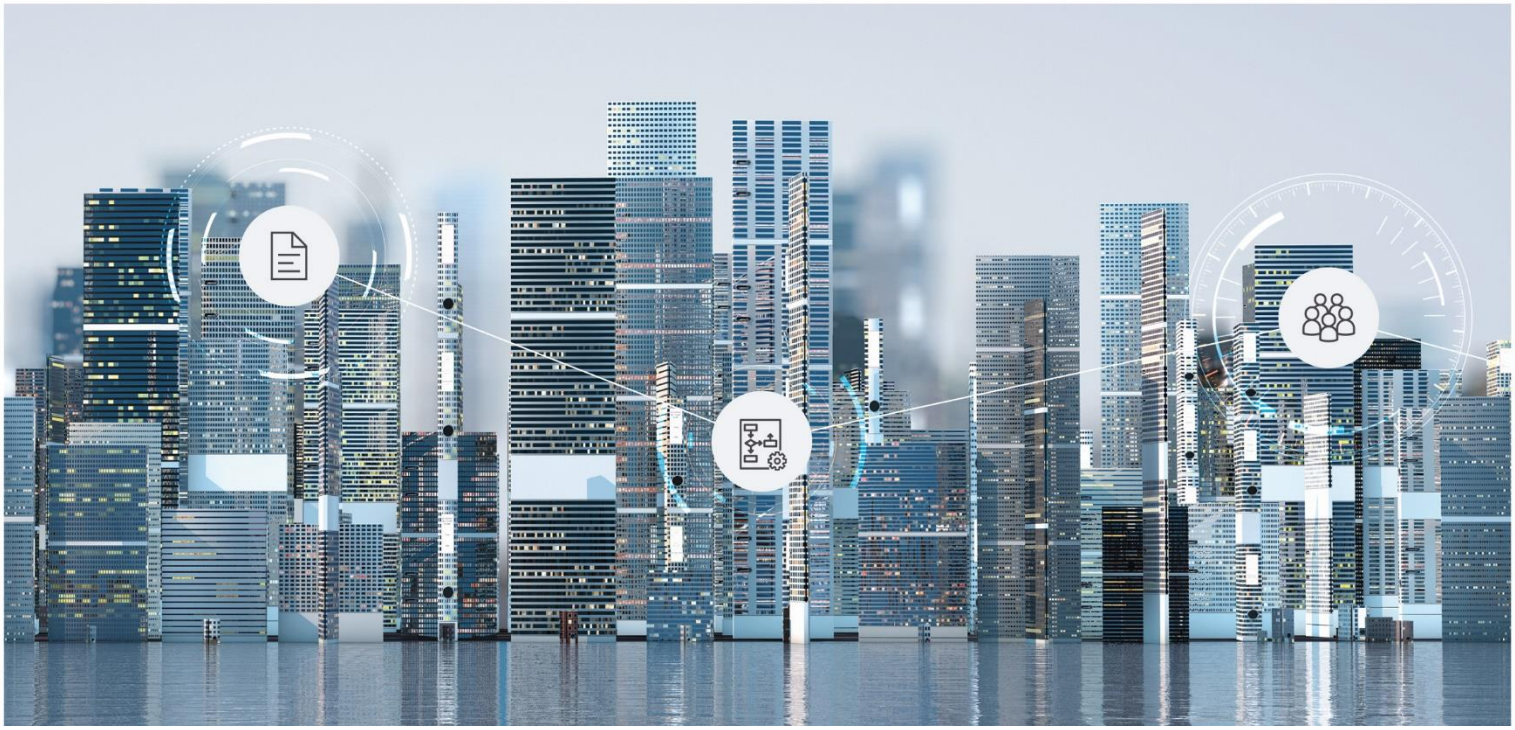
Fabasoft



White Paper

Fabasoft Folio Web Client Interfaces

Fabasoft Folio 2019

Contents

1 Introduction

This document describes the client interfaces and their parameters.

2 Software Requirements

System environment: All information contained in this document implicitly assumes a Microsoft Windows environment or Linux environment.

Supported platforms: For detailed information on supported operating systems and software see the software product information on the Fabasoft distribution media.

3 Control Interfaces

- `Init()`
  Set some member variables. Usually the function `RequestableControlInit` can be used. The most important variables are:
  - `data`
    The object containing the data from server.
  - `id`
    The ID of the control.
- `portlet`
  See chapter 4 “Portlet Object”.
- `Render(output)`
  Writes the HTML of the control into the output object and potentially adds events to the portlet object (see chapter 4.2 “Events”).
- `GetFscArgs()`
  Defines the arguments e.g. needed to show a context menu.

Example:

```
function ajaxMyControl()
{
  this.Init = RequestableControlInit;

  this.Render = function MyControlRender(output)
  {
    var portletid = this.portlet.GetId();
    output.Push("<div id=\"MyControlContainer"+this.id+portletid+
      "\" class=\"FscMyControl\">...</div>");
  };

  this.GetFscArgs = function MyControlGetFscArgs()
  {
    if (this.fscargs === undefined) {
      var container = window.document.getElementById(this.id+
        this.portlet.GetId());
      this.fscargs = new VAPPArgs(null, ";"+this.portlet.GetObject()+";;"+
        this.data.GetAttrdef()+";"+this.data.GetFscargs(), this, container);
    }
    return this.fscargs;
  };
}
```

4 Portlet Object

Within a control object the portlet object is stored in a member variable called `this.portlet`.

4.1 General information

The portlet object contains some information about the current application.

- `this.portlet.GetId()`
  Returns the ID of the portlet object. Each ID in the HTML should end with the portlet ID to ensure that it is unique on the page.
- `this.portlet.GetObject()`
  Returns the COO address of the current object.
- `this.portlet.GetObjClass()`
  Returns the COO address of the object class of the current object.
- `this.portlet.IsSingleControl()`
  Returns `true` if the portlet contains only one control.
- `this.portlet.GetView()`
  If the portlet contains only one control, this function returns the COO address of the attribute.

4.2 Events

The following events are managed by the portlet object:

- `FSCEVT.LOAD`
  Is triggered after the HTML of the portlet is completely rendered.
- `FSCEVT.LAYOUT`
  Is triggered if the size of the portlet has changed, e.g. by resizing the web browser.
- `FSCEVT.CLEANUP`
  Is triggered before handling the response of a request.
- `FSCEVT.FINALCLEANUP`
  Is triggered before removing the portlet, e.g. when closing an overlay dialog.

To add an event you have to call `this.portlet.AddLocalEvent(ident, callbackfunction)`. `this` of the callback function is the portlet object.

Example:

```
var id = this.id;
this.portlet.AddLocalEvent(FSCEVT.LOAD, function() {
  var elem =
    window.document.getElementById("MyControlContainer"+id+this.GetId());
  elem.style.backgroundColor = "blue";
});
```

4.3 Layout

If your control is the only one in the portlet and you want it to use all available space and resize automatically you can do this by calling the following code in the render function.

```
if (this.portlet.IsSingleControl()) {
  this.portlet.SetResizeArea("MyControlContainer"+this.id);
}
```

`this.portlet.SetResizeArea` expects the ID (excluding the portlet ID) of the HTML element which should be resized as parameter.

```
<div id="MyControlContainer0_01234" class="FscMyControl">
  <!-- control html -->
</div>
```

The CSS class of the resize container could look like this:

```
DIV.FscMyControl
{
  height:100%;
  overflow:auto;
}
```