

Gadget-veCenter.js

Aptean Ltd
Copyright © 2011-2026.

Contents

1 MediaWiki:Gadget-veCenter.js.....**1**

1 MediaWiki:Gadget-veCenter.js

```
mw.loader.using( [ 'ext.visualEditor.core', 'ext.visualEditor.mwtransclusion' ] ).then(function () {

// ----- (start of ve.ui.CenterAction definition) ----- // This is based on
[lib/ve/src/ui/actions/ve.ui.BlockquoteAction.js] from Extension:VisualEditor.

ve.ui.CenterAction = function VeUiCenterAction() { ve.ui.CenterAction.super.apply( this, arguments ); }; OO.inheritClass(
ve.ui.CenterAction, ve.ui.Action );

ve.ui.CenterAction.static.name = 'center'; ve.ui.CenterAction.static.methods = [ 'wrap', 'unwrap', 'toggle' ];

ve.ui.CenterAction.prototype.isWrapped = function () { var fragment = this.surface.getModel().getFragment(); return
fragment.hasMatchingAncestor( 'center' ); }; ve.ui.CenterAction.prototype.toggle = function () { return this[
this.isWrapped() ? 'unwrap' : 'wrap' ](); }; ve.ui.CenterAction.prototype.wrap = function () { var surfaceModel =
this.surface.getModel(), selection = surfaceModel.getSelection(), fragment = surfaceModel.getFragment( null, true ),
leaves, leavesRange;

if ( !( selection instanceof ve.dm.LinearSelection ) ) { return false; }

leaves = fragment.getSelectedLeafNodes(); leavesRange = new ve.Range( leaves[ 0 ].getRange().start, leaves[
leaves.length - 1 ].getRange().end ); fragment = surfaceModel.getLinearFragment( leavesRange, true );

fragment = fragment.expandLinearSelection( 'siblings' );

while ( fragment.getCoveredNodes().some( function ( nodeInfo ) { return !nodeInfo.node.isAllowedParentNodeType(
'center' ) || nodeInfo.node.isContent(); } ) ) { fragment = fragment.expandLinearSelection( 'parent' ); }

// Wrap everything in a

tag

fragment.wrapAllNodes( { type: 'center' } );

return true; }; ve.ui.CenterAction.prototype.unwrap = function () { var surfaceModel = this.surface.getModel(), selection =
surfaceModel.getSelection(), fragment = surfaceModel.getFragment( null, true ), leaves, leavesRange;

if ( !( selection instanceof ve.dm.LinearSelection ) ) { return false; }

if ( !this.isWrapped() ) { return false; }

leaves = fragment.getSelectedLeafNodes(); leavesRange = new ve.Range( leaves[ 0 ].getRange().start, leaves[
leaves.length - 1 ].getRange().end ); fragment = surfaceModel.getLinearFragment( leavesRange, true );

fragment

// Expand to cover entire

tag

.expandLinearSelection( 'closest', ve.dm.CenterNode ) // Unwrap it .unwrapNodes( 0, 1 );

return true; }; ve.ui.actionFactory.register( ve.ui.CenterAction );

// ----- (end of ve.ui.CenterAction definition) -----

ve.ui.CenterFormatTool = function VeUiCenterFormatTool() { ve.ui.CenterFormatTool.super.apply( this, arguments ); };
OO.inheritClass( ve.ui.CenterFormatTool, ve.ui.FormatTool );

ve.ui.CenterFormatTool.static.name = 'center'; ve.ui.CenterFormatTool.static.group = 'format';
ve.ui.CenterFormatTool.static.title = 'Center'; ve.ui.CenterFormatTool.static.format = { type: 'center' };
ve.ui.CenterFormatTool.static.commandName = 'center'; ve.ui.toolFactory.register( ve.ui.CenterFormatTool );

ve.ui.commandRegistry.register( new ve.ui.Command( 'center', 'center', 'toggle', { supportedSelections: [ 'linear' ] } ) );
```



```
ve.ui.triggerRegistry.register( 'center', { mac: new ve.ui.Trigger('cmd+j'), pc: new ve.ui.Trigger('ctrl+j') } );  
    } );
```

