Visualize.Js

Introduction

Visualize.js is a javascript library that works as an extension to JasperReports Server. It allows you to:

* search the repository
* embed and interact with reports on any page
* load and visualize data from AdhocViews on any page (currently in development)

Getting Started

To get started with visualize.js you will need to do the following:

1. Include the javascript file on your webpage.
2. Initialize the framework
3. Including the Javascript File

You only need to include 1 file on your webpage. This file comes with your JasperReports Server. To include the javascript file add the following line of code to your HTML:

<script type="text/javascript" src="http://www.example.com/jasperserver-pro/bif/visualize.js"></script>

Make sure to replace [www.example.com/jasperserver-pro](http://www.example.com/jasperserver-pro) with your actual domain and location of your JasperReports Server instance.

1. Initialize the framework

After including the javascript file you need to initialize the framework by calling BIF.init().

BIF.init() takes 3 parameters:

* + A configuration object.
  + A callback function called when login succeeds.
  + A callback function called when login fails.

The configuration object requires the following properties to be set:

* + domain: (string) set this property to the URL of your JasperReports Server instance, i.e. <http://www.example.com/jasperserver-pro>
  + username: (string) JasperReports Server  username.
  + password: (string) JasperReports Server  password.
  + mods: (array of strings) list of modules you want to load. Available modules are listed in the next section.

An optional parameter keepAlive can be set to false if you would like JasperReports Server to close the session after a certain amount of time has passed without activity.

BIF.init({

domain: 'http://www.example.com/jasperserver-pro',

username: 'superuser',

password: 'superuser',

mods: ['repository']

}, **function**(Repo) {

});

Security Hooks

The configuration object used for initialization can take 2 optional parameters commonly related to security:

* beforeInit: a function to be called first in the init sequence. The function is passed a deferred object which must be resolved for the init sequence to resume. A use case for this option would be security.
* rpcExtraUrlParameter: (string) URL parameter in the form of "parm=value". If provided this string will be appended to the URLs used to open a channel to JRS. A use case for this option would be security.

BIF.init({

domain: 'http://www.example.com/jasperserver-pro',

username: 'superuser',

password: 'superuser',

beforeInit: **function**(deferredObject) {

*/\* do something before BIF initialization sequence starts... \*/*

deferredObject.resolve();

},

rpcExtraUrlParameter: 'token=123',

mods: ['repository']

}, **function**(Repo) {

});

Modules

The following modules are currently available:

* [repository](http://community.jaspersoft.com/wiki/visualizejs-repository)
* [reports](http://community.jaspersoft.com/wiki/visualizejs-reports)

BIF.init({

domain: 'http://www.example.com/jasperserver-pro',

username: 'superuser',

password: 'superuser',

mods: ['repository', 'reports']

}, **function**(Repo, Reports) {

});