ExtJS form field - a text area with integrated TinyMCE WYSIWYG Editor

Ext.ux.form.TinyMCETextArea

ExtJS form field - a text area with integrated TinyMCE WYSIWYG editor.

Version: 5.0 Release date: 22.09.2014 ExtJS Version: 5.0.0 TinyMCE Version: 4.1.5 License: LGPL v2.1 or later, Sencha License Author: Oleg Schildt

Implementation details

This component is a standard text area field, which is extended with the TinyMCE editor.

The TinyMCE editor is applied to the internal text area element of the Ext.form.field.TextArea.

Following tasks had to be solved:

- Time for starting of the editor initialization.
- Initialization in an initially invisible tab.
- Correct resizing of the editor by resizing of the underlying text in any layout.
- Enabling and disabling of the editor. Keeping of the cursor position by switching to the HTML text modus.
- Focusing of the editor field.
- Marking invalid.
- Activation and deactivation of the editor.

Initialization and Sizing

To correctly initialize the editor, we need the information of the actual size of the underlying text area. The size is not known at the time of the afterRender event. I add the 'resize' event handler only after rendering and start the initialization, when the first 'resize' event is fired. The initialization is done only once, in subsequent firing of the 'resize' event, only the adjustment of the editor size is done.

I retrieve the height of the tool bar and that of the status bar, then, I calculate the correct height for the contents iframe and set it.

Dynamic loading of the tinymce.js

If you do not include the tinymce.js code statically, but it is loaded dynamically only if it is required, you have to do some additional actions. You have to place somewhere in your code this initialization statement

```
window.tinyMCEPreInit = {
    suffix : '',
    base : '/js/tinymce' // your path to TinyMCE
};
```

ExtJS form field - a text area with integrated TinyMCE WYSIWYG Editor

This is necessary, because the TinyMCE is initialized upon "document ready" event, and in the case of dynamic loading, this event already occurred before and is not fired upon loading of the TinyMCE script. The author of the TinyMCE have provided this option – tinyMCEPreInit – for such cases.

Setting and Getting Value

You can set and get value by using usual methods getValue() and setValue().

The editor does pre-formatting of the entered HTML text. It is important, for instance, to let the editor convert absolute or relative urls, make the entered HTML valid etc.

If you enter the HTML text manually in the HTML modus, the WYSIWYG editor does the pre-formatting also.

Enabling and Disabling

The TinyMCE does not support disabling and enabling. I solved it through usage of setting ReadOnly.

Marking invalid

When the editor is active, the standard marking does not work. I had to add an invalid mark manually.

For marking invalid, I add the class "tinymce-error-field" to the most outer container.

Following rule should be written somewhere in a CSS file or style:

```
.tinymce-error-field
{
   border: 1px dashed red !important;
}
```

It should be defined with a stronger rule than just

```
.tinymce-error-field {}
```

otherwise, other rules may override this.

Class for removing the border of the textarea

In the ExtJS 5.x, the border of the fields is designed not in the fields themselves, but in the wrappers. Thus, when the WYSIWYG editor is initialized, the border of the wrapper should be removed, because the WYSIWYG editor has own border. To remove border, the class "tinymce-hide-border" is used. Define this class in your CSS file as follows:

```
.tinymce-hide-border
{
   border-width: 0px;
}
```

Changing CSS files for the contents on the fly

This is a very useful feature. The editor allows specifying of the CSS file, which will be applied for the contents. Let's assume a CMS. If the user changes the design template for

ExtJS form field - a text area with integrated TinyMCE WYSIWYG Editor

the page that is opened for editing in the WYSWYG editor, the CSS file should be also changed and applied to the editable contents.

I implemented the method reinitEditor that does it. It is performed through removing the editor and initialization it again with the adjusted options.

Integration into the Sencha Architect

The file TinyMCETextArea.xdc, kindly offered by Gregory, is included in the package of the TinyMCETextArea.

Open the Sencha Architect and in the menu "Edit" select "Import Component". Select the file "TinyMCETextArea.xdc":

File Ed	it View Actions H	lelp		
	Undo	Ctrl+Z	1	4
Net	Redo	Ctrl+Y	t De	ploy I
Tool	Find	Ctrl+F		Refre
Filto	Find Next	F3		111111
TICE	Find Previous	Shift+F3		
Evei	Replace	Ctrl+H		
DEFA	New Component			
Beha	Import Component			
Butt	Delete Component	Shift+Del		
Char	Architect Settings	Ctrl+		
Cont	Project Settings Ctrl+P			
Data	37 10 CUSIC	omevent		
Forms	25 Carlos Static	Function		
	T Char	t Axis	1	

Choose the category of the component and provide the name for the component:

Import to Toolbox			
What name	would you like to give this component in t	he Toolbox?	
Name:	TinyMCETextArea		
Contra a series of	Fauna Fielda	~	

ExtJS form field - a text area with integrated TinyMCE WYSIWYG Editor

The component should have been added:

Forms	25	Zml Writer
Grids	22	▼ Form Fields
Menus	6	Checkbox
Models	22	Checkbox Group
Resources	3	ComboBox
Resources		📰 Date Field
Trees	16	abc Display Field
Views	54	File Upload
CUSTOM	+	Hidden Field
New	0	D HTML Editor
New 2	0	abc Label
HOIT L	~ 4	뒢 Multi Slider
		123 Number Field
		Radio
		Radio Group
		di⊨ Slider
		Text Area
		abc Text Field
		Time Field
	7	📑 Trigger Field
		TinyMCETextArea

Attention!

For some reason, the component is visible in the "Form Fields" branch only over the general view. If you click "Forms", the component is not listed there.

ExtJS form field - a text area with integrated TinyMCE WYSIWYG Editor

Now, we create the project and copy the following files and directories to the project directory:

- tinymce
- TinyMCETextArea.js

Next, add the following JS files to the project resources:



- 1. tinymce\tiny_mce_src.js
- 2. TinyMCETextArea.js

B	Resources	
	Library	
	🔒 tiny_mce_src.js	
	TinyMCETextArea.js	
Config		

Attention!

The order of the file including is important!

ExtJS form field - a text area with integrated TinyMCE WYSIWYG Editor

Put the component to the desired place:

Toolbox - Ext	JS 4.1.)	K 4	Refresh CSS
Filter			
Everything	146	Hidden Field	
DEFAULT		Ab HTML Editor	
Behaviors	8	He Multi Slider	
Buttons	4	123 Number Field	
Charts	21	Radio	
Containers	10	Radio Group	
Data	37	∰ Slider	
Forms	25	Text Area	
Grids	22	Time Field	
Menus	6	📩 TinyMCETextArea	
Models	22	Trigger Field	
Resources	3	▼ Grid	
Trees	16	Cell Editing Plugin	tinymcetextarea
Views	54 🗸	Property Grid	
CUSTOM	+	Row Editing Plugin	

Set the tinyMCEConfig property for the component.

tinyMCEConfig	🛓 📋 Design Code	Project Inspector +		
tinyMCEConfig - object		Application Controllers		
14 {}		Views		
		🔻 🖶 MyViewpo	rt	Q
		Config	ladaata	
		• tinymcetextarea Ext	t.form.field.Te.	xtArea 🕜
		Filter or Quick Valu	e Set	Add
		Property 🛦 (#) lastneight	Value u	
		(#) lastWidth	0	
		🕢 noWysiwyg		
		positionBeforeBlur		
		storedCursorPosi	(none)	
		tinyMCEConfig	{ 🖪 }	
		wysiwygIntialized		
		▼ Ext.AbstractCompo	nent	
		autoRender		
		autoShow		
		border	(none)	
		cls	(none)	

ExtJS form field - a text area with integrated TinyMCE WYSIWYG Editor

Now, we can compile the project and see the result:

() MyTinyDemo	+
B I U ABC ≣ ≣ ≣ Styles ∷ i = ∰ ∰ ∽ (∾ ∽ ∯ Ů ♥ ⊲ — 2 ⊞ ×, x ³ Ω	▼ Paragraph ▼ Ø @ нтт.
Path: p	

ExtJS form field - a text area with integrated TinyMCE WYSIWYG Editor

Reference

Generally, the component derives all properties and methods from the TextArea. Here, only the new and overridden properties and methods are described.

Properties

noWysiwyg

This property forces that the editor is not initially active.

disabled

This property let make the control initially disabled.

readOnly

This property let make the control initially readOnly.

tinyMCEConfig

This property contains the TinyMCE settings which are passed to the editor by the initialization.

Methods

enable([Boolean silent])

Enable the component

Parameters:

silent : Boolean (optional)

Passing true will supress the 'enable' event from being fired.

Defaults to: false

disable([Boolean silent])

Disable the component

Parameters:

silent : Boolean (optional)

Passing true will supress the 'disable' event from being fired.

ExtJS form field - a text area with integrated TinyMCE WYSIWYG Editor

Defaults to: false

setReadOnly([Boolean readOnly])

Set the control ReadOnly or not ReadOnly.

Parameters:

readOnly : Boolean

The desired state of the control.

showEditor()

Activate the editor, WYSIWYG modus.

hideEditor()

Deactivate the editor, HTML text modus.

toggleEditor()

Toggle the modus WYSIWYG versus HTML text.

isEditorHidden()

Return whether the WYSIWYG editor is hidden or not.

removeEditor()

Completely remove the WYSIWYG editor. The control becomes a normal text area.

focus([Boolean selectText], [Boolean/Number delay]) : Ext.Component

Try to focus this component.

Parameters:

selectText : Boolean (optional)

If applicable, true to also select the text in this component

delay : Boolean/Number (optional)

Delay the focus this number of milliseconds (true for 10 milliseconds).

Page 9 of 11

ExtJS form field - a text area with integrated TinyMCE WYSIWYG Editor

Returns:

Ext.Component (this)

storeCurrentSelection()

Store the current cursor position or the current selection for later restoring. This method should be called before calling any dialog for inserting of a placeholder or other text into the current cursor position or instead of the current selection.

restoreCurrentSelection()

Restore the cursor position or the current selection for later text insertion. This method should be called immediately before inserting of a placeholder or other text into the current cursor position or instead of the current selection.

insertText(txt)

Insert the text "txt" at the current cursor position or instead of the current selection.

getValue() : Object

Returns:

The actual HTML text

ExtJS form field - a text area with integrated TinyMCE WYSIWYG Editor

setValue(Object value) : Ext.ux.form.TinyMCETextArea

Sets a data value into the field and runs the change detection and validation. The value is pre-formatted with the WYSIWYG editor.

Parameters:

value : Object

The value to set

Returns:

Ext.ux.form.TinyMCETextArea (this)

reinitEditor([cfg]) : Ext.ux.form.TinyMCETextArea

Reinitialize the editor with the adjusted settings.

Parameters:

cfg : Object (optional)

The configuration options which differ from the initial option. All options do not need be repeated.

Returns:

Ext.ux.form.TinyMCETextArea (this)