

# Modules

- Introduction
- System Configuration

# Introduction

The chapter focuses strictly on the development section. For administrative information check out an [administration tutorial](#).

## Main assumptions

The customer panel comes with an easy way of developing and deploying new modules to the panel. It provides you an out-of-the-box configuration with a few steps needed for proper running.

# System Configuration

## XML Configuration File

Configuration determines the main aspects of the module. Each one regardless of future different configurations has the same core model:

```
<?xml version="1.0" encoding="UTF-8" ?>
<otrs_config version="2.0" init="Application">
  <Setting Name="OTRSFrontendModule::<NAVIGATION_NAME>" Required="0" Valid="0">
    <Description Translatable="1">Module random description</Description>
    <Navigation>CustomerFrontend</Navigation>
    <Value>
      <Hash>
        <Item Key="id">LatestActivity</Item>
        <Item Key="type">[ module/redirect/hidden] </Item>
        <Item Key="text">Latest Activity</Item>
        <Item Key="to">/module/LatestActivity</Item>
        <Item Key="icon">timeline</Item>
        <Item Key="priority">110</Item>
        <!-- OPTIONAL -->
        <Item Key="mode">[ newWindow/tab] </Item>
        <Item ValueType="Textarea" Key="widgets">
          [
            {
              "props": {
            },
              "id": "5",
              "widgetFile": "WIDGET_Timeline.js",
              "compiled": false,
              "view" : "Dashboard",
              "slot" : "slot3",
              "name": "Latest Activity"
            }
          ]
        </Item>
      </Hash>
    </Value>
  </Setting>
</otrs_config>
```

```
</Hash>
</Value>
</Setting>
</otrs_config>
```

`<Hash>` specifies main configurations settings:

- **`<id>`**  
An **identification key** for your module. Should be unique for every module.
- **`<type>`**  
Determines the type of the module. For further information check out [the administration guide for modules](#).
- **`<text>`**  
Display label for frontend panel.
- **`<to>`**  
Specify the URL path to the module. If redirect was chosen as a type this field specifies redirection link.
- **`<icon>`**  
Specify the name of the icon to display. Available icons can be found in this [icon library](#).
- **`<priority>`**  
The higher the priority the higher the module is on the menu.
- **`<widgets>`**  
Set an array of **widget configurations** that should be placed in the module. More on widget configuration [here](#).
- **[Optional] `<mode>`**  
Active if type set to **redirect**. Specifies the way in which the system will display the redirection page.  
Available options are:
  - *newWindow*
  - *tab*

## Available modules

Another step is to include a module name in the `CustomerOTRSFrontendModule::AvailableModules` in system configuration.

It's important to pass the module's **id** as a configuration key.

## Permissions

The last step is to set visibility of the module for predefined CustomerUser groups in the `CustomerOTRSFrontendModule::Permissions`

The screenshot shows a configuration window titled "CustomerOTRSFrontendModule::Permissions" with a sub-header "OTRSFrontend.xml". The window is divided into two main sections. The left section contains a list of modules with their respective user group permissions. The right section is titled "Modules available to customers." and contains a checkbox and a close button.

Module	User Group	Permission
CUWCommitments	users	-
CUWLedger	users	-
CUWReceivables	users	-
CustomerDashboard	users	-
CustomerForm	users	-
CustomerUserSettings	users	-
ITDepartment	users	-
LatestActivity	users	-
NewTicket	users	-
Tickets	users	-
		✓

The right section, "Modules available to customers.", has a checkbox and a close button. A yellow tooltip "Add new entry" is visible near the "CustomerUserSettings" module.

If your module is not visible in the menu try to initiate the configuration rebuild from within

the OTRS console.

1. Login on your server using SSH
2. Change user to otrs
3. Run config rebuild command -> `/opt/otrs/bin/otrs.Console.pl Maint::Config::Rebuild`

After completion of these three steps, your module is ready to go!