

GWT - RICHTEXTAREA WIDGET

http://www.tutorialspoint.com/gwt/gwt_richtextarea_widget.htm

Copyright © tutorialspoint.com

Introduction

The **RichTextArea** widget represents a rich text editor that allows complex styling and formatting. Because some browsers do not support rich text editing, and others support only a limited subset of functionality, there are two formatter interfaces, accessed via `getBasicFormatter` and `getExtendedFormatter`. A browser that does not support rich text editing at all will return null for both of these, and one that supports only the basic functionality will return null for the latter `getExtendedFormatter`.

Class declaration

Following is the declaration for **com.google.gwt.user.client.ui.RichTextArea** class:

```
public class RichTextArea
    extends FocusWidget
    implements HasHTML, HasInitializeHandlers, HasSafeHtml
```

CSS style rules

Following default CSS Style rules will be applied to all the TextBox widget. You can override it as per your requirements.

```
.gwt-RichTextArea {}
```

Class constructors

S.N.	Constructor & Description
1	RichTextArea Creates a new, blank RichTextArea object with no stylesheet.

Class methods

S.N.	Function name & Description
1	HandlerRegistration addInitializeHandler <i>InitializeHandlerhandler</i> Adds an InitializeEvent handler.
2	RichTextArea.BasicFormatter getBasicFormatter Deprecated. use getFormatter instead.
3	RichTextArea.ExtendedFormatter getExtendedFormatter Deprecated. use getFormatter instead.
4	

RichTextArea.Formatter getFormatter

Gets the rich text formatting interface.

- 5 **java.lang.String getHTML**
Gets this object's contents as HTML.
- 6 **java.lang.String getText**
Gets this object's text.
- 7 **boolean isEnabled**
Gets whether this widget is enabled.
- 8 **protected void onAttach**
This method is called when a widget is attached to the browser's document.
- 9 **protected void onDetach**
This method is called when a widget is detached from the browser's document.
- 10 **void setEnabledbooleanenabled**
Sets whether this widget is enabled.
- 11 **void setFocusbooleanfocused**
Explicitly focus/unfocus this widget.
- 12 **void setHTMLjava.lang.StringsafeHtml**
Sets this object's contents via safe HTML.
- 13 **void setHTMLjava.lang.Stringhtml**
Sets this object's contents via HTML.
- 14 **void setTextjava.lang.Stringtext**
Sets this object's text.

Methods inherited

This class inherits methods from the following classes:

- com.google.gwt.user.client.ui.UIObject
- com.google.gwt.user.client.ui.Widget
- com.google.gwt.user.client.ui.FocusWidget
- java.lang.Object

RichTextBox Widget Example

This example will take you through simple steps to show usage of a RichTextBox Widget in GWT. Follow the following steps to update the GWT application we created in *GWT - Create Application* chapter:

Step	Description
1	Create a project with a name <i>HelloWorld</i> under a package <i>com.tutorialspoint</i> as explained in the <i>GWT - Create Application</i> chapter.
2	Modify <i>HelloWorld.gwt.xml</i> , <i>HelloWorld.css</i> , <i>HelloWorld.html</i> and <i>HelloWorld.java</i> as explained below. Keep rest of the files unchanged.
3	Compile and run the application to verify the result of the implemented logic.

Following is the content of the modified module descriptor **src/com.tutorialspoint/HelloWorld.gwt.xml**.

```
<?xml version="1.0" encoding="UTF-8"?>
<module rename-to='helloworld'>
  <!-- Inherit the core Web Toolkit stuff. -->
  <inherits name='com.google.gwt.user.User' />

  <!-- Inherit the default GWT style sheet. -->
  <inherits name='com.google.gwt.user.theme.clean.Clean' />

  <!-- Specify the app entry point class. -->
  <entry-point class='com.tutorialspoint.client.HelloWorld' />

  <!-- Specify the paths for translatable code -->
  <source path='client' />
  <source path='shared' />

</module>
```

Following is the content of the modified Style Sheet file **war/HelloWorld.css**.

```
body{
  text-align: center;
  font-family: verdana, sans-serif;
}
h1{
  font-size: 2em;
  font-weight: bold;
  color: #777777;
  margin: 40px 0px 70px;
  text-align: center;
}
.gwt-RichTextArea {
  padding:10px;
}
```

Following is the content of the modified HTML host file **war/HelloWorld.html**.

```
<html>
```

```

<head>
<title>Hello World</title>
  <link rel="stylesheet" href="HelloWorld.css"/>
  <script language="javascript" src="helloworld/helloworld.nocache.js">
  </script>
</head>
<body>

<h1>RichTextArea Widget Demonstration</h1>
<div ></div>

</body>
</html>

```

Let us have following content of Java file **src/com.tutorialspoint/HelloWorld.java** which will demonstrate use of TextBox widget.

```

package com.tutorialspoint.client;

import com.google.gwt.core.client.EntryPoint;
import com.google.gwt.user.client.ui.RichTextArea;
import com.google.gwt.user.client.ui.RootPanel;
import com.google.gwt.user.client.ui.VerticalPanel;

public class HelloWorld implements EntryPoint {
    public void onModuleLoad() {
        //create RichTextArea elements
        RichTextArea richTextArea = new RichTextArea();

        richTextArea.setHeight("200");
        richTextArea.setWidth("200");

        //add text to text area
        richTextArea.setHTML("<b>Hello World!</b> <br/> <br/>" +
            "<i>Be Happy!</i> <br/> <br/> <u>Stay Cool!</u>");

        // Add text boxes to the root panel.
        VerticalPanel panel = new VerticalPanel();
        panel.add(richTextArea);

        RootPanel.get("gwtContainer").add(panel);
    }
}

```

Once you are ready with all the changes done, let us compile and run the application in development mode as we did in [GWT - Create Application](#) chapter. If everything is fine with your application, this will produce following result:



HELLO WORLD!

Be Happy!

Stay Cool!

Loading [Mathjax]/jax/output/HTML-CSS/jax.js