XML Configuration

Widget's development process consists of a few simple steps to let your widget start work for you.

XML configuration file:

The core of the whole widget. This one specifies visibility for specific groups, determines in which module should widget be located and how should it be called in the panel. The syntax and layout configuration is similar to every XML OTRS configuration.

There are two ways to configure widgets in the module:

- **Specifying the widgets array directly in the module's configuration file:**

```
   <Setting Name="OTRSFrontendModule::LatestActivity" Required="0" Valid="0">
        <Description Translatable="1">Customer user latest activity menu
module.</Description>
        <Navigation>CustomerFrontend</Navigation>
        <Value>
            <Hash>
                <Item Key="id">LatestActivity</Item>
                <Item Key="type">module</Item>
                <Item Key="text">Latest Activity</Item>
                <Item Key="to">/module/LatestActivity</Item>
                <Item Key="icon">timeline</Item>
                <Item Key="priority">110</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>
```

- **Specifying widget as a separate configuration:**

```
<?xml version="1.0" encoding="UTF-8" ?>
<otrs_config version="2.0" init="Application">


    <!--Core configuration containing module settings and additional properties-->
    <!--Setting location's name should be set in "OTRSFrontendModule::XYZ-->
    <Setting Name="OTRSFrontendModule::<MODULE_NAME>Widget#1000" Required="0"
Valid="1">
        <!--Description visible in system configuration-->
        <Description Translatable="1">Additional Ticket module widget for Intalio Customer
Panel.</Description>
        <Navigation>CustomerFrontend</Navigation>
        <Value>
            <Hash>
                <!--JSON configuration file-->
                <Item ValueType="Textarea" Key="schema">
                    {
                        "props": {
                        },
                        "id": "1000",
                        "widgetFile": "WIDGET_XYZ.js",
                        "compiled": false,
                        "view" : "TicketPreview",
                        "slot" : "slot4",
                        "name": "XYZ"
                    }
                </Item>
            </Hash>
        </Value>
```

```
    </Setting>


</otrs_config>
```

Schema

The schema file is a JSON file. It specifies the most important aspects of the whole widget.

```
{
  "props": {
    "predefined_property": true
  },
  "id": "1000",
  "widgetFile": "WIDGET_Contract.js",
  "compiled": false,
  "view" : "TicketPreview",
  "slot" : "slot4",
  "name": "Contracts"
}
```

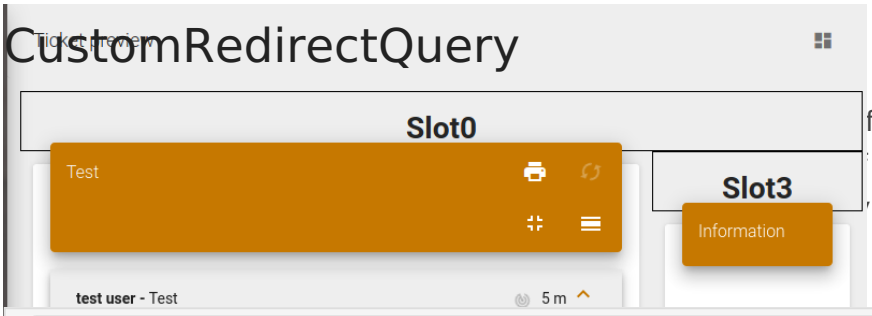
- ***<props>***
  Specify predefined properties that could be referenced in the template file
- ***<id>***
  An **identification key** for your widget. Should be unique for every module**.** Id is responsible for creating unique widget settings for every user.
- ***<widgetFile>***

  Template file's name - see more on the widget's template file.
- ***<view>***
  Specify the primary module where should widget be located. You can choose between multiple options starting, from predefined modules ending up to your own custom modules.

  You can find a more detailed explanation on the administration section,
- ***<compiled>***
  ????????,
- ***<slot>***
  This label specifies a place on the layout where should widget be put in the module.

<Slot> property is required while creating a widget in the TicketPreview module.

You can display slot layout by adding query parameter - "dev=1" to the URL parameters eg. '/customer-panel/ticketPreview/000014/?dev=1'



for the list of tickets, you can define
you add a new parameter in
manage them in the TicketPreview

```
[
   {
      "id" : "8",
      "widgetFile" : "ticket-list",
      "name" : "Tickets: Example queue",
      "view" : "module",
      "compiled" : true,
      "props" : {
         "defaultConfig" : {
            "customRedirectQuery" : {
               "queue" : "ExampleQueryParam"
            }
         }
      }
   }
]
```

In this example clicking on the specific ticket in the ticket list will result with URL like this:

`https://<OTRS_URL>/otrs/route.pl/customer-panel/ticketPreview/<TICKET_NUMBER>?queue=ExampleQueryParam`

Revision #10

Created Sun, Jul 12, 2020 9:30 AM by editor

Updated Tue, Aug 11, 2020 1:37 PM by editor