

Geomajas Codemirror GWT wrapper project

Geomajas Developers and Geosparc

Geomajas Codemirror GWT wrapper project

by Geomajas Developers and Geosparc

3.1.3

Copyright © 2010-2015 Geosparc nv

Table of Contents

1. Introduction	1
2. How to include the project	2
3. Development guidelines	3

Chapter 1. Introduction

The Geomajas Codemirror GWT Project is a stand-alone project under the Geomajas banner. Its goal is to provide an easy way to allow you to integrate the excellent Codemirror Javascript library into your own GWT applications

Chapter 2. How to include the project

The codemirror project can be included as a GWT module. Add dependency in pom.xml:

```
<dependency>
  <groupId>org.geomajas.project</groupId>
  <artifactId>geomajas-project-codemirror-gwt</artifactId>
  <version>latestVersion</version>
</dependency>
```

In your application .gwt.xml file, include the codemirror module:

```
<inherits name='org.geomajas.codemirror.CodeMirror' />
```

As from version 3.13.0, GWT 2.7.0 is used. This requires codemirror script files to be included directly into the index.html page of your application. Add this section to the html page, where gwtModuleReference needs to be replaced by your GWT application module's name (e.g. for example project, gwtModuleReference is org.geomajas.codemirror.Example)

```
<!-- scripts for codemirror START -->
<script type="text/javascript" language="javascript" src="gwtModuleReference/li
<script type="text/javascript" language="javascript" src="gwtModuleReference/mo
<script type="text/javascript" language="javascript" src="gwtModuleReference/mo
<script type="text/javascript" language="javascript" src="gwtModuleReference/mo
<script type="text/javascript" language="javascript" src="gwtModuleReference/ad
<script type="text/javascript" language="javascript" src="gwtModuleReference/ad
<!-- scripts for codemirror END -->
```

Remark: be sure to include end tags as provided. Do not use <script.../>, as some browsers will fail on this.

Chapter 3. Development guidelines

Look at the `ExampleEntryPoint` class in the example project for a basic use of the codemirror panel.

The initial configuration can be set with the `Config` object.

</br>

It behaves like a key-value set, most common keys are provided as static strings in the `Config` class. See the codemirror website for a detailed description of what all these options do.

You can also change these options live once the widget is created by getting the wrapper object from the panel:

```
CodeMirrorPanel panel = new CodeMirrorPanel(); ... CodeMirrorWrapper editor = panel.getEditor(); editor.setOption(Config.MODE, "json");
```

Some convenience setters and getters are provided. The most important one of which will be the `getContent()` which retrieves the text that was typed into the editor.

```
CodeMirrorPanel panel = new CodeMirrorPanel(); ... CodeMirrorWrapper editor = panel.getEditor(); String result = editor.getContent();
```

If you wish to use a custom theme or addon, the best place to add the is your `'Application.gwt.xml'`. Add them like you would any other javascript or css resource.

</br>

Look at the codemirror samples and just add the same js and css resources to your module.

```
<?xml version="1.0" encoding="UTF-8"?>
<module> <inherits name='com.google.gwt.user.User' /> <inherits name="com.google.gwt.i18n.I18N"/> <script src="lib/codemirror.js" />
<script src="mode/xml/xml.js" /> <script src="mode/css/css.js" /> <script src="mode/htmlmixed/htmlmixed.js" />
<script src="addon/edit/closetag.js" /> <script src="addon/fold/collapserange.js" /> <stylesheet src="lib/codemirror.css" />
<stylesheet src="codemirrorCustom.css" /> </module>
```

PS: these are the resources that are already provided, no need to include them again.