

# pt\_extbase

## Content

<b>General Information</b>	<b>1</b>
<b>Administration</b>	<b>2</b>
<b>Integration</b>	<b>2</b>
<b>Development</b>	<b>3</b>
ViewHelpers	3
Backend	3
Content	3
RenderPage	3
Form	3
FormToken	3
Format	3
RemoveLineBreaks	3
FileSize	3
StringToLower	3
TimeStamp	4
Javascript	4
Rbac	4
HasAccess	4
Tree	5
Path	5
Misc	5
Comment	5
Explode	5
Captcha	5
CObjectConfig	6
IfValueChanges	6
RequestArguments	6
4. Schnittstellen	7

## General Information

**Administration**

**Integration**

# Development

## ViewHelpers

### ***Backend***

### ***Content***

#### ***RenderPage***

Renders the complete content of a given pageUid.

Arguments:

**pageUid:** The page Uid

### ***Form***

#### ***FormToken***

Just returns a formToken to be used in backend form links

Arguments:

Example:

```
{ptx:be.formToken( ) }
```

Returns:

&formToken=<formTokenHash>

### ***Format***

#### ***RemoveLineBreaks***

Remove line breaks from the string.

Arguments:

**string:** The string to remove the linebreaks from.

Example

```
{ptx:format.removeLineBreaks(string:'bla')}
```

#### ***FileSize***

Formats a given integer as a fileSize.

Arguments:

**labels:** Labels for each degree in the notation "B| KB| MB| GB"

Example

```
<ptx:format.fileSize labels="B| KB| MB| GB">122132</ptx:format.fileSize>
```

#### ***StringToLower***

Formats a string to lower characters.

Arguments:

**string:** Input string

## ***TimeStamp***

Formats an integer timestamp as date

Arguments:

**timestamp:** The timestamp integer  
**format:** The format string for strftime()

## ***Javascript***

### ***Rbac***

#### ***HasAccess***

Checks the access right for a given role and object action.

Arguments:

**object:** Object to check if user has access rights for  
**action:** Action to check if user has access rights for  
**hasAny:** Check if user has access rights for any of these object:action combinations  
**hasAll:** Check if user has access rights for all of these object:action combinations

Examples:

```
<code title="Basic usage">
  <rbac:hasAccess object="rbac_object_name" action="rbac_action_name">
    This is being shown in case user has access to action on object
  </rbac:hasAccess>
</code>
```

Everything inside the <rbac:access> tag is being displayed if the frontend user has access to action on object. If no user is given, the currently logged in fe user will be used.

```
<code title="hasAccess / access / noAccess">
  <rbac:hasAccess object="rbac_object_name" action="rbac_action_name">
    <f:then>
      This is being shown in case the user has access.
    </f:then>
    <f:else>
      This is being displayed in case the user has NO access.
    </f:else>
  </rbac:hasAccess>
</code>
```

Everything inside the "access" tag is displayed if the user has access to action on object. Otherwise, everything inside the "noAccess"-tag is displayed.

```
<code title="inline notation">
  {rbac:hasAccess(object: 'objectName', action: 'actionName'
    then: 'user has access', else: 'access is denied')}
</code>
```

The value of the "then" attribute is displayed if access is granted for user on object and action. Otherwise, the value of the "else"-attribute is displayed.

## Tree

### Path

Iterates over the path from a given node Id to the root, to draw a path or a rootline menu.

Adds the following variables to the template:

**node:** current node  
**firstNode:** Boolean, true if first node

Arguments:

**repository:** Specifies the node repository  
**namespace:** Specifies the tree namespace  
**node:** The node uid  
**skipRoot:** Skip the root node

Example:

```
<f:for each="{ptx:explode(delimiter: ', ', string:listRow.categoryUid.value.categoryUid)}" as="categoryUid">
  <div>
    <ptx:tree.path node="{categoryUid}" skipRoot="1" namespace="tx_ptcertification_domain_model_category"
      repository="Tx_PtCertification_Domain_Repository_CategoryRepository" >
    <f:if condition="{firstNode}">
      <f:then>{node.label}</f:then>
      <f:else>&raquo; {node.label}</f:else>
    </f:if>
    </ptx:tree.path>
  </div>
</f:for>
```

## Misc

### Comment

Just removes everything between the tags.

Example:

```
<ptx:comment>
<!--
  Here comes the comment which is never rendered.
-->
</ptx:comment>
```

### Explode

Explodes a string by the given delimiter.

Arguments:

**delimiter:** The delimiter character  
**string:** The string to explode.

Example

```
{ptx:explode(delimiter: ', ', '1,2,3,4')}
```

### Captcha

Uses the *captcha* Extension to render a captcha image.

Example

```
<ptx:captcha />
```

### ***CObjectConfig***

Renders the given cObject through cObjGetSingle.

Arguments:

- config:** The cObject config
- data:** Optional data to use for rendering the cObejct.

Example

```
{ptx:cObjectConfig(config: config)}
```

### ***IfValueChanges***

ViewHelper evaluates to true every time a new value is given. This can be used in a loop for example to render a header when one dimension of the objectcollection changes.

Example:

```
<f:for each="{things}" as="thing">
    <ptx:ifValueChanges value="{thing.category}">
        <h1>{thing.category}</h1>
    </ptx:ifValueChanges>
</f:for>
```

### ***RequestArguments***

Shows a submitted extension argument.

Arguments:

- key:** The argument array key

Example:

```
<ptx:requestArguments key="action" />
```

#### **4. Schnittstellen**

Die Extension bindet keine Schnittstellen ein und stellt keine Schnittstellen zur Verfügung.