Documentation of jQuery-Widget for structural information display

1 Requirements

1.1 Purpose

1.2 Functional Requirements

1. A protein is selected by providing an Id.
2. All available PDB files for an Id can be selected in an Dropdown-Box.
3. Each PDB file is shown in the Jmol applet.
4. A region or single positions can be selected.
5. By selecting a region or single positions the available PDB files are filtered, so that only the ones that include this region or positions are selectable.
6. Additional information for the protein and the selected region is displayed.
7. A warning message is displayed if there is no structural information available for a protein or a selected region/position.

1.3 Libraries

- jQuery (http://docs.jquery.com/Downloading_jQuery#Download_jQuery)
- jQuery.ui (http://jqueryui.com/download)
- Jmol (java + javascript) (http://jmol.sourceforge.net/download/)

2 Design and Implementation

2.1 Integration

To integrate the widget in a .html page the following steps have to be followed:
- Include the file "ui.Jmol.js" in your .html file.
- Include the file "Jmol.js", the Jmol Javascript library. This file has to be in the same folder as all .jar and other files from the Jmol Javascript project (You don't have to include the other files) . (see "resources/jmol-12.0.48")
- Include the following javascript-libraries:
  - jquery-1.4.3.min.js
  - jquery-ui-1.8.10.custom.min.js
- Create a proxy (e.g. like the proxy.php in "resources/proxy")
- Create a container (e.g. a <div> for the the widget).
- Instantiate the widget like this:

```javascript
<script type="text/javascript">
  $(document).ready(function() {
    var jmolPlugin = $("#myDiv").Jmol(
     {width: 200,
      height: 1000,
      jmolFolder: 'resources/jmol-12.0.48',
      proxyUrl: 'resources/proxy/proxy.php'});
  });
</script>
```

2.1.1 Options

There are the following options to set, the displayed values are the default values:

```javascript
options: {
 width: 200,
 height: 1000,
 proxyUrl: '',
 jmolFolder: '',
 warningImageUrl: '',
 loadingImageUrl: ''
},
```

Explanation

width:	the width of the widget (Jmol applet + control elements)
height:	the height of the widget
proxyUrl:	a url for a proxy that can be used in Javascript Ajax requests
jmolFolder:	the folder where Jmol.js and all other Jmol files can be found
warningImageUrl:	if there is an error while loading a pdb file a warning will be displayed; set this property and the given image will be shown at the beginning of the text
loadingImageUrl:	if this property is set the given image will be shown while loading data

2.2 Public Methods

There are five public methods to interact with the widget. You can call a public method of a widget like this:

```
jmolPlugin.Jmol('functionName', parameter1, parameter2);
```

```
getSelectedRegion():
```
Returns information about the currently selected PDB file.
The same parameters and conditions as with the triggered event apply (see 2.4 Events).

```
reset():
```
Resets all selected regions or positions and removes the displayed PDB file from the applet.

```
selectProtein({interactorId: '', interactorName: ''}):
```
Provides all PDB files provided for the given `interactorId` for selection in a drop-down box.
The parameter `interactorId` is mandatory, the `interactorName` optional.
If the `interactorName` is provided it is used in warning messages and as additionally displayed information.

```
selectRegion({coordinates: { x: '', x2: ''}}):
```
Filters the currently provided PDB files: Only PDB files containing a part of the requested region are selectable. x and $x2$ have to be numbers, where $x2$ is greater than or equal to x. The specified region is highlighted in the displayed PDB file.

```
selectPositions({coordinates:{ positionArray: []}}):
```
Filters the currently provided PDB files: Only PDB files containing a part of the requested region are selectable. `positionArray` has to be an array of numbers. The specified positions are highlighted in the displayed PDB file.

```
unselectRegion():
```
Unhighlights the currently highlighted region and resets all selected regions or positions.

2.3 Managing Data

2.3.1 Data
To get the PDB files associated to the UniprotId given in the call to the public method
`selectProtein` (see 2.2 Public Methods) a request is made to
http://www.rcsb.org/pdb/rest/das/pdb_uniprot_mapping/alignment?query= .
As query the UniprotId is used. The response from the web service is an .XML file with the
following structure:

```
<dasalignment>
  <alignment alignType="PDB_SP">
    <alignObject  dbAccessionId="1AAP" intObjectId="1AAP"
                  objectVersion="unknown" type="STRUCTURE" dbSource="PDB"
                  dbVersion="latest" dbCoordSys="PDBresnum,Protein Structure">
      <alignObjectDetail dbSource="PDB" property="citation">
        X-ray crystal structure of the protease inhibitor domain of
        Alzheimer's amyloid beta-protein precursor.
      </alignObjectDetail>
    </alignObject>
    <alignObject  dbAccessionId="P05067" intObjectId="P05067"
                  objectVersion="unknown" type="PROTEIN" dbSource="UniProt"
                  dbVersion="latest" dbCoordSys="UniProt,Protein Sequence"/>
    <block blockOrder="1">
      <segment intObjectId="1AAP.A" start="1" end="58"/>
      <segment intObjectId="P05067" start="287" end="344"/>
    </block>
  </alignment>
</dasalignment>
```

All the information needed can be found in the "`<block>`" section of the .XML file. These
sections are parsed and the `intObjectId` of the first segment is used as the value in a drop-
down box.

2.3.2 Filtering PDB files
If one of the public methods `selectRegion` or `selectPositions` (see 2.2 Public Methods)
is called the PDB files are filtered to just provide the ones that contain the specified region/
positions. Also all atoms in the structure except the ones specified are made slightly translucent
to highlight the specified region/positions.

2.3.3 Messages and Displayed Information
There are two messages possibly displayed by the widget - one if there are no PDB files for the
given term or if there are no PDB files containing the specified region/positions.
The first one uses the interactorName if specified and otherwise the given interactorId:
```
No structural information available for ''.
No structural information available for this region.
```

If an image is given for the option "`warningImageUrl`" this picture is displayed at the beginning
of a warning message.
If a protein is displayed the given `interactorId` and `interactorName` (if specified) is provided
at the right side of the applet. For the region the requested positions and the actually highlighted
positions are provided, since also PDB files partly covering the requested positions are

displayed.

2.3.4 PDBe Redirect

If an PDB file is deprecated and replaced by another one, there is a redirect on the EBI-page. Since the page itself is given to the Java applet the redirect can not be followed automatically. If this happens the applet displays an error message:

"unrecognized file format for file http://.../1rw6.pdb
 <html><body>You are being redirected.</body></html>"

To catch this error and redirect manually, the method `jmolSetCallback("callbackName", "functionName")` provided by the applet is used to catch this message and load the specified file.
Since the function is called by the applet, it has to be globally available and therefore specified outside the widget (the name of the function is `pdbLoadingError` and can be found at the bottom of the .js file containing the widget).

2.4 Events

If a PDB file is selected an event is triggered. This event can easily be caught with jQuery:

`$(document).bind(eventName, function(event, params){//do something});`

Event Name	Parameter	
pdb_selected		
	Parameter	**Description**
	proteinId	UniprotId associated with the selected PDB file
	start	first position in the PDB file
	end	last position in the PDB file

If there was an error while loading the file, the parameter of the event are `null`.