, undef ],
            [ 'UserCity',        Translatable('City'),                'city',       ␣
↪    1, 0, 'var', '', 0, undef, undef ],
            [ 'UserCountry',     Translatable('Country'),             'country',    ␣
↪    1, 0, 'var', '', 0, undef, undef ],
            [ 'UserComment',     Translatable('Comment'),             'comments',   ␣
↪    1, 0, 'var', '', 0, undef, undef ],
            [ 'ValidID',         Translatable('Valid'),               'valid_id',   ␣
↪    0, 1, 'int', '', 0, undef, undef ],

            # Dynamic field example
#           [ 'DynamicField_Name_X', undef, 'Name_X', 0, 0, 'dynamic_field', undef, 0,
↪  undef, undef ],
        ],

        # default selections
        Selections => {

#           UserTitle => {
#               'Mr.' => Translatable('Mr.'),
#               'Mrs.' => Translatable('Mrs.'),
#           },
        },
    };
```

If you want to customize the customer user data, change the columns or add new ones to the
customer_user table in the database.

For example, to add a new field for room number:

1. Add a new column `room` to table `customer_user`.

   MySQL or MariaDB:

   ```
   root> mysql -u root -p -e 'ALTER TABLE otobo.customer_user ADD room VARCHAR (250)'
   ```

   PostgreSQL (from the `/opt/otobo` directory):

   ```
   otobo> psql -c 'ALTER TABLE customer_user ADD COLUMN room varchar(250)'
   ```

2. Copy the `$Self->{CustomerUser}` section from `Kernel/Config/Defaults.pm` into `Kernel/Config.pm`.

3. Add the new column to the `Map` array.

   ```
   [ 'UserRoom', 'Room', 'room', 0, 1, 'var', '', 0, undef, undef ],
   ```

   You can set the HTTP link target and link class (the last two keys) to `undef` in map array elements, if they are not to be used. These keys add `target=""` and `class=""` attributes to the HTTP link element, respectively. They are ignored if HTTP link is not set (it is `''` in this example).

   Note: It is recommended to always use English words for names.

   See also:

   Names can be translated into other languages with custom translation files. See the Custom Translation File chapter in the developer manual.

LDAP

If you have an LDAP directory with your customer user data, you can use it as the customer user back end. The example below shows the configuration of a LDAP customer user back end.

```
# CustomerUser
# (customer user ldap backend and settings)
    $Self->{CustomerUser} = {
        Name => 'LDAP Backend',
        Module => 'Kernel::System::CustomerUser::LDAP',
        Params => {
            # ldap host
            Host => 'bay.csuhayward.edu',
            # ldap base dn
            BaseDN => 'ou=seas,o=csuh',
            # search scope (one|sub)
            SSCOPE => 'sub',
            # The following is valid but would only be necessary if the
            # anonymous user does NOT have permission to read from the LDAP tree
            UserDN => '',
            UserPw => '',
            # in case you want to add always one filter to each ldap query, use
            # this option. e. g. AlwaysFilter => '(mail=*)' or AlwaysFilter =>
→'(objectclass=user)'
            AlwaysFilter => '',
            # if the charset of your ldap server is iso-8859-1, use this:
#            # SourceCharset => 'iso-8859-1',
```

```
            # die if backend can't work, e. g. can't connect to server
            Die => 0,
            # Net::LDAP new params (if needed - for more info see perldoc Net::LDAP)
            Params => {
                port    => 389,
                timeout => 120,
                async   => 0,
                version => 3,
            },
        },
        # customer unique id
        CustomerKey => 'uid',
        # customer #
        CustomerID => 'mail',
        CustomerUserListFields => ['cn', 'mail'],
        CustomerUserSearchFields => ['uid', 'cn', 'mail'],
        CustomerUserSearchPrefix => '',
        CustomerUserSearchSuffix => '*',
        CustomerUserSearchListLimit => 250,
        CustomerUserPostMasterSearchFields => ['mail'],
        CustomerUserNameFields => ['givenname', 'sn'],
        # Configures the character for joining customer user name parts. Join single␣
↪space if it is not defined.
        CustomerUserNameFieldsJoin => '',
        # show customer user and customer tickets in the external interface
        CustomerUserExcludePrimaryCustomerID => 0,
        # add a ldap filter for valid users (expert setting)
#        # CustomerUserValidFilter => '(!(description=gesperrt))',
        # admin can't change customer preferences
        AdminSetPreferences => 0,
        # cache time to live in sec. - cache any ldap queries
#        CacheTTL => 0,
        Map => [
            # note: Login, Email and CustomerID needed!
            # var, frontend, storage, shown (1=always,2=lite), required, storage-type,
↪ http-link, readonly, http-link-target, link class(es)
            [ 'UserTitle',      Translatable('Title or salutation'), 'title',      ␣
↪      1, 0, 'var', '', 1, undef, undef ],
            [ 'UserFirstname',   Translatable('Firstname'),          'givenname',  ␣
↪      1, 1, 'var', '', 1, undef, undef ],
            [ 'UserLastname',    Translatable('Lastname'),           'sn',         ␣
↪      1, 1, 'var', '', 1, undef, undef ],
            [ 'UserLogin',       Translatable('Username'),           'uid',        ␣
↪      1, 1, 'var', '', 1, undef, undef ],
            [ 'UserEmail',       Translatable('Email'),              'mail',       ␣
↪      1, 1, 'var', '', 1, undef, undef ],
            [ 'UserCustomerID',  Translatable('CustomerID'),         'mail',       ␣
↪      0, 1, 'var', '', 1, undef, undef ],
            # [ 'UserCustomerIDs', Translatable('CustomerIDs'),       'second_
↪customer_ids', 1, 0, 'var', '', 1, undef, undef ],
            [ 'UserPhone',       Translatable('Phone'),              'telephonenumber
↪',    1, 0, 'var', '', 1, undef, undef ],
            [ 'UserAddress',     Translatable('Address'),            'postaladdress',
↪      1, 0, 'var', '', 1, undef, undef ],
            [ 'UserComment',     Translatable('Comment'),            'description', ␣
↪      1, 0, 'var', '', 1, undef, undef ],

            # this is needed, if "SMIME::FetchFromCustomer" is active
```

```
        # [ 'UserSMIMECertificate', 'SMIMECertificate', 'userSMIMECertificate', 0, 1,
↪ 'var', '', 1, undef, undef ],

        # Dynamic field example
        # [ 'DynamicField_Name_X', undef, 'Name_X', 0, 0, 'dynamic_field', undef,␣
↪0, undef, undef ],
    ],
};
```

To activate and configure the LDAP back end:

1. Copy the `$Self->{CustomerUser}` section from `Kernel/Config/Defaults.pm` into `Kernel/Config.pm`.

2. Remove the comments (# characters) from the beginning of the lines.

If additional customer user attributes are stored in your LDAP directory, such as a manager name, a mobile phone number, or a department, this information can be displayed in OTOBO.

To display additional customer user attributes from LDAP directory:

1. Expand the `Map` array in `Kernel/Config.pm` with the entries for these attributes.

```
[ 'UserMobilePhone', 'Mobile Phone', 'mobilephone', 1, 0, 'var', '', 1, undef,␣
↪undef ],
```

Note: It is recommended to always use English words for names.

See also:

Names can be translated into other languages with custom translation files. See the Custom Translation File chapter in the developer manual.

Multiple Customer User Back Ends

If you want to use more than one customer user data source, the `CustomerUser` configuration parameter should be expanded with a number, like `CustomerUser1` and `CustomerUser2`.

The following configuration example shows usage of both a database and an LDAP customer user back end.

```
# Data source 1: customer user database back end and settings.
    $Self->{CustomerUser1} = {
        Name    => 'Database Backend',
        Module => 'Kernel::System::CustomerUser::DB',
        Params => {
            DSN => 'DBI:odbc:yourdsn',
            DSN => 'DBI:mysql:database=customerdb;host=customerdbhost',
            User => '',
            Password => '',
            Table => 'customer_user',
        },
        # Other setting here.
    };

# Data source 2: customer user LDAP back end and settings.
    $Self->{CustomerUser2} = {
```

```
        Name => 'LDAP Backend',
        Module => 'Kernel::System::CustomerUser::LDAP',
        Params => {
            Host => 'bay.csuhayward.edu',
            BaseDN => 'ou=seas,o=csuh',
            SSCOPE => 'sub',
            UserDN => '',
            UserPw => '',
            AlwaysFilter => '',
            Die => 0,
            Params => {
                port    => 389,
                timeout => 120,
                async   => 0,
                version => 3,
            },
        },
        # Other setting here.
    };
```

It is possible to integrate up to 10 different customer back ends. Use the Customer Users screen to view or edit (assuming write access is enabled) all customer user data.

Customer User Data in Dynamic Fields

Sometimes it can be useful to also store customer user data directly in dynamic fields of a ticket, for example to create special statistics on this data.

The dynamic field values are set, when a ticket is created or when the customer user of a ticket is changed. The values of the dynamic fields are taken from the customer user data. This works for all back ends, but is especially useful for LDAP back ends.

To activate this optional feature:

1. Activate the setting `Ticket::EventModulePost###4100-DynamicFieldFromCustomerUser`.

2. Activate the setting `DynamicFieldFromCustomerUser::Mapping`. This setting contains the configuration of which customer user field entry should be stored in which ticket dynamic field.

3. Create the dynamic fields, if the dynamic fields are not present in the system yet.

4. Enable the dynamic fields in setting `Ticket::Frontend::AgentTicketFreeText###DynamicField`, so that they can be set manually.

Note: The dynamic field must not be enabled in the following settings:

- `Ticket::Frontend::AgentTicketPhone###DynamicField`

- `Ticket::Frontend::AgentTicketEmail###DynamicField`

- `Ticket::Frontend::AgentTicketCustomer###DynamicField`

If they were, they would have precedence over the automatically set values.

1.3.7 Customer Users ⧄ Customers

In an organization, for example, which views its customers as it's departments and teams, a customer user may have to have access to multiple customers tickets for controlling purposes. Maybe you have partners who represent several different companies, or a corporation wants to have a look at all the requests of their subsidiaries.

For all situations, OTOBO provides the means. Aside from a primary customer, your customer users can gain access to multiple customer tickets as defined by you.

Use this screen to add one or more customer users to one or more customers. To use this function, at least one customer user and one customer need to have been added to the system. The management screen is available in the Customer Users ⧄ Customers module of the Users, Groups & Roles group.

Fig. 1.130: Manage Customer User-Customer Relations

Manage Customer Users ⧄ Customers Relations

Note: This module is for assigning a customer user to additional customer records. The primary customer is assigned via the Customer User Settings.

To assign some customers to a customer user:

1. Click on a customer user in the Customer Users column.
2. Select the customers you would like to assign to the customer user.
3. Click on the Save or Save and finish button.

Fig. 1.131: Change Customer Relations for Customer User

To assign some customer users to a customer:

1. Click on a customer in the Customers column.
2. Select the customer users you would like to assign to the customer.
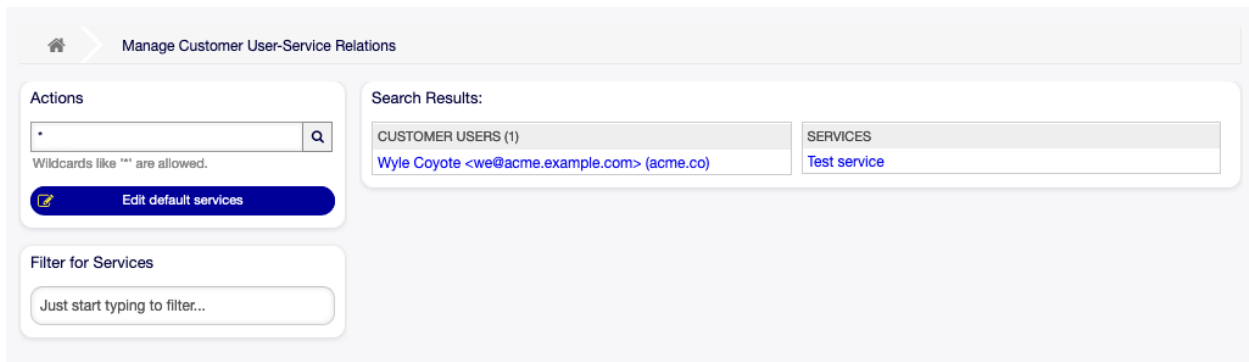3. Click on the Save or Save and finish button.
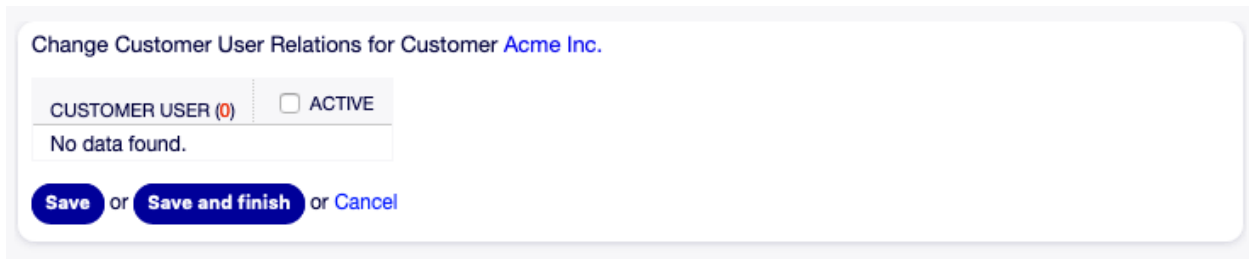
Fig. 1.132: Change Customer User Relations for Customer
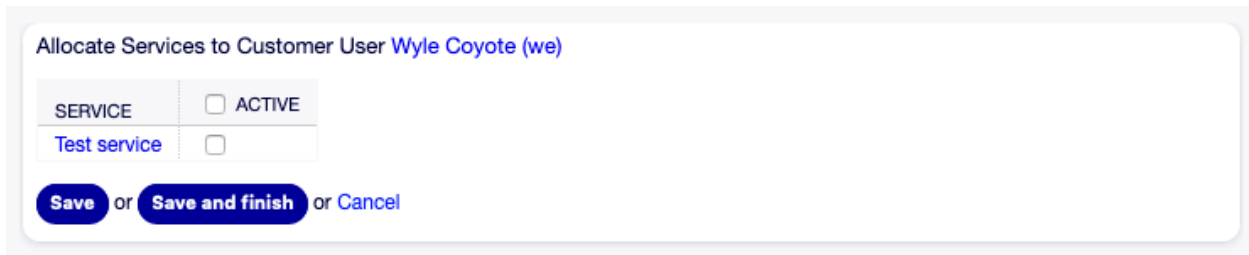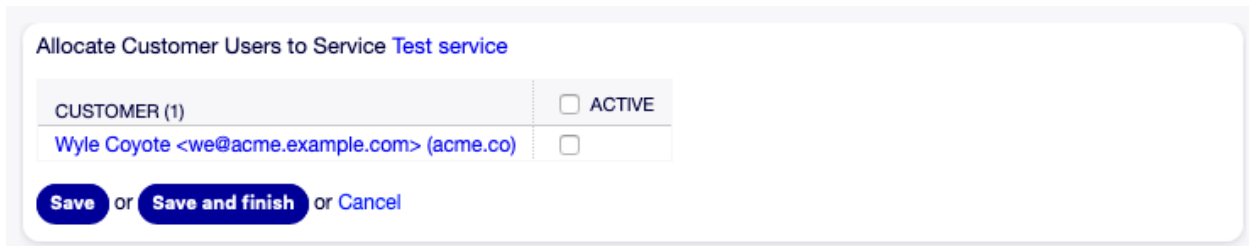

Note: If several customer users or customers are added to the system, use the search box to find a particular customer user or customer. Only the first 1000 customer users and customers are listed by default.

Multiple customer users or customers can be assigned in both screens at the same time. Additionally clicking on a customer user or clicking on a customer in the relations screen will open the Edit Customer User screen or the Edit Customer screen accordingly.

Warning: Accessing a customer user or a customer provides no back link to the relations screen.

Note: By setting a checkbox in the header of a column will set all the checkboxes in the selected column.

1.3.8 Customer Users ⮃ Groups

Customer users shouldn't need to be bothered with the internal workings of your service desk. A single point of contact request can trigger several processes within your organization, all of which having the customer user information attached and are visible to the customer.

OTOBO allows you to assign group permissions to customer users. Access works just the same as for agents, preventing a customer from modifying and viewing a request. Thus allowing the customer to focus on the results of the original communication and funneling the discussion through one ticket.

See also:

Assign a group to an entire customer using Customers ⮃ Groups.

Use this screen to add one or more customer users to one or more groups. To use this function, at least one customer user and one group need to have been added to the system. The management screen is available in the Customers Users ⮃ Groups module of the Users, Groups & Roles group.

Customer group support needs to be enabled in at least one customer user back end to use this function. For the default OTOBO back end, this can be enabled in the system configuration by clicking on the Enable it here! button.

Note: To enable this feature in systems using a directory server or multiple non-default back ends, a custom configuration file needs to be placed in `Kernel/Config/Files` (for example named

---
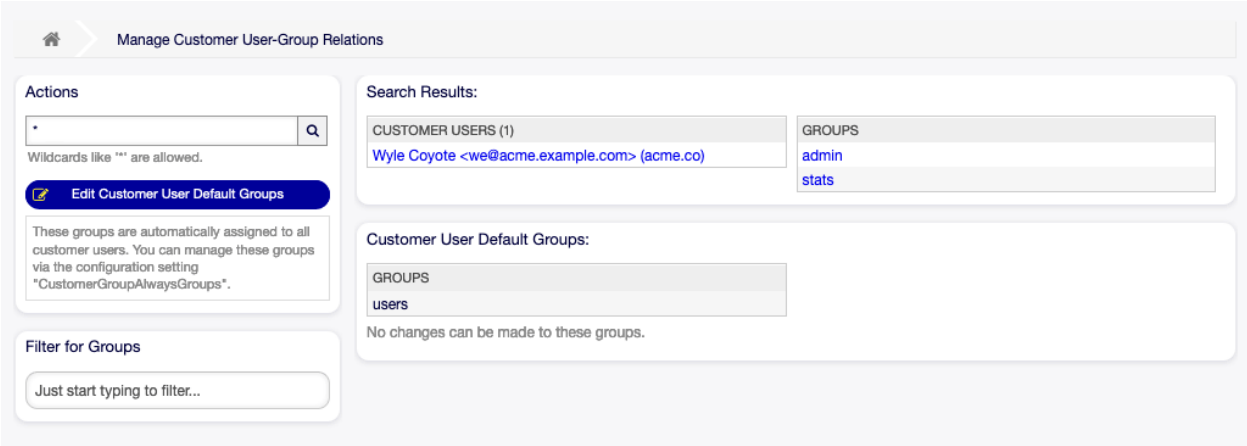
Fig. 1.133: Manage Customer User-Group Relations
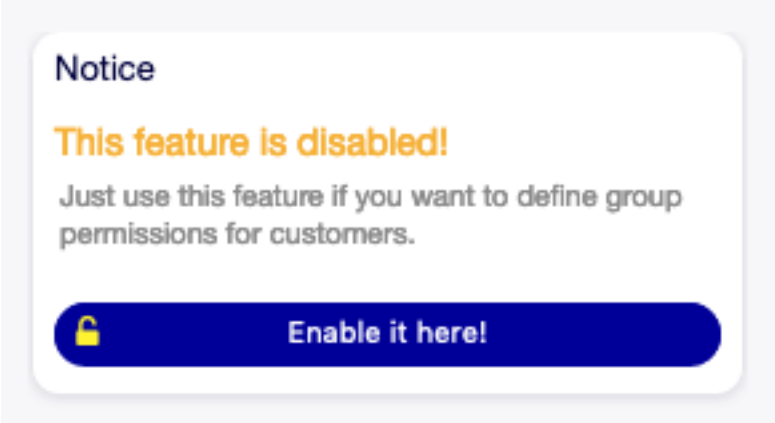
Fig. 1.134: Enable Customer Group Feature

`ZZZ_CustomerBackend.pm`). Once activated, all customer users from this back end will require group assignment.

Warning: After making changes to the back end, the server cache will be deleted, which may cause a temporary drop in performance.

Manage Customer Users ⬚ Groups Relations

To assign some groups to a customer user:

1. Click on a customer user in the Customer Users column.
2. Select the permissions you would like to connect the customer user to groups with.
3. Click on the Save or Save and finish button.
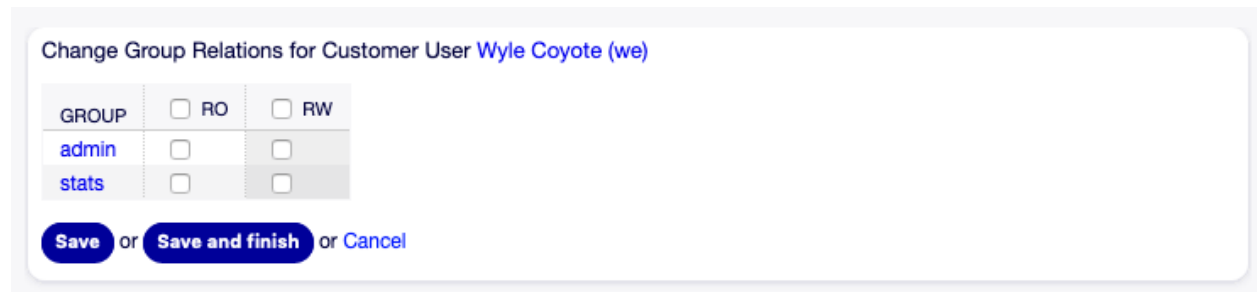
Fig. 1.135: Change Group Relations for Customer User

To assign some customer users to a group:

1. Click on a group in the Groups column.
2. Select the permissions you would like to connect the group to customer users with.
3. Click on the Save or Save and finish button.
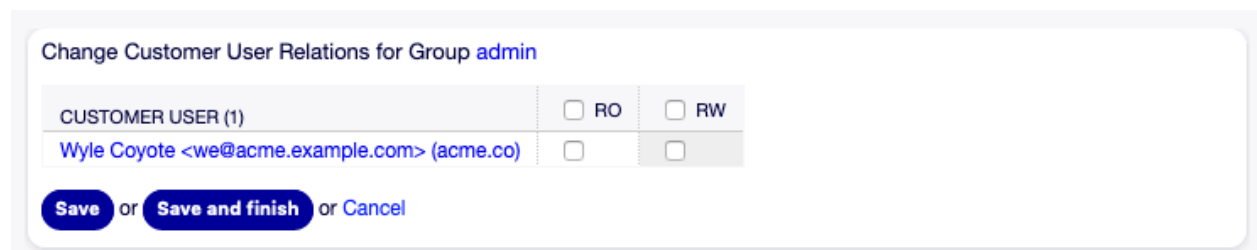
Fig. 1.136: Change Customer User Relations for Group

To change customer user default groups:

1. Click on the Edit Customer User Default Groups button in the left sidebar.
2. Add or modify groups in setting `CustomerGroupAlwaysGroups`.
3. Deploy the modified system configurations.
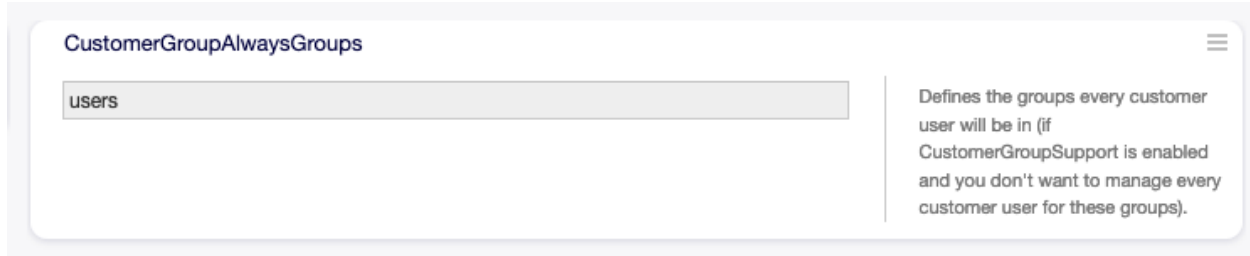
Fig. 1.137: `CustomerGroupAlwaysGroups` System Configuration Screen

These groups are automatically assigned to all customer users.

Note: If several customer users or groups are added to the system, use the search box to find a particular customer user or use the filter box to find a particular group by just typing the name to filter.

Multiple customer users or groups can be assigned in both screens at the same time. Additionally clicking on a customer user or clicking on a group in the relations screen will open the Edit Customer User screen or the Edit Group screen accordingly.

Warning: Accessing a customer user or a group provides no back link to the relations screen.

Customer Users ⮂ Groups Relations Reference

When assigning a customer user to a group or vice versa, several permissions can be set as connection between a customer user and a group. The following permissions are available by default:

ro Read only access to the resource.

rw Full read and write access to the resource.

Note: By setting a checkbox in the header of a column will set all the checkboxes in the selected column.

1.3.9 Customer Users ⮂ Services

Use this screen to add one or more customer users to one or more services. To use this function, at least one customer user and one service need to have been added to the system. The management screen is available in the Customers Users ⮂ Services module of the Users, Groups & Roles group.

Manage Customers Users ⮂ Services Relations

To allocate some services to a customer user:

1. Click on a customer user in the Customer Users column.

2. Select the services you would like to allocate to the customer user.

3. Click on the Save or Save and finish button.

1.3. Users, Groups & Roles

Fig. 1.138: Manage Customer User-Service Relations

Fig. 1.139: Change Service Relations for Customer User

To allocate some customer users to a service:

1. Click on a service in the Services column.

2. Select the customer users you would like to allocate to the service.

3. Click on the Save or Save and finish button.

Fig. 1.140: Change Customer User Relations for Service

Note: If several customer users or services are added to the system, use the search box to find a particular customer user or use the filter box to find a particular service by just typing the name to filter.

Multiple customer users or services can be assigned in both screens at the same time. Additionally clicking on a customer user or clicking on a service in the relations screen will open the Edit Customer User screen or the Edit Service screen accordingly.

> **Warning:** Accessing a customer user or a service provides no back link to the relations screen.

> **Note:** By setting a checkbox in the header of a column will set all the checkboxes in the selected column.

Manage Default Services

It is possible to add default services, so that all customer users may access them. This prevents having to add each service to each customer user.

To edit the default services:

1. Click on the Edit default services button in the left sidebar.

2. Select the services which should be selectable for all customer users.

3. Click on the Save or Save and finish button.
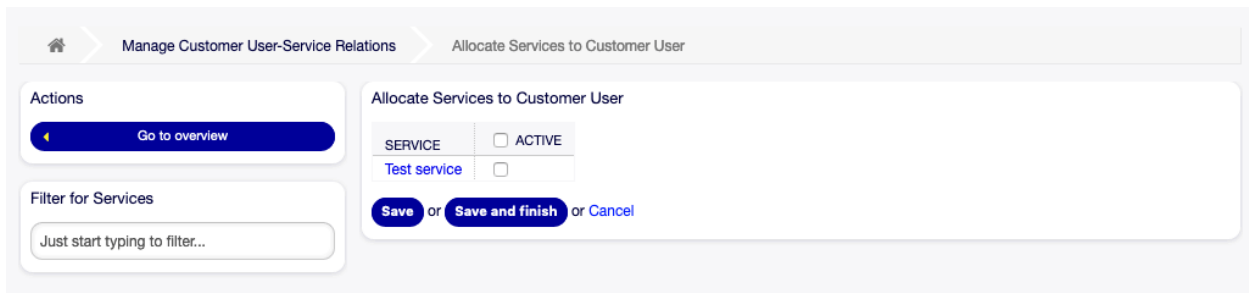
Fig. 1.141: Allocate Services to Customer User Screen

> **Warning:** Mixing default services and customer specific services can be confusing. If a customer user has specific services assigned, then the default services will be not applied.

1.3.10 Groups

Use this screen to add groups to the system. A fresh OTOBO installation contains some default groups. The group management screen is available in the Groups module of the Users, Groups & Roles group.

Manage Groups

To add a group:

1. Click on the Add Group button in the left sidebar.

2. Fill in the required fields.

3. Click on the Save button.

Fig. 1.142: Group Management Screen

Fig. 1.143: Add Group Screen

> **Warning:** Groups can not be deleted from the system. They can only be deactivated by setting the Validity option to invalid or invalid-temporarily.

To edit a group:

1. Click on a group in the list of groups.

2. Modify the fields.

3. Click on the Save or Save and finish button.

Fig. 1.144: Edit Group Screen

> **Note:** If several groups are added to the system, use the filter box to find a particular group by just typing the name to filter.

Group Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

> **Note:** Renaming a group does not affect permissions previously given. When group1 is now called group2, then all the permissions are the same for the users which used to be assigned to group1. This result is because OTOBO uses IDs for the relationship, and not the name.

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

> **Note:** Invalidating a group does not remove the permissions from the user, but only makes them invalid. If you reactivate this group, even with a new name, the permissions take effect.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

Default Groups

Every agent's account should belong to at least one group or role. In a fresh installation, there are some pre-defined groups available:

admin Allowed to perform administrative tasks in the system.

stats Qualified to access the stats module of OTOBO and generate statistics.

users Agents should belong to this group, with read and write permissions. They can then access all functions of the ticket system.

1.3.11 Roles

Use this screen to add roles to the system. A fresh OTOBO installation contains no roles by default. The role management screen is available in the Roles module of the Users, Groups & Roles group.
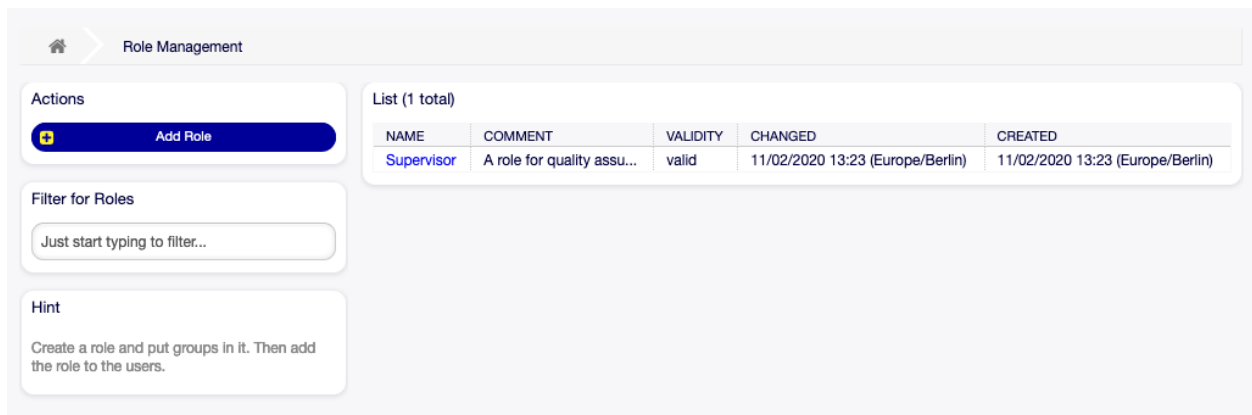
Fig. 1.145: Role Management Screen

Manage Roles

To add a role:

1. Click on the Add Role button in the left sidebar.
2. Fill in the required fields.
3. Click on the Save button.

> **Warning:** Roles can not be deleted from the system. They can only be deactivated by setting the Validity option to invalid or invalid-temporarily.

To edit a role:

1. Click on a role in the list of roles.
2. Modify the fields.
3. Click on the Save or Save and finish button.

Fig. 1.146: Add Role Screen

Fig. 1.147: Edit Role Screen

Note: If several roles are added to the system, use the filter box to find a particular role by just typing the name to filter.

Role Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

1.3.12 Roles ⬌ Groups

Use this screen to add one or more roles to one or more groups. To use this function, at least one role and one group need to have been added to the system. The management screen is available in the Roles ⬌ Groups module of the Users, Groups & Roles group.
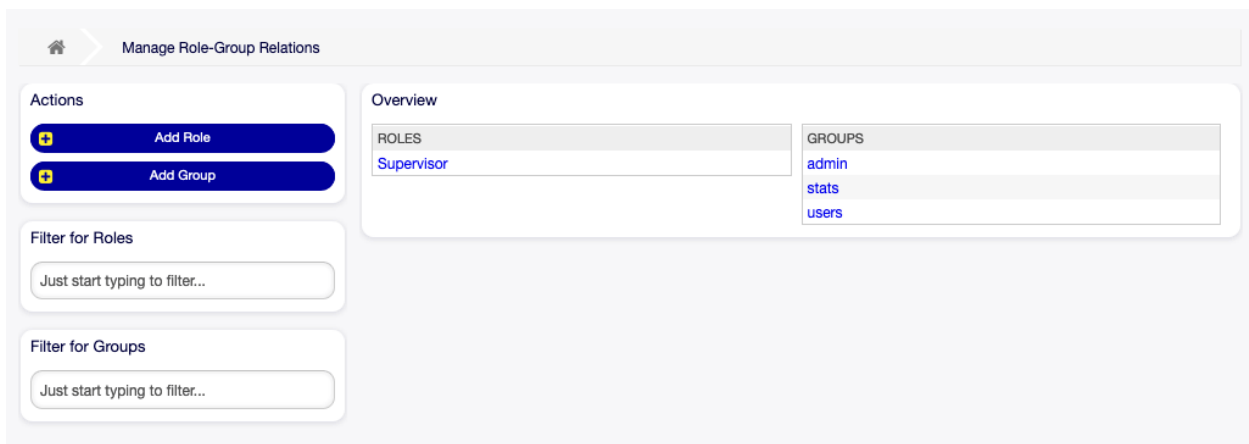
Fig. 1.148: Manage Role-Group Relations

Manage Roles ⮀ Groups Relations

To assign some groups to a role:

1. Click on a role in the Roles column.

2. Select the permissions you would like to connect the role to groups with.

3. Click on the Save or Save and finish button.
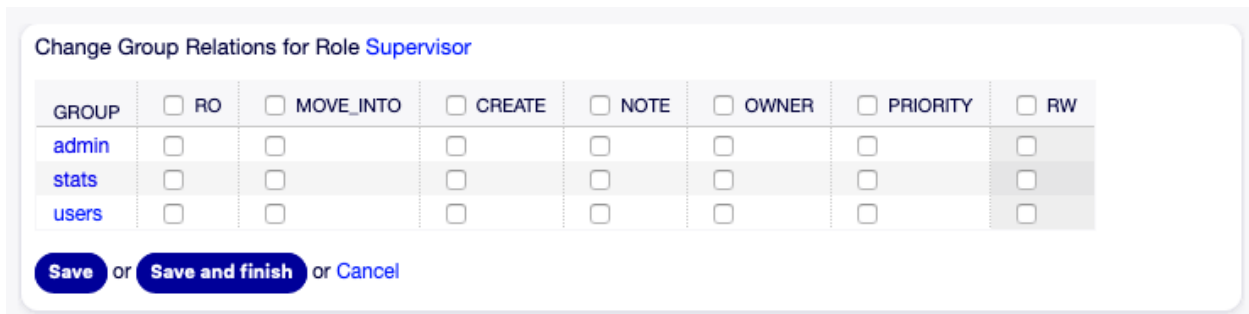
Fig. 1.149: Change Group Relations for Role

To assign some roles to a group:

1. Click on a group in the Groups column.

2. Select the permissions you would like to connect the group to roles with.

3. Click on the Save or Save and finish button.

Note: If several roles or groups are added to the system, use the filter box to find a particular role or group by just typing the name to filter.

Multiple roles or groups can be assigned in both screens at the same time. Additionally clicking on a role or clicking on a group in the relations screen will open the Edit Role screen or the Edit Group screen accordingly.

Fig. 1.150: Change Role Relations for Group

Warning: Accessing a role or a group provides no back link to the relations screen.

Roles ⧖ Groups Relations Reference

When assigning a role to a group or vice versa, several permissions can be set as connection between a role and a group. The following permissions are available by default:

ro Read only access to the ticket in this group/queue.

move_into Permissions to move tickets into this group/queue.

create Permissions to create tickets in this group/queue.

note Permissions to add notes to tickets in this group/queue.

owner Permissions to change the owner of tickets in this group/queue.

priority Permissions to change the ticket priority in this group/queue.

chat_observer Users with this permission type will only be able to observe chats in a channel after they have been invited.

chat_participant Users with this permission type will be able to take part in a chat, but only after they get invited to it.

chat_owner Users with this permission type will be able to accept chat customer/public requests and do all kinds of observer and participant actions on a chat.

rw Full read and write access to the tickets in this group/queue.

See also:

Not all available permissions are shown by default. See `System::Permission` setting for permissions that can be added. These additional permissions can be added:

stats Gives access to the stats page.

bounce Permissions to bounce an email message (with bounce button in Ticket Zoom screen).

compose Permissions to compose an answer for a ticket.

customer Permissions to change the customer of a ticket.

forward Permissions to forward a message (with the forward button).

pending Permissions to set a ticket to pending.

phone Permissions to add a phone call to a ticket.

responsible Permissions to change the responsible agent for a ticket.

Note: By setting a checkbox in the header of a column will set all the checkboxes in the selected column. By setting the checkbox in the last rw column will set all the checkboxes in the selected row.

1.4 Processes & Automation

Working effectively with tickets requires more than just the possibility to manually change their state, add information, communicate with other persons and finally close the tickets.

Automation frees agents from reoccurring and time-consuming tasks and allows them to focus on activities, where their interaction is required.

Process management guides customer users and agents through ticket creation to closure ensuring that tickets take defined workflows at any time.

OTOBO offers many options to automate tasks based on events, time, external systems and defined processes. OTOBO also includes the possibility to add individual information types to tickets and help agents to lower their error rate when working with tickets by allowing only defined activities for tickets in specific states.

1.4.1 Access Control Lists (ACL)

Working with tickets can become a bewildering task. Many options are given to process, or close tickets, even if they are not needed in the current state of a ticket or due to the role of the current agent. Hiding unneeded entries cleans up the menu bar and gets it easier to work with, hiding values from dynamic fields or next queues lowers chance of human error.

OTOBO uses access control lists (ACL) to restrict agents and customer users on ticket options, allowing only correct and meaningful activities with a ticket. OTOBO administrators can easily generate ACLs in the graphical interface to prevent ticket closure until meeting specific requirements, prevent tickets from being moved to queues before adding the defined information and much more.

Use this screen to manage access control lists in the system. A fresh OTOBO installation contains no access control lists by default. The access control lists management screen is available in the Access Control Lists (ACL) module of the Processes & Automation group.
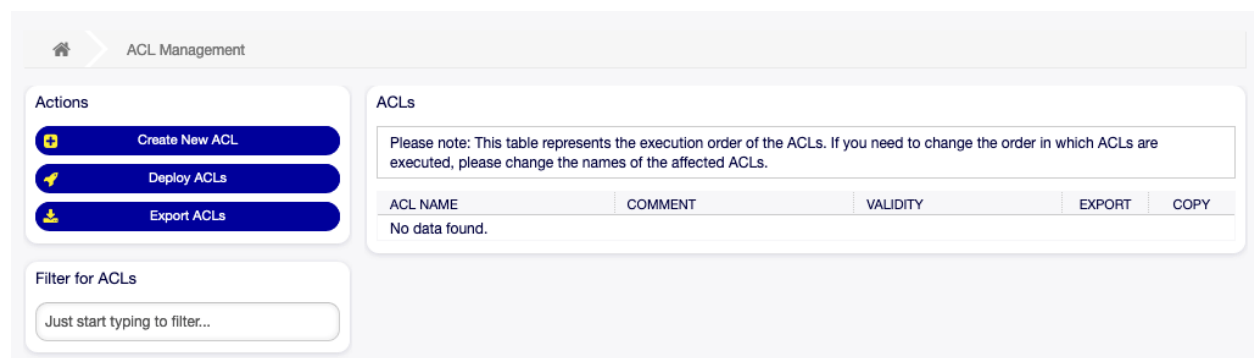
Fig. 1.151: ACL Management Screen
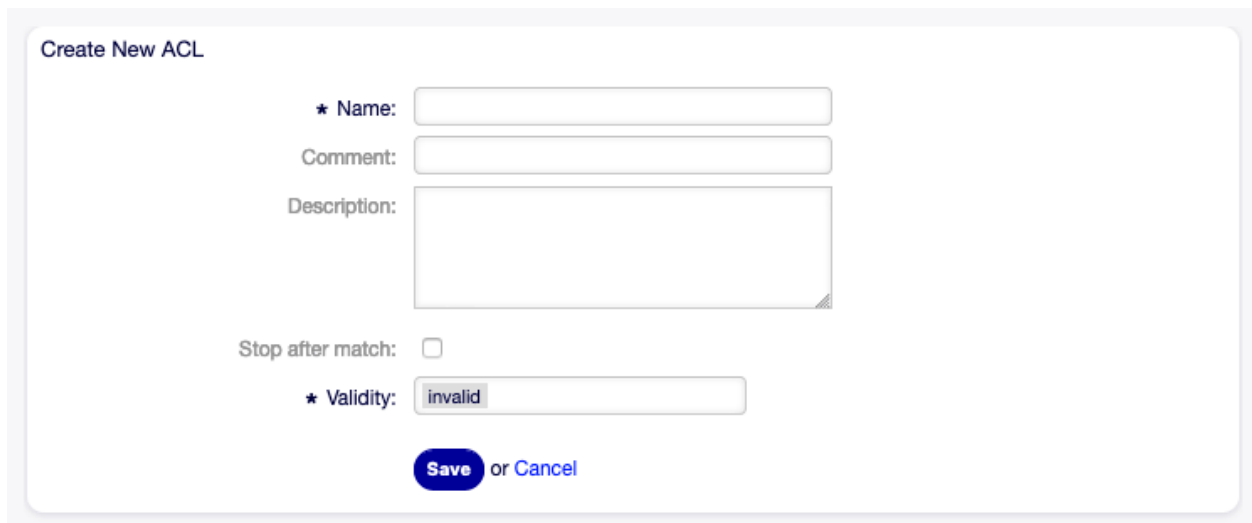
Manage Access Control Lists

Note: When creating some access control lists, please keep in mind that they are executed alphabetically as displayed in the access control lists overview.

> **Warning:** ACL restrictions will be ignored for the superuser account (UserID 1).

To create a new ACL:

1. Click on the Create New ACL button in the left sidebar.

2. Fill in the required fields.

3. Click on the Save button.

4. You will be redirected to Edit ACL screen to edit the ACL structure.

Fig. 1.152: Create New ACL Screen

To edit an ACL:

1. Click on an ACL in the list of ACLs or you are already redirected here from Create New ACL screen.

2. Modify the fields and the ACL structure.

3. Click on the Save or Save and finish button.

4. Deploy all ACLs.

To delete an ACL:

1. Click on an ACL in the list of ACLs.

2. Set the Validity option to invalid or invalid-temporarily.

3. Click on the Save button. A new Delete Invalid ACL button will appear in the left sidebar.

4. Click on the Delete Invalid ACL button.

5. Click on the Delete button in the confirmation screen.

Fig. 1.153: Edit ACL Structure Screen

6. Deploy all ACLs.

> **Warning:** ACLs are written into `ZZZACL.pm` file in Perl format. Without deploying, all ACLs are still in this cache file even if they are deleted or the Validity option is set to invalid or invalid-temporarily. Don't forget to deploy all ACLs after modifications!

To deploy all ACLs:

1. Click on the Deploy ACLs button in the left sidebar.

> **Note:** New or modified ACLs have to deploy in order to make affect the behavior of the system. Setting the Validity option to valid just indicates, which ACLs should be deployed.

To export all ACLs:

1. Click on the Export ACLs button in the left sidebar.
2. Choose a location in your computer to save the `Export_ACL.yml` file.

To import ACLs:

1. Click on the Browse… button in the left sidebar.
2. Select a previously exported `.yml` file.
3. Click on the Overwrite existing ACLs? checkbox, if you would like to overwrite the existing ACLs.
4. Click on the Import ACL configuration(s) button.
5. Deploy the imported ACLs with Deploy ACLs button.

> **Note:** If several ACLs are added to the system, use the filter box to find a particular ACL by just typing the name to filter.

> **Warning:** Changing the name of this object should be done with care, the check only provides verification for certain settings and ignores things where the name can't be verified. Some examples are dashboard filters, action control lists (ACLs), and processes (sequence flow actions) to name a few. Documentation of your setup is key to surviving a name change.

ACL Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

Description Like comment, but longer text can be added here.

Stop after match ACLs are evaluated in alphabetical order. This setting disables the evaluation of the subsequent ACLs.

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

Edit ACL Structure

The ACL definition can be split into two big parts, Match settings and Change settings. In the matching sections the ACLs contain attributes that has to be met in order to use the ACL. If the attributes defined in the ACL do not match with the attributes that are sent, then the ACL does not take any affect, but any other match ACL will. The change sections contain the rules to reduce the possible options for a ticket.

Match settings

`Properties` This section contains matching options that can be changed on the fly. For example on a ticket creation time the data of the ticket changes dynamically as the agent sets the information. If an ACL is set to match a ticket attribute then only when the matching attribute is selected the ACL will be active and might reduce other ticket attributes, but as soon as another value is selected the ACL will not take any affect.

`PropertiesDatabase` This section is similar to `Properties` but does not take changes in ticket attributes that are not saved into the database, this means that changing an attribute without submit will not make any effect. This section is not use for ticket creation screens (as tickets are not yet created in the database).

Change settings

`Possible` This section is used to reset the data to be reduce to only the elements that are set in this section.

`PossibleAdd` This section is used to add missing elements that were reduced in other ACLs. This section is only used in together with other ACLs that have `Possible` or `PossibleNot` sections.

`PossibleNot` This section is used to remove specific elements from the current data. It could be used stand alone or together with other ACLs with a `Possible` or `PossibleAdd` sections.

Modifiers

In order to make the development of ACLs easier and more powerful there is a set of so called modifiers for the attributes on each section. This modifiers are explained below:

`[Not]` This modifier is used to negate a value, for example `[Not]2 low`. Talking about priorities will be the same as to have: 1 very low, 3 normal, 4 high, 5 very high.

`[RegExp]` It is used to define a regular expression for matching several values, for example `[RegExp]low`. In this case talking about priorities is the same as 1 very low, 2 low.

`[regexp]` It is very similar to `[RegExp]` but it is case insensitive.

`[NotRegExp]` Negated regular expressions, for example `[NotRegExp]low`. Talking about priorities is the same as 3 normal, 4 high, 5 very high.

`[Notregexp]` It is very similar to `[NotRegExp]` but it is case insensitive.

ACL Examples

Move Ticket to Queue Based on Priority

This example shows you how to allow movement into a queue of only those tickets with ticket priority 5 very high.

First, it needs to have a name. In this case, it is 100-Example-ACL. Note that the ACLs will be numerically sorted before execution, so you should use the names carefully. The comment and the description fields are optional.

Fig. 1.154: 100-Example-ACL - Basic Settings

Secondly, you have a Properties section which is a filter for your tickets. All the criteria defined here will be applied to a ticket to determine if the ACL must be applied or not. In our example, a ticket will match if it is in the queue Raw and has priority 5 very high. This is also affected by changes in the form (e. g. if the ticket is the queue Raw and had a priority 3 normal at this moment the ACL will not match, but then priority drop-down is selected and the priority is changed now to 5 very high then will also match).

Lastly, a section Possible defines modifications to the screens. In this case, from the available queues, only the queue Alert can be selected in a ticket screen.

Note: Don't forget to set Validity to valid and deploy the newly created ACL.

Move Ticket to Queue Based on Priority Stored in Database

This example is very similar to the first one, but in this case only tickets in the queue Raw and with a priority 5 very high, both stored in the database will match. This kind of ACLs does not consider changes in the form before the ticket is really updated in the database.

Disable Ticket Close in Queue and Hide Close Button

This example shows how to disable the closing of tickets in the queue Raw, and hide the close button. It is possible to filter a ticket field (state) with more than one possible values to select from. It is also

---
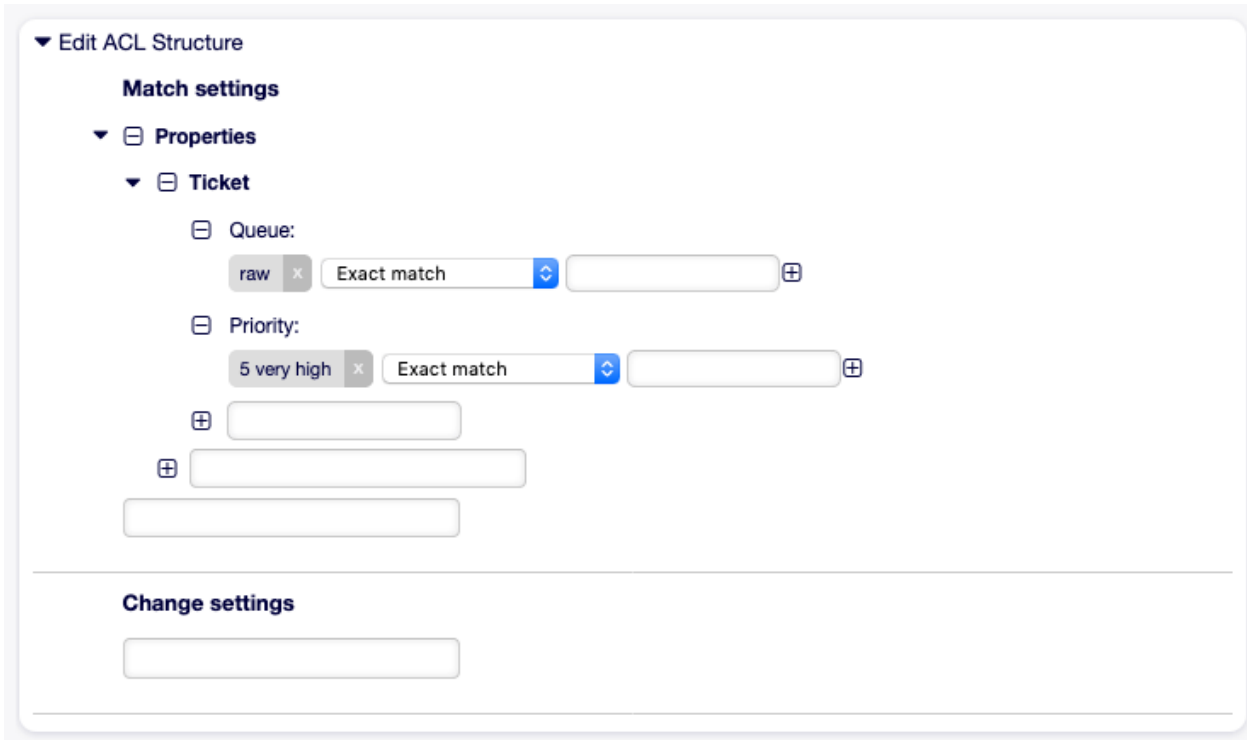
Fig. 1.155: 100-Example-ACL - Match Settings

possible to limit the actions that can be executed for a certain ticket. In this case, the ticket cannot be closed.

Remove State

This example shows how it is possible to define negative filters (the state closed successful will be removed). You can also see that not defining match properties for a ticket will match any ticket, i. e. the ACL will always be applied. This may be useful if you want to hide certain values by default, and only enable them in special circumstances (e. g. if the agent is in a specific group).

Using Regular Expressions

This example shows you how you can use regular expressions for matching tickets and for filtering the available options. This ACL only shows Hardware services for tickets that are created in queues that start with HW.

Disallow Process For Customer

This ACL restricts the process P14 in the external interface using the customer ID TheCustomerID.

Fig. 1.156: 100-Example-ACL - Change Settings

Fig. 1.157: 101-Example-ACL

Fig. 1.158: 102-Example-ACL

Fig. 1.159: 103-Example-ACL

Fig. 1.160: 104-Example-ACL

Fig. 1.161: 105-Example-ACL

Disallow Standard Reply Function For Customers in Process

This ACL prohibits using the reply button in CustomerTicketZoom if the ticket is a process ticket of the process with entity ID Process-0123456789abcdef.

Fig. 1.162: 106-Example-ACL

ACL Reference

Properties, keys and values that can be used in ACLs are highly dependent on the OTOBO installation. For example the possibilities can be extended by installing extension modules, as well as it can depend on the customer user mapping set in `Config.pm`. Therefore it is not possible to provide a full ACL reference, that contains all settings.

For properties, keys and values that can be used in ACLs, see the following example ACL in YAML format.

```
---
- ChangeBy: root@localhost
  ChangeTime: 2019-01-07 10:42:59
  Comment: ACL Reference.
  ConfigMatch:
    Properties:
      # Match properties (current values from the form).
      CustomerUser:
        UserLogin:
        - some login
        UserCustomerID:
```

```
        - some customer ID
      Group_rw:
      - some group
    DynamicField:
      # Names must be in DynamicField_<field_name> format.
      # Values for dynamic fields must always be the untranslated internal
      #   data keys specified in the dynamic field definition and not the
      #   data values shown to the user.
      DynamicField_Field1:
      - some value
      DynamicField_OtherField:
      - some value
      DynamicField_TicketFreeText2:
      - some value
      # more dynamic fields
    Frontend:
      Action:
      - AgentTicketPhone
      - AgentTicketEmail
      - ...
      - CustomerTicketZoomReply
      Endpoint:
      - ExternalFrontend::PersonalPreferences
      - ExternalFrontend::ProcessTicketCreate
      - ExternalFrontend::ProcessTicketNextStep
      - ExternalFrontend::TicketCreate
      - ExternalFrontend::TicketDetailView
      - ...
    Owner:
      UserLogin:
      - some login
      Group_rw:
      - some group
      Role:
      - admin
      # more owner attributes
    Priority:
      ID:
      - some ID
      Name:
      - some name
      # more priority attributes
    Process:
      ProcessEntityID:
      # the process that the current ticket is part of
      - Process-9c378d7cc59f0fce4cee7bb9995ee3eb
      ActivityEntityID:
      # the current activity of the ticket
      - Activity-f8b2fdebe54eeb7b147a5f8e1da5e35c
      ActivityDialogEntityID:
      # the current activity dialog that the agent/customer is using
      - ActivityDialog-aff0ae05fe6803f38de8fff6cf33b7ce
    Queue:
      Name:
      - Raw
      QueueID:
      - some ID
      GroupID:
```

```
        - some ID
      Email:
      - some email
      RealName:
      - OTOBO System
      # more queue attributes
  Responsible:
    UserLogin:
    - some login
    Group_rw:
    - some group
    Role:
    - admin
    # more responsible attributes
  Service:
    ServiceID:
    - some ID
    Name:
    - some name
    ParentID:
    - some ID
    # more service attributes
  SLA:
    SLAID:
    - some ID
    Name:
    - some name
    Calendar:
    - some calendar
    # more SLA attributes
  State:
    ID:
    - some ID
    Name:
    - some name
    TypeName:
    - some state type name
    TypeID:
    - some state type ID
    # more state attributes
  Ticket:
    Queue:
    - Raw
    State:
    - new
    - open
    Priority:
    - some priority
    Lock:
    - lock
    CustomerID:
    - some ID
    CustomerUserID:
    - some ID
    Owner:
    - some owner
    DynamicField_Field1:
    - some value
```

```
          DynamicField_MyField:
          - some value
          # more ticket attributes
        Type:
          ID:
          - some ID
          Name:
          - some name
          # more type attributes
        User:
          UserLogin:
          - some_login
          Group_rw:
          - some group
          Role:
          - admin
    PropertiesDatabase:
      # Match properties (existing values from the database).
      # Please note that Frontend is not in the database, but in the framework.
      # See section "Properties", the same configuration can be used here.
  ConfigChange:
    Possible:
      # Reset possible options (white list).
      # Hide or show Dynamicfields
      # Please just enter the name of the Dynamic Fields, they don't need any prefix.
      Form:
      - SomeDynamicFieldName
      - someOtherDynamicFieldName
      # Hide or show Standard Fields like Article (in the future Queue, Service, etc.)
      FormStd:
      # Hide or show Article in CustomerTicketMessage (including all three: Subject,␣
→Body and Attachment)
      - Article
      Action:
      # Possible action options (white list).
      - AgentTicketBounce
      - AgentTicketPhone    # only used to show/hide the Split action
      - AgentLinkObject     # only used to show/hide the Link action
      - ...
      ActivityDialog:
      # Limit the number of possible activity dialogs the agent/customer can use in a␣
→process ticket.
      - ActivityDialog-aff0ae05fe6803f38de8fff6cf33b7ce
      - ActivityDialog-429d61180a593414789a8087cc4b3c6f
      - ...
      Endpoint:
      # Limit the functions on external interface.
      - ExternalFrontend::PersonalPreferences
      - ExternalFrontend::ProcessTicketCreate
      - ExternalFrontend::ProcessTicketNextStep
      - ExternalFrontend::TicketCreate
      - ExternalFrontend::TicketDetailView
      - ...
      Process:
      # Limit the number of possible processes that can be started.
      - Process-9c378d7cc59f0fce4cee7bb9995ee3eb
      - Process-123456789012345678901223456789012
      - ...
```

```
      Ticket:
      # Possible ticket options (white list).
        Queue:
        - Raw
        - some other queue
        State:
        - some state
        Priority:
        - 5 very high
        DynamicField_Field1:
        - some value
        DynamicField_MyField:
        - some value
        # more dynamic fields
        NewOwner:
        # For ticket action screens, where the Owner is already set.
        - some owner
        OldOwner:
        # For ticket action screens, where the Owner is already set.
        - some owner
        Owner:
        # For ticket create screens, because Owner is not set yet. Please make sure,␣
→that you enter "Firstname Lastname".
        - some owner
        SLA:
        - some sla-name
        # more ticket attributes
    PossibleAdd:
      # Add options (white list).
      # See section "Possible", the same configuration can be used here.
    PossibleNot:
      # Remove options (black list).
      # See section "Possible", the same configuration can be used here.
  CreateBy: root@localhost
  CreateTime: 2019-01-07 10:42:59
  Description: This is the long description of the ACL to explain its usage.
  ID: 1
  Name: 200-ACL-Reference
  StopAfterMatch: 0
  ValidID: 3
```

1.4.2 Dynamic Fields

Beside general information required for all tickets, organizations have individual needs to add specific details to tickets. This needed information takes various formats like texts, integers, date-time and more.

OTOBO supports adding a so-called dynamic field to handle texts, integers, drop-down lists, multi-select fields, date-time, checkboxes and more. OTOBO administrators can define where those fields should be visible or editable, and of course, the dynamic fields are also available in statistics and reports.

Use this screen to manage dynamic fields in the system. A fresh OTOBO installation contains two dynamic fields by default. The dynamic field management screen is available in the Dynamic Fields module of the Processes & Automation group.
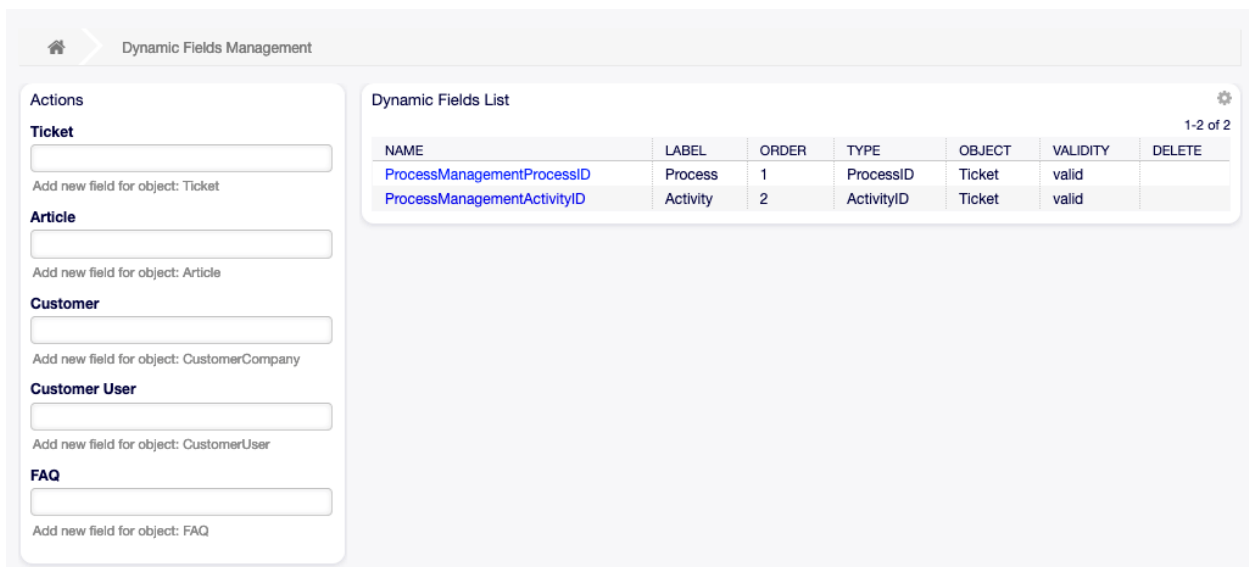
Fig. 1.163: Dynamic Field Management Screen

Manage Dynamic Fields

To create a new dynamic field:

1. Choose an object type in the left actions sidebar and select a dynamic field type from its drop-down.

2. Fill in the required fields.

3. Click on the Save button.

To edit a dynamic field:

1. Click on a dynamic field in the dynamic fields list.

2. Modify the fields.

3. Click on the Save or Save and finish button.

To delete a dynamic field:

1. Click on the trash icon in the last column of the overview table.

2. Click on the Confirm button.

Note: If several dynamic fields are added to the system, use the filter box to find a particular dynamic field by just typing the name to filter.

Dynamic Field Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Fig. 1.164: Create New Dynamic Field Screen

Fig. 1.165: Edit Dynamic Field Screen

Fig. 1.166: Delete Dynamic Field Screen

General Dynamic Field Settings

These settings are the same for all types of dynamic fields.

Fig. 1.167: Dynamic Field General Screen

Name * The name of this resource. The value should be alphabetic and numeric characters only. The name will be displayed in the overview table.

Label * This is the name to be shown on the screens where the field is active.

> **See also:**

It is possible to add translations for a dynamic field label. Label translations have to be added manually to language translation files.

Field order * This is the order in which this field will be shown on the screens where is active.

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

Field type This type has been selected in the left sidebar and can not be changed here anymore. This is a read-only field.

Object type This type has been selected in the left sidebar and can not be changed here anymore. This is a read-only field.

Note: The object type determines where the dynamic field can be used. For example a dynamic field with object type Ticket can be used only in tickets, and can not be used in articles.

The following settings are relevant only for the particular type of dynamic fields.

Checkbox Dynamic Field Settings

Checkbox dynamic field is used to store true or false value.
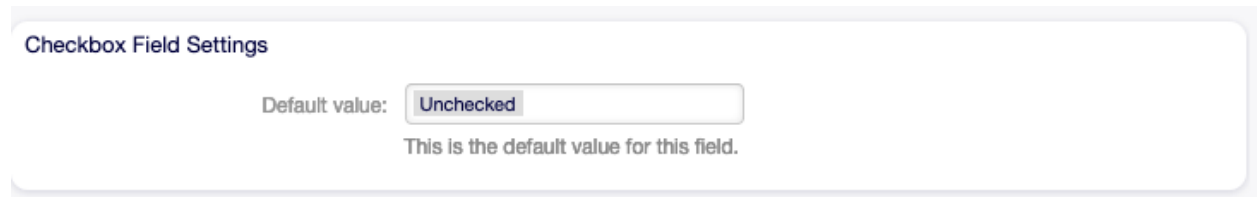
Fig. 1.168: Checkbox Dynamic Field Settings

Default value * The default value for the checkbox.

> **Checked** The checkbox is checked by default.

> **Unchecked** The checkbox is unchecked by default.

Contact With Data Dynamic Field Settings

This dynamic field allows to add contacts with data to tickets.

Name Field * The attribute `Name` is always mandatory and it is not automatically added, so for each new data source this attribute must be added manually. Within the data source definition (or dynamic field configuration) this must be represented by the key `Name` and the value could be Name for example.

ValidID Field * The attribute `ValidID` is always mandatory and it is not automatically added, so for each new data source this attribute must be added manually. Within the data source definition (or dynamic field configuration) this must be represented by the key `ValidID` and the value could be Validity for example.

Other Fields These are the possible data attributes for contacts. Clicking on the ⊞ button of the Add Fields will add two new fields, where a Key (internal value) and a Value (displayed value) can be set. With the button you can add multiple key-value pairs.

> **Warning:** The keys `Name` and `ValidID` are already used by Name Field and ValidID Field. Do not use these keys again!

Add Fields Use this button to add more fields to dynamic field.

Mandatory fields Comma separated list of mandatory keys.

> **Note:** Keys `Name` and `ValidID` are always mandatory and doesn't have to be listed here.

Sorted fields Comma separated list of keys in sort order. Keys listed here come first, all remaining fields afterwards and sorted alphabetically.

Searchable fields Comma separated list of searchable keys.

> **Note:** Key `Name` is always searchable and doesn't have to be listed here.

Fig. 1.169: Contact With Data Dynamic Field Settings

Translatable values If you activate this option the values will be translated to the user defined language.

Note: You need to add the translations manually into the language translation files.

When the dynamic field was saved, click on the name of the newly created dynamic field in the overview table. The Edit Dynamic Field screen will open. There is a button Add or edit contacts, that points to Tickets → Edit contacts with data to add some data.
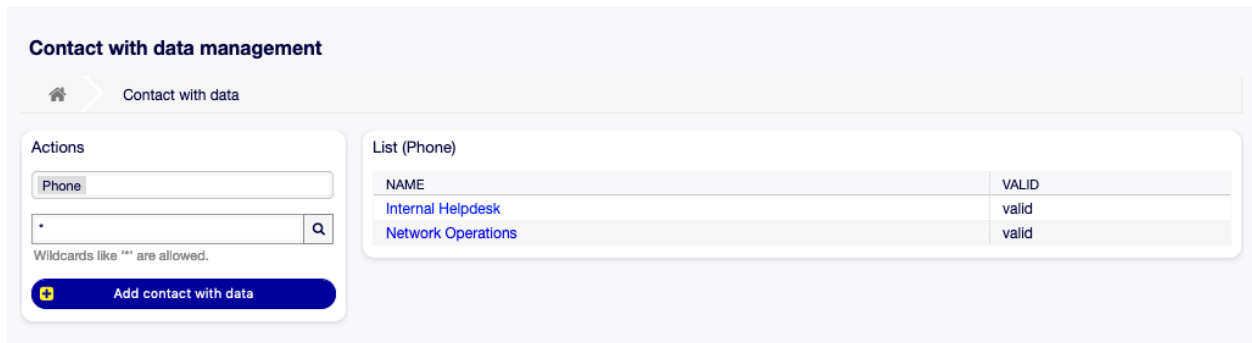
Fig. 1.170: Contact With Data Management Screen

To add a new contact with data:

1. Select a dynamic field from the drop-down list in Actions widget of the left sidebar.

2. Click on the Add contact with data button in the left sidebar.

3. Fill in the required fields.

4. Click on the Save button.
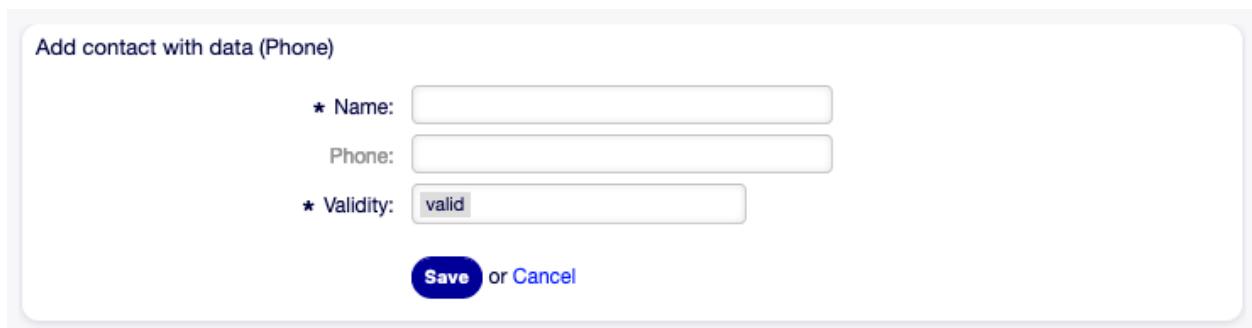
Fig. 1.171: Add Contact With Data Screen

To edit a contact with data:

1. Click on a contact with data in the list of contact with data entries.

2. Modify the fields.

3. Click on the Save or Save and finish button.

The usage of this type of dynamic field is more complex then the others. An exemplary usage of contacts with data is as follows:

1. Create a new dynamic field of type contact with data.

Fig. 1.172: Edit Contact With Data Screen

2. Set the possible contact attributes (possible values).

   - Type Name into Name Field.

   - Type Validity into ValidID Field.

   - Add any other attribute with Add Fields button such as `Telephone` attribute (key: `Telephone`, value: Phone).

3. Add the list of mandatory attribute keys comma separated (`Name` and `ValidID` are not needed).

4. Set the attribute key order list comma separated as: `Name, Telephone, ValidID`.

5. Add the list of searchable attribute keys comma separated (`Name` is not needed).

6. Populate the data source by adding at least one contact in the newly created data source by using Tickets → Edit contacts with data screen from the main navigation bar.

7. Add the new dynamic field to the screen's configuration where it should be shown. For example in New Phone Ticket screen by updating the system configuration setting `Ticket::Frontend::AgentTicketPhone###DynamicField` and do the same for `Ticket::Frontend::AgentTicketZoom###DynamicField`.

8. Go to New Phone Ticket screen, and notice that the new field is there. Add all needed information to the ticket.

9. Select an existing contact using autocomplete and choosing a contact.

10. The assigned contact and its attributes will be shown in the Ticket Zoom screen.

11. It is possible to update the attributes of the contact by clicking the Edit contact data button that appears in the right side of the title of the contact data box (if the current user is a member of the groups defined in system configuration setting `Frontend::Module###AdminDynamicFieldContactWithData`).

12. If it's necessary to change the contact for this ticket, it can be done via any other ticket action where the dynamic field is configured for display.

Date Dynamic Field Settings

Date dynamic field is used to store a date value.

Default date difference The difference from **now** (in seconds) to calculate the field default value (e.g. 3600 or -60).

Define years period Activate this feature to define a fixed range of years (in the future and in the past) to be displayed on the year part of the field. If set to Yes the following options will be available:

Fig. 1.173: Date Dynamic Field Settings

Years in the past Define the number of years in the past from the current day to display in the year selection for this dynamic field in edit screens.

Years in the future Define the number of years in the future from the current day to display in the year selection for this dynamic field in edit screens.

Show link Here you can specify an optional HTTP link for the field value in overviews and zoom screens. Example:

```
http://some.example.com/handle?query=[% Data.Field1 | uri %]
```

Link for preview If filled in, this URL will be used for a preview which is shown when this link is hovered in ticket zoom. Please note that for this to work, the regular URL field above needs to be filled in, too.

Restrict entering of dates Here you can restrict the entering of dates of tickets.

Prevent entry of dates in the future Selecting this option will prevent entering a date that is after the current date.

Prevent entry of dates in the past Selecting this option will prevent entering a date that is before the current date.

Date / Time Dynamic Field Settings

Date / time dynamic field is used to store a date time value.

The settings for this type of dynamic field is the same as for date dynamic field.

Fig. 1.174: Date / Time Dynamic Field Settings

Dropdown Dynamic Field Settings

Drop-down dynamic field is used to store a single value, from a closed list.

Possible values These are the possible data attributes for contacts. Clicking on the ⊞ button will add two new fields, where a key (internal value) and a value (displayed value) can be set. With the button you can add multiple key-value pairs.

Default value This is the default value for this field and this will be shown on the edit screens.

Add empty value If this option is activated an extra value is defined to show as a - in the list of possible values. This special value is empty internally.

Tree View Activate this option to display values as a tree.

Translatable values If you activate this option the values will be translated to the user defined language.

Note: You need to add the translations manually into the language translation files.

Show link Here you can specify an optional HTTP link for the field value in overviews and zoom screens. Example:

```
http://some.example.com/handle?query=[% Data.Field1 | uri %]
```
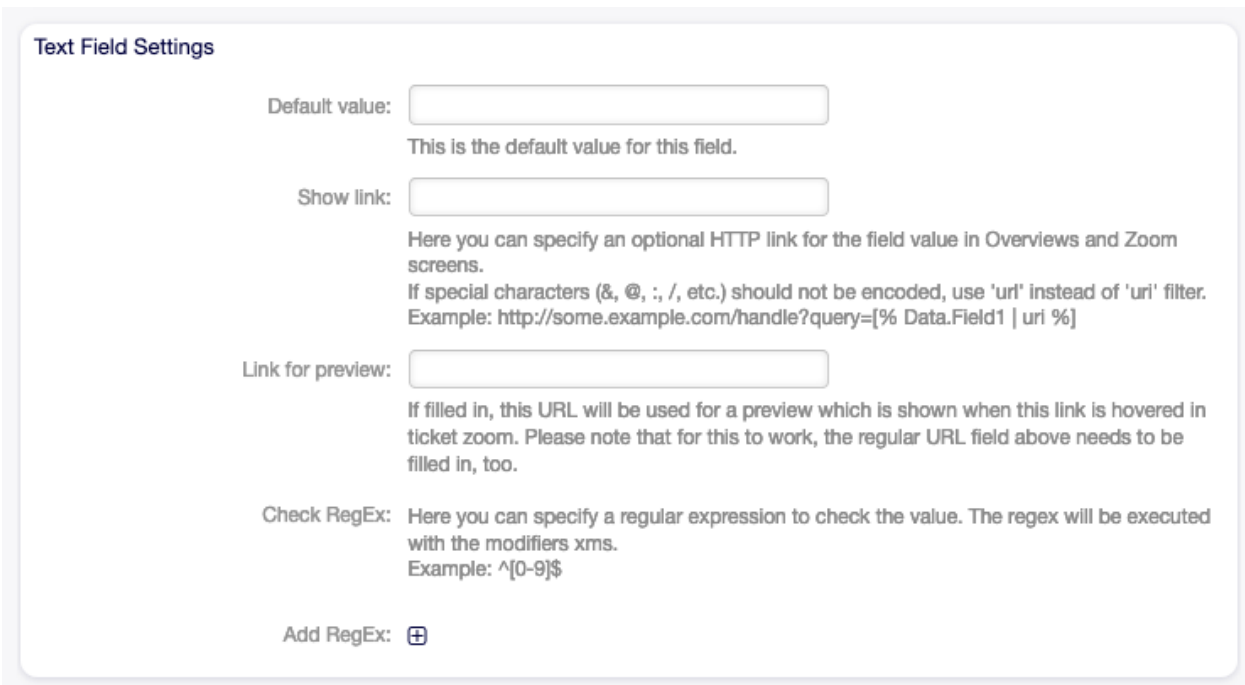
Link for preview If filled in, this URL will be used for a preview which is shown when this link is hovered in ticket zoom. Please note that for this to work, the regular URL field above needs to be filled in, too.

Fig. 1.175: Dropdown Dynamic Field Settings

Multiselect Dynamic Field Settings

Fig. 1.176: Multiselect Dynamic Field Settings

Possible values These are the possible data attributes for contacts. Clicking on the ⊞ button will add two new fields, where a key (internal value) and a value (displayed value) can be set. With the button you can add multiple key-value pairs.

Default value This is the default value for this field and this will be shown on the edit screens.

Add empty value If this option is activated an extra value is defined to show as a - in the list of possible values. This special value is empty internally.

Tree View Activate this option to display values as a tree.

Translatable values If you activate this option the values will be translated to the user defined language.

Note: You need to add the translations manually into the language translation files.

Text Dynamic Field Settings

Text dynamic field is used to store a single line string.

Default value This is the default value for this field and this will be shown on the edit screens.

Show link Here you can specify an optional HTTP link for the field value in overviews and zoom screens. Example:

```
http://some.example.com/handle?query=[% Data.Field1 | uri %]
```

Fig. 1.177: Text Dynamic Field Settings

Link for preview If filled in, this URL will be used for a preview which is shown when this link is hovered in ticket zoom. Please note that for this to work, the regular URL field above needs to be filled in, too.

Check RegEx Here you can specify a regular expression to check the value. The regex will be executed with the modifiers `xms`. Example:

```
^[0-9]$
```

Add RegEx Clicking on the ⊞ button will add two new fields, where a regular expression and an error message can be added.

Textarea Dynamic Field Settings

Textarea dynamic field is used to store a multiple line string.

Number of rows The height (in lines) for this field in the edit mode.

Number of cols The width (in characters) for this field in the edit mode.

Default value This is the default value for this field and this will be shown on the edit screens.

Check RegEx Here you can specify a regular expression to check the value. The regex will be executed with the modifiers `xms`. Example:

```
^[0-9]$
```

Add RegEx Clicking on the ⊞ button will add two new fields, where a regular expression and an error message can be added.

Fig. 1.178: Textarea Dynamic Field Settings

Database Dynamic Field

Note: This dynamic field is already included in OTOBO 10 Standard! The documentation on the field types still needs to be completed.

Web Service Dynamic Field

Note: This dynamic field is already included in OTOBO 10 Standard! The documentation on the field types still needs to be completed.

Display Dynamic Fields on Screens

To display a dynamic field on a screen:

1. Make sure that the dynamic field Validity is set to valid.

2. Open the System Configuration module in the admin interface.

3. Navigate to Frontend → Agent → View and select a view to add the dynamic field to.

4. Find the setting ends with `###DynamicField` and click on the Edit this setting button.

5. Click on the + button to add the dynamic field.

6. Enter the name of the dynamic field to the text box and click on the tick button.

7. Select 1 - Enabled or 2 - Enabled and required.

8. Click on the tick button on the right to save the setting.

9. Deploy the modified system configuration.
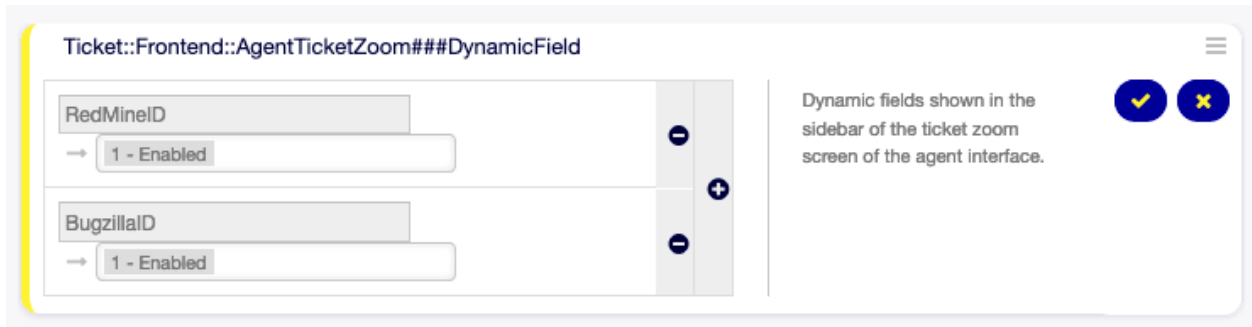
Fig. 1.179: Display Dynamic Fields on Screen

Note: It is possible to add multiple dynamic fields at the same time. To do this, repeat steps 5-7.

Set Default Value via Ticket Event Module

A ticket event (e.g. `TicketCreate`) can trigger a value set for a certain field, if the field does not have a value yet.

1. Open the System Configuration module in the admin interface.

2. Navigate to Core → Event → Ticket and search for the setting `Ticket::EventModulePost###9600-TicketDynamicFieldDefault`.

3. Click on the Edit this setting button to activate the setting.

4. Click on the tick button on the right to save the setting.

5. Deploy the modified system configuration.
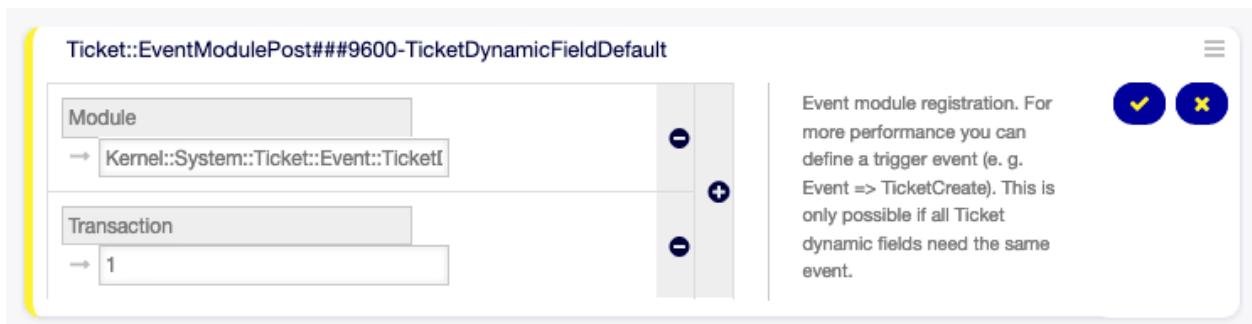
Fig. 1.180: Activate Ticket Event Module

Example: activate Field1 in `TicketCreate` event:

1. Open the System Configuration module in the admin interface.

2. Navigate to Core → Ticket → DynamicFieldDefault and search for the setting `Ticket::TicketDynamicFieldDefault###Element1`.

3. Click on the Edit this setting button to activate the setting.

4. Click on the tick button on the right to save the setting.

5. Deploy the modified system configuration.

Ticket::TicketDynamicFieldDefault###Element1 ≡

Event	→	TicketCreate
Name	→	Field1
Value	→	Default

Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (https://doc.otobo.org/doc/), chapter "Ticket Event Module".
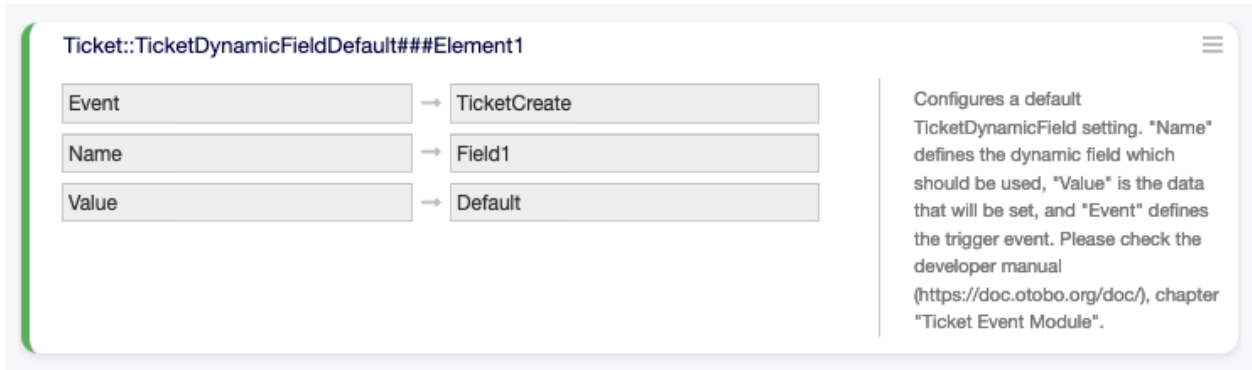
Fig. 1.181: Activate Dynamic Field in Ticket Create Event

Note: This configuration can be set in any of the 16 `Ticket::TicketDynamicFieldDefault###Element` settings.

See also:

If more than 16 fields needs to be set up, a custom XML file must be placed in `$OTOBO_HOME/Kernel/Config/Files/XML` directory to extend this feature.

Set Default Value via User Preferences

The dynamic field default value can be overwritten with a user defined value stored in the personal preferences.

1. Open the System Configuration module in the admin interface.

2. Navigate to Frontend → Agent → View → Preferences and search for the setting `PreferencesGroups###DynamicField`.

3. Click on the Edit this setting button to activate the setting.

4. Click on the tick button on the right to save the setting.

5. Deploy the modified system configuration.

Click on your avatar on the top left corner, and select Personal Preferences → Miscellaneous to add a default value for the dynamic field.

This setting is an example of how to create an entry in the user preferences screen to set an exclusive dynamic field `Name_X` default value for the selected user. The limitation of this setting is that it only permits the use of one dynamic field. If two or more fields will use this feature, it is necessary to create a custom XML configuration file to add more settings similar to this one.

Note: If more settings are added in a new XML each setting name needs to be unique in the system and different than `PreferencesGroups###DynamicField`. For example:

Fig. 1.182: Activate Dynamic Field in Personal Preferences

Fig. 1.183: Dynamic Field in Personal Preferences

- `PreferencesGroups###101-DynamicField-Field1`
- `PreferencesGroups###102-DynamicField-Field2`
- `PreferencesGroups###My-Field1`
- `PreferencesGroups###My-Field2`

1.4.3 Generic Agent

Processing tickets requires often a workflow. Let's say "if-then" activities.

If specific conditions match like:

- A ticket is from one particular customer.
- A ticket is assigned to an appropriate queue.
- A ticket has a defined priority.
- A ticket contains defined keywords.

Outlined activities must be performed like changing the ticket priority, moving the ticket to another group, assigning a service to a ticket, and many more.

Also time-based activities can be required like cleaning up the spam-queue once a week.

OTOBO supports this with the Generic Agent. Here, simple or compound time and event-based tasks are configurable in the OTOBO front end without the requirement to learn a scripting language. Depending on search criteria, and time or event criteria, tickets will automatically be acted upon.

Use this screen to manage generic agent jobs in the system. A fresh OTOBO installation contains no generic agent jobs by default. The generic agent job management screen is available in the Generic Agent module of the Processes & Automation group.
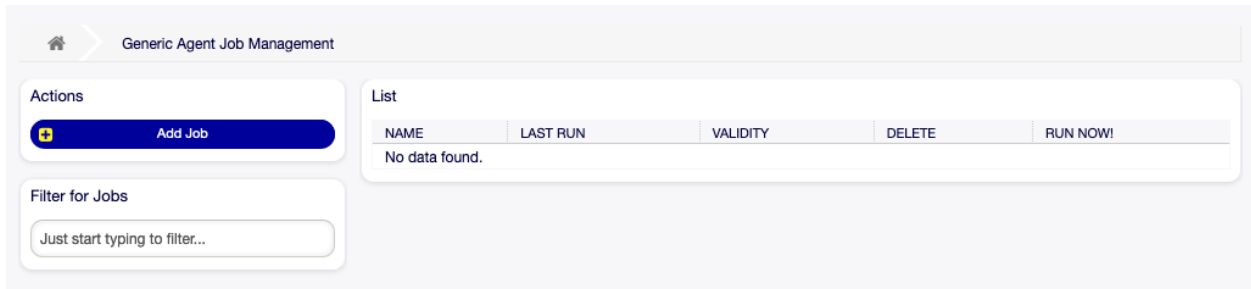
Fig. 1.184: Generic Agent Management Screen

Manage Generic Agent Jobs

To create a new generic agent job:

1. Click on the Add Job button in the left sidebar.
2. Fill in the required fields.
3. Click on the Save button.

To edit a generic agent job:

Job Settings

★ Job name: [   ]

Validity: [Yes]

▶ Automatic Execution (Multiple Tickets)

▶ Event Based Execution (Single Ticket)

▶ Select Tickets

▶ Update/Add Ticket Attributes

▶ Add Note

▶ Execute Ticket Commands

▶ Execute Custom Module

Save Changes

Save or Cancel

Fig. 1.185: Create New Generic Agent Job Screen

1. Click on a generic agent job in the list of generic agent jobs.

2. Modify the fields.

3. Click on the Save or Save and finish button.

Fig. 1.186: Edit Generic Agent Job Screen

To delete a generic agent job:

1. Click on the trash icon in the fourth column of the overview table.

2. Click on the Confirm button.

Fig. 1.187: Delete Generic Agent Job Screen

Note: If several generic agent jobs are added to the system, use the filter box to find a particular generic agent job by just typing the name to filter.

Generic Agent Job Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

General Job Settings

Fig. 1.188: Job Settings - General

Job Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Validity Set the validity of this resource. This resource can be used in OTOBO only, if this field is set to Yes. Setting this field to No will disable the use of the resource.

Automatic Execution

Fill in this section to set the times for automatic execution of the job.

Fig. 1.189: Job Settings - Automatic Execution

Schedule minutes Select the minutes in which the job has to be executed. For example if 10 is selected, the job will be executed in every hour in 00:10, 01:10, 02:10, etc.

Schedule hours Select the hours in which the job has to be executed. For example if 10 is selected for minutes and 02 is selected for hours, the job will be executed in every day in 02:10.

Schedule days Select the days in which the job has to be executed. For example if 10 is selected for minutes, 02 is selected for hours and Fri is selected for days, the job will be executed in every week in Friday 02:10.

Event Based Execution

Fill in this section to set events that trigger the execution of the job.
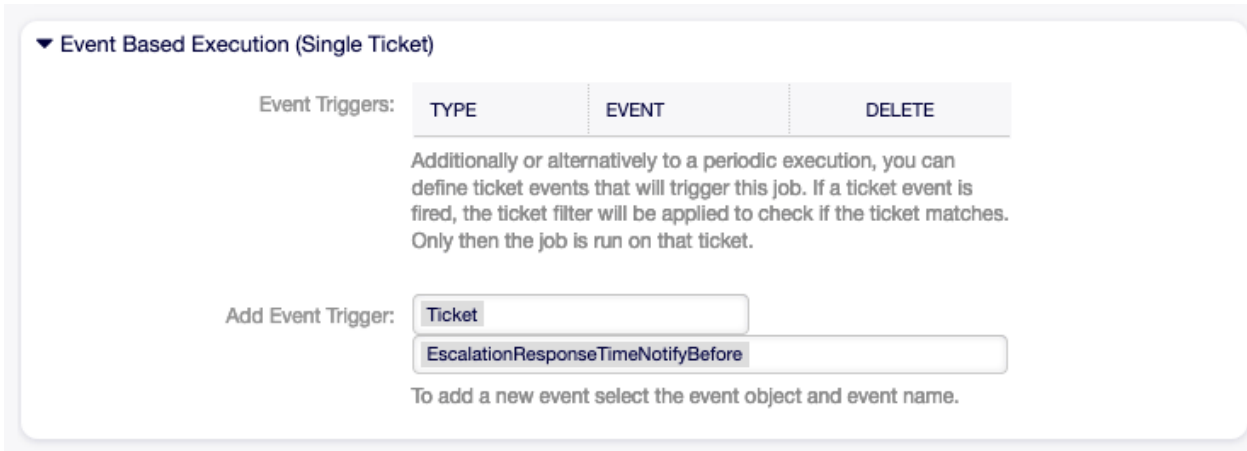
Fig. 1.190: Job Settings - Event Based Execution

Event Triggers This is a list of already added events. If no events are added yet, the list contains no elements. Elements can be deleted by clicking on the trash icon in the last column.

Add Event Trigger Select an object and a related event that will trigger the job.

Select Tickets

This section contains all the attributes, that you can use to select the affected tickets.

Fields are self-explained, so there is no additional explanation here.

Update/Add Ticket Attributes

Fill in this section to update and/or add ticket attributes.

Fields are self-explained, so there is no additional explanation here.

Add Note

Fill in this section to add a note to tickets that are affected by job execution.

Execute Ticket Commands

Fill in this section, if you would like to make execute a custom command with the job.

Send agent/customer notifications on changes If Yes is selected, a notification will be sent to agents and customers about ticket changes.

CMD Enter a command here, that will be executed. ARG[0] will be the ticket number. ARG[1] the ticket ID.

Note: Use `: :` as directory separator, if the value contains a path.

Fig. 1.191: Job Settings - Add Note

Fig. 1.192: Job Settings - Execute Ticket Commands

Delete tickets If Yes is selected, the generic agent job will delete the matched tickets.

> **Warning:** All affected tickets will be removed from the database and cannot be restored!

Execute Custom Module

Fill in this section, if you would like to make execute a custom module with the job.

Fig. 1.193: Job Settings - Execute Custom Module

Module This is the path for the module to be executed.

> **Note:** Use `::` as directory separator, if the value contains a path.

Param key Enter the key of the parameter, that should be passed to the module.

Param value Enter the value of the parameter, that should be passed to the module.

1.4.4 Process Management

Maximizing performance while minimizing human error is a requirement of organizations of all sizes. This need is covered by defined processes and workflows, for recurring tasks. Ensuring all required information is available in the right place, and contacts are informed about their responsibilities like adding information, approving requests, etc.

OTOBO supports this requirement by process management. Process tickets help by using the required mandatory and optional fields (see Dynamic Fields) that information is not forgotten upon ticket creation or during later steps of the process. Process tickets are simple to handle for customer users and agents, so no intensive training is required.

Processes are designed completely and efficiently within the OTOBO front end to fit the requirements of your organization.

Use this screen to manage processes in the system. The process management screen is available in the Process Management module of the Processes & Automation group.

Manage Processes

To create a new process:

1. Click on the Create New Process button in the left sidebar.
2. Fill in the required fields.
3. Click on the Save button.
4. Add activities, activity dialogs, transitions and transition actions.
5. Set State to Active.
6. Deploy all processes.

Fig. 1.194: Create New Process Screen

To edit a process:

1. Click on a process in the list of processes.
2. Modify the fields and the process path.
3. Click on the Save or Save and finish button.
4. Deploy all processes.

To copy a process:

1. Click on the copy icon in the fifth column of list of processes.
2. Click on the newly created process to edit it.

To delete a process:

1. Click on a process in the list of processes.
2. Set the State option to Inactive.
3. Click on the Save button. A new Delete Inactive Process button will appear in the left sidebar.
4. Click on the Delete Inactive Process button.
5. Click on the Delete button in the confirmation screen.
6. Deploy all processes.

> **Warning:** Processes are written into file in Perl format. Without deploying, all processes are still in this cache file even if they are deleted or the State option is set to Inactive or FadeAway. Don't forget to deploy all processes after modifications!

Give some time for agents to complete the running process tickets before the process will be deleted. It is possible to mark a process for deletion, i. e. set the process as not to be selected anymore. Process states can be:

Active Processes can be used in new process tickets.

FadeAway Processes cannot be selected any more for new tickets, but existing tickets still can use the process.

Inactive Processes are deactivated and cannot be used for new or existing tickets.

To deploy all processes:

1. Click on the Deploy All Processes button in the left sidebar.

> **Note:** New or modified processes have to deploy in order to make affect the behavior of the system. Setting the State option to Active just indicates, which processes should be deployed.

To export a process:

1. Click on the export icon in the fourth column of list of processes.

2. Choose a location in your computer to save the `Export_ProcessEntityID_xxx.yml` file.

> **Warning:** The exported file contains only the process itself, and doesn't contain the Queues, Agents, Dynamic Fields, etc. needed for the process.

To import a process:

1. Click on the Browse... button of the Configuration Import widget in the left sidebar.

2. Select a previously exported `.yml` file.

3. Click on the Import process configuration button.

4. Deploy all processes.

> **Note:** Before import of a process, it is still necessary to create all Queues, Agents and Dynamic Fields, as well as to set System Configuration, that are needed by each process before the import. If the process requires the use of Access Control Lists (ACL) those are also needed to be set manually.

> **Note:** If several processes are added to the system, use the filter box to find a particular process by just typing the name to filter.

Process Elements

The element names of the process modeler have been adapted to the Business Process Model and Notation (BPMN) ISO naming convention. The following task activities can be used as basic elements

---

of the process.

Script task activity Script task activity is executed by the process management module and it can set dynamic field values or manage tickets automatically.
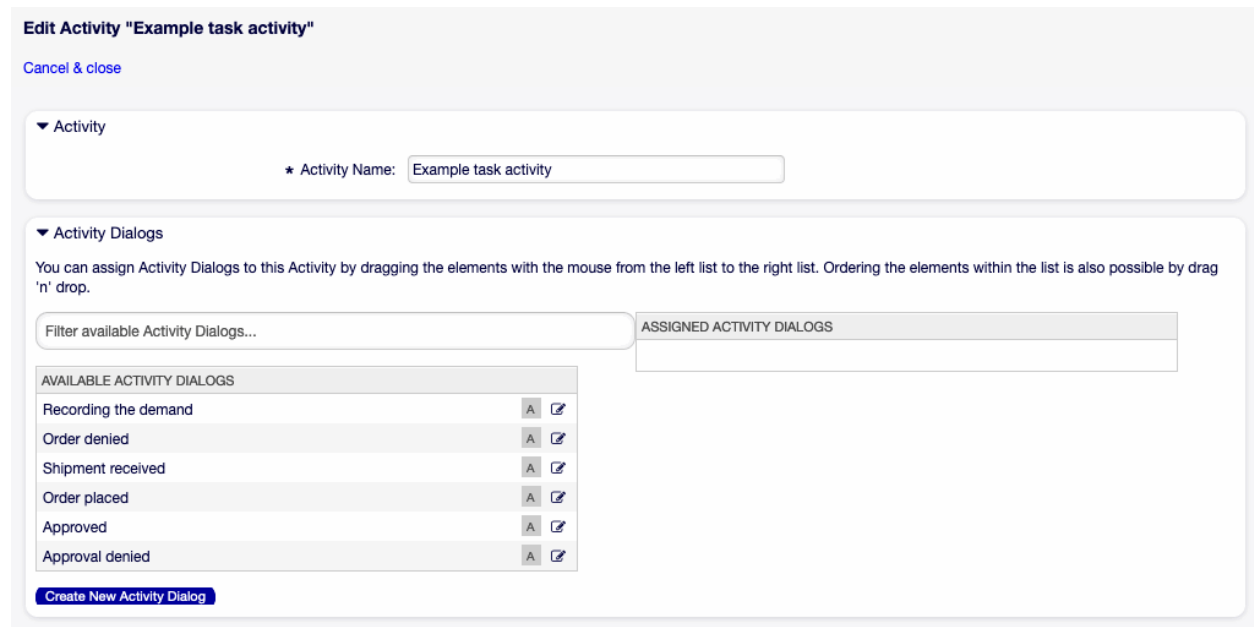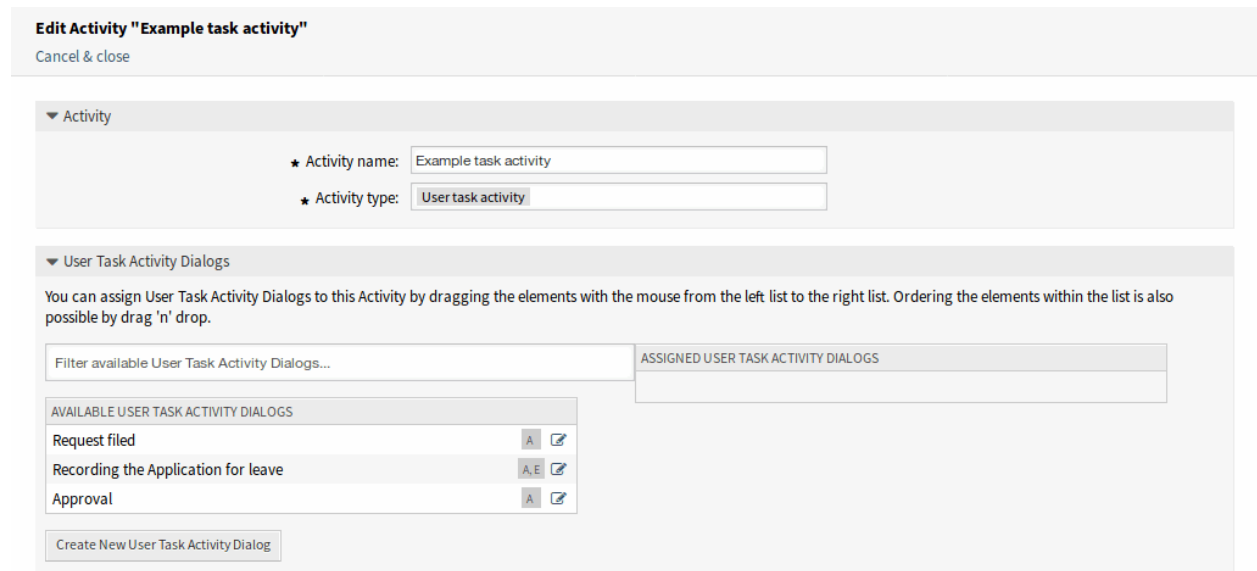

Fig. 1.195: Script Task Activity Window

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Activity name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces.

Activity type * The following types of task activities can be used:

- Script task activity (selected for now)

- Service task activity

- User task activity

Activity description Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity.

Activity error code Define a custom error code for script or service task activities. The custom error code must be a positive integer number.

Can start processes in Defines, where can the processes be started by the agents or customer users. A process can be started in the following interfaces:

- Agent Interface

- Agent and External Interface

- External Interface

Script * In this drop-down can be selected which script should be triggered immediately if the activity is set. Click on the Configure button to add parameters (key-value pairs) for the script.

Fig. 1.196: Service Task Activity Window

Service task activity Service task activity uses a web service to complete the task.

> The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

> **Activity name** * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces.

> **Activity type** * The following types of task activities can be used:

>> • Script task activity

>> • Service task activity (selected for now)

>> • User task activity

> **Activity description** Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity.

> **Activity error code** Define a custom error code for script or service task activities. The custom error code must be a positive integer number.

> **Can start processes in** Defines, where can the processes be started by the agents or customer users. A process can be started in the following interfaces:

>> • Agent Interface

>> • Agent and External Interface

>> • External Interface

> **Web Service** * Select one of the Web Services from the drop-down list.

> **Invoker** * Select an invoker for the web service. Click on the Configure button to add parameters for the invoker.

User task activity User task activity can be used when the task is being done by an agent or a customer user.

Fig. 1.197: User Task Activity Window

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Activity name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces.

Activity type * The following types of task activities can be used:

- Script task activity

- Service task activity

- User task activity (selected for now)

Activity Dialogs You can assign activity dialogs to this activity by dragging the elements with the mouse from the left list to the right list. Ordering the elements within the list is also possible by drag and drop.

Click on the Create New Activity Dialog button to create new dialogs.

Example process

Processes are more complex than other resources in OTOBO. To create a process, you need to do several steps. The following chapters show you, how to define a process from the specification and create the needed resources. Let's see an example to make it more demonstrative. We will define a book order process.

Process Specification

The book order process has four states.

Recording the demand Before an order will be placed, the demand for literature by an employee will be recorded. The following book is needed in our example:

```
Title: Prozessmanagement für Dummies
Autor: Thilo Knuppertz
ISBN: 35210.03713
```

Approval by manager The head of the employee's department needs to decide on the order. In case of a denial, a reason should be recorded by the manager. In case of approval, the order is passed to the purchasing department.

Processing by purchasing department Purchasing now has the task to find out where the book can be ordered with the best conditions. If it is out of stock, this can be recorded in the order. In case of a successful order purchasing will record the supplier, the price and the delivery date.

Processing by the mail room The shipment will arrive at the company. The incoming goods department checks the shipment and records the date of receipt. Now the employee will be informed that their order has arrived and is ready to be collected.

Introduce The Process Elements

If we assume that a ticket acts in this workflow like an accompanying document that can receive change notes, we already have a clear picture of process tickets.

From the analysis of the example process we can identify the following necessary items:

- Possibility to record data, let's call this activity dialog.

- Check which can react to changed data automatically, let's call this transition.

- Change which can be applied to a process ticket after successful transitions of a process ticket, let's call this transition action.

- A possibility to offer more than just one activity dialog to be available. In our example this is needed when the manager must have the choice between Approve and Deny. Let's call this activity.

Now, with activities, activity dialogs, transitions and transition actions we have the necessary tools to model the individual steps of our example. What is still missing is an area where for each workflow the order of the steps can be specified. Let's call this process.

Create Necessary Resources

Before the creation of the process and its parts is necessary to prepare the system. We need to define some Queues, Agents and Dynamic Fields as well as set some System Configuration settings.

Create the following Queues:

- Management

- Employees

- Purchasing

- Post office

Create the following Agents:

- Manager

- Employee

Create the following Dynamic Fields:
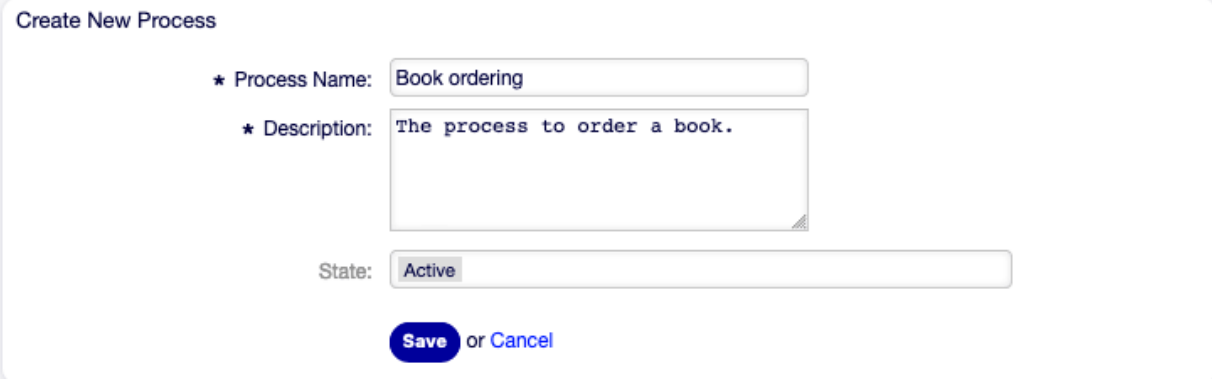
Object	Type	Name	Label	Possible values
Ticket	Text	Title	Title	
Ticket	Text	Author	Author	
Ticket	Text	ISBN	ISBN	
Ticket	Dropdown	Status	Status	- Approval - Approval denied - Approved - Order denied - Order placed - Shipment received
Ticket	Text	Supplier	Supplier	
Ticket	Text	Price	Price	
Ticket	Date	DeliveryDate	Delivery date	
Ticket	Date	DateOfReceipt	Date of receipt	

Set the following System Configuration settings:

- `Ticket::Responsible`

    - Enabled

- `Ticket::Frontend::AgentTicketZoom###ProcessWidgetDynamicFieldGroups`

    - `Book` → `Title, Author, ISBN`

    - `General` → `Status`

    - `Order` → `Price, Supplier, DeliveryDate`

    - `Shipment` → `DateOfReceipt`

- `Ticket::Frontend::AgentTicketZoom###ProcessWidgetDynamicField`

    - `Author` → 1 - Enabled

    - `DateOfReceipt` → 1 - Enabled

    - `DeliveryDate` → 1 - Enabled

    - `ISBN` → 1 - Enabled

    - `Price` → 1 - Enabled

    - `Status` → 1 - Enabled

    - `Supplier` → 1 - Enabled

    - `Title` → 1 - Enabled

Note: Don't forget to deploy the modified system configuration settings.

Now, go back to the Process Management screen and click on the Create New Process. Fill in the required fields.
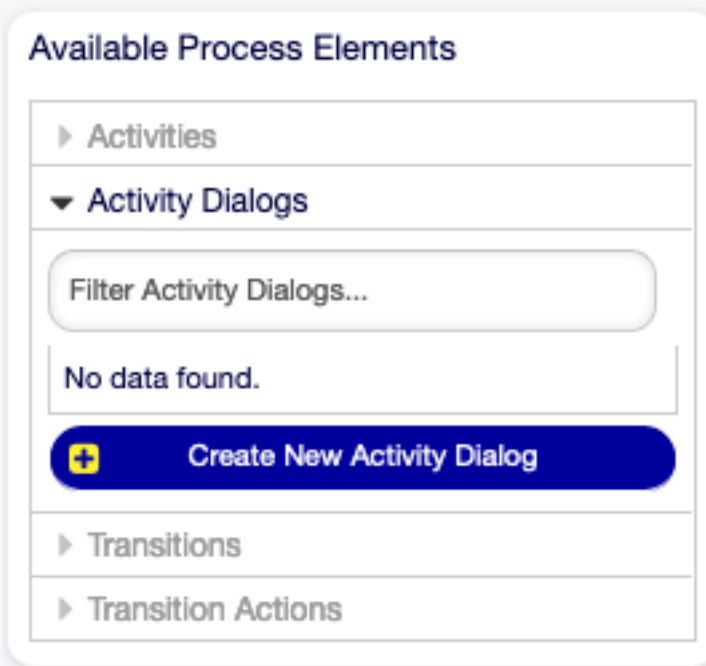
Fig. 1.198: Book Ordering - Create New Process

The new process is created. You can add some process element now.

Create Activity Dialogs

Click on the Activity Dialogs item in the Available Process Elements widget in the left sidebar. This action will expand the Activity Dialogs options and will collapse all others doing an accordion like effect. Click on the Create New Activity Dialog button.

Fig. 1.199: Book Ordering - Activity Dialogs

In the opened popup screen fill in the Dialog Name as well as the Description (short) fields. For this example we will leave all other fields as the default.

Fig. 1.200: Book Ordering - Add Activity Dialog

To assign fields to the activity dialog simple drag the required field from the Available Fields pool and drop into the Assigned Fields pool. The order in the Assigned Fields pool is the order as the fields will have in the screen. To modify the order simply drag and drop the field within the pool to rearrange it in the correct place.

In this example we will use:

- `Article` field for comments.
- `DynamicField_Title`, `DynamicField_Author`, `DynamicField_ISBN` fields for the data to be collected for the order.
- `DynamicField_Status` with the possibility to choose Approval.

Drag these fields from the Available Fields pool and drop into the Assigned Fields pool.

Note: In this screen all dynamic fields has the prefix `DynamicField_` as in `DynamicField_Title`. Do not confuse with the field `Title` that is the ticket title.

As soon as the fields are dropped into the Assigned Fields pool another popup screen is shown with some details about the field. We will leave the default options and only for `Article` fields we should make sure that the Communication Channel field is set to OTOBO and that the Is visible for customer is not checked.

After all fields are filled in, click on the Save and finish button to save the changes and go back to the process management screen.

Create the following activity dialogs with fields:

- Recording the demand (already created before)
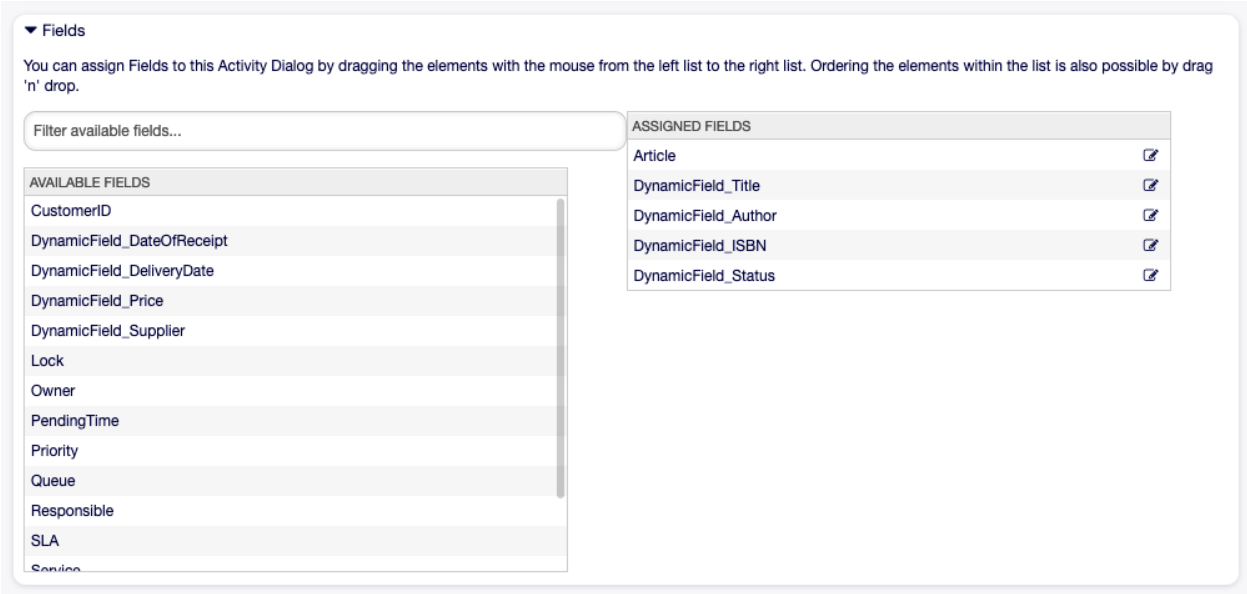
Fig. 1.201: Book Ordering - Add Activity Dialog Fields
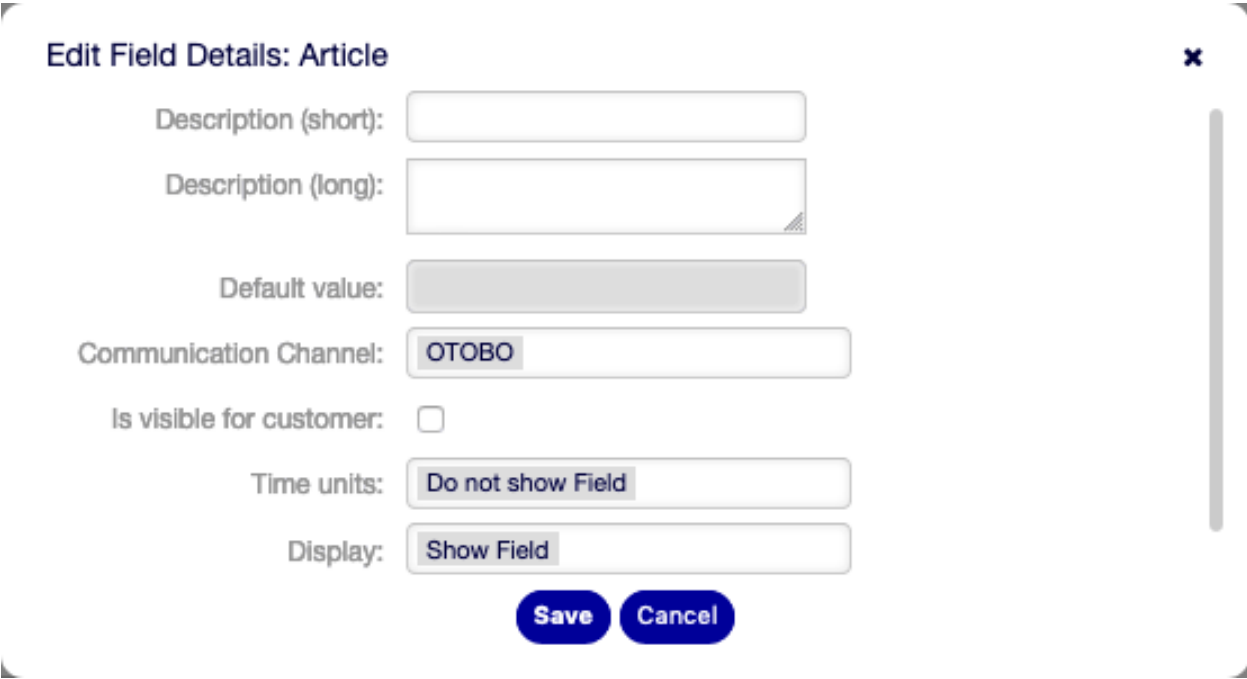
Fig. 1.202: Book Ordering - Edit Activity Dialog Fields

- – `Article` field for comments.
- – `DynamicField_Title`, `DynamicField_Author`, `DynamicField_ISBN` fields for the data to be collected for the order.
- – `DynamicField_Status` with the possibility to choose Approval.

- Approval denied
    - – `Article` field for comments.
    - – `DynamicField_Status` with the possibility to choose Approval denied.

- Approved
    - – `DynamicField_Status` with the possibility to choose Approved.

- Order denied
    - – `Article` field for comments.
    - – `DynamicField_Status` with the possibility to choose Order denied.

- Order placed
    - – `DynamicField_Supplier`, `DynamicField_Price`, `DynamicField_DeliveryDate` fields for purchasing.
    - – `DynamicField_Status` with the possibility to choose Order placed.

- Shipment received
    - – `DynamicField_DateOfReceipt` for the mail room.
    - – `DynamicField_Status` with the possibility to choose Shipment received.

Create Transitions

Click on the Transitions item in the Available Process Elements widget in the left sidebar. This action will expand the Transitions options and will collapse all others doing an accordion like effect. Click on the Create New Transition button.

In the opened popup screen fill in the Transition Name. For this example in the Condition Expressions we will use just one condition expression and just one field. For both we can leave the Type of Linking as and and we will use the filed match type value as String.

After all fields are filled in, click on the Save and finish button to save the changes and go back to the process management screen.

Create the following transitions:

- Approval (already created before)

    Check if the `DynamicField_Status` is set to Approval.

- Approval denied

    Check if the `DynamicField_Status` field is set to Approval denied.

- Approved

    Check if the `DynamicField_Status` field is set to Approved.

- Order denied

    Check if the `DynamicField_Status` field is set to Order denied.
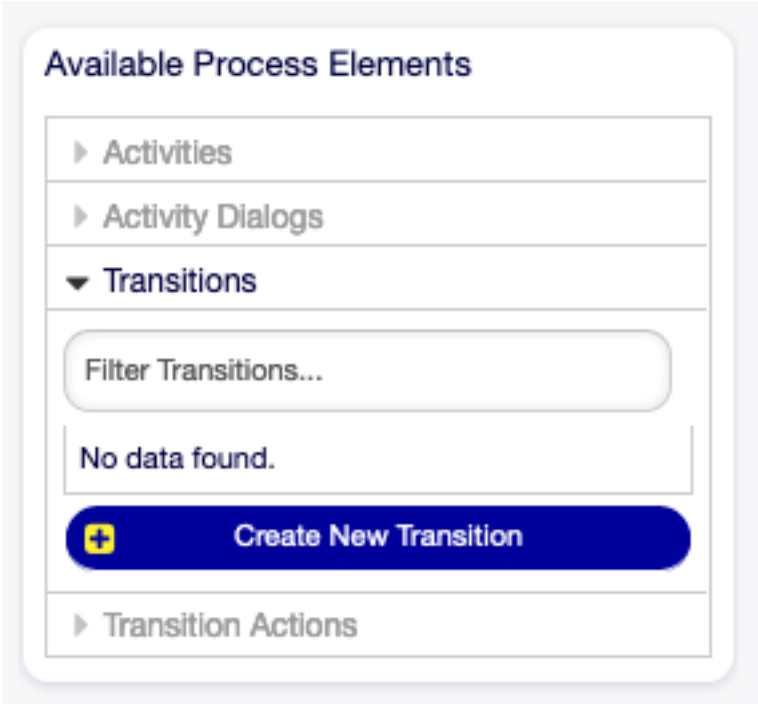
Fig. 1.203: Book Ordering - Transitions
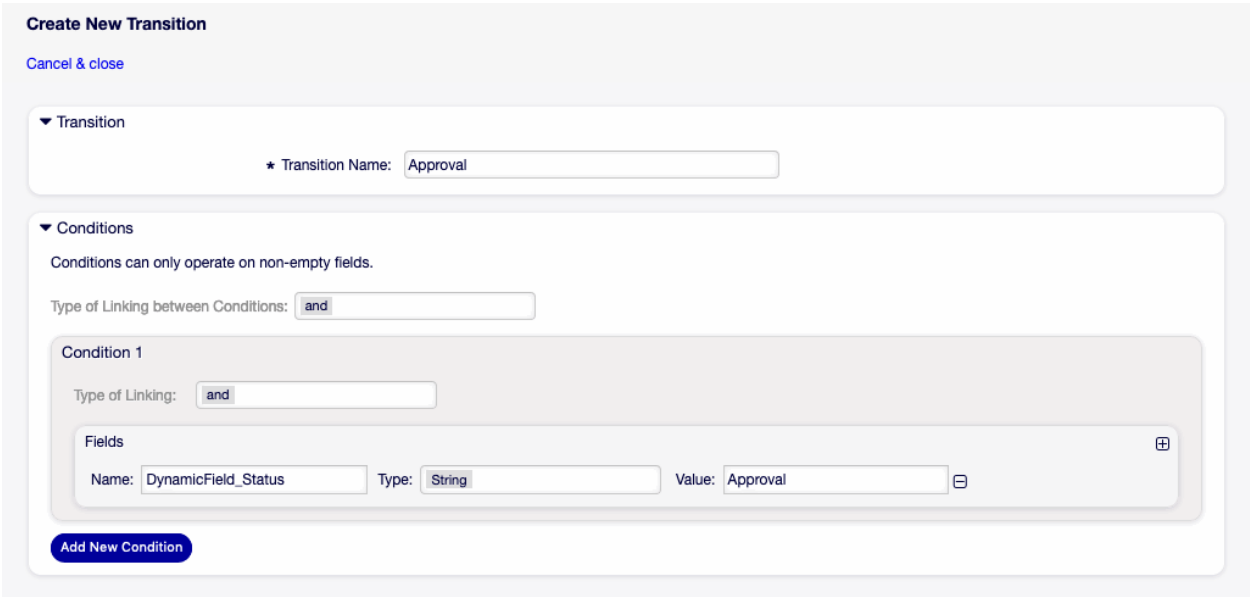
Fig. 1.204: Book Ordering - Add Transition

- Order placed

    Check if the `DynamicField_Status` field is set to Order placed.

- Shipment received

    Check if the `DynamicField_Status` field is set to Shipment received.

Create Transition Actions

Click on the Transition Actions item in the Available Process Elements widget in the left sidebar. This action will expand the Transition Actions options and will collapse all others doing an accordion like effect. Click on the Create New Transition Action button.
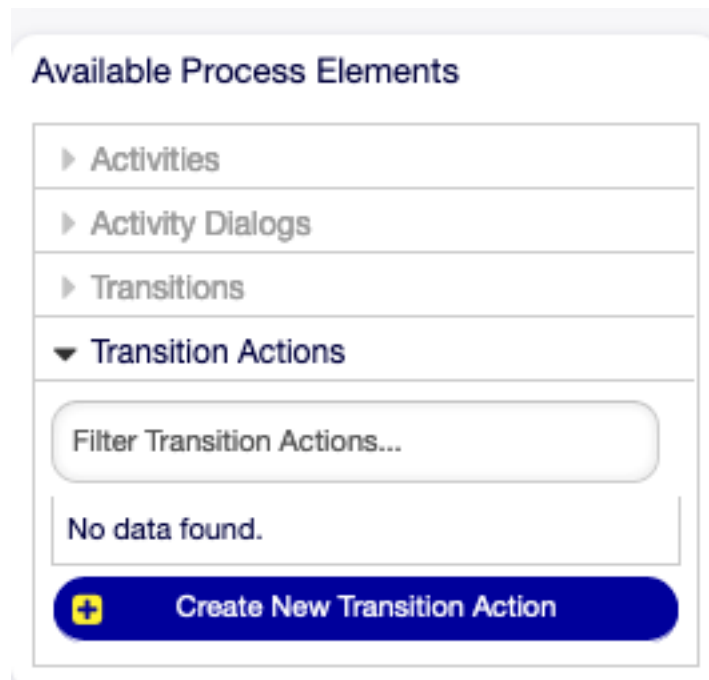
Fig. 1.205: Book Ordering - Transition Actions

In the opened popup screen fill in the Transition Action Name and the Transition Action module then click on the Save button. A new Configure button will appear next to the module field.
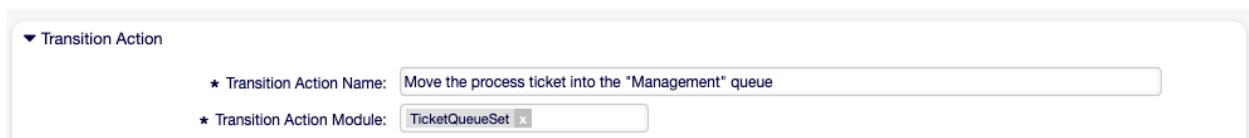
Fig. 1.206: Book Ordering - Add Transition Action

Click on the Configure button and add the needed configuration parameter keys and values.

After all fields are filled in, click on the Save and finish button to save the changes and go back to the process management screen.

See also:

Fig. 1.207: Book Ordering - Transition Action Parameters

Each module has its own and different parameters. As soon as you add a TransitionAction, the options are displayed directly. Further information can currently only be found in the source code:

- DynamicFieldSet
- TicketArticleCreate
- TicketCreate
- TicketCustomerSet
- TicketLockSet
- TicketOwnerSet
- TicketQueueSet
- TicketResponsibleSet
- TicketServiceSet
- TicketSLASet
- TicketStateSet
- TicketTitleSet
- TicketTypeSet

All transition action modules are located in the legacy named `Kernel/System/ProcessManagement/TransitionAction`.

Create the following transition actions:

- Move the process ticket into the "Management" queue (already created before)

    To be executed when the transition Approval applied.

- Change ticket responsible to "Manager"

    To be executed when the transition Approval applied.

- Move process ticket into the "Employees" queue

    To be executed when:

    – The transition Approval denied applied.

    – The transition Order denied applied.

    – The transition Shipment received applied.

- Change ticket responsible to "Employee"

    To be executed when:

    – The transition Approval denied applied.

    – The transition Order denied applied.

    – The transition Shipment received applied.

- Move process ticket into the "Purchasing" queue

    To be executed when the transition Approved applied.

- Move process ticket into the "Post office" queue

    To be executed when the transition Order placed applied.

- Close ticket successfully

    To be executed when the transition Shipment received applied.

- Close ticket unsuccessfully

    To be executed when:

    - The transition Approval denied applied.

    - The transition Order denied applied.

There are places where the same transition actions should be executed. Therefore it is reasonable to make it possible to link transition actions freely with transitions to be able to reuse them.

Create Activities

Click on the Activities item in the Available Process Elements widget in the left sidebar. This action will expand the Activities options and will collapse all others doing an accordion like effect. Click on the Create New Activity button.

In the opened popup screen fill in the Activity name field and select User task activity from the Activity type drop-down.

To assign dialogs to the activity simple drag the required dialogs from the Available Activity Dialogs pool and drop into the Assigned Activity Dialogs pool. The order in the Assigned Activity Dialogs pool is the order as the dialogs will be presented in the Ticket Zoom screen. To modify the order simply drag and drop the dialog within the pool to rearrange it in the correct place.

Note: This order is specially important in the first activity, since the first activity dialog for this activity is the only one that is presented when the process starts.

In this example we need to assign only the Recording the demand activity dialog. Drag this dialog from the Available Activity Dialogs pool and drop into the Assigned Activity Dialogs pool.

After all fields are filled in, click on the Save and finish button to save the changes and go back to the process management screen.

Create the following activities:

- Recording the demand (already created before)

    Assign the activity dialog Recording the demand.

- Approval

    Assign the activity dialogs Approval denied and Approved.

- Order

    Assign the activity dialogs Order denied and Order placed.

- Incoming

    Assign the activity dialog Shipment received.
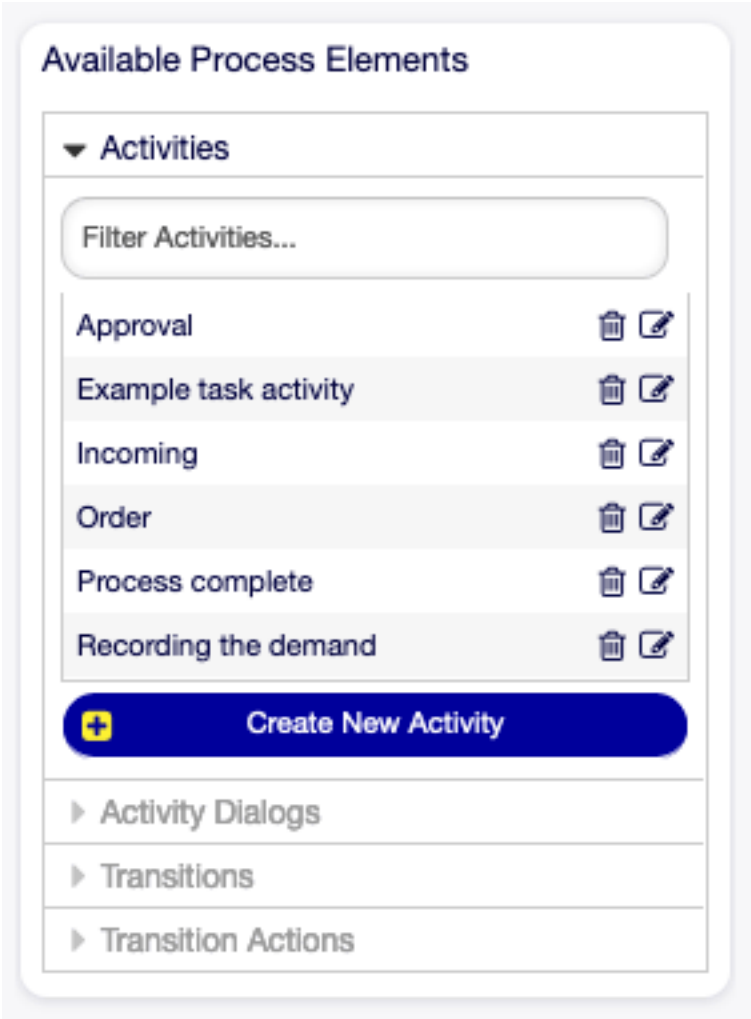
Fig. 1.208: Book Ordering - Activities
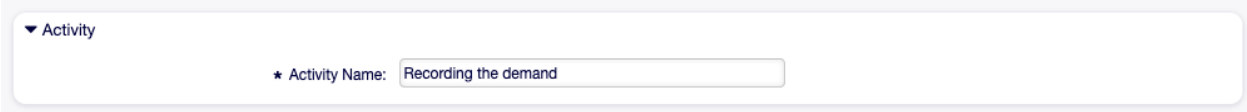
Fig. 1.209: Book Ordering - Add Activity
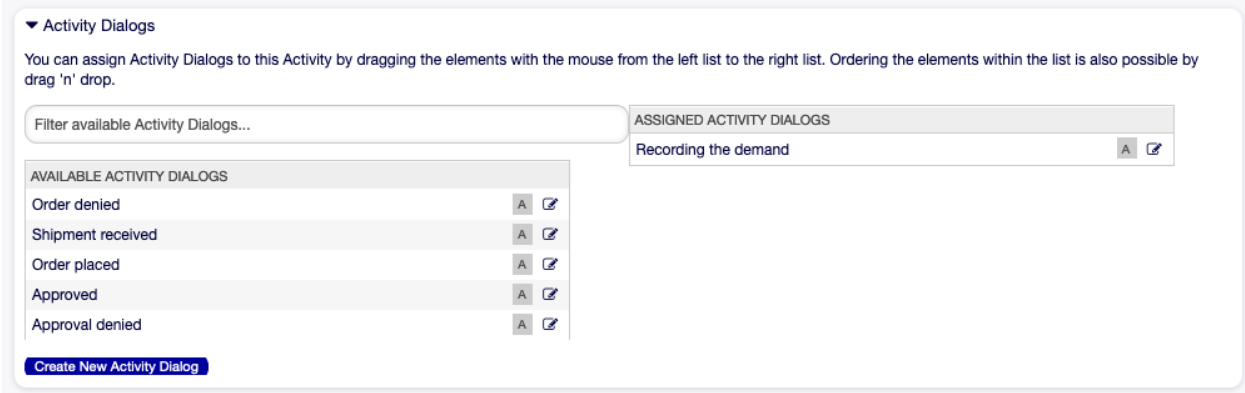
Fig. 1.210: Book Ordering - Assign Activity Dialog

- Process complete

    This is an activity without possible activity dialogs. It will be set after Approval denied, Order denied or Shipment received and represents the end of the process.

Now we can clearly see that activities are precisely defined states of a process ticket. After a successful transition a process ticket moves from one activity to another.

Create Process Path

Let us conclude our example with the last missing piece in the puzzle, the process as a flow describer. In our case this is the whole ordering workflow. Other processes could be office supply ordering or completely different processes.

The process has a starting point which consists of the start activity and the start activity dialog. For any new book order, the first activity dialog of the first activity is the first screen that is displayed. If this is completed and saved, the process ticket will be created and can follow the configured workflow.

The process also contains the directions for how the process ticket can move through the process. Let's call this process path. It consists of the start activity, one or more transitions (possibly with transition actions) and other activities.

Assuming that the activities have already assigned their activity dialogs, drag an activity from the accordion in the Available Process Elements widget in the left sidebar and drop it into the canvas area below the process information. Notice that an arrow from the process start (white circle) to the activity is placed automatically. This is the first activity and its first activity dialog is the first screen that will be shown when the process starts.

Next, drag another activity into the canvas too. Now we will have two activities in the canvas. The first one is connected to the start point and the second has no connections. You can hover the mouse over each activity to reveal their own activity dialogs.

Then let's create the process path (connection) between this two activities. For this we will use the transitions. Click on transition in the accordion, drag a transition and drop it inside the first activity. As soon as the transition is dropped the end point of the transition arrow will be placed next to the process start point. Drag the transition arrow end point and drop it inside the other activity to create the connection between the activities.

Now that the process path between the actions is defined, then we need to assign the transition actions to the transition. Double click the transition label in the canvas to open a new popup window.

Fig. 1.211: Book Ordering - First Activity On Canvas

Fig. 1.212: Book Ordering - Second Activity On Canvas

Fig. 1.213: Book Ordering - First Transition On Canvas

Fig. 1.214: Book Ordering - Assign First Transition Action

After the transition actions are assigned, click on the Save button to go back to the main process edit screen. Click on Save button below the canvas to save all other changes.

Complete the process path by adding the following activities, transitions and transition actions:

- Recording the demand (already created before)

    Possible transition: Approval

    Starting activity: Recording the demand

    Next activity: Approval

    If the condition of this activity is fulfilled, the ticket will move to activity Approval.

    Additionally, the following transition actions are executed:

    - Move the process ticket into the "Management" queue

    - Change ticket responsible to "Manager"

    The activity Recording the demand is a defined step of the process ticket, where there is the possibility for the transition Approval. If this applies, the ticket will move to the next activity Approval, and the transition actions Move the process ticket into the "Management" queue and Change ticket responsible to "Manager" are executed. In the activity Approval, the activity dialogs Approval denied and Approved are available.

- Approval

    Possible transition: Approval denied

    Starting activity: Approval

    Next activity: Process complete

    If this matches, the process ticket will move to activity Process complete.

    Additionally, the following transition actions are executed:

    - Move process ticket into the "Employees" queue

    - Change ticket responsible to "Employee"

    - Close ticket unsuccessfully

    Possible transition: Approved

    Starting activity: Approval

    Next activity: Order

    If this matches, the process ticket will move to activity Order.

    Additionally, the following transition actions are executed:

    - Move process ticket into the "Purchasing" queue

    We can see that from the current activity, which defines a step of the process ticket, there are one or more possibilities for transition which have exactly one target activity (and possibly one or more transition actions).

- Order

    Possible transition: Order denied

    Starting activity: Order

    Next activity: Process complete

If this matches, the process ticket will move to activity Process complete.

Additionally, the following transition actions are executed:

- Move process ticket into the "Employees" queue

- Change ticket responsible to "Employee"

- Close ticket unsuccessfully

Possible transition: Order placed

Starting activity: Order

Next activity: Incoming

If this matches, the process ticket will move to activity Incoming.

Additionally, the following transition actions are executed:

- Move process ticket into the "Post office" queue

- Incoming

Possible transition: Shipment received

Starting activity: Incoming

Next activity: Process complete

If this matches, the process ticket will move to activity Process complete.

Additionally, the following transition actions are executed:

- Move process ticket into the "Employees" queue

- Change ticket responsible to "Employee"

- Close ticket successfully

The complete process path for the book ordering process will then look like this:

After you finish the process path, click on Save and finish button below the canvas to go back to the process management screen.

Click on the Deploy All Processes button in the left sidebar. This will gather all processes information form the database and create a cache file (in Perl language). This cache file is actually the processes configuration that the system will use to create or use process tickets.

Note: Any change that is made on the process will require to re-deploy the process in order to get the change reflected in the system.

1.4.5 Ticket Masks

1.4.6 Web Services

In a connected world, a ticket system needs to be able to react to requests from other systems and also to send requests or information to other systems:

- CRM systems

- Project management systems

- Documentation management systems
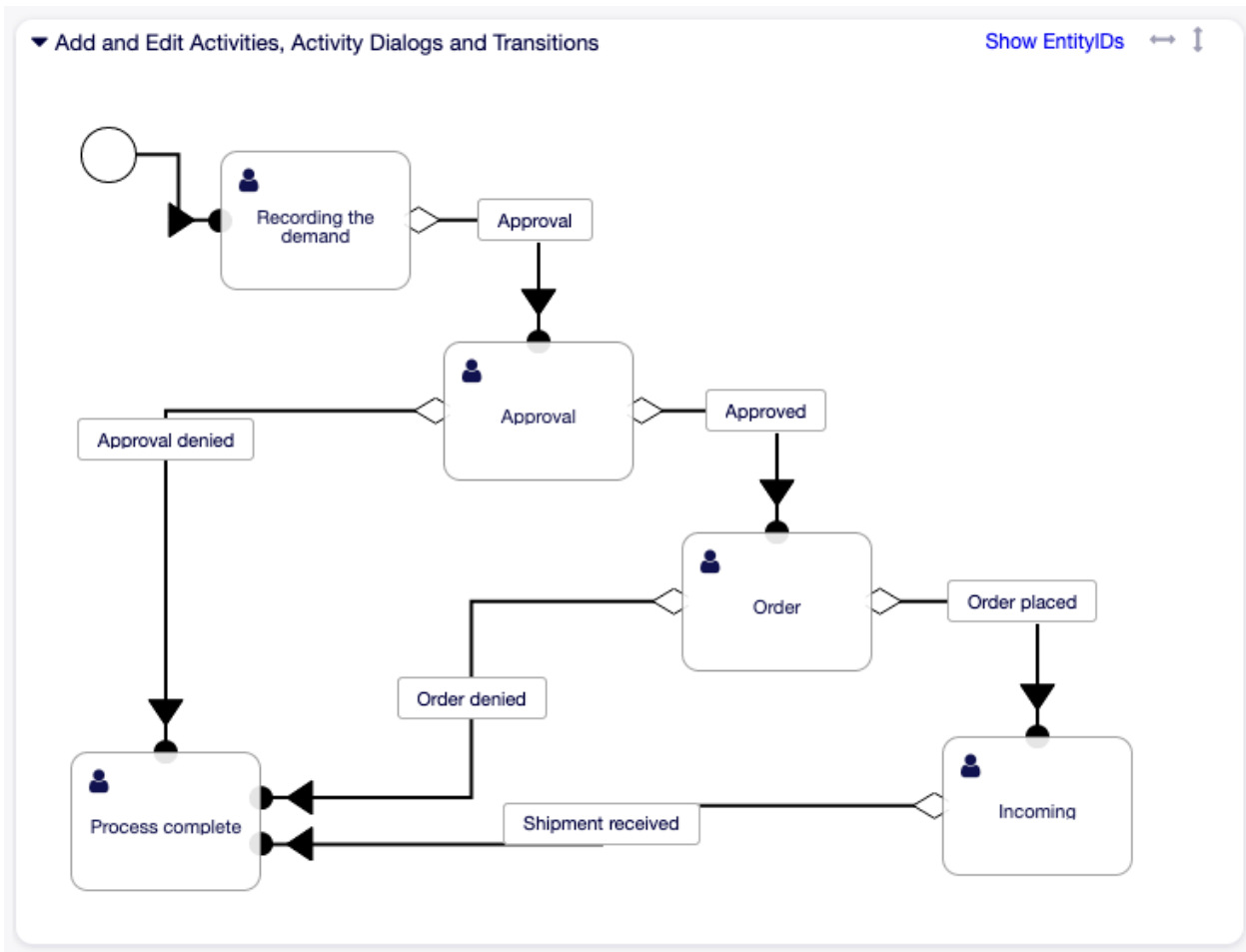
Fig. 1.215: Book Ordering - Process Complete

- and many more

The ticket system must be reachable by other services without manual intervention by an agent.

OTOBO supports this requirement by the Generic Interface. It empowers the administrator to create a web service for a specific task without scripting language knowledge. OTOBO reacts on incoming REST or SOAP requests and creates objects or provides object data to other systems transparently.

A web service is a communication method between two systems, in our case OTOBO and a remote system. In its configuration, the operation or invoker determine the direction of communication, and the mapping and transport take care of how the data is received and interpreted.

By configuration, you can define what actions the web service can perform internally (operation), what actions the OTOBO request can perform on a remote system (invokers), how data is converted from one system to the other (mapping), and which protocol is used for communication (transport).

The generic interface is the framework which makes it possible to create web services for OTOBO in a predefined way, using ready-made building blocks that are independent from each other and interchangeable.

Use this screen to manage web services in the system. A fresh OTOBO installation contains no web service by default. The web service management screen is available in the Web Services module of the Processes & Automation group.
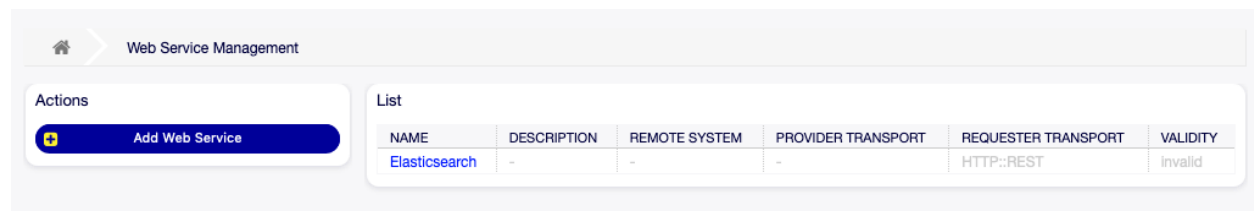
Fig. 1.216: Web Service Management Screen

Manage Web Services

To create a web service:

1. Click on the Add Web Service button in the left sidebar.
2. Fill-in the required fields.
3. Click on the Save button.

To edit a web service:
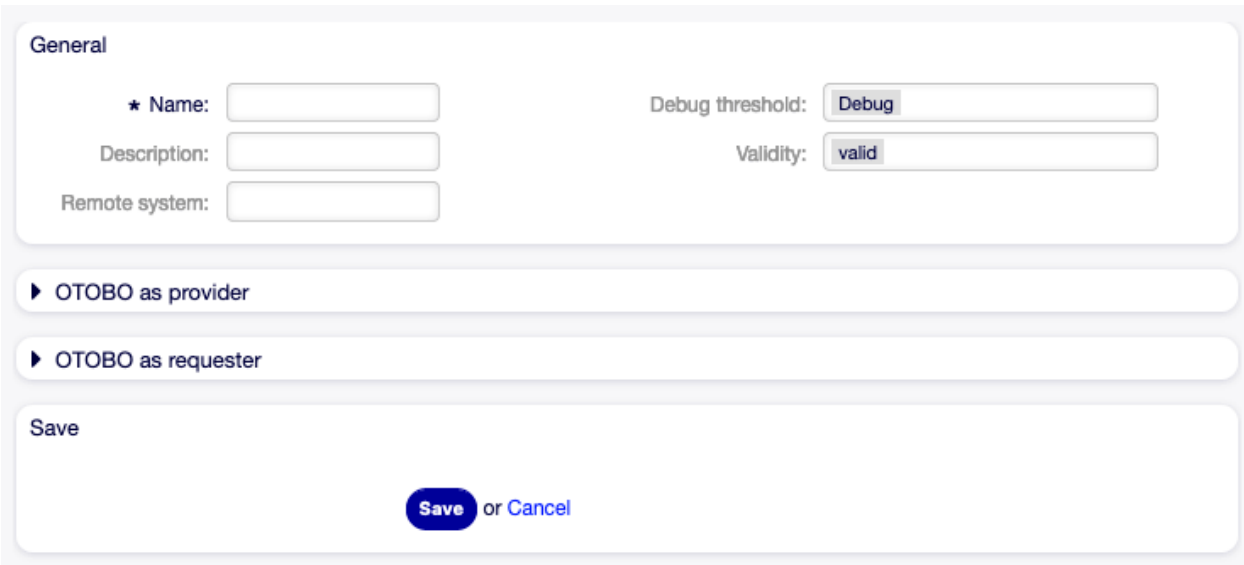
1. Click on a web service in the list of web services.
2. Modify the fields.
3. Click on the Save or Save and finish button.

To delete a web service:

1. Click on a web service in the list of web services.
2. Click on the Delete Web Service button in the left sidebar.
3. Click on the Delete button in the confirmation dialog.
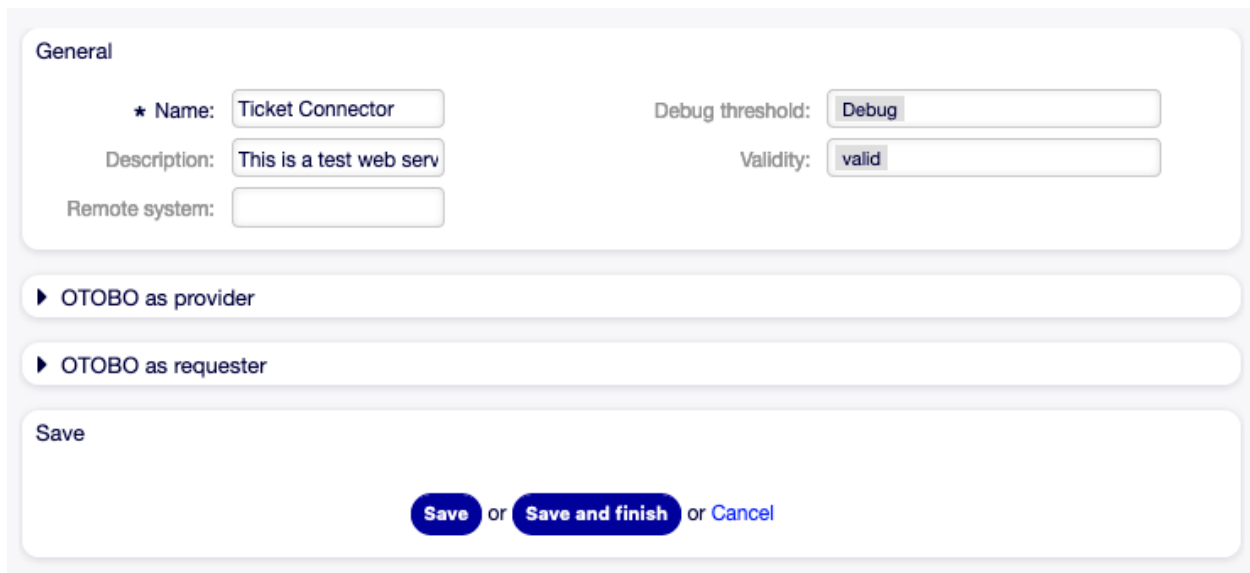
To clone a web service:

1. Click on a web service in the list of web services.
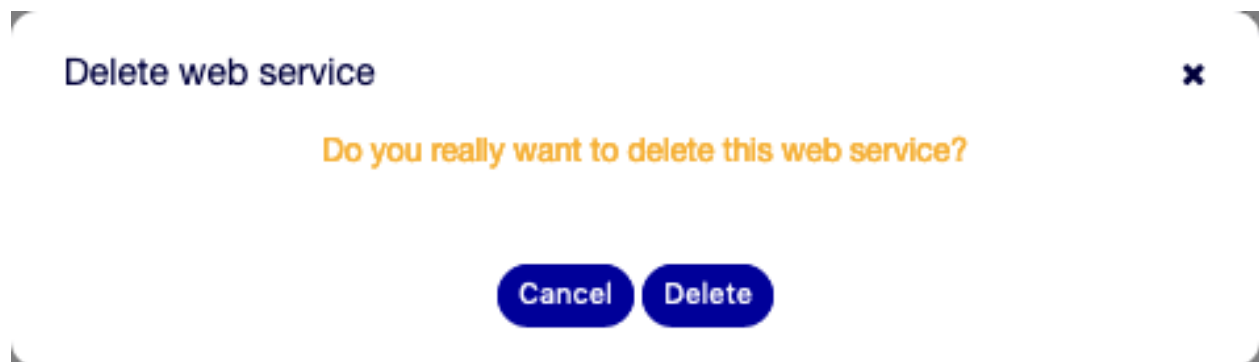
Fig. 1.217: Create New Web Service Screen

Fig. 1.218: Edit Web Service Screen

Fig. 1.219: Delete Web Service Screen

2. Click on the Clone Web Service button in the left sidebar.

3. Enter a new name for the web service.

Fig. 1.220: Clone Web Service Screen

To export a web service:

1. Click on a web service in the list of web services.

2. Click on the Export Web Service button in the left sidebar.

3. Choose a location in your computer to save the `Export_ACL.yml` file to.

Warning: All stored passwords in the web service configuration will be exported in plain text format.

To see the configuration history of a web service:

1. Click on a web service in the list of web services.

2. Click on the Configuration History button in the left sidebar.

Fig. 1.221: Web Service Configuration History Screen

To use the debugger for a web service:

1. Click on a web service in the list of web services.

2. Click on the Debugger button in the left sidebar.

To import a web service:

1. Click on the Add Web Service button in the left sidebar.
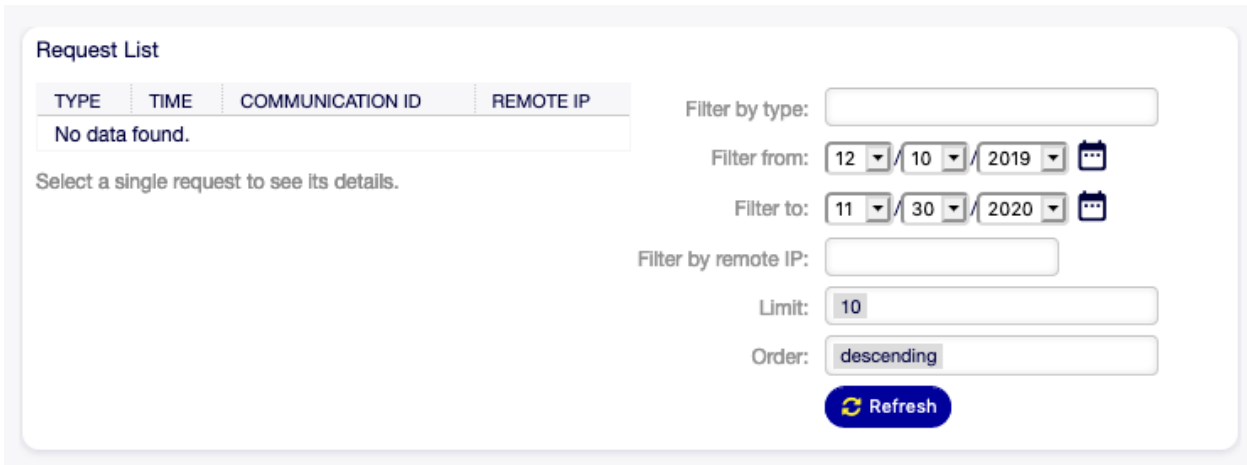
Fig. 1.222: Web Service Debugger Screen

2. Click on the Import Web Service button in the left sidebar.

3. Click on the Browse… button in the dialog.

4. Select a previously exported `.yml` file.

5. Add a name for the imported web service (optional). Otherwise the name will be taken from the configuration file name.

6. Click on the Import button.

Web Service Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

General Web Service Settings
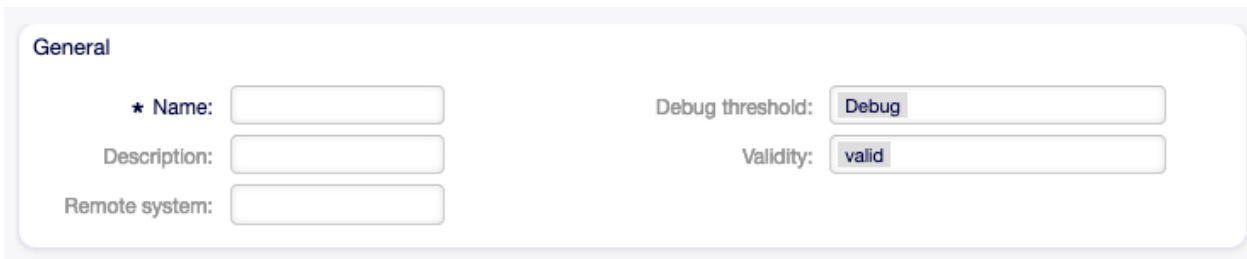
Fig. 1.223: Web Service Settings - General

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Description Like comment, but longer text can be added here.

Remote system

Debug threshold The default value is Debug. When configured in this manner, all communication logs are registered in the database. Each subsequent debug threshold value is more restrictive and discards communication logs of lower order than the one set in the system.

Debug threshold levels (from lower to upper):

- Debug
- Info
- Notice
- Error

Validity Set the validity of this resource. Resources can only be used in OTOBO if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

Provider Web Service Settings

Note: To access the otobo webservice, please use the following URL: https://SERVERADDRESS/otobo/nph-genericinterface.pl/Webservice/WEBSERVICENAME/OPERATION

Fig. 1.224: Web Service Settings - OTOBO as Provider

Network transport Select which network transport you would like to use with the web service. Possible values are HTTP::REST and HTTP::SOAP.

Note: After selecting the transport method, you have to save the configuration by clicking on the Save button. A Configuration button will be displayed next to this field.

Configuration The Configuration button is visible only after a network transport was selected and saved. See the configuration for OTOBO as Provider - HTTP::REST and OTOBO as Provider - HTTP::SOAP below.

Add Operation This option is visible only after a network transport was selected and saved. Selecting an operation will open a new screen for configuration.

Fig. 1.225: Web Service Settings - OTOBO as Provider - Operation

OTOBO as Provider - HTTP::REST

To use the OTOBO REST interface, choose the network transport method "HTTP::REST". Save and reload the screen to load the ticket operations.

> Web Service Settings - OTOBO as Provider - HTTP::REST

You should now be able to select an operation.

Operations

There are different Ticket Operations which all serve a specific job:

- Ticket::TicketCreate
- Ticket::TicketGet
- Ticket::TicketSearch
- Ticket::TicketUpdate
- Ticket::TicketHistoryGet

In this example, we are going to use the Ticket::TicketCreate operation. Click on "Add Operation" and choose the "Ticket::TicketCreate" operation. Choose a descriptive name, save the operation and go back to the webservice overview.

You now should see a new entry "Route mapping for Operation 'TicketCreate'". Enter for example "/TicketCreate"

This will define the route, which will translate to: `https://YOURSERVER/otobo/nph-genericinterface.pl/Webservice/<WEBSERVICE_NAME>/TicketCreate`

Click "Save and finish".

Now you can send a request to the endpoint.

Here is an example using curl:

```
curl -X POST --header "Content-Type: application/json"
--data '{
        "UserLogin": "AgentUser",
        "Password": "Password",
        "Ticket": {
                "Title": "created by Webservice request",
                "QueueID":5,
                "CustomerUser": "CustomerUser",
                "State": "new",
                "PriorityID": 1
        },
        "Article": {
                "CommunicationChannel": "Email",
                "From": "test@test.de",
                "Subject": "Webservice Create Example",
                "Body": "This was created by a Webservice request!",
                "ContentType": "text/html charset=utf-8"
        }
}'
https://YOURSERVER/otobo/nph-genericinterface.pl/Webservice/<WEBSERVICE_NAME>
↪/TicketCreate
```

This request is using the least amount of attributes needed to create a new Ticket.

A full list of all attributes needed for operations can be found here:

- TicketCreate: https://github.com/RotherOSS/otobo/blob/rel-10_1/Kernel/GenericInterface/Operation/Ticket/TicketCreate.pm#L70

- TicketGet: https://github.com/RotherOSS/otobo/blob/rel-10_1/Kernel/GenericInterface/Operation/Ticket/TicketGet.pm#L70

- TicketUpdate: https://github.com/RotherOSS/otobo/blob/rel-10_1/Kernel/GenericInterface/Operation/Ticket/TicketUpdate.pm#L70

- TicketSearch: https://github.com/RotherOSS/otobo/blob/rel-10_1/Kernel/GenericInterface/Operation/Ticket/TicketSearch.pm#L70

- TicketHistoryGet: https://github.com/RotherOSS/otobo/blob/rel-10_1/Kernel/GenericInterface/Operation/Ticket/TicketHistoryGet.pm#L70

XLST-Mapping for OTOBO as Provider - HTTP::REST

The XLST standard can be used to transform XML, JSON and CSV data.

In this example, we are going to use the XLST mapping to transform the response from the webservice into Dynamic Fields.

Create a Dynamic Field of Type Ticket->Text and name it for example "RemoteTicketID".

Given the incoming data: { "incidentID" : "12345", "incidentTitle" : "Test Ticket" }

We can save the data in the Dynamic Field as follows: <example code here>

OTOBO as Provider - HTTP::SOAP

Requester Web Service Settings

Network transport Select which network transport would you like to use with the web service. Possible values are HTTP::REST and HTTP::SOAP.

> **Note:** After selecting the transport method, you have to save the configuration by clicking the Save button. A Configuration button will be displayed next to this field.

Configuration The Configuration button is visible only after a network transport was selected and saved. See the configuration for OTOBO as Requester - HTTP::REST and OTOBO as Requester - HTTP::SOAP below.

Add error handling module This option is visible only after a network transport was selected and saved. Selecting an operation will open a new screen for its configuration.

OTOBO as Requester - HTTP::REST

OTOBO as Requester - HTTP::SOAP

1.5 External Interface

> **Warning:** TODO: This part of the documentation is out out of scope and needs someone to help updating the changes introduced with the latest releases.

The need to transport valuable information about service to users is a major one for a service desk. Not always is it possible or plausible to do this via a corporate internet. The customers of a service desk need a one-stop solution for all their service needs.

OTOBO provides access to create and manage demands, as well as receive valuable information such as the service catalog and knowledge base.

The following chapter describes the administration tools needed to implement corporate identity and manage language-based content.

1.5.1 Customer Service Catalogue

Fig. 1.226: Web Service Settings - OTOBO as Provider - HTTP::SOAP
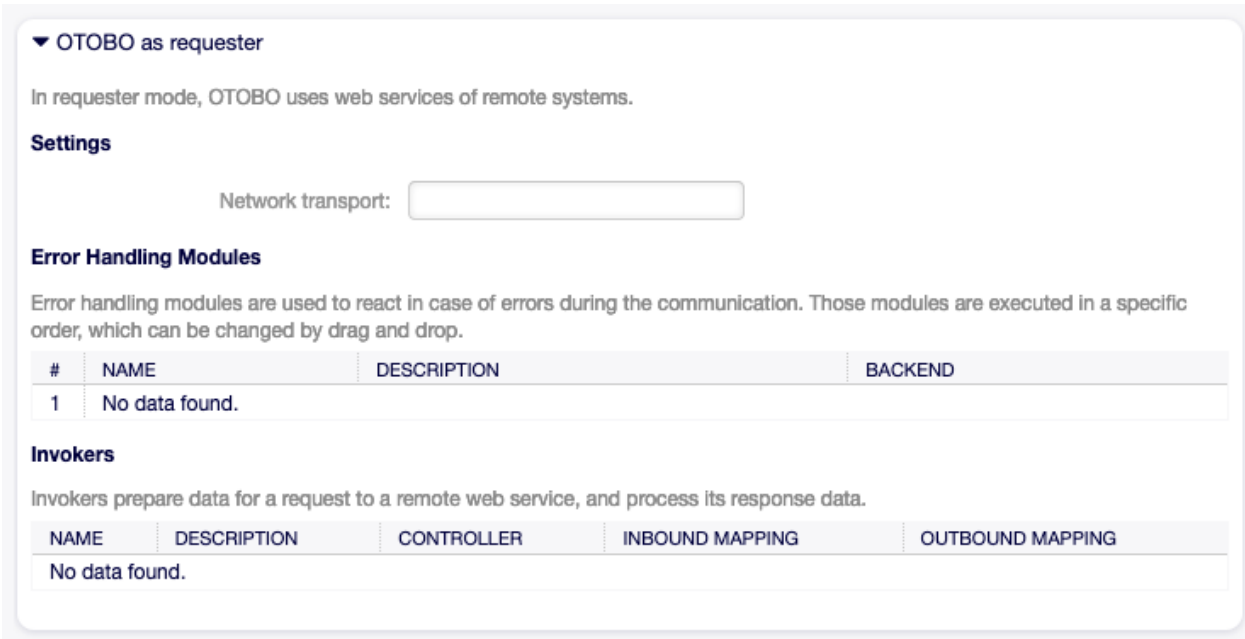
Fig. 1.227: Web Service Settings - OTOBO as Requester

Note: Rother OSS TODO: This feature will be included in one of the next OTOBO releases. If you need it beforehand, please contact sales@otobo.de, we will find a solution.

Use this screen to add categories and items for use in external interface. A fresh OTOBO installation doesn't contain any categories or items by default. The catalogue management screen is available in the Customer Service Catalogue module of the External Interface group.

This module consists of two management screens: a category management screen and an item management screen.

Manage Categories

Use this screen to add categories to collect the same items into groups. The Category Management screen in available via the Go to category management button or via the Category Management module.

To add a category:

1. Click on the Add Category button in the left sidebar.

2. Fill in the required fields.

3. Click on the Save button.

To edit a category:

1. Click on a category in the list of categories.

2. Modify the fields.

3. Click on the Save or Save and finish button.

To delete a category:

▼ General options

* Name:

The name can be used to distinguish different error handling configurations.

Description:

Error handling module backend: RequestRetry

This OTOBO error handling backend module will be called internally to process the error handling mechanism.

▼ Processing options

Configure filters to control error handling module execution.

Only requests matching all configured filters (if any) will trigger module execution.

Invoker filter:

Only execute error handling module for selected invokers.

Error message content filter:

Enter a regular expression to restrict which error messages should cause error handling module execution.
Error message subject and data (as seen in the debugger error entry) will considered for a match.
Example: Enter '^.*401 Unauthorized.*$' to handle only authentication related errors.

Error stage filter:

Only execute error handling module on errors that occur during specific processing stages.
Example: Handle only errors where mapping for outgoing data could not be applied.

Error code:

An error identifier for this error handling module.
This identifier will be available in XSLT-Mapping and shown in debugger output.

Error message:

An error explanation for this error handling module.
This message will be available in XSLT-Mapping and shown in debugger output.

Stop after match:

Define if processing should be stopped after module was executed, skipping all remaining modules or only those of the same backend.
Default behavior is to resume, processing the next module.

▼ Request retry options

Retry options are applied when requests cause error handling module execution (based on processing options).

* Schedule retry: No

Should requests causing an error be triggered again at a later time?

Submit

Save or Cancel

Fig. 1.228: Web Service Settings - OTOBO as Provider - Error Handling Module

Fig. 1.229: Web Service Settings - OTOBO as Requester - HTTP::REST

Network Transport

Properties

Type:	HTTP::SOAP
★ Endpoint:	[　　　　　　　　　　　]

URI to indicate specific location for accessing a web service.
e.g. https://local.otobo.de:8000/Webservice/Example

★ Timeout:	120

Timeout value for requests.

★ Set SOAPAction:	Yes

Set to "Yes" in order to send a filled SOAPAction header.
Set to "No" in order to send an empty SOAPAction header.

★ SOAPAction scheme:	<NameSpace><Separator><Operation>

Select how SOAPAction should be constructed.
Some web services require a specific construction.

★ SOAPAction separator:	#

Character to use as separator between name space and SOAP operation.
Usually .Net web services use "/" as separator.

★ Namespace:	[　　　　　　　　　　　]

URI to give SOAP methods a context, reducing ambiguities.
e.g urn:otobo-com:soap:functions or http://www.otobo.de/GenericInterface/actions

★ Request name scheme:	<FunctionName>DATA</FunctionName>

Select how SOAP request function wrapper should be constructed.
'FunctionName' is used as example for actual invoker/operation name.
'FreeText' is used as example for actual configured value.

★ Response name scheme:	<FunctionNameResponse>DATA</FunctionNameResponse>

Select how SOAP response function wrapper should be constructed.
'FunctionName' is used as example for actual invoker/operation name.
'FreeText' is used as example for actual configured value.

Encoding:	[　　　　　　　　　　　]

The character encoding for the SOAP message contents.
e.g utf-8, latin1, iso-8859-1, cp1250, Etc.

Authentication:	[　　　　　　　　　　　]

An optional authentication mechanism to access the remote system.

★ Use Proxy Options:	No

Show or hide Proxy options to connect to the remote system.

★ Use SSL Options:	No

Show or hide SSL options to connect to the remote system.

Sort options:	

Add new first level element: [　　　　　　　] **Add**

Outbound sort order for xml fields (structure starting below function name wrapper) - see documentation for SOAP transport.

Save or **Save and finish** or Cancel

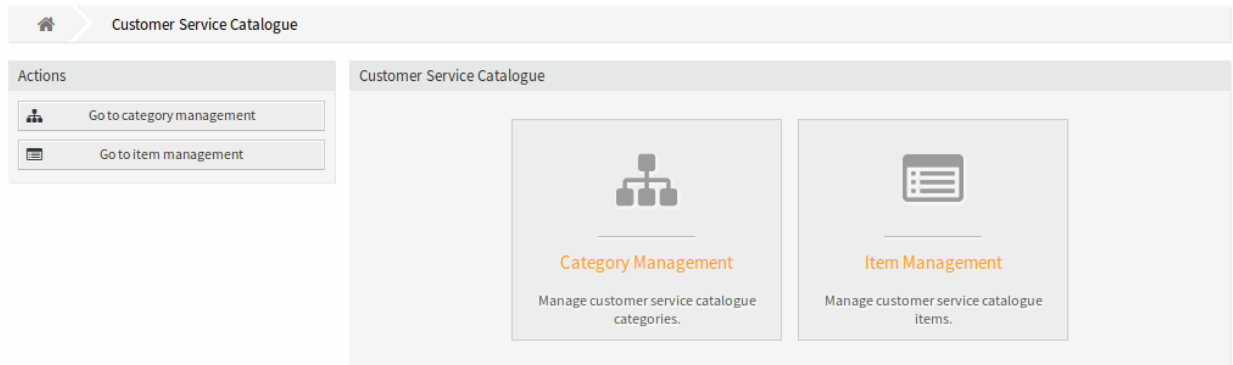Fig. 1.230: Web Service Settings - OTOBO as Requester - HTTP::SOAP

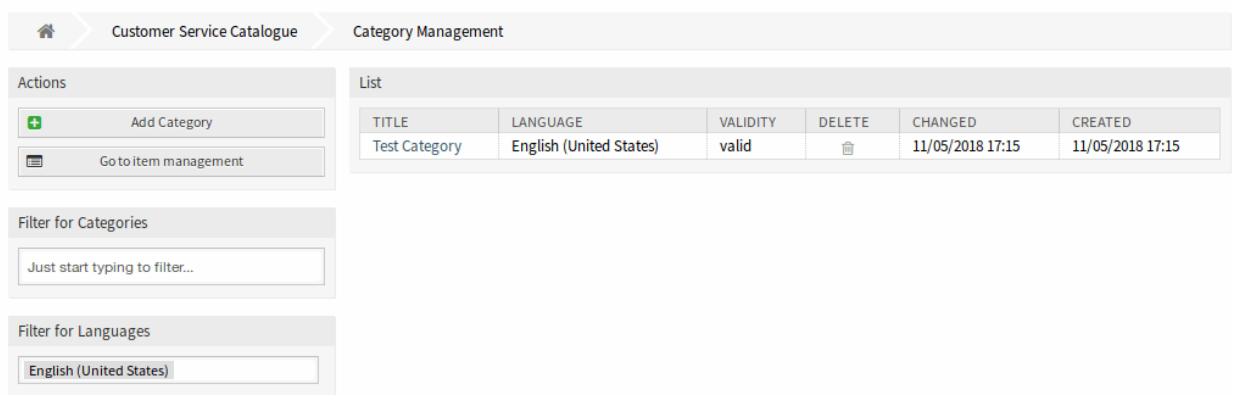Fig. 1.231: Customer Service Catalogue Management Screen
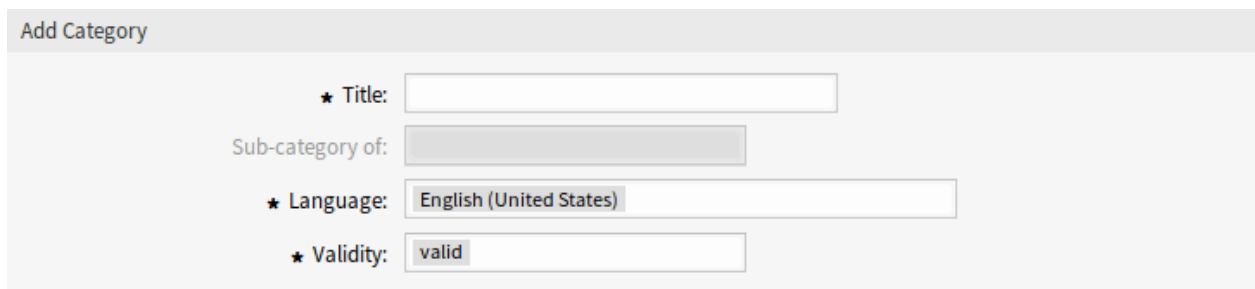

Fig. 1.232: Category Management Screen

Fig. 1.233: Add Category Screen
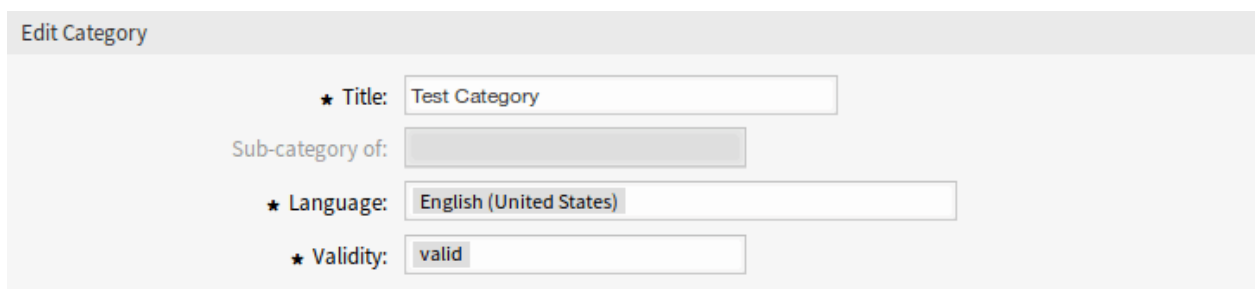

Fig. 1.234: Edit Category Screen

1. Click on the trash icon in the Delete column of the overview table.

2. Click on the Confirm button.

Fig. 1.235: Delete Category Screen

Note: If several categories are added to the system, use the filter box to find a particular category by just typing the name to filter.

Category Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Title * The name of this resource. Any type of characters can be entered to this field including upper-case letters and spaces. The name will be displayed in the overview table.

Sub-category of It is possible to add the new category under an existing one as sub-category. This will be displayed as Parent Category::Child Category.

Language * Select a language from the available languages of the system.

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

Manage Items

Use this screen to add items to the catalogue. Items can be collected into categories. The Item Management screen in available via the Go to item management button or via the Item Management module.

To add an item:

1. Click on the Add Item button in the left sidebar.

2. Fill in the required fields.

3. Click on the Save button.

To edit an item:

1. Click on an item in the list of items.

2. Modify the fields.

3. Click on the Save or Save and finish button.

To delete an item:

1. Click on the trash icon in the Delete column of the overview table.

2. Click on the Confirm button.

Fig. 1.236: Item Management Screen

Fig. 1.237: Add Item Screen

Fig. 1.238: Edit Item Screen

Fig. 1.239: Delete Item Screen

Note: If several items are added to the system, use the filter box to find a particular item by just typing the name to filter.

Item Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Internal Title * The name of this resource, that is only displayed in the admin interface. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

Item Content In this widget can be added some localized content for the item.

  Title * The name of this resource in the given language. Any type of characters can be entered to this field including uppercase letters and spaces.

  Text * The text for this item in the given language.

  Categories One ore more categories can be selected for the item in which the item should be visible.

  Note: Only those categories can be selected, that have the same language as the selected language for this widget.

  Link * A link to an internal or an external URL.

  Add new item content Select which languages should be added to create localized item content. All added languages can hold its own localized content, that are explained above.

1.5.2 Custom Pages

Note: Rother OSS TODO: This feature will be included in one of the next OTOBO releases. If you need it beforehand, please contact sales@otobo.de, we will find a solution.

Use this screen to add custom pages for use in external interface. A fresh OTOBO installation already contains some custom pages by default. The custom page management screen is available in the Custom Pages module of the External Interface group.

Manage Custom Pages

Warning: Make sure to save your changes when you finish. The new configuration will be immediately deployed.

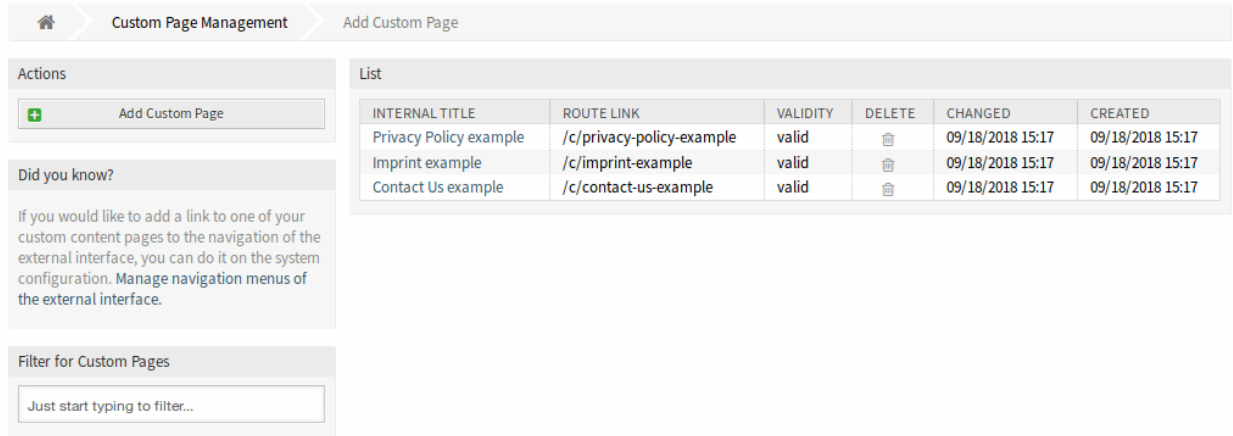---

To add a custom page:

Fig. 1.240: Custom Page Management Screen

1. Click on the Add Custom Page button in the left sidebar.

2. Fill in the required fields.

3. Click on the Save button.

To edit a custom page:

1. Click on a custom page in the list of custom pages.

2. Modify the fields.

3. Click on the Save or Save and finish button.

To delete a custom page:

1. Click on the trash icon in the fourth column of the overview table.

2. Click on the Confirm button.

Note: If several custom pages are added to the system, use the filter box to find a particular custom page by just typing the name to filter.

Custom Page Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Internal Title * The name of this resource, that is only displayed in the admin interface. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Slug * This will be the URL of the custom page. Recommended characters are lowercase letters, numbers and minus sign.

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

Custom Page Content In this widget can be added some localized content for the item.

Fig. 1.241: Add Custom Page Screen

Fig. 1.242: Edit Custom Page Screen

Fig. 1.243: Delete Custom Page Screen

Title * The name of this resource in the given language. Any type of characters can be entered to this field including uppercase letters and spaces.

Content * The text for this item in the given language.

Add new custom page content Select which languages should be added to create localized item content. All added languages can hold its own localized content, that are explained above.

1.5.3 Home Page

Note: Rother OSS TODO: This feature will be included in one of the next OTOBO releases. If you need it beforehand, please contact sales@otobo.de, we will find a solution.

Use this screen to define home page configuration for different user languages, that are displayed in the external interface. The home page management screen is available in the Home Page module of the External Interface group.

This screen contains several widget for each languages, where localized content can be added.

Warning: Make sure to save your changes when you finish. The new configuration will be immediately deployed.

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Hero Unit

Title This is the most important sentence or motto of the home page.

Background Style Define the behavior of background image and background color. The following options are available:

- Only use the background image

- Only use the background color

- Use the background image and overlay it with the selected color

Background Image Select an image used as background image for the hero unit. To select an image, click on the Select image to upload button and chose an image from your file system.

Background Color To change the background color, just select a new color from the color palette. You can chose from the pre-selected colors or define other colors by choosing it from the color selector or typing the hexadecimal value.

Ticket List

Show a ticket list for logged in users This setting defines if the ticket list should be visible for logged in users or not.

Row Title This is an optional title for the row, that contains this widget in the external interface.

Fig. 1.244: Hero Unit Widget

Fig. 1.245: Ticket List Widget

Image Teasers

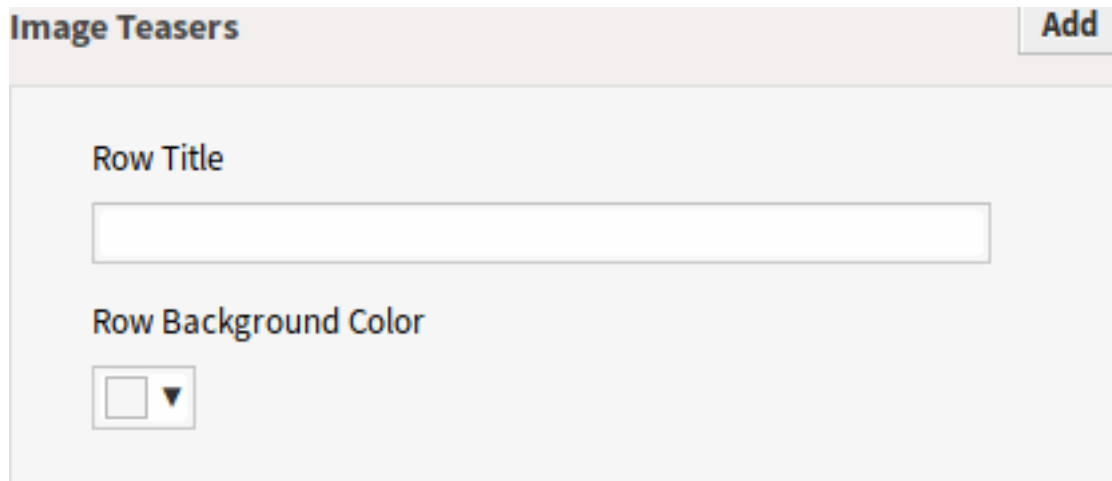The image teasers are displayed in a separate row in the external interface.

Fig. 1.246: Image Teasers Widget

Row Title This is an optional title for the row, that contains this widget in the external interface.

Row Background Color To change the background color, just select a new color from the color palette. You can chose from the pre-selected colors or define other colors by choosing it from the color selector or typing the hexadecimal value.

You can add up to three image teasers by clicking on the Add button in the top right corner of the widget.

Image Select an image to display it in the image teaser widget. To select an image, click on the Select image to upload button and chose an image from your file system.

Title * The heading text that are displayed in this item.

Text * The text for this item.

Link Target * An URL that will be opened after clicking on this item in the external interface.

Link Lists

The link lists are displayed in a separate row in the external interface.

Row Title This is an optional title for the row, that contains this widget in the external interface.

You can add up to four link lists by clicking on the Add button in the top right corner of the widget.

Title * The heading text that are displayed in this item.
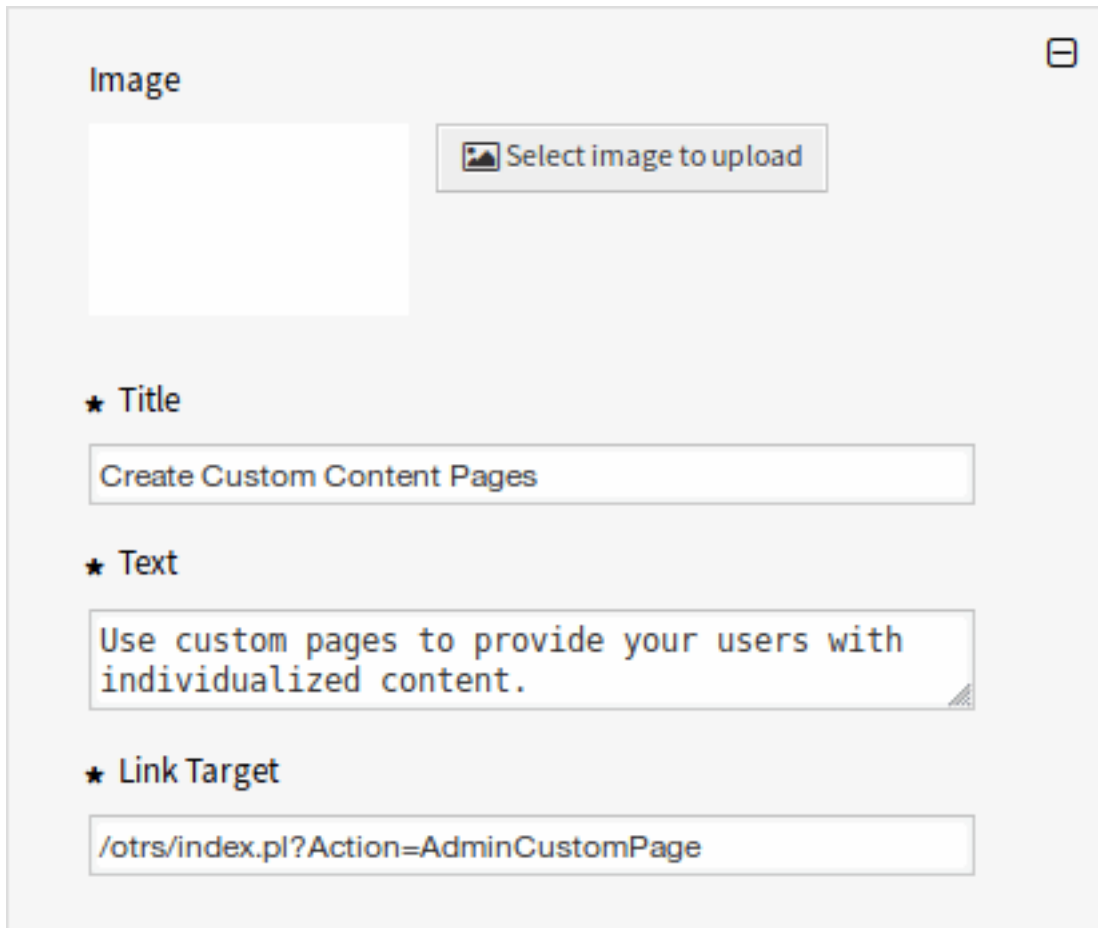
Link Text Add the text for the show more link.

Link Target Add the URL for the show more link.

Link List Type * Specifies the functionality how the item will be added.

> **Add the Items Manually** Clicking on the edit button a new widget will be visible to add the items.
>
> **Show the Results of a Search** Clicking on the edit button a new widget will be visible to add the items.

Fig. 1.247: Image Teaser

Fig. 1.248: Link Lists Widget

Fig. 1.249: Link List

Fig. 1.250: Add the Items Manually

Fig. 1.251: Show the Results of a Search

Content Cards

The link lists are displayed in a separate row in the external interface.

Fig. 1.252: Content Cards Widget

Row Title This is an optional title for the row, that contains this widget in the external interface.

You can add up to three content cards by clicking on the Add button in the top right corner of the widget.

Title * The heading text that are displayed in this item.

Text * The text for this item.

Link Text Add the text for the show more link.

Link Target An URL that will be opened after clicking on this item in the external interface.

Fig. 1.253: Content Card

1.5.4 Layout

Note: Rother OSS TODO: This feature will be included in one of the next OTOBO releases. If you need it beforehand, please contact sales@otobo.de, we will find a solution.

Use this screen to edit the layout displayed in external interface. The layout management screen is available in the Layout module of the External Interface group.
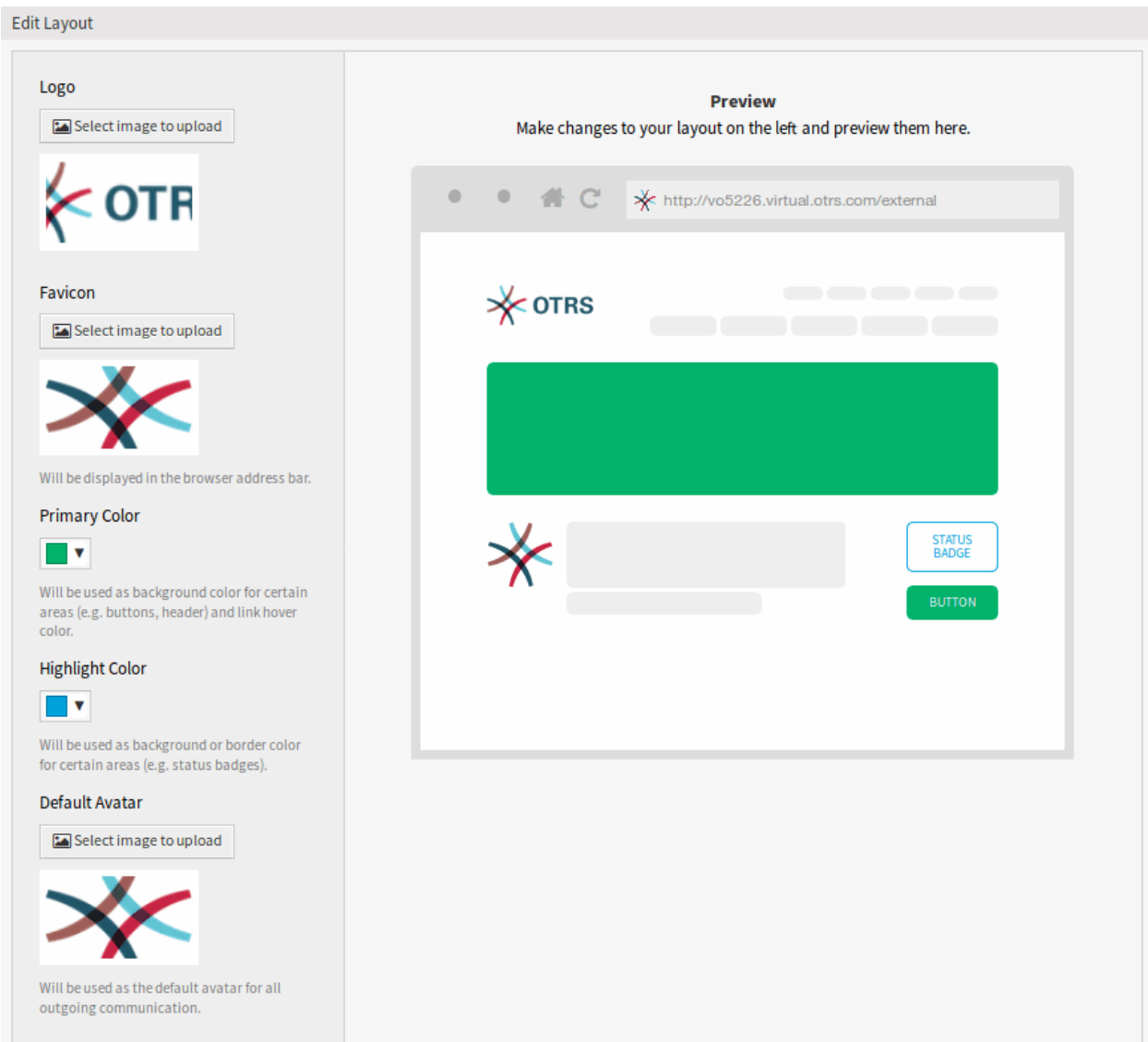

Fig. 1.254: Edit Layout Screen

A fresh OTOBO installation already contains a default layout. In this screen all parameters can be changed.

> **Warning:** Make sure to save your changes when you finish. The new configuration will be immediately deployed.

Logo The logo is a small image that is displayed in the top left corner of the external pages.

> To change the logo, click on the Select image to upload button, and select a new logo image. Recommended file format is PNG.

Favicon The favorite icon is an icon that is displayed in the URL bar of the web browser.

> To change the favorite icon, click on the Select image to upload button, and select a new icon. This is usually a 16×16 pixel image in PNG or ICO format.

Primary Color Primary color is the most important color of the external pages (see the preview screen).

> To change the primary color, just select a new color from the color palette. You can chose from the pre-selected colors or define other colors by choosing it from the color selector or typing the hexadecimal value. The new color will be displayed in the preview widget immediately.

Highlight Color Highlight color is the second color of the external pages using for status badges, links, etc.

> To change the highlight color, just select a new color from the color palette. You can chose from the pre-selected colors or define other colors by choosing it from the color selector or typing the hexadecimal value. The new color will be displayed in the preview widget immediately.

Default Avatar Will be used as the default avatar for all outgoing communication.

> To change the avatar, click on the Select image to upload button, and select a new avatar image.

Custom CSS Use this text area to add custom CSS to be applied in the external interface.
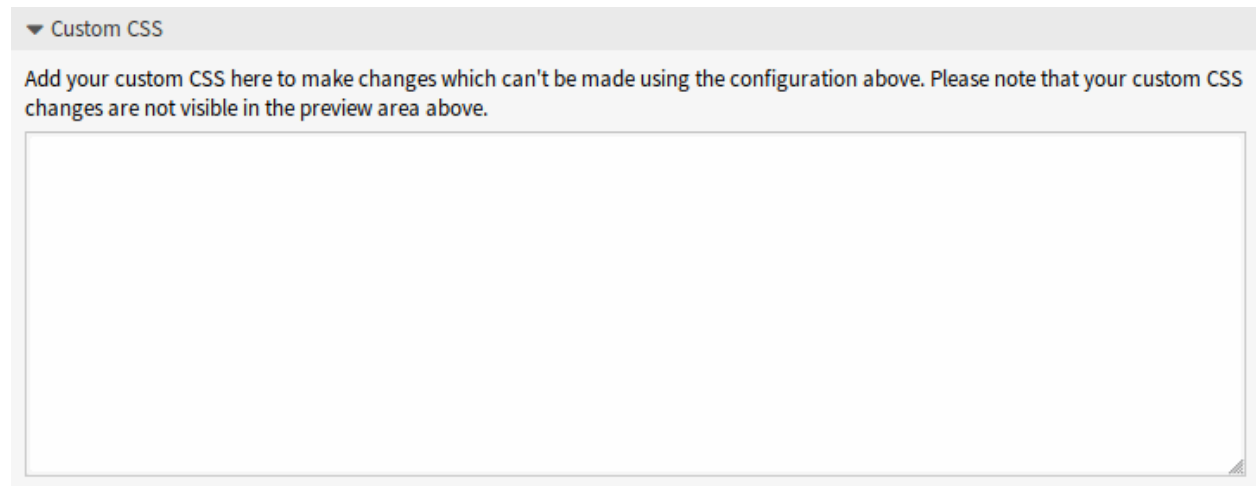

Fig. 1.255: Custom CSS Widget

Include Custom Scripts

External interface application has been built with security-first mindset in place. It contains several mechanisms to make sure that all code in the front end is provided directly by the OTOBO framework.

In addition, a security standard called Content Security Policy headers is now leveraged by the built-in web server. It serves as a last line of defense and in case some unexpected code still makes it to the client, it will make sure that the user's browser simply refuses to run it.

However, a use-case still exists in which even a modern web application might still need to run some additional code, provided it is vetted and included by the administrators. One example could be an external integrated chat solution, but could also be an inline code snippet used for external web analytics.

See also:

Including custom scripts can be done in the System Configuration with the following settings:

- `ExternalFrontend::ExternalScripts`
- `ExternalFrontend::InlineScripts`
- `WebApp::Server::AdditionalOrigins`

Web Analytics Example

In this example, we outline the steps to include custom scripts used by a web analytics provider in the external interface application. They will be run each time the application is used in order to track and report the behavior of the user for later analysis.

We start with the provided code snippet below, from a third party provider.

```
<script async src="https://www.example.com/track/js?id=UA-XXXXXX-Y"></script>
<script>
   window.dataLayer = window.dataLayer || [];
   function track(){dataLayer.push(arguments);}
   track('js', new Date());

   track('config', 'UA-XXXXXX-Y');
</script>
```

First line of the snippet refers to an external resource which should be loaded by the `script` tag.

In the script block below, we have some inline code which is run immediately when the page is loaded.

Adding External Script Resources

First, we need to include the external script location in the appropriate configuration:

1. Go to the System Configuration screen.
2. Search for the setting `ExternalFrontend::ExternalScripts`.
3. Click on the setting in order to edit it.
4. In case a value is already present, click on the plus button. Otherwise, proceed below.
5. Enter the full location of the external resource in the text field. For example: `https://www.example.com/track/js?id=UA-XXXXXX-Y`.
6. Click on the check mark in order to save the modified setting.
7. Deploy the modified system configuration.

Adding Inline Code Snippets

Next, we have to also include the inline part of the original snippet:

1. Go to the System Configuration screen.

2. Search for the setting `ExternalFrontend::InlineScripts`.

3. Click on the setting in order to edit it.

4. In case a value is already present, click on the plus button. Otherwise, proceed below.

5. Enter the full code snippet in the text area, minus any `script` tags. For example:

```javascript
window.dataLayer = window.dataLayer || [];
function track(){dataLayer.push(arguments);}
track('js', new Date());

track('config', 'UA-XXXXXX-Y');
```

6. Click on the check mark in order to save the modified setting.

7. Deploy the modified system configuration.

Rebuilding the External Interface Application

In order to apply the changes, we also need to rebuild the external interface application. Drop down to shell, and execute the following command:

```
bin/otobo.WebServer.pl --deploy-assets
```

Whitelisting Additional Origins in the Security Headers

If you now try to access the external interface application, you will be able to verify that the scripts are included in the code. However, your browser will probably block access to all inline and external resources, therefore the code might fail with some errors.

This behavior is by design, since external resources can only be loaded if they are specifically whitelisted in the Content Security Policy headers.

To check for blocked code, please use suitable web browser inspection tools. In our example, we will be using Mozilla Firefox and its web console available via Tools → Web Developer → Web Console menu item, or via the F12 shortcut key.

For the example code snippet, you might receive following errors in the console when the application is accessed:

From the console errors we can see that the external script resource was prevented from being loaded (lines 1 and 3). In addition to that, two evaluation calls were also blocked (lines 2 and 5). All errors reference a Content Security Policy rule under the name of `script-src`, which signals script resources.

We need to add both the external resource and the evaluation calls to the additional origins list of the Content Security Policy headers:

1. Go to the System Configuration screen.

2. Search for the setting `WebApp::Server::AdditionalOrigins`.

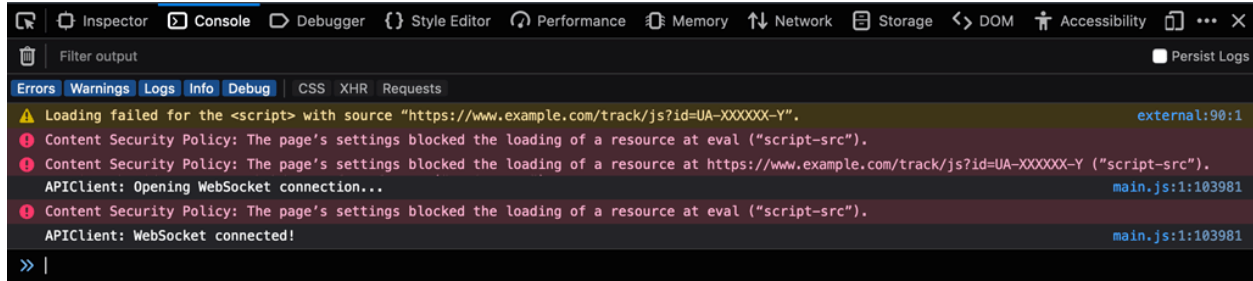3. Click on the setting in order to edit it.

Fig. 1.256: Browser Console Errors and Warnings

4. In case a value for `script-src` is already present, click on the plus button next to it. Otherwise, proceed below.

5. Enter the domain part only of the blocked resource in the text field. For example: `https://www.example.com`. This allows the external resource to be loaded.

6. Click on the plus button next to the field, so another value is added.

7. Enter the following directive in the new field, including the quotes: `'unsafe-eval'`. This allows the evaluation calls to be executed.

8. Click on the check mark in order to save the modified setting.

9. Deploy the modified system configuration.

There is no need to rebuild the external interface application at this point, as the additional origins configuration should be immediately in effect.

If you reload the external interface application, you might get some additional errors. In our example, it might be the following:
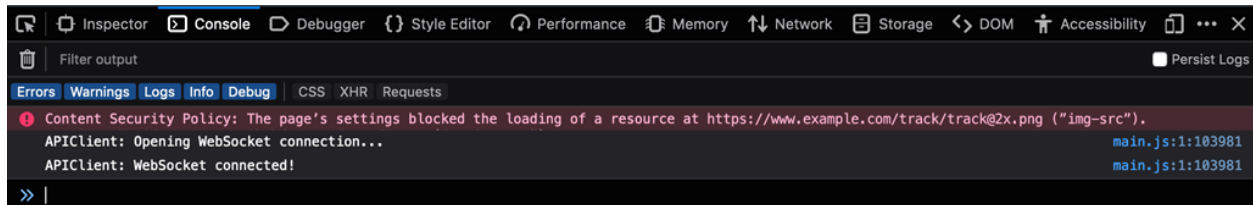
Fig. 1.257: Additional Browser Console Errors and Warnings

This error points that an additional resource that was also blocked, an image at a specific location (line 1). We can deduce this via the name of the Content Security Policy rule `img-src`, which references an image resource. In order to add it to the whitelist, try the following:

1. Go to the System Configuration screen.

2. Search for the setting `WebApp::Server::AdditionalOrigins`.

3. Click on the setting in order to edit it.

4. In case a value for `img-src` is already present, click on the plus button next to it. Otherwise, proceed below.

5. Enter the domain part only of the blocked resource in the text field. For example: `https://www.example.com`. This allows the external image resource to be loaded.

6. Click on the check mark in order to save the modified setting.

7. Deploy the modified system configuration.

Try again to reload the external interface application and check if there are more errors. If not, your scripts are now probably working as expected.

Unfortunately, it is not possible to predict what kind of resources your scripts might be requiring. But, no worries, you can whitelist most of them, just make sure to follow the trail of hints shown in the browser console log. Find a corresponding header rule in the configuration and update it accordingly.

Note: Some resources might only be requested by others, hence several iterations might be needed until everything is configured properly.

Warning: Whitelisting external resources opens potential security risks in your OTOBO application! Only allow those resources that you are sure are not malicious and come from reputable sources. Keep in mind that if something is secure today, does not mean it will be tomorrow. Stay safe!

1.6 Rother OSS Services

A strong partner is good to have when dealing with mission-critical systems. Whether it be maintenance or for consuming cloud services, your partner should be tightly integrated.

The following chapter describes the tools you have available to integrate your system with the powerful cloud service offered by Rother OSS.

1.6.1 Cloud Services

Note: Rother OSS TODO: This feature will be included in one of the next OTOBO releases. If you need it beforehand, please contact sales@otobo.de, we will find a solution.

Use this screen to add cloud services to the system. A fresh OTOBO installation doesn't contain any configured cloud services by default. The cloud service management screen is available in the Cloud Services module of the Rother OSS Services group.
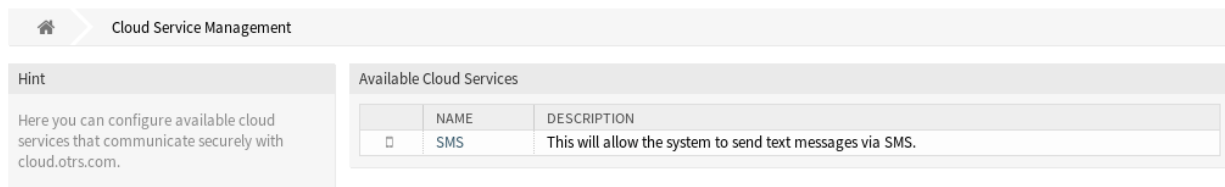
Fig. 1.258: Cloud Service Management Screen

Manage Cloud Services

Activate SMS Cloud Service

Note: Rother OSS TODO: This feature will be included in one of the next OTOBO releases. If you need it beforehand, please contact sales@otobo.de, we will find a solution.

To be able to use SMS cloud service in OTOBO, you have to activate it first. To activate the SMS cloud service:

1. Click on the Activate SMS Cloud Service button in the left sidebar.

2. Fill in the required fields.

3. Click on the Save button.

Fig. 1.259: Add Cloud Service Screen

Configuration

Note: Rother OSS TODO: This feature will be included in one of the next OTOBO releases. If you need it beforehand, please contact sales@otobo.de, we will find a solution.

The following settings are available when adding an SMS cloud service. The fields marked with an asterisk are mandatory.

Phone field for agent * Agent data field from which the mobile phone number for sending messages via SMS should be taken.

Phone field for customer * Customer data field from which the mobile phone number for sending messages via SMS should be taken.

Sender string * Will be shown as sender name of the SMS (Not longer than 11 characters).

Allowed role members If selected, only agents assigned to these roles will be able to receive notifications via SMS.

Perform URL shortening Perform URL shortening before sending the message.

Phone black list A blacklist of telephone numbers where it is forbidden to send SMS messages to. Phone numbers must be added in international format without spaces, e.g. +491791234567, one number per field.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

Validity Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

Data Protection Information In order to be able to use the SMS transmission feature of the Rother OSS GmbH, the Data Protection Information needs to be read and understood.

1.6.2 Support Data Collector

Support data collector is used to collect some data and sent to Rother OSS on a regular basis.

Use this screen to review the data to be sent to Rother OSS.

Manage Support Data Collector

Support data collector is used to collect some data and sent to Rother OSS on a regular basis, if the system is registered. To register your system:

1. Click on the System Registration button in the left sidebar.

2. Follow the registration instructions.

To show what kind of data will be sent:

1. Click on the Show transmitted data button in the left sidebar.

2. Review the System Registration Data and Support Data in the newly opened screen.

To manually trigger the support data sending:

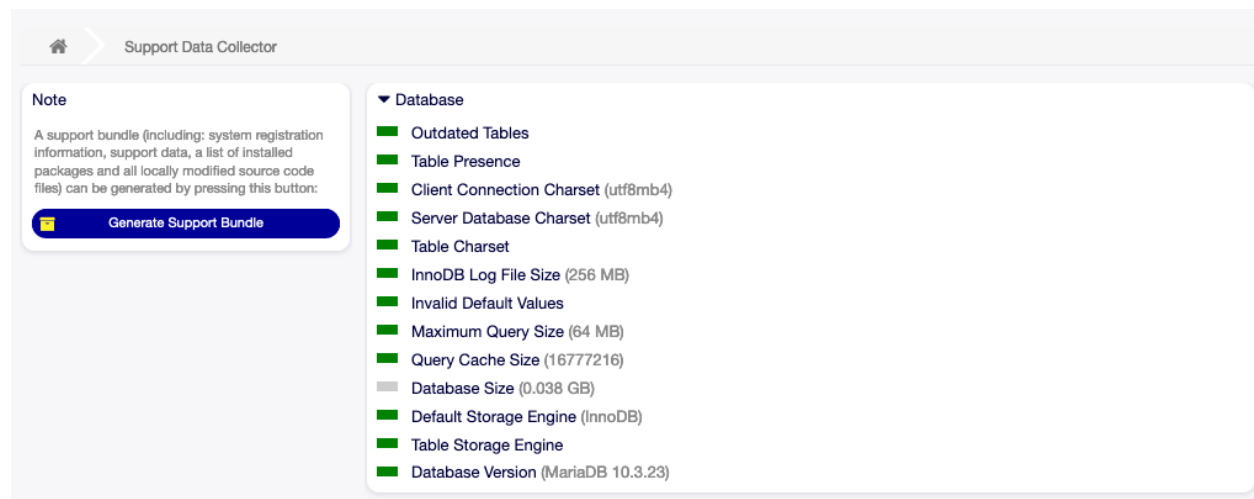1. Click on the Send Update button in the left sidebar.

Fig. 1.260: Download Support Bundle Screen

To generate a support bundle:

1. Click on the Generate Support Bundle button in the left sidebar.

2. Download the generated support bundle.

3. Open it with an archive manager and review the content.

Collected Data

The screen contains several sections. Each section has some entries with a traffic light, that indicates the following:

- Gray LED means information, just displays a value.

- Green LED means OK, the entry has a good value.

- Yellow LED means notification, you have to check the value, but it is not an error.

- Red LED means error, you have to do something to solve the issue.

Cloud Services Section

This section displays information about OTOBO cloud services.

Available SMS This entry shows information about your available SMS messages. If they are getting low, the LED changes to red.

Database Section

This section displays information about the database used by OTOBO.

Outdated Tables Display the outdated database tables. Green LED means, there are no outdated tables.

Table Presence Display whether all needed tables exist in the database or not.

Client Connection Charset Display the character set for the client connection. It must be utf8.

Support Bundle

The Support Bundle has been Generated

Please choose one of the following options.

Send by Email

Name: Admin OTOBO

Email: admin@acme.com

✈ Send

The support bundle will be sent to OTOBO Team via email automatically.

Download File

⬇ Download

A file containing the support bundle will be downloaded to the local system. Please save the file and send it to the OTOBO Team, using an alternate method.

Close this dialog

Fig. 1.261: Download Support Bundle Dialog

Server Database Charset Display the character set of the database server. It must be `utf8`.

Table Charset Display the character set of the database table. It must be `utf8`.

InnoDB Log File Size Display the log file size for InnoDB driver. It must be at least `512 MB`.

Invalid Default Values Display the invalid default values. Green LED means, there are no invalid default values.

Maximum Query Size Display the maximum size of a database query. It must be at least `1024 MB`.

Database Size Display the size of database. This is just an information.

Default Storage Engine Display the default storage engine of the database. It must be `InnoDB`.

Table Storage Engine Display the storage engine of the database tables. It must be `InnoDB`.

Database Version Display the database driver version. Green LED means, the version is high enough.

Document Search Section

This section displays information about document search and the used cluster.

Cluster The name of the used cluster.

Cluster Health Details Display some internal variables of the used cluster.

Indices Health Display information about indices.

Indices Size Display the size of each index.

Node Health Display information about the used node.

Operating System Section

This section displays information about the running operating system and installed software components.

Environment Dependencies Display information about environment dependencies.

OTOBO Disk Partition Display the disk partition to where OTOBO is installed.

Information Disk Partitions Usage Display the used space per disk partitions.

Distribution Display the distribution name of the operating system.

Kernel Version Display the kernel version of the operating system.

System Load Display the system load of the operating system. The system load should be at maximum the number of CPUs the system has (e.g. a load of 8 or less on a system with 8 CPUs is OK).

Perl Version Display the version of Perl.

Free Swap Space (%) Display the free swap space as percentages. There should be more than 60% free swap space.

Used Swap Space (MB) Display the used swap space in megabytes. There should be no more than 200 MB swap space used.

OTOBO Section

This section displays information about the OTOBO instance.

Article Search Index Status Display information about indexed articles.

Articles Per Communication Channel Display the number of articles per communication channels.

Communication Log Display aggregated information about communications.

Communication Log Account Status (last 24 hours) Display information about communication log account status in the last 24 hours.

Concurrent Users Details Display information about the logged in users at the same time separated by hourly.

Concurrent Users Display information about the number of maximum logged in users in the same time.

Config Settings Display some important configuration settings from system configurations.

Daemon Display whether the OTOBO daemon is running or not.

Database Records Display the main OTOBO object and the related number of records in the database.

Default Admin Password Green LED means, that the default administrator password was changed.

Email Sending Queue Display the number of emails that are queued for sending.

Domain Name Display the fully qualified domain name set in system configuration setting `FQDN`.

File System Writable Display whether the file system is writable or not.

Legacy Configuration Backups Green LED means, there are no legacy configuration backup files found.

Package Installation Status Green LED means, that all packages are installed correctly.

Package Framework Version Status Green LED means, that the OTOBO framework version is suitable for the installed packages.

Package Verification Status Green LED means, that all installed packages are verified by the Rother OSS.

Session Config Settings Display the maximum allowed sessions per agents and customers.

Spooled Emails Display the number of emails that are in the sending pool.

SystemID Display the system identifier set in system configuration setting `SystemID`.

Ticket Index Module Display the ticket index module set in system configuration setting `Ticket::IndexModule`.

Invalid Users with Locked Tickets Display the number of users, who are set to invalid, but have some ticket locked for him.

Open Tickets Display the number of open tickets in the system. You will not have performance trouble until you have about 60,000 open tickets in your system.

Ticket Search Index Module Display the ticket search index module set in system configuration setting `Ticket::SearchIndex::ForceUnfilteredStorage`.

Orphaned Records In ticket_index Table Display the number of orphaned records in `ticket_index` table.

Orphaned Records In ticket_lock_index Table Display the number of orphaned records in `ticket_lock_index` table.

Time Settings Display timezone information for OTOBO, for the calendars and for users.

UI - Agent Skin Usage Display the used skins per agents.

UI - Agent Theme Usage Display the used theme on the agent interface.

UI - Special Statistics Display some statistics about personal modifications like using favorites, custom menu ordering, etc.

OTOBO Version Display the version number of OTOBO.

1.7 Administration

Any system requires configuration. Configuring a system should be an easy task and the tools for configuration fit-for-purpose.

OTOBO offers several administration tools to configure, monitor, control and extend OTOBO.

1.7.1 Calendars

When working with customers, resource planning and scheduling can be a complex task. Appointments enable you to meet your customers where and whenever needed.

OTOBO supports this requirement with calendars. Calendars allow management of appointments and resources inside the ticket system. You can connect your tickets to scheduled tasks and make them available to all users to see. This feature adds transparency to show your teams workload and prevent users from promising resources which are not available.

Use this screen to manage calendars in the system. A fresh OTOBO installation contains no calendars by default. The calendar management screen is available in the Calendars module of the Administration group.
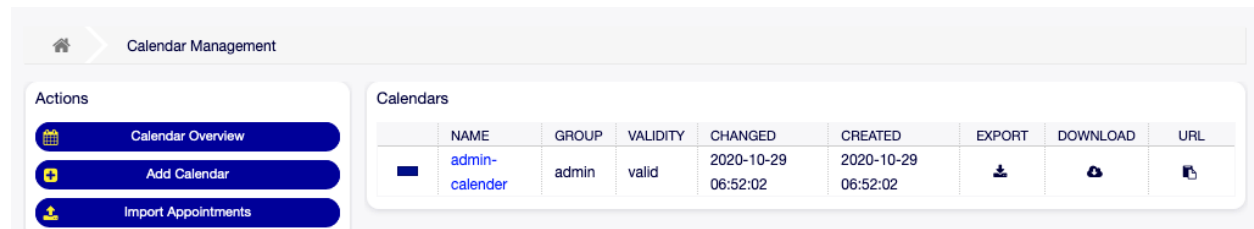
Fig. 1.262: Calendar Management Screen

Manage Calendars

To add a new calendar:

1. Click on the Add Calendar button in the left sidebar.

2. Fill in the required fields.

3. Click on the Save button.

> **Warning:** Calendars can not be deleted from the system. They can only be deactivated by setting the Validity option to invalid or invalid-temporarily.

Fig. 1.263: Add New Calendar Screen

To edit a calendar:

1. Click on a calendar in the list of calendars.

2. Modify the fields.

3. Click on the Save or Save and finish button.

Fig. 1.264: Edit Calendar Screen

To export a calendar:

1. Click on the export icon in the list of calendars.

2. Choose a location in your computer to save the `Export_Calendar_CalendarName.yml` file.

To import calendars:

1. Click on the Browse… button in the left sidebar.

2. Select a previously exported `.yml` file.

3. Click on the Overwrite existing entities checkbox, if you would like to overwrite the existing calendars.

4. Click on the Import Calendar button.

Note: If several calendars are added to the system, use the filter box to find a particular calendar by just typing the name to filter.

Calendar Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

General Calendar Settings

Calendar name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Color * The calendar color that will be displayed in the calendar overview screens.

To change the calendar color, just select a new color from the color palette. You can chose from the pre-selected colors or define other colors by choosing it from the color selector or typing the hexadecimal value.

Permission group * Select which Groups can access the calendar.

Depending on the group field, the system will allow users the access to the calendar according to their permission level.

- Read only: users can see and export all appointments in the calendar.
- Move into: users can modify appointments in the calendar, but without changing the calendar selection.
- Create: users can create and delete appointments in the calendar.
- Read/write: users can manage the calendar itself.

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

Calendar Ticket Appointments

Define rules for creating automatic appointments in this calendar based on ticket data. To add a new rule, click on the Add Rule button.

Start date Select a start date for the ticket appointment.

End date Select the end date for the ticket appointment.

Queues * Select one or more queues to narrow down for which tickets appointments will be automatically created.

Search attributes Additional search attributes can be added for further filtering by selecting an attribute and clicking on the ⊞ button.

Import Appointments

If at least one calendar have been added to the system, it is possible to import some appointments into the calendar.

To import some appointments:

1. Click on the Import Appointments button in the left sidebar.
2. Upload an iCal file and select a calendar.
3. Click on the Import appointments button.

Fig. 1.265: Calendar Settings - Ticket Appointments

Fig. 1.266: Import Appointments Screen

Upload * Click on the Browse... button, and select a valid iCal (`.ics`) file to upload.

Calendar * Select an available calendar.

> **Note:** If desired calendar is not listed here, please make sure that you have at least create permissions.

Update existing appointments? If checked, all existing appointments in the calendar with same `UniqueID` will be overwritten.

1.7.2 FAQ Category

Use this screen to manage categories available in FAQ articles. The FAQ package can be installed from the OTOBO addons repository with the Package Manager. A fresh OTOBO installation already contains a category by default. The category management screen is available in the FAQ Category module of the Administration group.
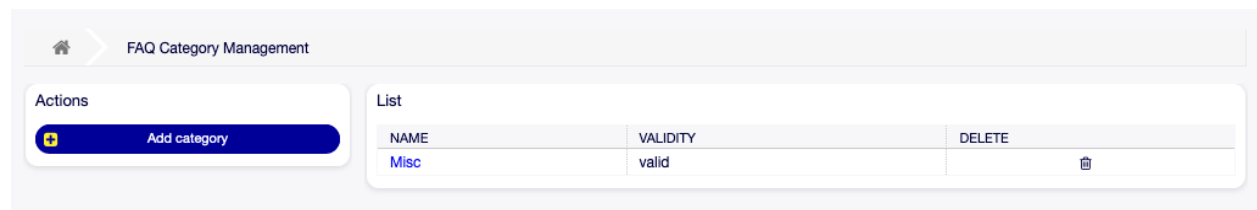
Fig. 1.267: FAQ Category Management Screen

Manage FAQ Categories

To add a category:

1. Click on the Add category button in the left sidebar.
2. Fill in the required fields.
3. Click on the Submit button.

To edit a category:

1. Click on a category in the list of categories.
2. Modify the fields.
3. Click on the Submit button.

To delete a category:

1. Click on the trash icon in the list of categories.
2. Click on the Yes button in the confirmation dialog.

FAQ Category Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Fig. 1.268: Add Category Screen

Fig. 1.269: Edit Category Screen

Fig. 1.270: Delete Category Screen

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Subcategory of It is possible to add the new category under an existing one as sub-category. This will be displayed as Parent Category::Child Category.

Permission * Agent groups that can access articles in this category.

Validity Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

Comment * Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

1.7.3 FAQ Language

Use this screen to manage languages available in FAQ articles. A fresh OTOBO installation already contains some languages by default. The language management screen is available in the FAQ Language module of the Administration group.
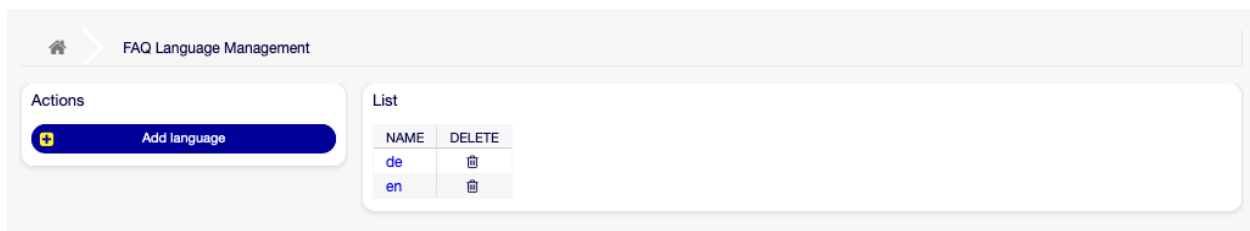
Fig. 1.271: FAQ Language Management Screen

Manage FAQ Languages

To add a language:

1. Click on the Add language button in the left sidebar.
2. Fill in the required field.
3. Click on the Submit button.
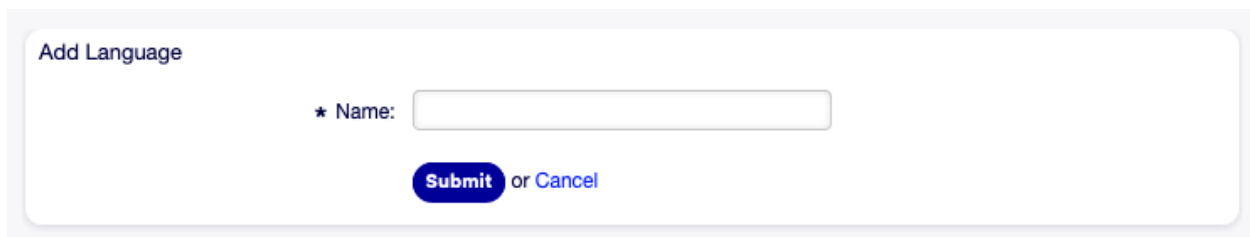
Fig. 1.272: Add Language Screen

To edit a language:

1. Click on a language in the list of languages.
2. Modify the field.

3. Click on the Submit button.

Fig. 1.273: Edit Language Screen

To delete a language:

1. Click on the trash icon in the list of languages.
2. Click on the Yes button in the confirmation dialog.

Fig. 1.274: Delete Language Screen

FAQ Language Settings

The following setting is available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Name * The ISO 639-1 code of a language.

1.7.4 Import/Export

1.7.5 Package Manager

Modular systems can be extended by adding additional software packages to the framework. Administrators need an easy way to see which features are installed in which version and for sure to add, update and remove packages.

OTOBO uses a package manager to perform all package-related activities as mentioned above in the graphical interface.

Use this screen to install and manage packages that extend the functionality of OTOBO. The package manager screen is available in the Package Manager module of the Administration group.
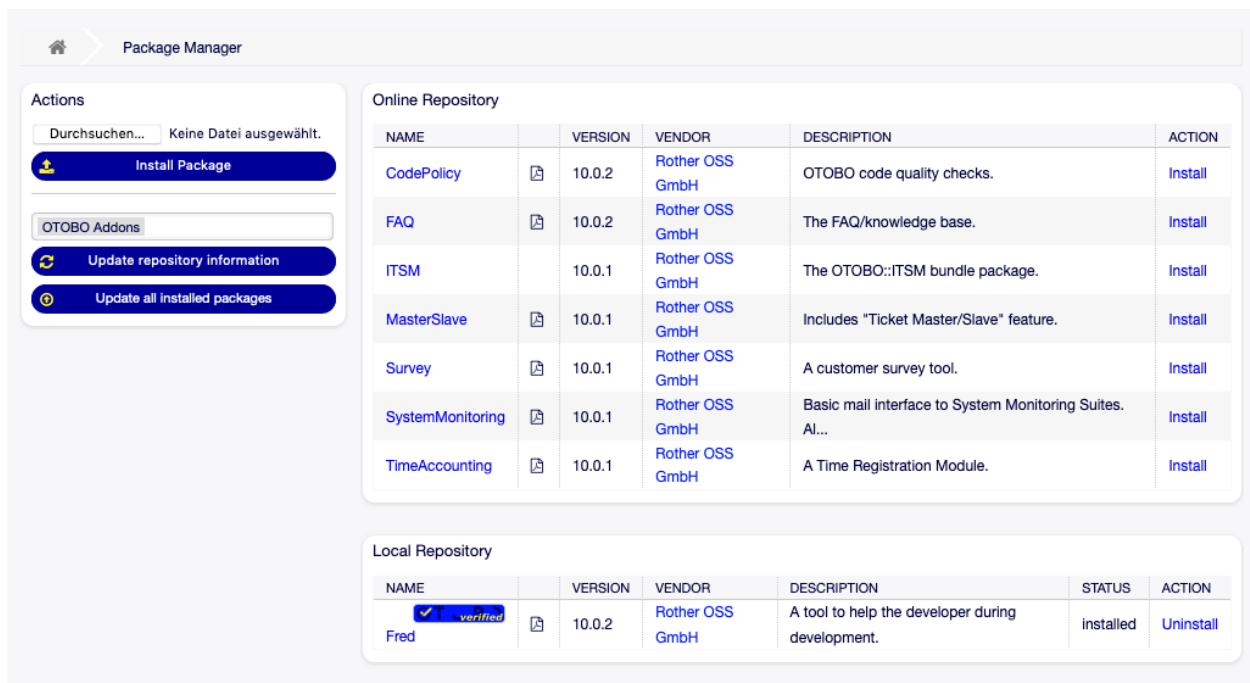
Fig. 1.275: Package Manager Screen

Manage Packages

> **Warning:** The installation of packages which are not verified by Rother OSS is possible by default, cause we love open source.

See also:

You can deactivate the installation of not verified packages in the system configuration setting `Package::AllowNotVerifiedPackages`.

Install Packages

To install a package from online repository:

1. Select an online repository from the drop-down in the left sidebar.
2. Click on the Update repository information button to refresh the available package list.
3. Select a package from the Online Repository widget and click on the Install in the last column.
4. Follow the installation instructions.
5. After installation, the package is displayed in the Local Repository widget.

See also:

The repository list can be changed in system configuration setting `Package::RepositoryList`.

To install a package from file:

1. Click on the Browse... button in the left sidebar.

Fig. 1.276: Online Repository Widget

2. Select an `.opm` file from your local file system.

3. Click on the Install Package button.

4. Follow the installation instructions.

5. After installation, the package is displayed in the Local Repository widget.

Fig. 1.277: Local Repository Widget

Update Packages

To update a package from online repository:

1. Check the available packages in the Online Repository widget if there is Update in the Action column.

2. Click on the Update link.

3. Follow the update instructions.

4. After updating, the package is displayed in the Local Repository widget.

To update a package from file:

1. Click on the Browse… button in the left sidebar.

2. Select an `.opm` file which is newer than the installed package.

3. Click on the Install Package button.

4. Follow the update instructions.

5. After updating, the package is displayed in the Local Repository widget.

To update all packages:

1. Click on the Update all installed packages button in the left sidebar.

2. Follow the update instructions.

3. After updating, the package is displayed in the Local Repository widget.

This feature reads the information of all defined package repositories and determines if there is a new version for every installed package in the system and calculates the correct order to update the packages respecting all other package dependencies, even if new versions of existing packages require new packages not yet installed in the system.

Note: If there are packages installed that do not have a corresponding repository defined in the system, they can not be updated by this feature and will be marked as failed (due to the missing online repository).

Reinstall Packages

If at least one of the package files are modified locally, the package manager marks the package as broken, and need to reinstall.

To reinstall a package:

1. Select the package from the Local Repository widget that are marked for reinstall.

2. Click on the Reinstall link in the Action column.

3. Follow the installation instructions.

Uninstall Packages

To uninstall a package:

1. Select the package from the Local Repository widget.

2. Click on the Uninstall link in the Action column.

3. Follow the uninstall instructions.

Fig. 1.278: Local Repository Widget

1.7.6 Performance Log

Performance is always a crucial topic for web-based applications. Administrators need to have the possibility to see which activities use which time to execute over time to identify potential increases over time and take proper measures.

OTOBO supports this requirement with the performance log. The performance log can, when it is activated, log activities and display various activity types and their min/max/average response time and a number of requests for different time frames.

Use this screen to view the performance log of OTOBO. The log overview screen is available in the Performance Log module of the Administration group.

Note: To be able to see performance log in OTOBO, you have to enable its setting first.
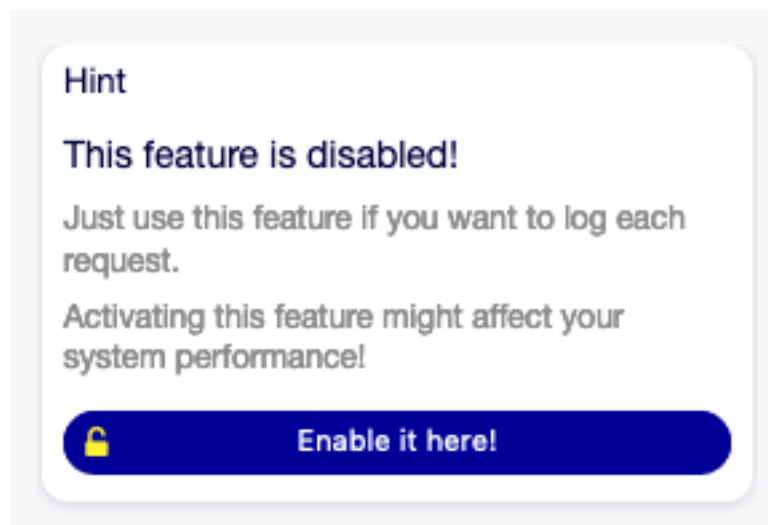
Fig. 1.279: Enable Performance Log Support

If the performance log is enabled, OTOBO collects all the information about requests and responses in an overview table.

Clicking on an entry will show the details.

1.7.7 Session Management

Administrators of web-based applications need to have access to the information who's connected to the system and, if required, delete unwanted sessions.

OTOBO offers the session management to quickly get an overview of agent and customer sessions, unique agents and customers currently logged in and the ability to kill sessions with just a mouse click.

Use this screen to manage logged in user sessions in the system. The session management screen is available in the Session Management module of the Administration group.

Manage Sessions

To see a logged in user session:

Overview

Range (last 5 m)

INTERFACE	REQUESTS	MIN RESPONSE	MAX RESPONSE	AVERAGE RESPONSE
Agent	14	0s	1s	0.07s
Admin	2	0s	0s	0s
AdminPackageManager&AJAXGetPackageUpg...	1	0s	0s	0s
AdminPerformanceLog	1	0s	0s	0s
AdminSystemConfiguration&AJAXNavigati...	1	0s	0s	0s
AdminSystemConfigurationDeployment;Su...	1	1s	1s	1s
AdminSystemConfigurationDeployment;Su...	1	0s	0s	0s
AdminSystemConfigurationDeployment;Su...	1	0s	0s	0s
AdminSystemConfigurationDeploymentHis...	1	0s	0s	0s
AdminSystemConfigurationGroup	1	0s	0s	0s
AdminSystemConfigurationGroup&Lock	1	0s	0s	0s
AdminSystemConfigurationGroup;Subacti...	2	0s	0s	0s
AdminSystemConfigurationGroup;Subacti...	1	0s	0s	0s

Range (last 30 m)

INTERFACE	REQUESTS	MIN RESPONSE	MAX RESPONSE	AVERAGE RESPONSE
Agent	55	0s	1s	0.07s
Admin	7	0s	0s	0s
AdminAppointmentCalendarManage	1	0s	0s	0s
AdminAppointmentImport	1	0s	0s	0s
AdminPackageManager	4	0s	1s	0.5s
AdminPackageManager&AJAXGetPackageUpg...	28	0s	0s	0s
AdminPackageManager&ChangeRepository	2	0s	0s	0s
AdminPerformanceLog	1	0s	0s	0s
AdminSystemConfiguration&AJAXNavigati...	1	0s	0s	0s
AdminSystemConfigurationDeployment;Su...	1	1s	1s	1s
AdminSystemConfigurationDeployment;Su...	1	0s	0s	0s
AdminSystemConfigurationDeployment;Su...	1	0s	0s	0s
AdminSystemConfigurationDeploymentHis...	1	0s	0s	0s
AdminSystemConfigurationGroup	1	0s	0s	0s
AdminSystemConfigurationGroup&Lock	1	0s	0s	0s
AdminSystemConfigurationGroup;Subacti...	2	0s	0s	0s
AdminSystemConfigurationGroup;Subacti...	1	0s	0s	0s
AgentDashboard	1	1s	1s	1s

Fig. 1.280: Performance Log Screen

Fig. 1.281: Performance Log Details Screen

Fig. 1.282: Session Management Screen

1. Select a logged in user from the list of sessions.

2. Click on the token.

3. See the details.

Fig. 1.283: Session Management Details Screen

To kill a session:

1. Select a logged in user from the list of sessions.

2. Click on the Kill this session link in the Kill column.

Warning: Clicking the Kill this session link removes the session immediately without confirmation. The unsaved work of the user will be lost!

To kill all sessions:

Fig. 1.284: Session Kill Screen

1. Click on the Kill all sessions button in the left sidebar.

> **Warning:** Clicking the Kill all session link removes all sessions immediately without confirmation. The unsaved work of the users will be lost!

> **Note:** If several users are logged in to the system, use the filter box to find a particular session by just typing the name to filter.

1.7.8 SQL Box

In a ticket system, it is usually possible to have statistics that show a summarized view of ticket information when needed. Sometimes, it is however required to access the database directly to have even more individual reports, allow external statistic tools to query information from the system or perform in-depth analysis of a ticket behavior.

Direct access to the database requires access to the command line which an administrator may not have. In addition to username and password for the command line access, which is not given by all organizations, the username and password for the database are needed. These hurdles can prevent an administrator from using the database for more complex searches and operations.

OTOBO offers application administrators the SQL Box in the GUI (graphical user interface). It allows read access to the database. All results can be seen in the GUI or exported to CSV/Excel files.

Use this screen to query SQL statements in the system. The SQL box screen is available in the SQL Box module of the Administration group.

Fig. 1.285: SQL Box Screen

Query SQL statements

Note: The SQL statements entered here are sent directly to the application database. By default, it is not possible to change the content of the tables, only `SELECT` queries are allowed.

See also:

It is possible to modify the application database via SQL box. To do this, you have to enabled the system configuration setting `AdminSelectBox::AllowDatabaseModification`. Activate it to your own risk!

To execute an SQL statement:

1. Enter the SQL statement into the SQL box.

2. Select the result format.

3. Click on the Run Query button.
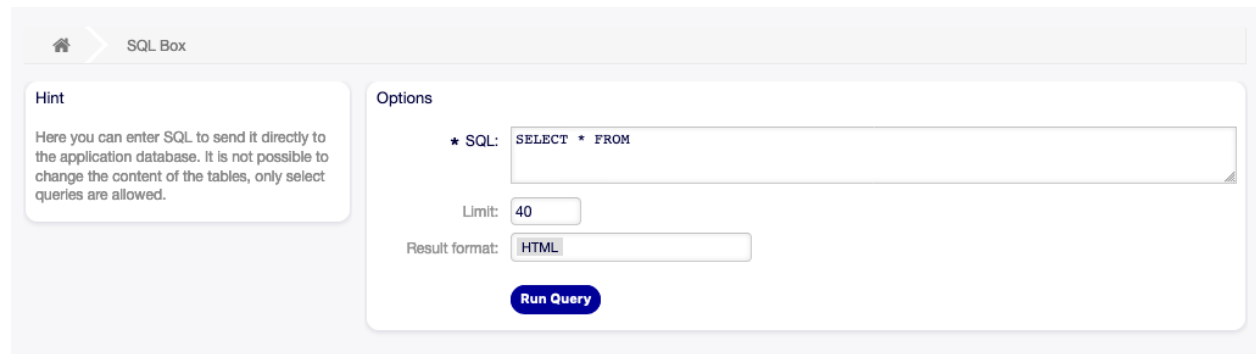
Fig. 1.286: SQL Box Widget

SQL Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

SQL * The SQL statement to be queried.

Limit Enter a number to limit the result to at most this number of rows. Leaving this field empty means there is no limit.

Note: Don't use `LIMIT` inside the SQL statement. Always use this field to limit the number of results.

Result format The format of the SQL statement result.

HTML The results are visible below the SQL box in a new widget.

CSV The results can be downloaded in comma separated plain text format.

Excel The results can be downloaded as Microsoft Excel spreadsheet.

SQL Examples

To list some information about agents and output the results as HTML:

```
SELECT id, login , first_name, last_name, valid_id FROM users
```

ID	LOGIN	FIRST_NAME	LAST_NAME	VALID_ID
1	root@localhost	Admin	OTOBO	1
2	sa	Super	Admin	1

2 Results

Fig. 1.287: SQL Box Result

To list all tables, you need to leave empty the Limit field and run the following query:

```
SHOW TABLES
```

To show the structure of the `users` table, you need to limit the results to 1 and run the following query (see the table header for the columns):

```
SELECT * FROM users
```

1.7.9 System Configuration

Modern systems have many ways to configure their behavior. Some use configuration files edited on the command line, and some use a graphical interface (and save the information to configuration files in the background), yet others use a database. Maintaining changes and auditing can sometimes be an issue, as it's not always clear who made a change. Making bulk changes is not always possible, and rolling back changes a chore.

OTOBO uses a comfortable graphical interface to configure the system. All changes to the default system configuration are stored in the database and can be audited (who changed a setting and when, what was the old and what is the new value) and rolled back to a previous state in case of misconfiguration.

Comfortable search allows finding the needed settings quickly and easily.

See also:

By using the System Configuration History package, you can easily roll back changes made by users. Contact sales@otobo.de to add this feature to your system. In one of the next OTOBO releases this feature is included in the standard framework.

Use this screen to manage the system configuration settings. OTOBO brings about 2200 configuration settings. The system configuration management screen is available in the System Configuration module of the Administration group.

Manage System Configurations

Note: For security reasons, the configuration settings for the database connection cannot be changed in the graphical user interface of the system configuration. These have to be set manually in `Kernel/Config.pm`.

To modify a system configuration, you need to do several steps. The following example shows you, how to find, modify, deploy and reset the system configuration `FirstnameLastnameOrder`.

1. Find the system configuration by entering a search term `lastname` into to search box.

   With the full-text search, all configuration settings can be scanned for one or more keywords. The full-text search not only searches through the names of the configuration settings, but also the descriptions and values. This allows an element to be found easily even if its name is unknown.
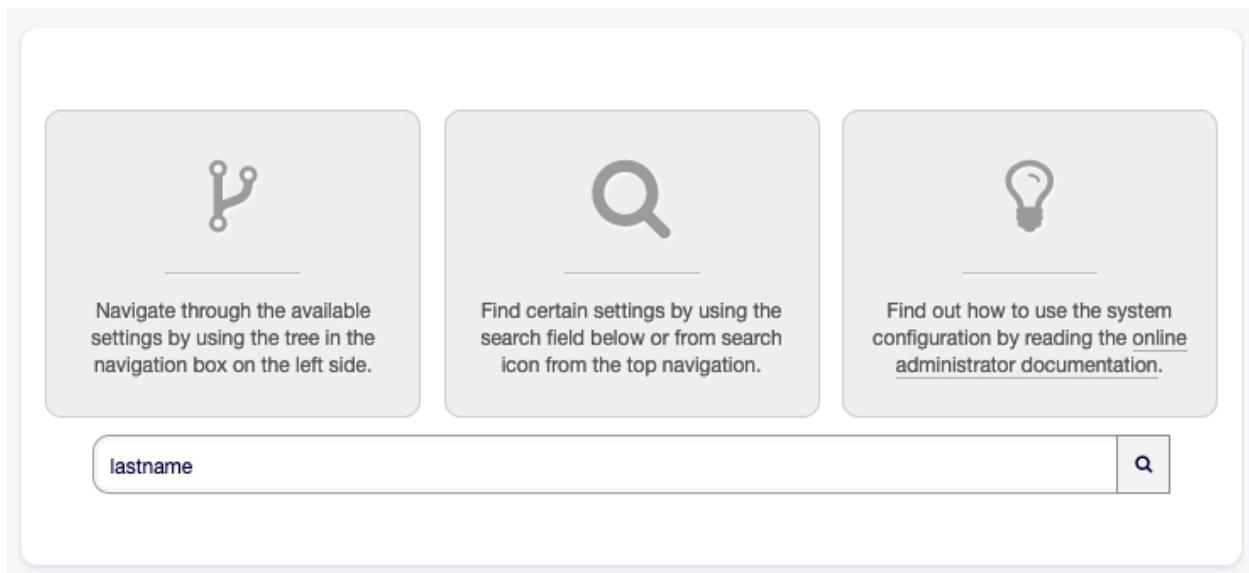
Fig. 1.288: System Configuration - Search For Setting

2. Select the setting from the search results.
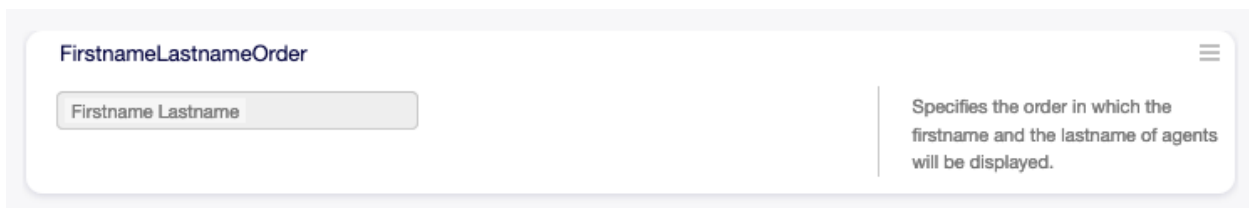
Fig. 1.289: System Configuration - Setting Found

3. Click on the header of the widget to see the options.

4. Hover the mouse over the widget body to see the Edit this setting button.

5. Click on the Edit this setting button to activate the edit mode. In edit mode the widget gets a yellow border on the left.

Note: If a setting is currently edited by an other administrator, it is not possible to get access to
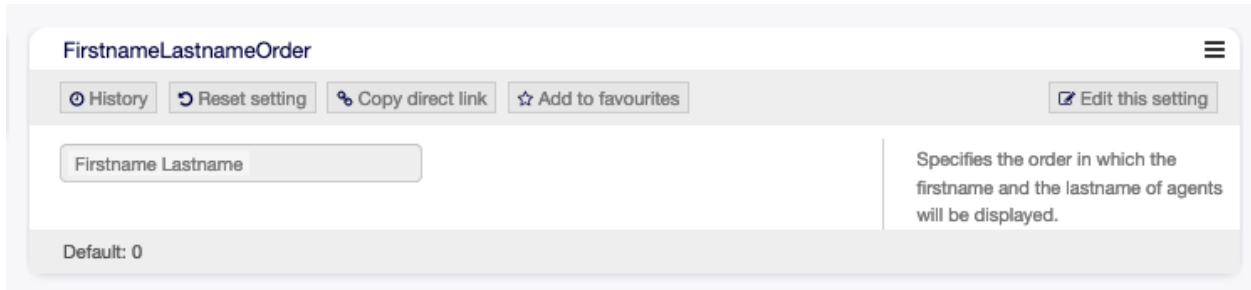
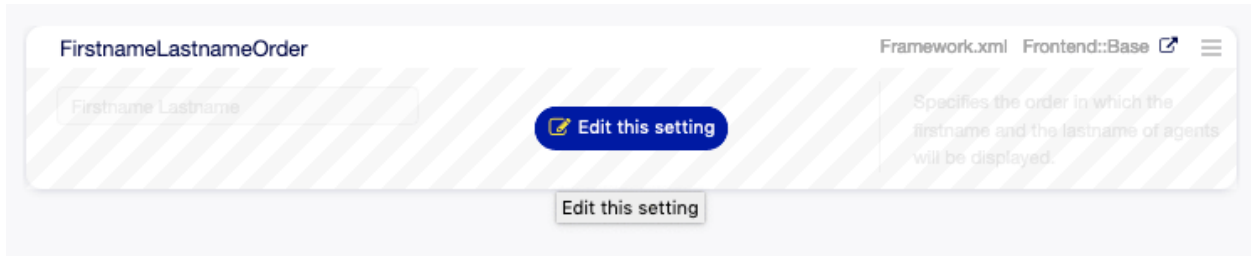Fig. 1.290: System Configuration - Setting Expanded

Fig. 1.291: System Configuration - Setting Hovered

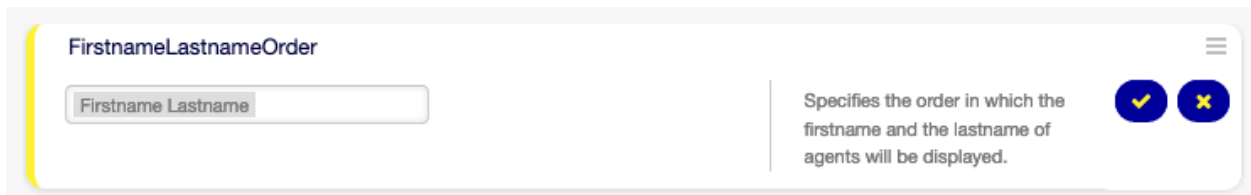the edit mode for that setting until the other administrator finished his work.

Fig. 1.292: System Configuration - Setting Clicked

6. Change the value of the setting. Editing can be cancelled by clicking the Cancel button on the right or hitting the Escape key on your keyboard. When editing is cancelled, all changes made during the current editing session are discarded.

7. Click on the Save button. If the modification is saved, the widget gets a green border on the left.

8. Go back and click on the Deployment button in the left sidebar. You are also notified in the notification bar, that you have undeployed settings.

9. Review the changes.

10. You can click on the ⬚ button in the top right corner to compare the changes side-by-side.

11. Click on the Deploy selected changes button. If several settings are changed, it is possible to deploy only the selected settings.

12. Add a deployment comment, that explain for other administrators, what is changed and why. Use full sentence here.
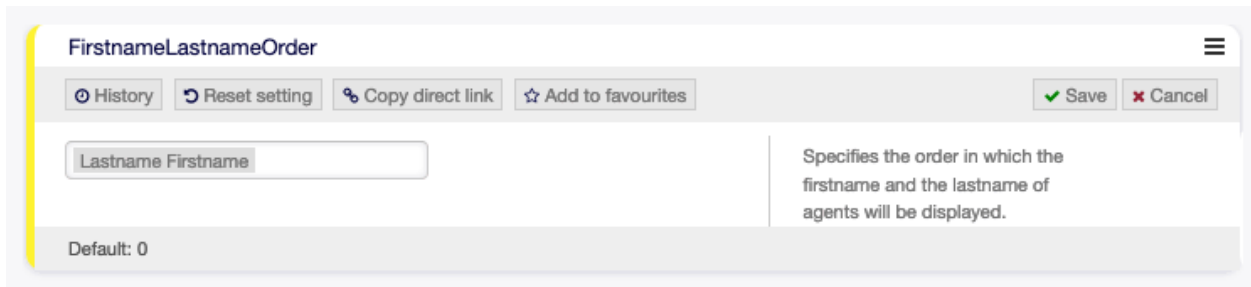
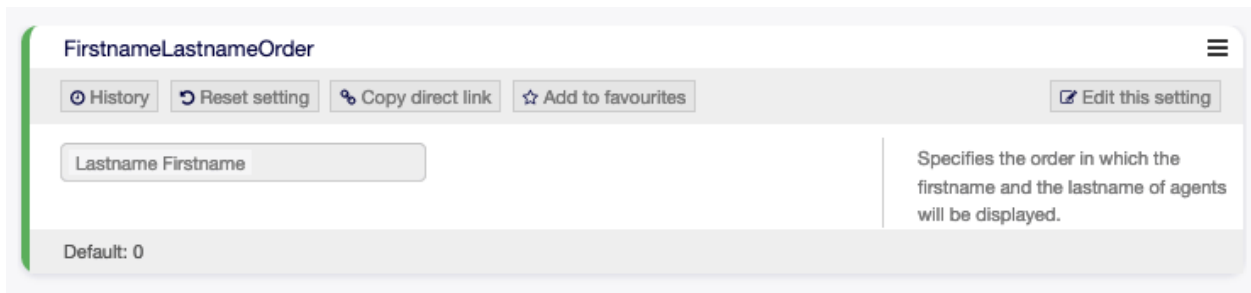Fig. 1.293: System Configuration - Setting Modified

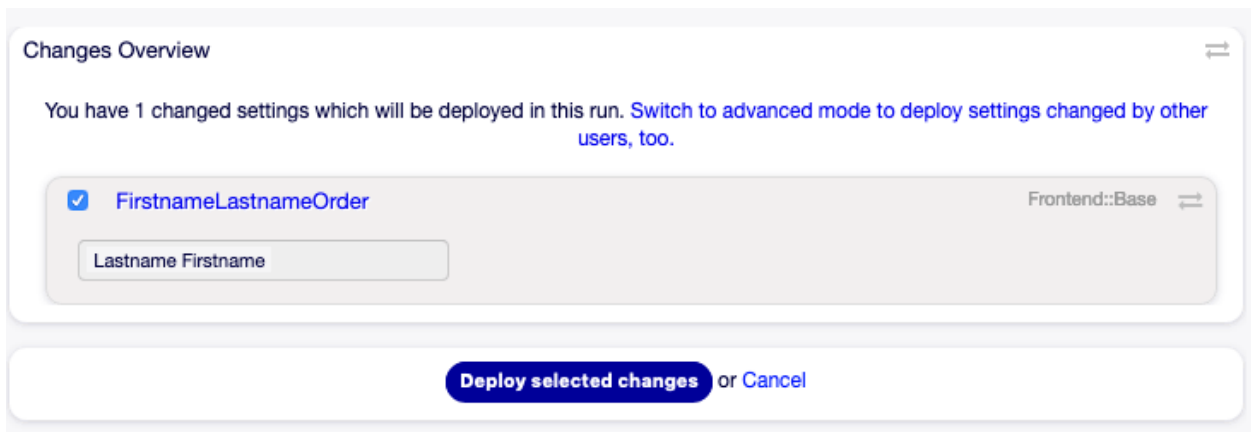Fig. 1.294: System Configuration - Setting Saved

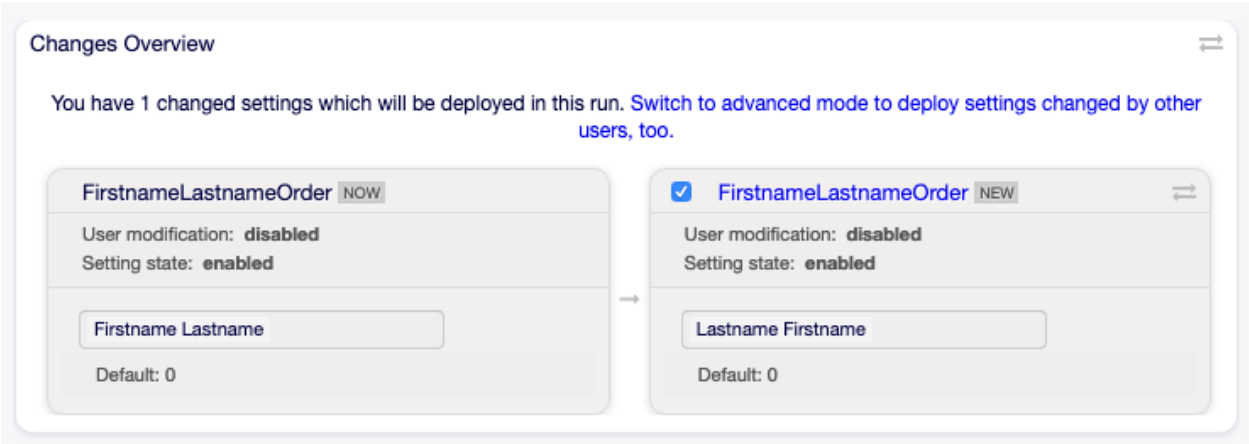Fig. 1.295: System Configuration - Setting Changes

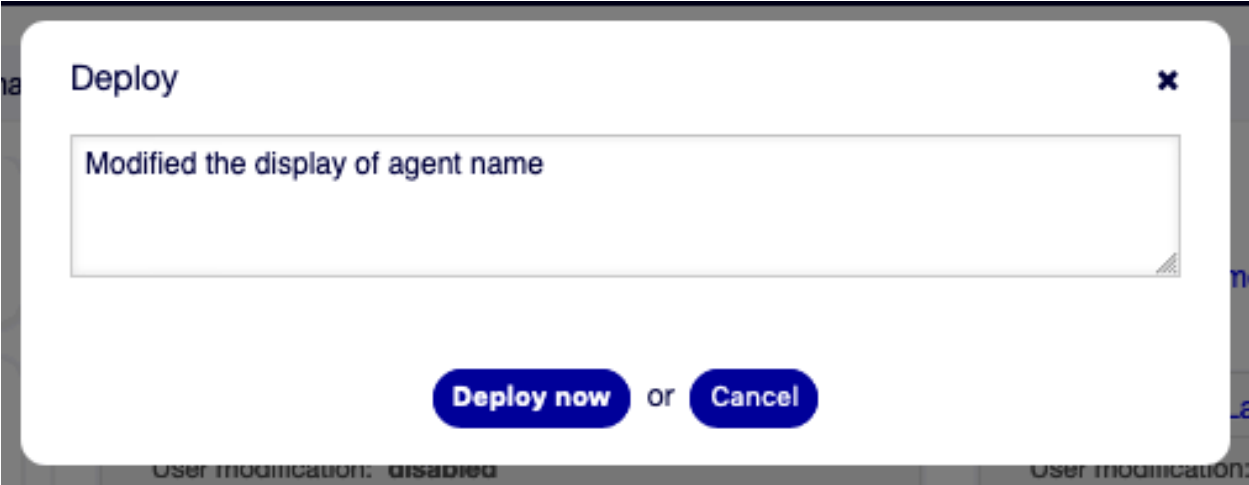Fig. 1.296: System Configuration - Setting Different

Fig. 1.297: System Configuration - Deploy Setting

13. Go back and search again the term `lastname` to find the modified setting. The widget has a gray border on the left to indicate, this setting is modified.
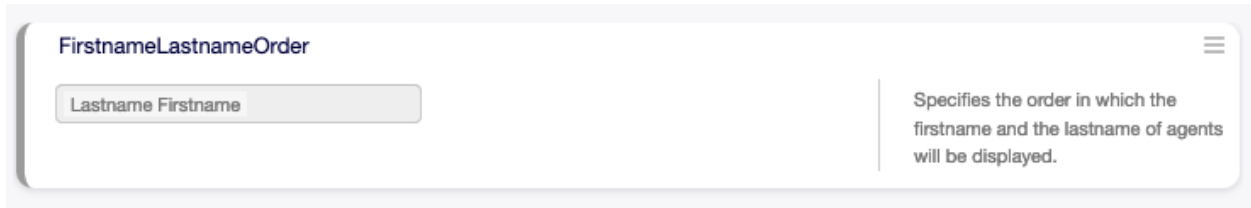

Fig. 1.298: System Configuration - Setting Deployed

14. To reset the setting, click on the header of the widget to see the options. Then click on the Reset setting button.
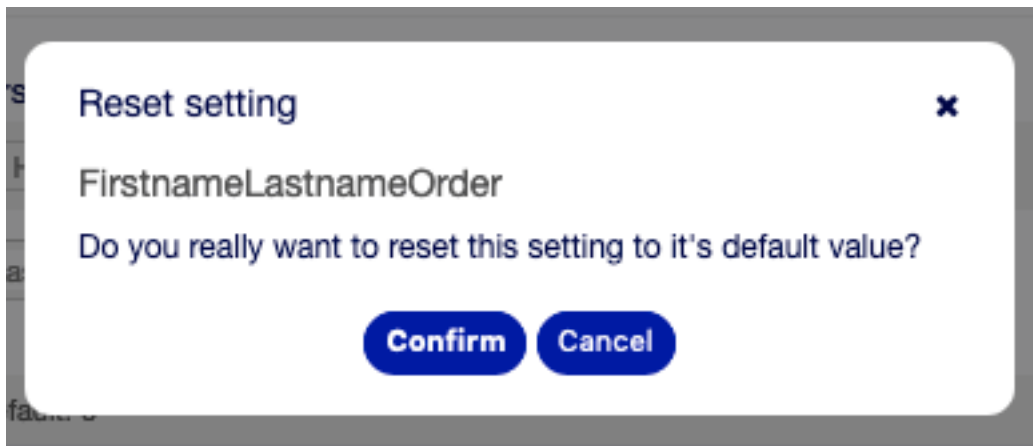

Fig. 1.299: System Configuration - Reset Setting

15. Click on the Confirm button.

16. Deploy the settings.

Using The Navigation Tree

Each configuration setting is classified by a category and a navigation group. Navigation groups are individual elements in the main navigation tree. By selecting one of these navigation entries, all settings assigned to the selected group will be shown. As long as no extensions are installed, the category selection is automatically hidden, but as soon as a package is installed which brings its own configuration settings (such as ITSM modules or Survey), the category selection will be revealed. Selecting a category makes the main navigation tree show only the navigation groups belonging to the selected category.

To expand an element, click on the arrow next to it. The number between the parentheses indicates how many settings belongs to this element. If an element has no number, this element is only a wrapper category. It doesn't have settings, it has only sub-categories.

Using the navigation tree results the same as search for a setting. If you would like to see for a setting to which group belongs to, expand it by clicking on the header of the widget.

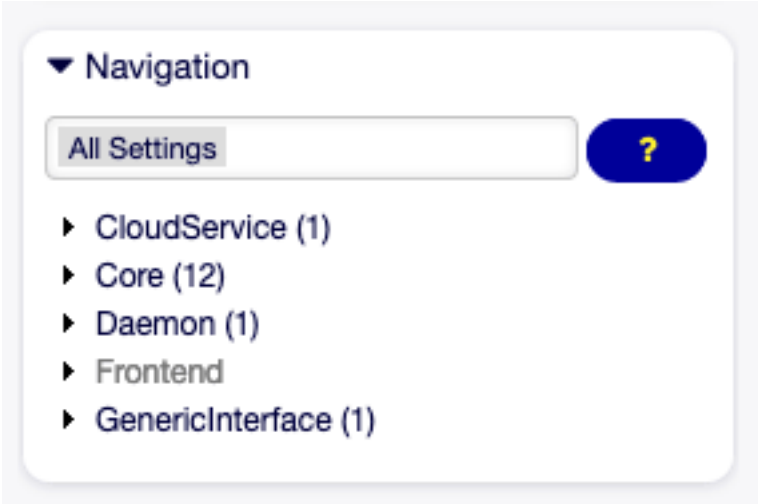For example `FirstnameLastnameOrder` can be found in Frontend → Base.

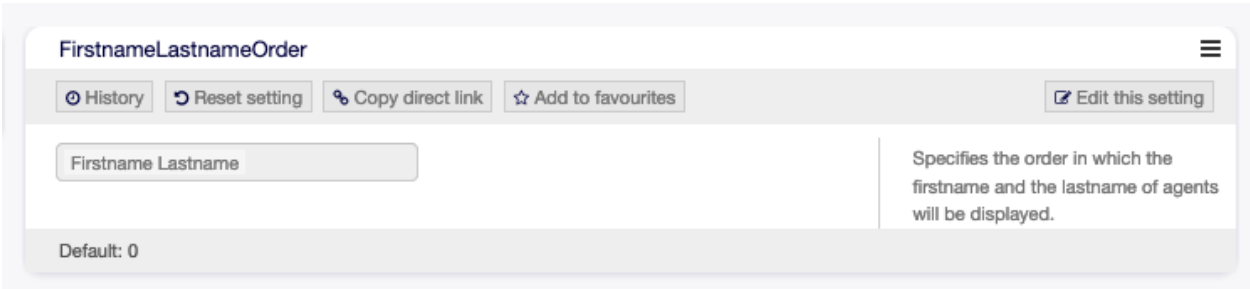Fig. 1.300: System Configuration Navigation Tree

Fig. 1.301: System Configuration - Setting Expanded

Import And Export System Configurations

Click on the Import & Export button in the left sidebar to access the import-export screen.
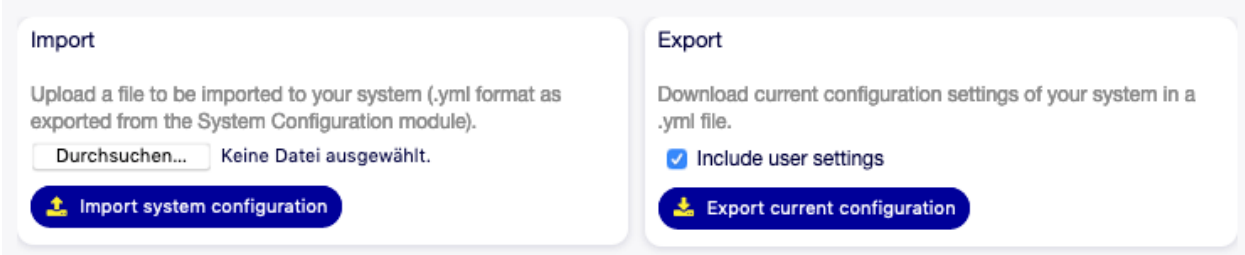

Fig. 1.302: System Configuration - Import and Export

To export the system configurations:

1. Click on the Export current configuration button in the Export widget.
2. Save the `Export_Current_System_Configuration.yml` file to your local file system.
3. Rename the file to a more descriptive name.

To import the system configurations:

1. Click on the Browse… button in the Import widget.
2. Select a previously exported `.yml` file.
3. Click on the Import system configuration button.

1.7.10 System Log

Professional systems log their activities in one or more log files to help administrators when troubleshooting or to get an overview of what is going on in their system on a detailed level.

These logs are usually not available to application administrators without a certain level of permissions, and skills on the operating system.

OTOBO allows application administrators to access the system log comfortably by using the graphical interface without the need to have access to the server's command shell. The administrator can decide which level of logging is needed, to make sure that the log files are not unnecessarily filled.

Use this screen to view log entries of OTOBO. The log overview screen is available in the System Log module of the Administration group.

Each line in the log contains a timestamp, the log priority, the system component and the log entry itself.

Note: If several log entries are displayed in the log, use the filter box to find a particular log entry by just typing the name to filter.

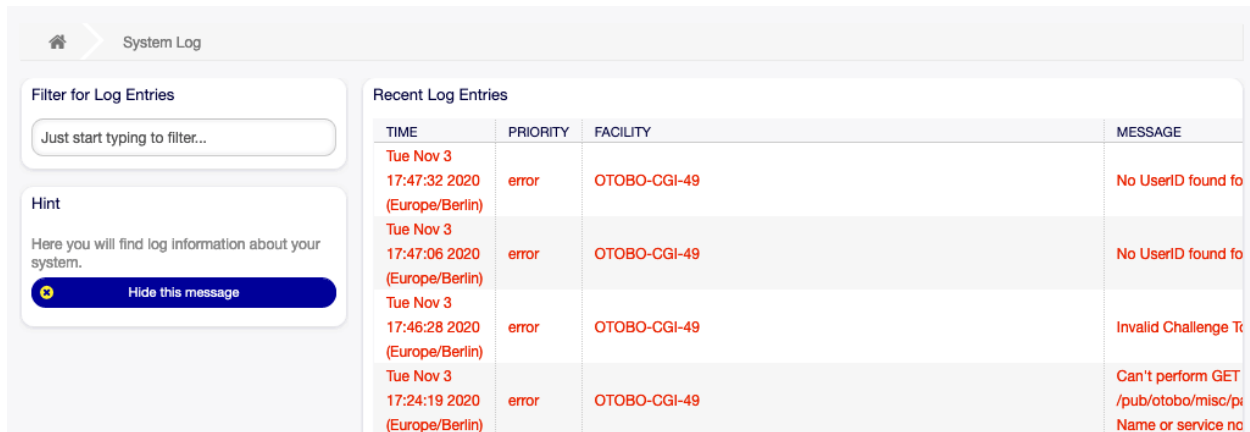### System Log Configuration Options

`Core` → `Log`

Fig. 1.303: System Log Screen

1.7.11 System Maintenance

System maintenance is a crucial part of the work done by all system administrators. Some maintenance tasks require to put the operation on hold. In such cases, pro-active information to users is essential to raise their awareness and understanding.

OTOBO supports this with the System Maintenance module, which allows administrators (agents who have access to the group "admin") to schedule maintenance windows in advance and inform users with login messages and notifications about the planned maintenance. Also, during a scheduled maintenance window, only administrators are allowed to log into the system.

Use this screen to schedule system maintenance for the system. The system maintenance management screen is available in the System Maintenance module of the Administration group.
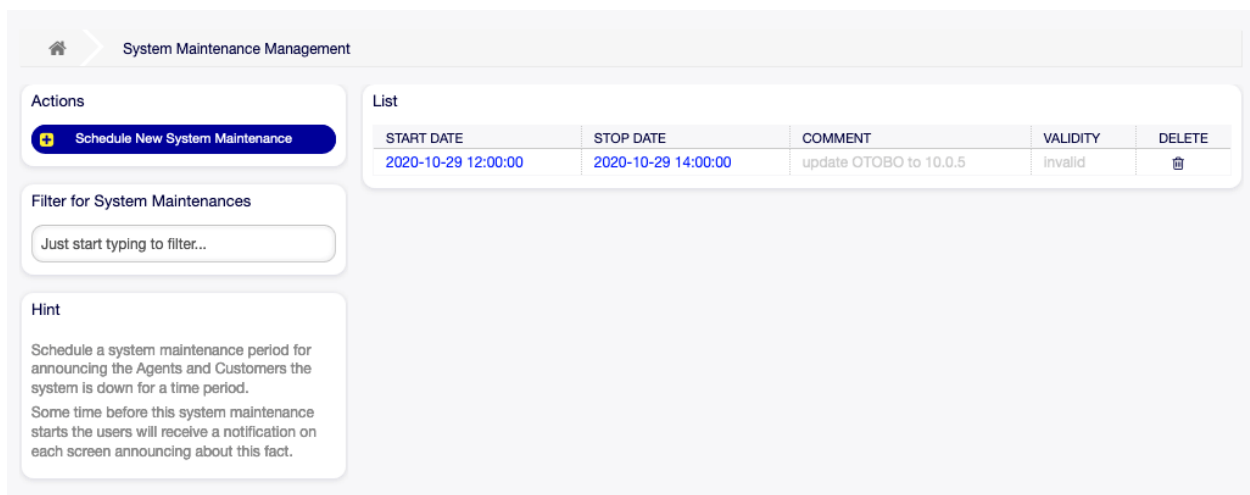

Fig. 1.304: System Maintenance Management Screen

Manage System Maintenance

To schedule a system maintenance:

1. Click on the Add New System Maintenance button in the left sidebar.

2. Fill in the required fields.

3. Click on the Save button.

Fig. 1.305: Schedule New System Maintenance Screen

To edit a system maintenance:

1. Click on a system maintenance entry in the list of system maintenance entries or you are already redirected here from Add New System Maintenance screen.

2. Modify the fields.

3. Click on the Save or Save and finish button.

To delete a system maintenance:

1. Click on the trash icon in the last column of the overview table.

2. Click on the Confirm button.

Note: If several system maintenance entries are added to the system, use the filter box to find a particular system maintenance by just typing the name to filter.

System Maintenance Settings

Start date Select a date from when the system maintenance is scheduled.

Stop date Select a date when the system maintenance is planned to finish.

Comment * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The comment will be displayed in the overview table.

Fig. 1.306: Edit System Maintenance Information Screen

Fig. 1.307: Delete System Maintenance Screen

Login message If this text is given, it will be displayed in the login screen, if the next option is checked.

> **See also:**
>
> There are some default system maintenance message set in the system configuration. For more information see the following system configuration settings:
>
> - `SystemMaintenance::IsActiveDefaultLoginErrorMessage`
> - `SystemMaintenance::IsActiveDefaultLoginMessage`
> - `SystemMaintenance::IsActiveDefaultNotification`

Show login message If checked, the login message is displayed in the login screen of OTOBO.

Notify message If this text is given, a notification will be displayed in the agent interface before the start of the system maintenance.

> **See also:**
>
> Set the minutes a notification is shown for notice about upcoming system maintenance period in system configuration setting:
>
> - `SystemMaintenance::TimeNotifyUpcomingMaintenance`

Validity * Set the validity of this resource. Each resource can be used in OTOBO only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

Manage Sessions This widget gives an overview about the currently logged in users. It is possible to kill each sessions by clicking on the Kill this session links. You can also kill all sessions except yours by clicking on the Kill all Sessions, except for your own button.

Fig. 1.308: Edit System Maintenance - Manage Sessions

1.7.12 Translation

How-Tos

How-Tos on configuring your OTOBO.

Glossary

Activity   Part of the process management.

Agent   An agent is a user which services customer users. The person working for your organization.

Back end   A configuration file, module, script or other functional item within OTOBO which is not reachable via the browser.

Calendar   Defines working hours, time-zone, and holidays to be applied when using escalation times.

Customer   Physical organizations to which customer users are assigned. These provide the function to group users together which belong to a single organization.

Customer User   Physical persons, and their contact data, which belong to a customer. These are the people raising tickets.

Dynamic Field   A field used in tickets and articles to extend the data which can be saved in a ticket or article. They are saved from values in the database or accessed from outside resources.

Escalation Times   The defined amount of working time before a ticket breaches an agreement.

Front end   The graphical interface to OTOBO as viewed in a browser.

Group   This is a resource within OTOBO. An agent or a customer can be assigned permissions to this resource as needed. Additionally, they can be used for access control, processes and web services.

Invoker   An invoker is a special Perl module which allows OTOBO to provide information to a remote system via REST or HTTP. Invokers must be developed in Perl by a back end developer.

Mapping   A mapping allows OTOBO to provide inbound and outbound translation of data allowing us to offer a specific data construction to rebuild incoming structures to meet our needs.

Operation   Similar to a web hook, a provider operation offers a specific sub-set of OTOBO Perl API functionality to external systems.

Process   Part of the process management. A process defines the workflow for a task and consists of activities, activity dialogs, transitions and transition actions.

Process Management   Part of OTOBO to support definition of workflows for recurring tasks.

Process Path   Part of the process management. A process path describes the connections between the parts of a process.

Resource   A resource is any object which has effective permissions. This can be, but is not limited to, a ticket, dashboard widget, statistic, or module.

Screen   A graphical interface to OTOBO viewed in a browser (see also Front end).

Service   A means of delivering added value to customers by making it easier for customers or helping them to achieve their desired results without them having to personally bear the responsibility for specific costs and risks.

Service Level Agreement   A service level agreement (SLA) is a contract between a service provider (either internal or external) and the end user that defines the level of service expected from the service provider. SLAs are output-based in that their purpose is specifically to define what the customer will receive.

Service Request   (see also Ticket).

Transition   Part of the process management.

Transition Action   Part of the process management.

SLA   (see also Service Level Agreement).

Ticket   A ticket is a collection of all communications with a customer during the course of a service request. A ticket contains articles, which are the communication received from or sent to customers, agents, external systems, etc. Tickets belong to a customer user, are assigned to agents and reside in queues.

Transport   A transport is the method chosen for communication. OTOBO supports REST and SOAP.

Activity Dialog   Part of the process management.

Working Time   A definition of minutes passed during the defined working hours of any calendar.

Documentation History

1.) 2019 - OTRS Administration Manual - OTRS AG (https://otrs.com)

2.) 2020 - OTOBO Administration Manual - Rother OSS GmbH (https://otobo.de)

Published by: Rother OSS GmbH, (https://otobo.de), Oberwalting 31, 94339 Leiblfing, Germany. Authors: OTRS AG (original version), Rother OSS GmbH (https://otobo.de)

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license can be found on the GNU website.

Index