
C1 adaptive images

Release 0.1.0

Manuel Munz

2018-07-03 19:13

CONTENTS

1	Introduction	1
1.1	What does it do?	1
2	Users Manual	3
3	Administrator Manual	5
3.1	Installation	5
3.2	Configuration	5
4	Configuration Reference	7
4.1	TypoScript Reference	7
4.1.1	Constants	7
4.2	FAQ	8
5	Developer Corner	9
5.1	Hooks	9
5.2	API	9
6	Known Problems	11
7	To-Do list	13
8	ChangeLog	15
9	Links	17

INTRODUCTION

1.1 What does it do?

This extension brings a set of Fluid ViewHelpers that are useful for adaptive image rendering.

By using these additional ViewHelpers in combination with Fluids `f:image` and `f:media` ViewHelpers rendering of adaptive images becomes much easier. Using this simple approach (just using ViewHelpers, not registering a custom ImageRenderer class, which would also be possible) the code stays really simple while at the same time the developer keeps maximum flexibility in generating the picture or image tag using the commun Fluid ViewHelpers.

Note: To use this extension, you should already have a good understanding of Fluid and adaptive images (i.e. know what picture, srcset and sizes are and how to use them).

USERS MANUAL

Target group: **Editors**

Editors shouldn't have to care too much about adaptive images except for crop settings, eventually for different breakpoints using the image editor.

ADMINISTRATOR MANUAL

Target group: **Administrators**

3.1 Installation

To install the extension, perform the following steps:

1. Go to the Extension Manager
2. Install the extension
3. Load the static template

3.2 Configuration

- overwrite default constants and default setup with your own values.

CONFIGURATION REFERENCE

Technical information: Installation, Reference of TypoScript options, configuration options on system level, how to extend it, the technical details, how to debug it and so on.

Language should be technical, assuming developer knowledge of TYPO3. Small examples/visuals are always encouraged.

Target group: **Developers**

4.1 TypoScript Reference

Possible subsections: Reference of TypoScript options. The construct below show the recommended structure for TypoScript properties listing and description.

Properties should be listed in the order in which they are executed by your extension, but the first should be alphabetical for easier access.

When detailing data types or standard TypoScript features, don't hesitate to cross-link to the TypoScript Reference as shown below. See the `Settings.yml` file for the declaration of cross-linking keys.

4.1.1 Constants

Properties

Property	Data type	<code>stdWrap</code>	Default
<code>allWrap</code>	<code>wrap</code>	yes	<code><div> </div></code>
<code>subst_elementUid</code>	<code>boolean</code>	no	0
<code>wrapItemAndSub</code>	<code>wrap</code>		

Property details

`allWrap`

```
plugin.tx_extensionkey.allWrap = wrap
```

Wraps the whole item.

`wrapItemAndSub`

```
plugin.tx_extensionkey.wrapItemAndSub = wrap
```

Wraps the whole item and any submenu concatenated to it.

subst_elementUid

`plugin.tx_extensionkey.subst_elementUid` = **boolean**

If set, all appearances of the string {elementUid} in the total element html-code (after wrapped in *allWrap*) are substituted with the uid number of the menu item. This is useful if you want to insert an identification code in the HTML in order to manipulate properties with JavaScript.

4.2 FAQ

Possible subsection: FAQ

DEVELOPER CORNER

Target group: **Developers**

Use this section for *providing code examples* or any **useful** information code wise.

5.1 Hooks

Possible hook examples. Input parameters are:

Parameter	Data type	Description
\$table	string	Name of the table
\$field	string	Name of the field

Use parameter `$table` to retrieve the table name...

5.2 API

How to use the API...

```
$stuff = \TYPO3\CMS\Core\Utility\GeneralUtility::makeInstance(  
    '\\Foo\\Bar\\Utility\\Stuff'  
);  
$stuff->do();
```

or some other language:

```
1 $(document).ready(  
2     function () {  
3         doStuff();  
4     }  
5 );
```


KNOWN PROBLEMS

Say where bugs can be reported / followed up. Is it a **bug tracker**? Use this section for informing about any type of problem that are not necessarily named in the bug tracker such as performance issues, ...

TO-DO LIST

Give a link pointing to a [roadmap](#). Alternatively, you can dress up a list of things you want to add or fix in this chapter or give a vision about where the extension is heading.

CHANGELOG

Providing a change log chapter is optional. You can also refer users to the ChangeLog file inside the extension or to some repository's commit listing.

LINKS

TER <https://typo3.org/extensions/repository/view/<extension key>>

Bug Tracker <https://forge.typo3.org/projects/extension-<extension key>/issues>

Git Repository <https://github.com/<username>/<extension key>>

Contact @<username>