EXT: tollwerk Open Device Lab

Extension Key: tw_odl
Language: en
Version: 0.2.0
Keywords: Open Device Lab, odl, tcpdf, wallpaper, json, jsonp, forAdministrators, forDevelopers, forIntermediates, forAdvanced
Copyright © 2013 Dipl.-Ing. Joschi Kuphal, <joschi@tollwerk.de>

This document is published under the Open Content License available from
http://www.opencontent.org/opl.shtml

The content of this document is related to TYPO3 – a GNU/GPL CMS/Framework available from www.typo3.org.

In case you are reading this manual online at the TYPO3 website, we strongly recommend that you also visit the TYPO3 Extensions & Manuals page respectively the page about the tollwerk Open Device Lab TYPO3-Extension at our own website. We provide a PDF version of this manual there, which probably renders more nicely than the online version on typo3.org. (Sorry for our website currently being available in German language only. However, the PDF extension manuals are in English of course.)

Table of Contents

Introduction

What does it do?

Let's use the words of opendevicelab.com to introduce the big idea behind Open Device Labs:

Open Device Labs (ODLs, #ODL) are a grass roots community movement. They establish shared community pools of internet connected devices for testing purposes of web and app developers. In result, ODLs lead to an ultimate improvement of the web & app experience both for developers and for consumers.

Since early 2012 there are lots of Open Device Labs evolving all over the world, most of them running their own, mostly small website, while opendevicelab.com and lab-up.org being something like the “big umbrella” above it all.

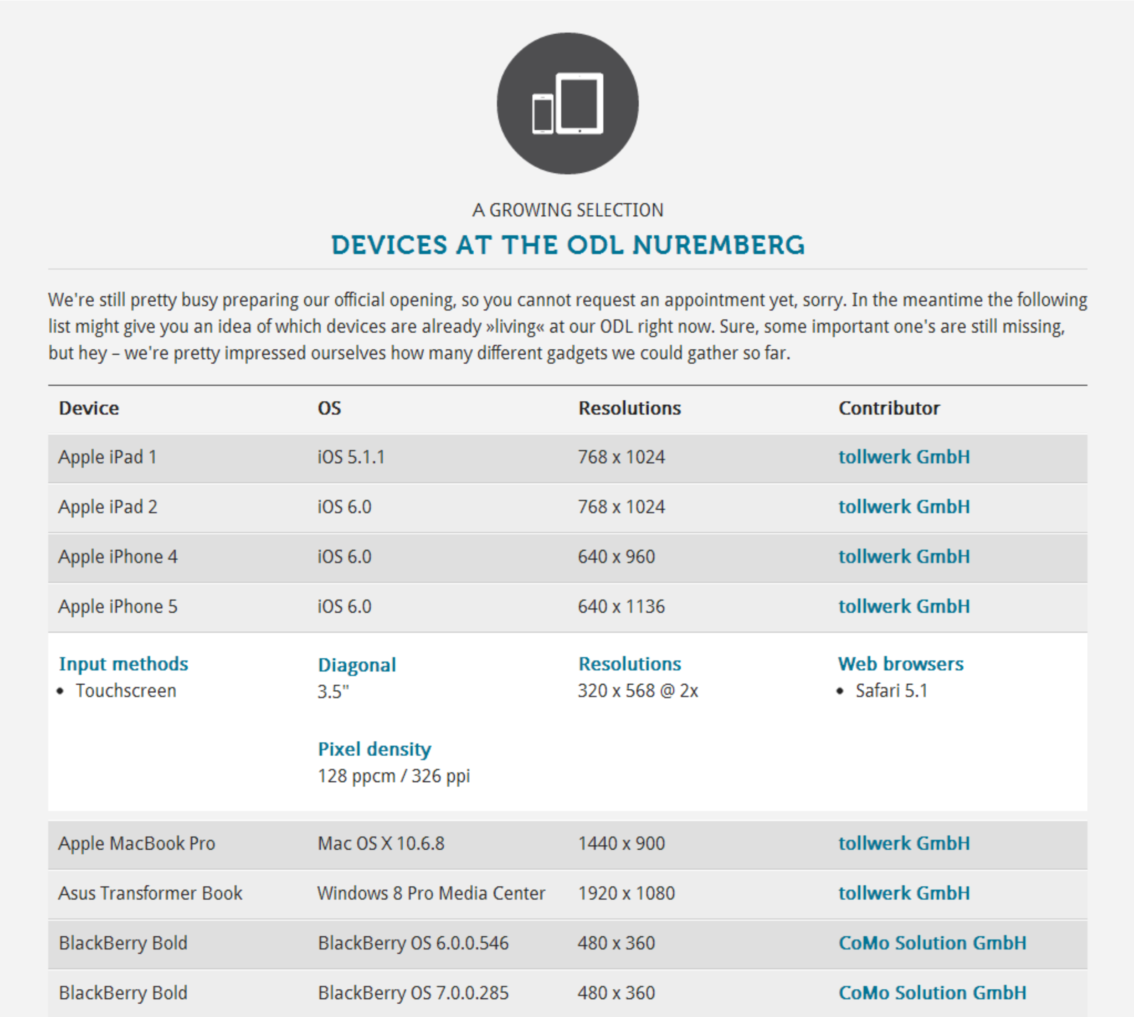
This extension would like to be of use for anyone running an ODL website with the help TYPO3. It provides some features that might come handy, like e.g.:

- Detailled device management including the rendering of a frontend device list
- Generation of multi-purpose device labels via PDF using TCPDF (http://www.tcpdf.org/)
- Creation of branded device wallpaper images
- JSON / JSONP API for registered ODL devices

The extension was written for TYPO3 version 6+ (no support for 4.x!) and is based on extbase / fluid.

Screenshots

The frontend plugin renders a list of registered ODL devices. It's based on a fluid template and can be tailored to your individual needs.



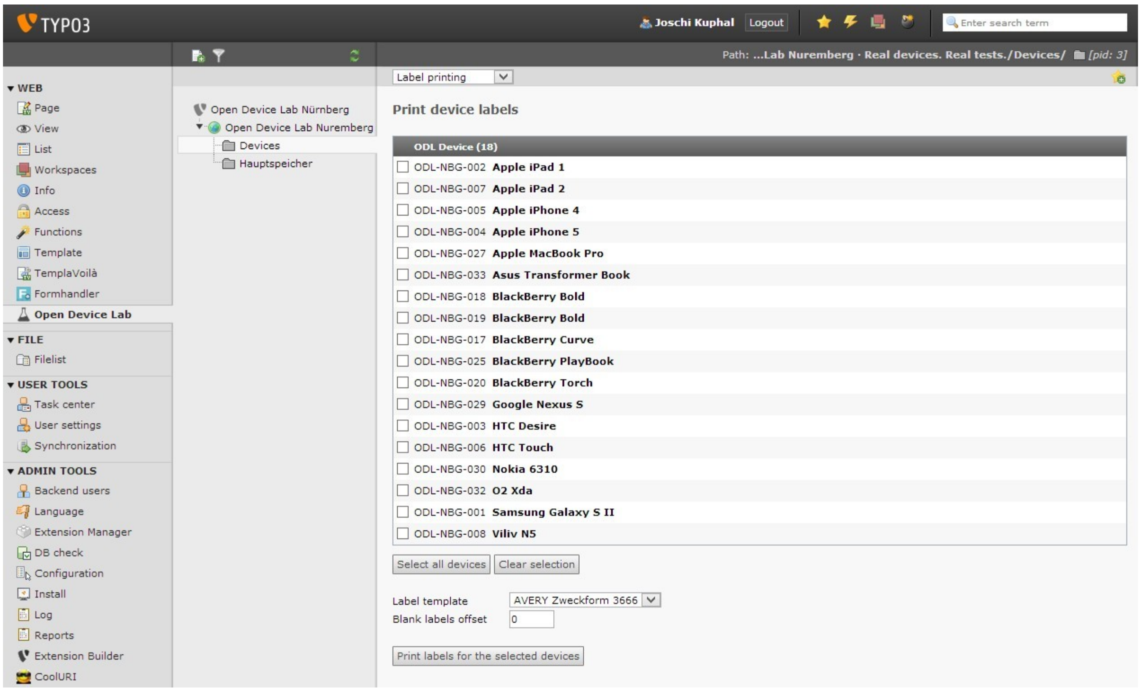
A GROWING SELECTION

DEVICES AT THE ODL NUREMBERG

We're still pretty busy preparing our official opening, so you cannot request an appointment yet, sorry. In the meantime the following list might give you an idea of which devices are already »living« at our ODL right now. Sure, some important one's are still missing, but hey – we're pretty impressed ourselves how many different gadgets we could gather so far.

Device	OS	Resolutions	Contributor
Apple iPad 1	iOS 5.1.1	768 x 1024	tollwerk GmbH
Apple iPad 2	iOS 6.0	768 x 1024	tollwerk GmbH
Apple iPhone 4	iOS 6.0	640 x 960	tollwerk GmbH
Apple iPhone 5	iOS 6.0	640 x 1136	tollwerk GmbH
Input methods • Touchscreen	**Diagonal** 3.5" **Pixel density** 128 ppcm / 326 ppi	**Resolutions** 320 x 568 @ 2x	**Web browsers** • Safari 5.1
Apple MacBook Pro	Mac OS X 10.6.8	1440 x 900	tollwerk GmbH
Asus Transformer Book	Windows 8 Pro Media Center	1920 x 1080	tollwerk GmbH
BlackBerry Bold	BlackBerry OS 6.0.0.546	480 x 360	CoMo Solution GmbH
BlackBerry Bold	BlackBerry OS 7.0.0.285	480 x 360	CoMo Solution GmbH

The backend provides a dedicated module for label printing and wallpaper image creation.



Print device labels

ODL Device (18)

- □ ODL-NBG-002 **Apple iPad 1**
- □ ODL-NBG-007 **Apple iPad 2**
- □ ODL-NBG-005 **Apple iPhone 4**
- □ ODL-NBG-004 **Apple iPhone 5**
- □ ODL-NBG-027 **Apple MacBook Pro**
- □ ODL-NBG-033 **Asus Transformer Book**
- □ ODL-NBG-018 **BlackBerry Bold**
- □ ODL-NBG-019 **BlackBerry Bold**
- □ ODL-NBG-017 **BlackBerry Curve**
- □ ODL-NBG-025 **BlackBerry PlayBook**
- □ ODL-NBG-020 **BlackBerry Torch**
- □ ODL-NBG-029 **Google Nexus S**
- □ ODL-NBG-003 **HTC Desire**
- □ ODL-NBG-006 **HTC Touch**
- □ ODL-NBG-030 **Nokia 6310**
- □ ODL-NBG-032 **O2 Xda**
- □ ODL-NBG-001 **Samsung Galaxy S II**
- □ ODL-NBG-008 **Viliv N5**

Label template: AVERY Zweckform 3666

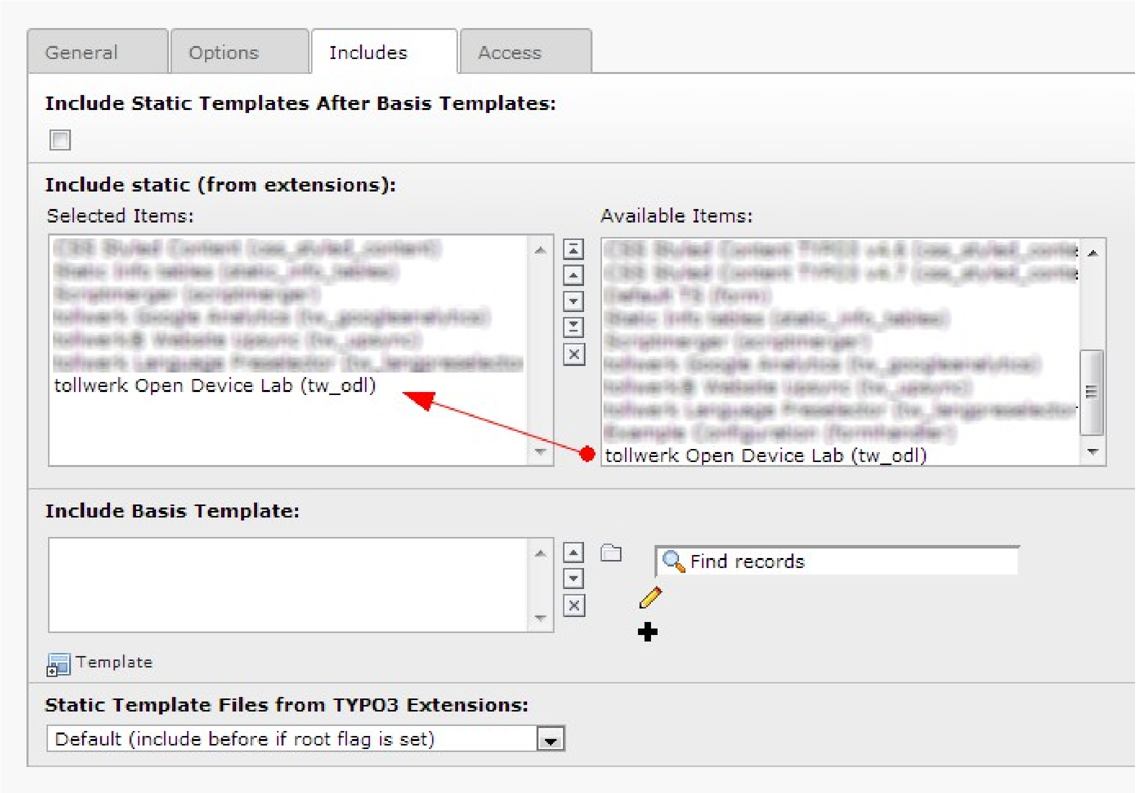
Blank labels offset: 0

Installation

To install the extension simply download it from the TYPO3 extension repository and enable it in the Extension Manager. There's nothing to configure in the Extension Manager.

Static TypoScript

Include the extension's static TypoScript into the TypoScript root template of your site.



Include Static Templates After Basis Templates:

□

Include static (from extensions):

Selected Items:

tollwerk Open Device Lab (tw_odl)

Available Items:

tollwerk Open Device Lab (tw_odl)

Include Basis Template:

Static Template Files from TYPO3 Extensions:

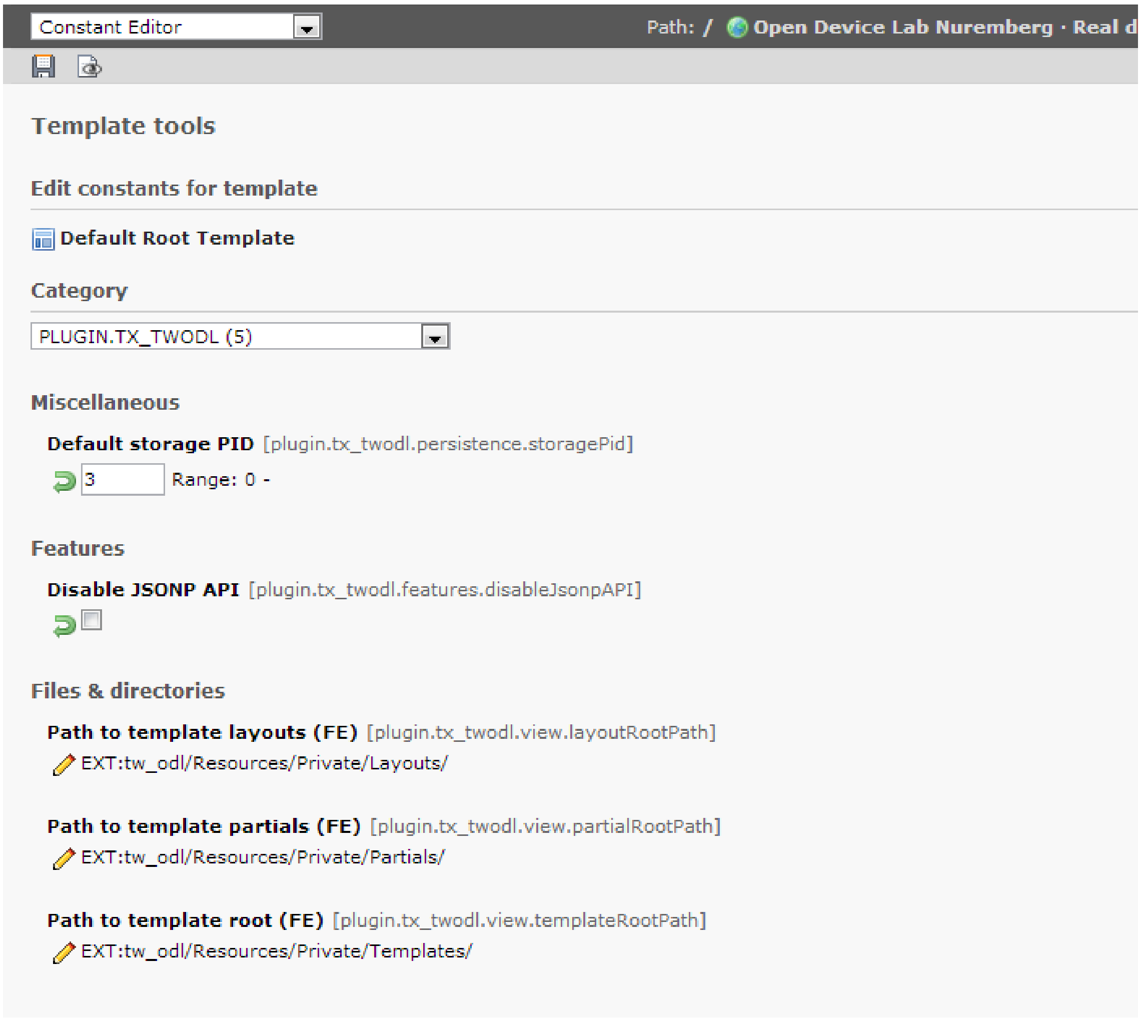
Default (include before if root flag is set)

Configuration

Constants

Most of the extension's features can be controlled via the constant editor (e.g. on the TypoScript root template). Almost any constant directly relates to a corresponding setup option that can also be set by manually crafted TypoScript. Please see the setup chapter for details on these options.

There are separate constant editor sections for the frontend plugin and the backend module. The frontend plugin parameters mainly consist of the usual settings for extbase extensions like root paths and the storage folder. Additionally you can disable the extension's JSONP API here.



Constant Editor

Path: / Open Device Lab Nuremberg · Real d

Template tools

Edit constants for template

Default Root Template

Category

PLUGIN.TX_TWODL (5)

Miscellaneous

Default storage PID [plugin.tx_twodl.persistence.storagePid]

3 Range: 0 -

Features

Disable JSONP API [plugin.tx_twodl.features.disableJsonpAPI]

□

Files & directories

Path to template layouts (FE) [plugin.tx_twodl.view.layoutRootPath]

EXT:tw_odl/Resources/Private/Layouts/

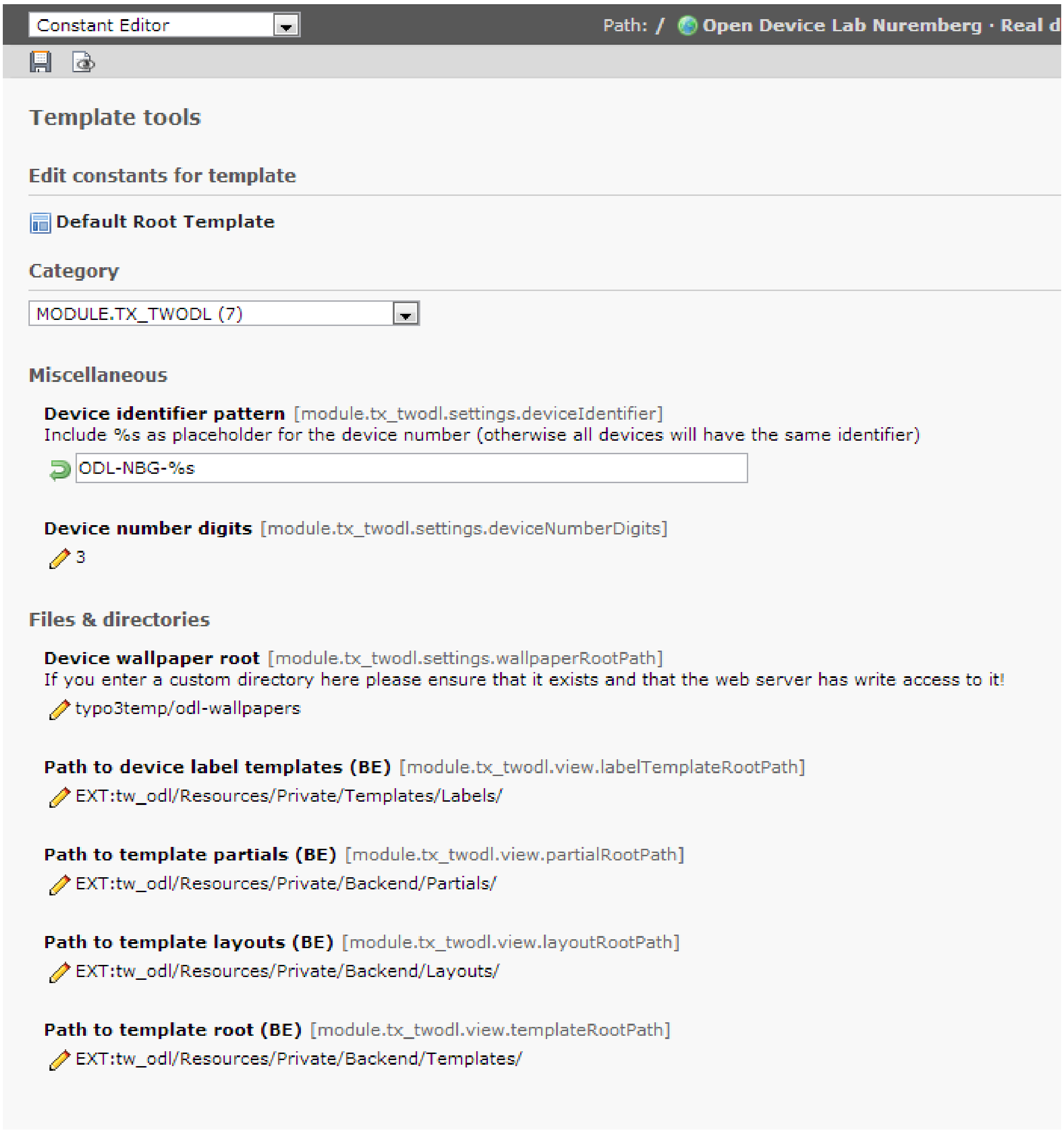
Path to template partials (FE) [plugin.tx_twodl.view.partialRootPath]

EXT:tw_odl/Resources/Private/Partials/

Path to template root (FE) [plugin.tx_twodl.view.templateRootPath]

EXT:tw_odl/Resources/Private/Templates/

The backend module constants feature additional parameters for specifying a **device identifier**, the root path for print label templates as well as the (publicly accessible) folder for device wallpapers.



Template tools

Edit constants for template

Default Root Template

Category

MODULE.TX_TWODL (7)

Miscellaneous

Device identifier pattern [module.tx_twodl.settings.deviceIdentifier]
Include %s as placeholder for the device number (otherwise all devices will have the same identifier)
ODL-NBG-%s

Device number digits [module.tx_twodl.settings.deviceNumberDigits]
3

Files & directories

Device wallpaper root [module.tx_twodl.settings.wallpaperRootPath]
If you enter a custom directory here please ensure that it exists and that the web server has write access to it!
typo3temp/odl-wallpapers

Path to device label templates (BE) [module.tx_twodl.view.labelTemplateRootPath]
EXT:tw_odl/Resources/Private/Templates/Labels/

Path to template partials (BE) [module.tx_twodl.view.partialRootPath]
EXT:tw_odl/Resources/Private/Backend/Partials/

Path to template layouts (BE) [module.tx_twodl.view.layoutRootPath]
EXT:tw_odl/Resources/Private/Backend/Layouts/

Path to template root (BE) [module.tx_twodl.view.templateRootPath]
EXT:tw_odl/Resources/Private/Backend/Templates/

Setup

Frontend plugin setup

There are several TypoScript setup options controlling the behavior of the frontend plugin:

Property:	Data type:	Description:	Default:
view.templateRootPath	dir	Root path for the **fluid templates** of the plugin. The plugin has just one controller (named "Device") with two actions (named "list" and "jsonp", where the latter is a special action that doesn't require a fluid template). Accordingly there has to be one fluid template at the following location (relative to the template root path): • *Device/List.html*	{$plugin.tx_twodl.view.templateRootPath}
view.partialRootPath	dir	Root path for the **fluid partials** of the plugin (not used by default)	{$plugin.tx_twodl.view.partialRootPath}
view.layoutRootPath	dir	Root path for the **fluid layouts** of the plugin (not used by default)	{$plugin.tx_twodl.view.layoutRootPath}
persistence.storagePid	dir	ID of the storage page containing the device and auxiliary records that should be shown in the frontend list	{$plugin.tx_twodl.persistence.storagePid}
settings.deviceIdentifier	string	Pattern for **device identifiers**. This one is identical to the corresponding backend module setting and get's inherited from there by default (see below for a description).	{$module.tx_twodl.settings.deviceIdentifier}
settings.deviceNumberDigits	integer	Total digits of **device numbers**. This one is identical to the corresponding backend module setting and get's inherited from there by default (see below for a description).	{$module.tx_twodl.settings.deviceNumberDigits}

[tsref:plugin.tx_twodl]

Backend module setup

There are several TypoScript setup options controlling the behavior of the backend module:

Property:	Data type:	Description:	Default:
view.templateRootPath	dir	Root path for the **fluid templates** of the model. The module has just one controller (named "Odl") with three actions (named "labels", "print" and "wallpaper"). Accordingly there have to be three fluid templates at the following locations (relative to the template root path): • *Odl/Labels.html* • *Odl/Print.html* • *Odl/Wallpapers.html* You are not supposed to change this setting.	{$module.tx_twodl.view.templateRootPath}
view.partialRootPath	dir	Root path for the **fluid partials** of the module (not used by default)	{$module.tx_twodl.view.partialRootPath}
view.layoutRootPath	dir	Root path for the **fluid layouts** of the module. You are not supposed to change this setting.	{$module.tx_twodl.view.layoutRootPath}
settings.labelTemplateRootPath	dir	Root path for **print label templates**. Templates for rendering print labels are supposed to be fluid templates with usage of special TCPDF view helpers.	{$plugin.tx_twodl.persistence.storagePid}

Property:	Data type:	Description:	Default:
settings.wallpaperRootPath	dir	Root path for **device wallpaper images**. Wallpapers are generated through the backend module and supposed to be publicly accessible for ease of installation on particular devices (reachable with regular web browsers), so make sure to enter an existing webserver directory here (relative to site root).	{$module.tx_twodl.settings.wallpaperRootPath}
settings.deviceIdentifier	string	Pattern for **device identifiers**. Each device will have a unique identifier, consisting of a device number (see below) and potentially some more letters or symbols. The (padded) device number will be substituted into this pattern via the PHP function *sprintf*, so make sure to include *%s* as marker in the pattern.	{$module.tx_twodl.settings.deviceIdentifier}
settings.deviceNumberDigits	integer	Total digits of **device numbers**. A device's number will be built by left-padding it's ***uid*** with the appropriate number of zeros. The padded device number will e.g. be used for the device identifiers (see above).	{$module.tx_twodl.settings.deviceNumberDigits}
settings.wallpaper	GIFBUILDER	TypoScript configuration for the rendering of **device wallpaper images**. This configuration will be used for the processing of a GIFBUILDER object and may contain some special device specific markers that will become substituted prior to the rendering process: • *###DEVICEWIDTH###*: Device screen width as given in the device record • *###DEVICEHEIGHT###*: Device screen height as given in the device record • *###DEVICEHOMEWIDTH###*: Device wallpaper width as given in the device record • *###DEVICEHOMEHEIGHT###*: Device wallpaper height as given in the device record • *###DEVICEPPCM###*: Device screen density in pixels per centimeter as given in the device record • *###DEVICEPPI###*: Device screen density in pixels per inch as given in the device record • *###DEVICENUMBER###*: Padded device number (see above) • *###DEVICEIDENTIFIER###*: Device identifier (see above)	See EXT:tw_odl/Configuration/TypoScript/setup.txt

[tsref:module.tx_twodl]

Usage

Device management

The extension introduces a couple of record types associated with the management of ODL devices. These are:

- Devices
- Manufacturers
- Operating systems
- Operating system versions
- Web browsers
- Web browser versions
- CSS screen resolutions
- Contributors

The central record type is – of course – the device, whereas the other record types serve as auxiliary records. Each device has

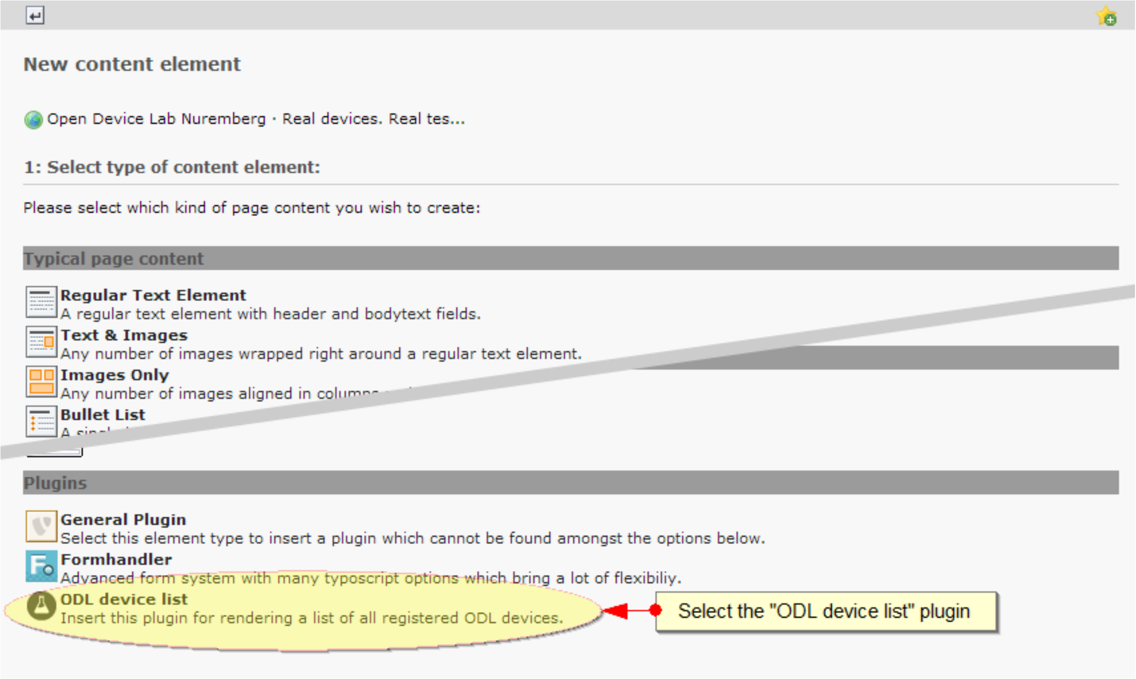
- one manufacturer,
- one operating system (with an optional version),
- one or more web browsers installed (each with an optional version),
- one or two CSS screen resolutions (they might differ from the physical screen resolution and might also be different in landscape and portrait orientation)
- and one contributor.

It is suggested that you create and manage the auxiliary records explicitly instead of creating them from within a device form. While the necessary wizards are included in the forms TYPO3's support for them is somewhat limited. Most importantly all the associated records (including the devices) have to be **contained on one single page** in the page tree.

The single properties of the different record types in the backend forms should be pretty self explanatory. Most of the properties are also being exported through the JSON / JSONP API as well, so it might be a good idea to study a JSON export result for further understanding.

Placing a device list in the frontend

To place a device list on a frontend page simply select the “ODL Device List” plugin from the “New content element” wizard:



New content element

Open Device Lab Nuremberg · Real devices. Real tes...

1: Select type of content element:

Please select which kind of page content you wish to create:

Typical page content

Regular Text Element
A regular text element with header and bodytext fields.

Text & Images
Any number of images wrapped right around a regular text element.

Images Only
Any number of images aligned in column

Bullet List

Plugins

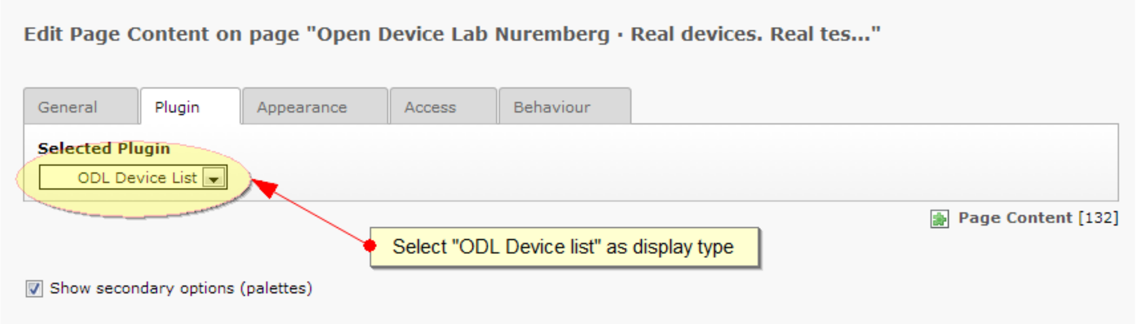
General Plugin
Select this element type to insert a plugin which cannot be found amongst the options below.

Formhandler
Advanced form system with many typoscript options which bring a lot of flexibiliy.

ODL device list
Insert this plugin for rendering a list of all registered ODL devices.

Select the "ODL device list" plugin

The content element created should feature the plugin type “ODL Device List”:



Edit Page Content on page "Open Device Lab Nuremberg · Real devices. Real tes..."

Selected Plugin: ODL Device List

Page Content [132]

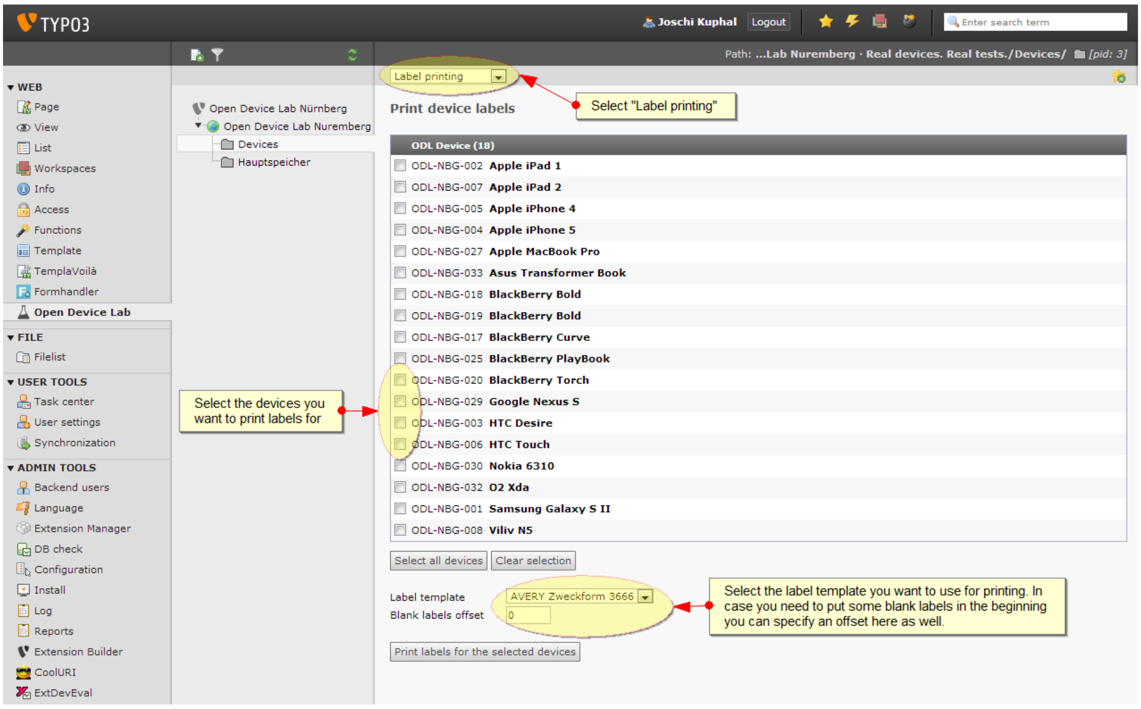
Select "ODL Device list" as display type

☑ Show secondary options (palettes)

Printing device labels

The extension provides a backend module which you can use for generating and printing device labels. You could use such labels e.g. for labeling your device display or your devices themselves. Labels are created as PDF files, as directly printing from the browser would potentially result in header and footer lines be printed as well.

Please select the **Web > Open Device Lab** module to the left as well as a page carrying device and auxiliary records in the page tree to print labels for those devices.



Select "Label printing"

Print device labels

ODL Device (18)	
□ ODL-NBG-002	Apple iPad 1
□ ODL-NBG-007	Apple iPad 2
□ ODL-NBG-005	Apple iPhone 4
□ ODL-NBG-004	Apple iPhone 5
□ ODL-NBG-027	Apple MacBook Pro
□ ODL-NBG-033	Asus Transformer Book
□ ODL-NBG-018	BlackBerry Bold
□ ODL-NBG-019	BlackBerry Bold
□ ODL-NBG-017	BlackBerry Curve
□ ODL-NBG-025	BlackBerry PlayBook
□ ODL-NBG-020	BlackBerry Torch
□ ODL-NBG-029	Google Nexus S
□ ODL-NBG-003	HTC Desire
□ ODL-NBG-006	HTC Touch
□ ODL-NBG-030	Nokia 6310
□ ODL-NBG-032	O2 Xda
□ ODL-NBG-001	Samsung Galaxy S II
□ ODL-NBG-008	Viliv N5

Select the devices you want to print labels for

Label template: AVERY Zweckform 3666
Blank labels offset: 0

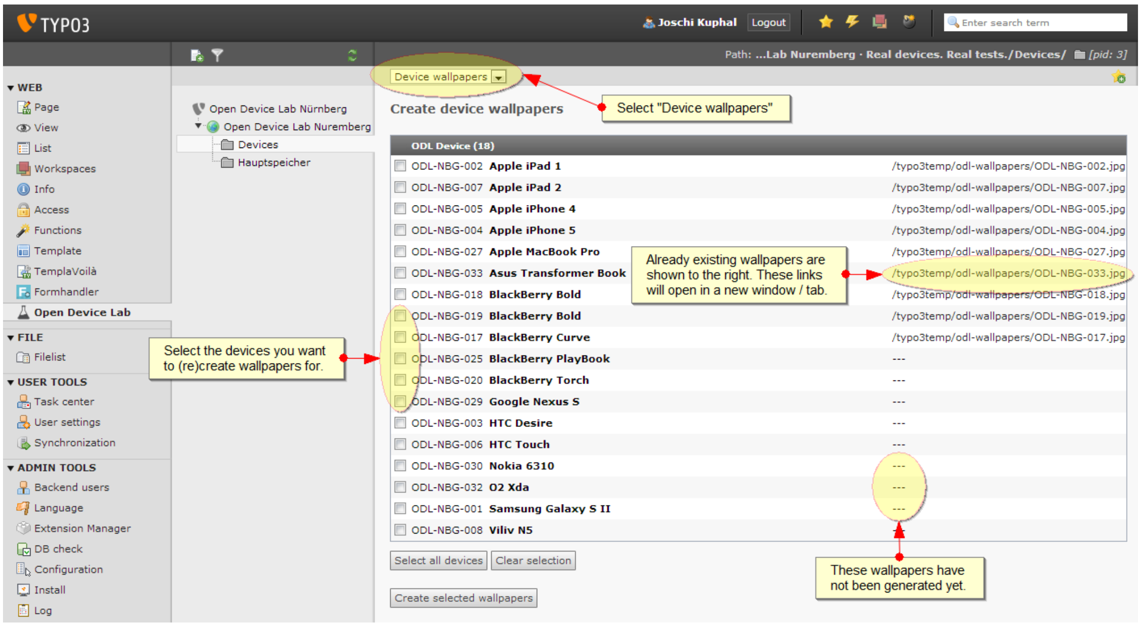
Select the label template you want to use for printing. In case you need to put some blank labels in the beginning you can specify an offset here as well.

There are two exemplary label templates coming with the extension, one for using a label sticker sheet with multiple columns and rows of labels ("AVERY Zweckform 3666") and another one for printing labels on endless paper with a dedicated label printer ("Endless paper"). The label templates are built with fluid (don't get confused by the ".pdf" file extension they carry, they're just ordinary XML) residing in the directory specified by the **settings.wallpaperRootPath** TypoScript parameter.

PDF generation itself is done via the freely available TCPDF library (http://www.tcpdf.org). The extension includes an unburdened version of TCPDF (essentially the examples and fonts have been stripped out in favor of significantly smaller file size). The extension comes with a couple of specialized view helpers for controlling TCPDF output. You can easily use your own label templates (or customize one of the default ones) by dropping appropriate fluid template files (with ".pdf" file extension) into the label template root directory.

Generating device wallpapers

As it might be desirable to give your devices an all-over consistent appearance the extension provides a backend module which you can use for generating wallpaper images for your devices. Please select the **Web > Open Device Lab** module to the left as well as a page carrying device and auxiliary records in the page tree to generate wallpapers for those devices.



Select "Device wallpapers"

Create device wallpapers

ODL Device (18)			
☐	ODL-NBG-002	**Apple iPad 1**	/typo3temp/odl-wallpapers/ODL-NBG-002.jpg
☐	ODL-NBG-007	**Apple iPad 2**	/typo3temp/odl-wallpapers/ODL-NBG-007.jpg
☐	ODL-NBG-005	**Apple iPhone 4**	/typo3temp/odl-wallpapers/ODL-NBG-005.jpg
☐	ODL-NBG-004	**Apple iPhone 5**	/typo3temp/odl-wallpapers/ODL-NBG-004.jpg
☐	ODL-NBG-027	**Apple MacBook Pro**	/typo3temp/odl-wallpapers/ODL-NBG-027.jpg
☐	ODL-NBG-033	**Asus Transformer Book**	/typo3temp/odl-wallpapers/ODL-NBG-033.jpg
☐	ODL-NBG-018	**BlackBerry Bold**	/typo3temp/odl-wallpapers/ODL-NBG-018.jpg
☐	ODL-NBG-019	**BlackBerry Bold**	/typo3temp/odl-wallpapers/ODL-NBG-019.jpg
☐	ODL-NBG-017	**BlackBerry Curve**	/typo3temp/odl-wallpapers/ODL-NBG-017.jpg
☐	ODL-NBG-025	**BlackBerry PlayBook**	---
☐	ODL-NBG-020	**BlackBerry Torch**	---
☐	ODL-NBG-029	**Google Nexus S**	---
☐	ODL-NBG-003	**HTC Desire**	---
☐	ODL-NBG-006	**HTC Touch**	---
☐	ODL-NBG-030	**Nokia 6310**	---
☐	ODL-NBG-032	**O2 Xda**	---
☐	ODL-NBG-001	**Samsung Galaxy S II**	---
☐	ODL-NBG-008	**Viliv N5**	

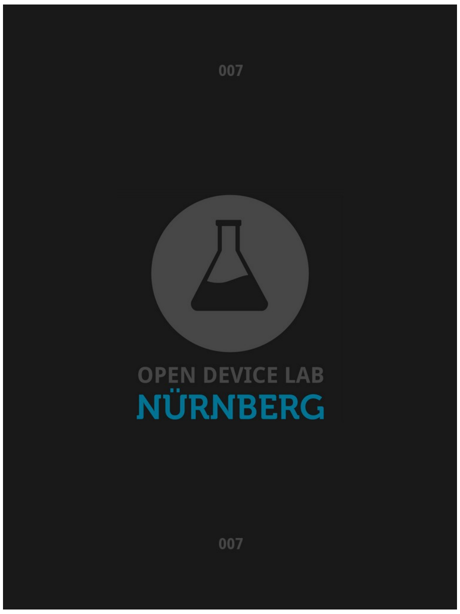
Already existing wallpapers are shown to the right. These links will open in a new window / tab.

Select the devices you want to (re)create wallpapers for.

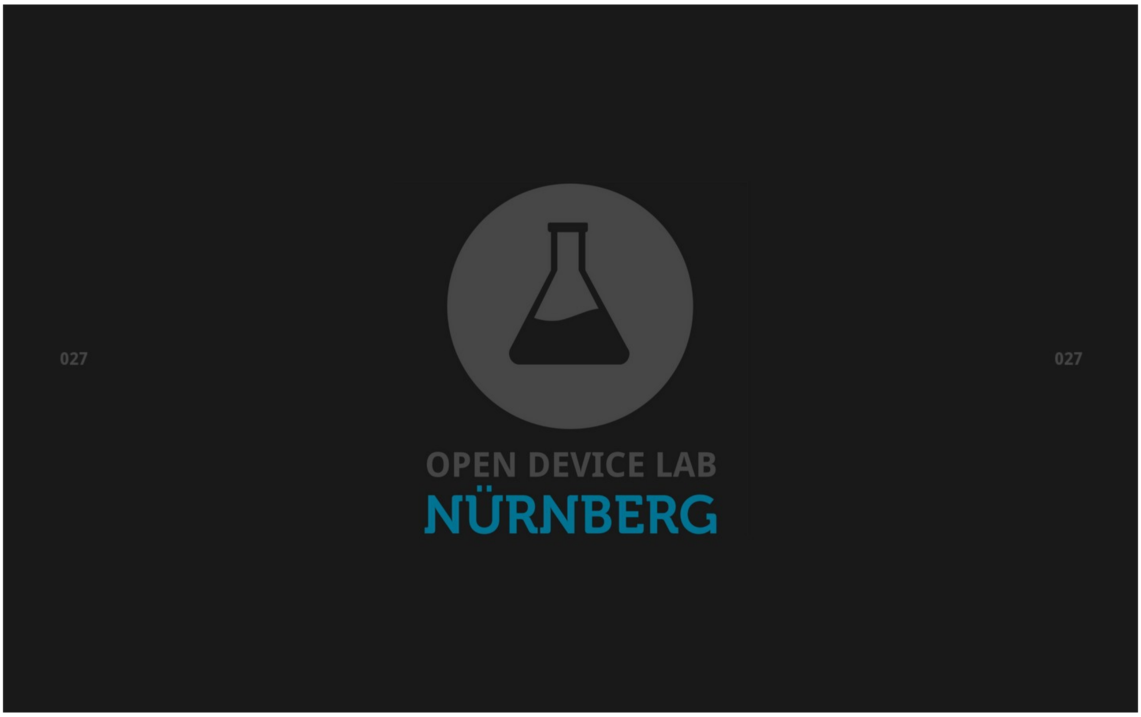
These wallpapers have not been generated yet.

The wallpaper images are rendered by means of regular TYPO3 frontend content element rendering using a GUIBUILDER TypoScript configuration which can be customized via the **settings.wallpaper** parameter. There are some custom markers you may use within the wallpaper configuration – see the TypoScript settings documentation for details. Most importantly, each wallpaper will be rendered in the dimensions defined by the corresponding device record (wallpaper width and height settings). Examples of wallpapers might e.g. look like:

Apple iPad 2 wallpaper



Apple MacBook Pro wallpaper



In many cases the easiest way of transferring a wallpaper to a mobile device is by retrieving and saving it with a regular web browser.In many cases the easiest way of transferring a wallpaper to a mobile device is by retrieving and saving it with a regular web browser. To accomplish this the extension generates the wallpaper images to a publicly accessible directory somewhere below your website root (by default into the directory typo3temp/odl-wallpapers – configure it via the settings.wallpaperRootPath TypoScript setting). Once generated the path to each device's wallpaper is shown in the backend module's device list. You can click on each wallpaper to open it in a separate browser window. To retrieve the wallpaper on your device simply browse to the URL shown in the address bar.

JSON / JSONP API

By default the extension exposes the registered devices through a JSON / JSONP API (you might disable this via the constant editor). You may query the API by requesting the TYPO3 frontend with a type parameter of value 1333:

http://your-device-lab.org/index.php?type=1333

The idea behind the JSON API is very well explained by Andre Jay Meissner (http://opendevicelab.com / http://lab-up.org) in response to a blog post of Africa's Nomad Device Lab: At some point in the future **opendevicelab.com** might start grabbing device lab data by pulling JSON files from the ODL's websites. Currently no such feature exists yet, but already having prepared a JSON API simply feels good, don't you think?

The export structure used right now is somewhat fictional and might lack certain information (e.g. the device lab residential data), but as soon as opendevicelab.com announces a standard to be used we will adapt the output. This is a current output example (excerpt):

```
[
   {
      "manufacturer":"Apple",
      "name":"iPad 1",
      "model":"MB292FD",
      "os":"iOS",
      "osversion":"5.1.1",
      "width":768,
      "height":1024,
      "diagonal":9.7,
      "ppcm":52,
      "ppi":132,
      "inputmethods":[
         "touchscreen"
      ],
      "resolutions":[
         {
            "width":768,
            "height":1024,
            "ratio":1,
            "orientation":3
         }
      ],
      "comment":"",
      "contributor":{
         "name":"tollwerk GmbH",
         "logo":"tollwerk-logo.png",
         "url":"tollwerk.de"
      },
      "available":1
   },
   ...
]
```

Use one of the GET parameters *callback* or *jsonp* to specifiy the callback function that should be used for padding the JSON data (otherwise you will get a simple JSON array as result). Example:

http://your-device-lab.org/index.php?type=1333&callback=myCallbackFunction

Advanced techniques

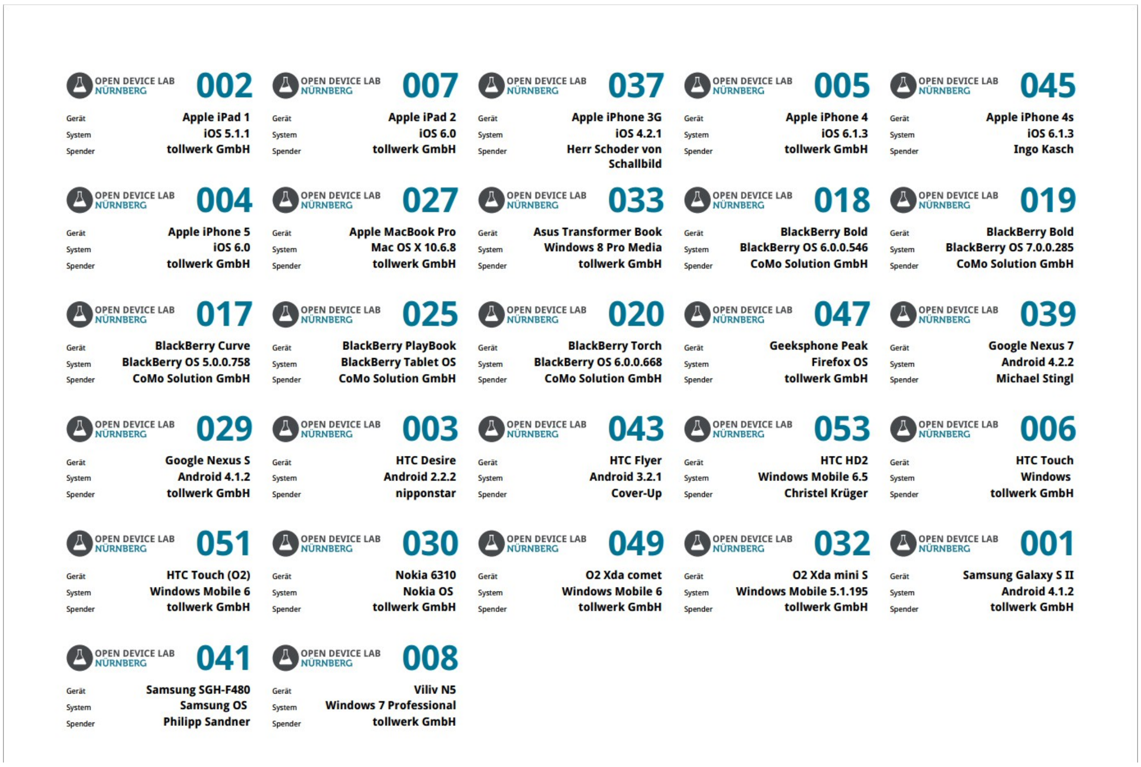
Creating a custom device label template

The fluid templates used for device label printing reside in the directory specified by the **settings.labelTemplateRootPath** TypoScript parameter and have to carry a ".pdf" file extension in order to be listed in the backend module (although they have ordinary XML content). Internally they make use of some special TCPDF view helpers documented below. One of the default label templates has the following content:

```
{namespace odl=Tollwerk\TwOdl\ViewHelpers}
<odl:tcpdf width="210" height="297" unit="mm" left="9.75" right="9.75"
    top="10.63" bottom="10.63" columns="5" rows="13" offset="{offset}">
    <odl:tcpdf.font family="DroidSans"
        fontfile="fileadmin/templates/fonts/DroidSans/DroidSans.ttf"/>
    <odl:tcpdf.font family="DroidSans"
        fontfile="fileadmin/templates/fonts/DroidSans/DroidSans-Bold.ttf"
        style="b"/>
    <f:for each="{devices}" as="device">
        <odl:tcpdf.label>
            <odl:tcpdf.image
                file="EXT:tw_odl/Resources/Public/Images/odl-nbg-hor-de-white.eps"
                width="20" top="2" left="2" right="2" align="left"/>
            <odl:tcpdf.text x="22" y="0.5" width="15.5" halign="right"
                fontfamily="DroidSans" fontstyle="b"
                fontcolor="{0:0, 1:116, 2:148}" fontsize="18">
                <odl:deviceIdentifier device="{device}" identifier="%s"
                    digits="{settings.deviceNumberDigits}"/>
            </odl:tcpdf.text>
            <odl:tcpdf.text x="1" y="9" width="10" height="2.5"
                fontfamily="DroidSans" fontsize="4" valign="bottom">
                <f:translate key="device.name"/>
            </odl:tcpdf.text>
            <odl:tcpdf.text x="10" width="27" height="3" fontfamily="DroidSans"
                fontsize="6" fontstyle="b" cursor="1" halign="right"
                >{device.manufacturer.name} {device.name}</odl:tcpdf.text>
            <odl:tcpdf.text x="1" width="10" height="2.5" fontfamily="DroidSans"
                fontsize="4" valign="bottom">
                <f:translate key="device.os"/>
            </odl:tcpdf.text>
            <odl:tcpdf.text x="10" width="27" height="3" fontfamily="DroidSans"
                fontsize="6" fontstyle="b" cursor="1" halign="right"
                >{device.os.name} {device.osversion.version}</odl:tcpdf.text>
            <odl:tcpdf.text x="1" width="10" height="2.5" fontfamily="DroidSans"
                fontsize="4" valign="bottom">
                <f:translate key="device.contributor"/>
            </odl:tcpdf.text>
            <odl:tcpdf.text x="10" width="27" height="6" fontfamily="DroidSans"
                fontsize="6" fontstyle="b" halign="right"
                >{device.contributor.name}</odl:tcpdf.text>
        </odl:tcpdf.label>
    </f:for>
</odl:tcpdf>
```

As you see there is no HTML, just TCPDF specific markup using fluid syntax and the TCPDF view helpers. One could describe the template with the following words:

On the top level a target medium is specified (in this case a DIN A4 sheet with 5 label columns and 13 label rows) using the **odl:tcpdf** view helper. Next, two custom fonts (Droid Sans) get declared via the **odl:tcpdf.font** view helper. Then, for each device in the list a label is created with the **odl:tcpdf.label** view helper, each consisting of an image (**odl:tcpdf.image**) and several text cells (**odl:tcpdf.text**). The PDF output of the template looks like this:



OPEN DEVICE LAB NÜRNBERG 002
Gerät Apple iPad 1
System iOS 5.1.1
Spender tollwerk GmbH

OPEN DEVICE LAB NÜRNBERG 007
Gerät Apple iPad 2
System iOS 6.0
Spender tollwerk GmbH

OPEN DEVICE LAB NÜRNBERG 037
Gerät Apple iPhone 3G
System iOS 4.2.1
Spender Herr Schoder von Schallbild

OPEN DEVICE LAB NÜRNBERG 005
Gerät Apple iPhone 4
System iOS 6.1.3
Spender tollwerk GmbH

OPEN DEVICE LAB NÜRNBERG 045
Gerät Apple iPhone 4s
System iOS 6.1.3
Spender Ingo Kasch

OPEN DEVICE LAB NÜRNBERG 004
Gerät Apple iPhone 5
System iOS 6.0
Spender tollwerk GmbH

OPEN DEVICE LAB NÜRNBERG 027
Gerät Apple MacBook Pro
System Mac OS X 10.6.8
Spender tollwerk GmbH

OPEN DEVICE LAB NÜRNBERG 033
Gerät Asus Transformer Book
System Windows 8 Pro Media
Spender tollwerk GmbH

OPEN DEVICE LAB NÜRNBERG 018
Gerät BlackBerry Bold
System BlackBerry OS 6.0.0.546
Spender CoMo Solution GmbH

OPEN DEVICE LAB NÜRNBERG 019
Gerät BlackBerry Bold
System BlackBerry OS 7.0.0.285
Spender CoMo Solution GmbH

OPEN DEVICE LAB NÜRNBERG 017
Gerät BlackBerry Curve
System BlackBerry OS 5.0.0.758
Spender CoMo Solution GmbH

OPEN DEVICE LAB NÜRNBERG 025
Gerät BlackBerry PlayBook
System BlackBerry Tablet OS
Spender CoMo Solution GmbH

OPEN DEVICE LAB NÜRNBERG 020
Gerät BlackBerry Torch
System BlackBerry OS 6.0.0.668
Spender CoMo Solution GmbH

OPEN DEVICE LAB NÜRNBERG 047
Gerät Geeksphone Peak
System Firefox OS
Spender tollwerk GmbH

OPEN DEVICE LAB NÜRNBERG 039
Gerät Google Nexus 7
System Android 4.2.2
Spender Michael Stingl

OPEN DEVICE LAB NÜRNBERG 029
Gerät Google Nexus S
System Android 4.1.2
Spender tollwerk GmbH

OPEN DEVICE LAB NÜRNBERG 003
Gerät HTC Desire
System Android 2.2.2
Spender nipponstar

OPEN DEVICE LAB NÜRNBERG 043
Gerät HTC Flyer
System Android 3.2.1
Spender Cover-Up

OPEN DEVICE LAB NÜRNBERG 053
Gerät HTC HD2
System Windows Mobile 6.5
Spender Christel Krüger

OPEN DEVICE LAB NÜRNBERG 006
Gerät HTC Touch
System Windows
Spender tollwerk GmbH

OPEN DEVICE LAB NÜRNBERG 051
Gerät HTC Touch (O2)
System Windows Mobile 6
Spender tollwerk GmbH

OPEN DEVICE LAB NÜRNBERG 030
Gerät Nokia 6310
System Nokia OS
Spender tollwerk GmbH

OPEN DEVICE LAB NÜRNBERG 049
Gerät O2 Xda comet
System Windows Mobile 6
Spender tollwerk GmbH

OPEN DEVICE LAB NÜRNBERG 032
Gerät O2 Xda mini S
System Windows Mobile 5.1.195
Spender tollwerk GmbH

OPEN DEVICE LAB NÜRNBERG 001
Gerät Samsung Galaxy S II
System Android 4.1.2
Spender tollwerk GmbH

OPEN DEVICE LAB NÜRNBERG 041
Gerät Samsung SGH-F480
System Samsung OS
Spender Philipp Sandner

OPEN DEVICE LAB NÜRNBERG 008
Gerät Viliv N5
System Windows 7 Professional
Spender tollwerk GmbH

By studying the example template as well as the view helper documentation below you should easily be able to write your own custom label template.

TCPDF view helpers

The extension comes with a couple of custom view helpers that you can use in your fluid label templates. If you want to do so, you have to register the appropriate **view helper namespace** in the beginning of your templates like this:

```
{namespace odl=Tollwerk\TwOdl\ViewHelpers}
```

odl:tcpdf

This view helper initializes TCPDF rendering and sets a target medium. It always has to be the outermost element in a label template. Example:

```
<odl:tcpdf width="210" height="297" unit="mm" left="9.75" right="9.75" top="10.63"
bottom="10.63" columns="5" rows="13" offset="{offset}">
    ...
</odl:tcpdf>
```

The possible arguments are as follows:

Attribute:	Data type:	Description:	Default:
width	float	**Page width** in user units (see below). This is the only mandatory attribute.	
height	float	**Page height** (not present = endless paper).	
unit	string	User defined **measurement unit** for pretty much everything (except font sizes).	mm
rows	integer	Number of **label rows**.	
columns	integer	Number of **label columns**.	1
top	float	**Top margin** (vertical distance between page border and first label).	0
bottom	float	**Bottom margin** (vertical distance between page border and last label).	0
left	float	**Left margin** (horizontal distance between page border and first label).	0
right	float	**Right margin** (horizontal distance between page border and last label).	0
spacing	float	General **label spacing** (horizontal and vertical distance between all labels).	0
labelheight	float	**Height of a single label** (necessary only if no page height is given = endless paper).	10
borderwidth	float	Label **border width** (0 = no border).	0
bordercolor	array	Label **border color**, given as fluid array of RGB values.	{0: 0, 1: 0, 2:0} (= black)
offset	integer	**Number of blank labels** at the beginning of the first page. This setting is only useful width non-endless target media (and will be ignored otherwise) when printing on a partially used sticker sheet.	0
fontsize	float	Default **font size** (may be overridden by single text cells).	11

odl:tcpdf.font

This view helper registers a TrueType font file for use with a template. Technically it doesn't matter where you put it in your template, as long as it's a descendant of the odl:tcpdf view helper. However, we suggest you to put it in the beginning. Example:

```
<odl:tcpdf.font family="DroidSans"
fontfile="fileadmin/templates/fonts/DroidSans/DroidSans.ttf"/>
```

The possible arguments are as follows:

Attribute:	Data type:	Description:	Default:
family	string	**Font family name**. This is the name you have to use with the odl:tcpdf.text view helper afterwards.	
fontfile	string	TrueType **font file**. You may give a relative path here (as measured from the site root), an absolute path or a path prefixed with "EXT:my_extension". TCPDF creates an intermediate font definition file for each TrueType font. These definition files are put into the directory *typo3temp/tcpdf-fonts*. Please make sure that the webserver has write access to this directory.	
style	string	The **font style** you would like to associate with the file given font file. May be one of • B (for bold) • I (for italic) • U (for underline) • D (for line through) • O (for overline) or an arbitrary combination / concatenation of these letters. An empty string means "regular" (default).	

odl:tcpdf.image

This view helper renders an image to the current label. Example:

```
<odl:tcpdf.image file="EXT:tw_odl/Resources/Public/Images/odl-nbg-hor-de-white.eps"
width="20" top="2" left="2" right="2" align="left"/>
```

The possible arguments are as follows:

Attribute:	Data type:	Description:	Default:
file	string	**Image file**. You may give a relative path here (as measured from the site root), an absolute path or a path prefixed with "EXT:my_extension". You may use the usual image file formats (JPEG, GIF, PNG) as well as Illustrator files (AI, EPS).	
halign	string	**Horizontal alignment** within the label. Valid values are • L: Left • C: Center • R: Right	L
width	float	**Image width** in measurement units. Zero means "original size" respectively "autocalculate" if height is greater than zero.	0
height	float	**Image height** in measurement units. Zero means "original size" respectively "autocalculate" if width is greater than zero.	0
top	float	**Top margin** within the label (in measurement units).	0
left	float	**Left margin** within the label (in measurement units). Only used if the image is left aligned.	0
right	float	**Right margin** within the label (in measurement units). Only used if the image is right aligned.	0

odl:tcpdf.label

This view helper denotes a single device label. Technically it is not necessary that the view helper encloses the text and image elements on the label, but for semantic reasons we suggest you to do so. Example:

```
<odl:tcpdf.label>
    ...
</odl:tcpdf.label>
```

The view helper has no arguments.

odl:tcpdf.page

This view helper denotes a page of device labels respectively enforces a page break. Technically it is not necessary that the view helper encloses the contained labels, but for semantic reasons we suggest you to do so. If you do not need to explicitly force page breaks you don't have to use this view helper at all. Example:

```
<odl:tcpdf.page>
    ...
</odl:tcpdf.page>
```

The view helper has no arguments.

odl:tcpdf.text

This view helper renders a text cell to the current label. Example:

```
<odl:tcpdf.text x="10" width="27" height="3" fontfamily="DroidSans" fontsize="6"
fontstyle="b" cursor="1" halign="right">{device.manufacturer.name}
{device.name}</odl:tcpdf.text>
```

The possible arguments are as follows:

Attribute:	Data type:	Description:	Default:
text	string	This argument is optional and denotes the **text to be printed** on the label. You may also give the text as children of the view helper element (which seems more semantic).	
width	float	The **width of the text cell** (in measurement units). Zero means automatic width (within the label dimensions).	0
height	float	The **height of the text cell** (in measurement units). Zero means automatic width (within the label dimensions).	0
x	float	The **horizontal position of the text cell** (in measurement units) within the current label. If this argument is not given, the text cell will be inserted at the current horizontal insertion point position (depending on your last insertion).	
y	float	The **vertical position of the text cell** (in measurement units) within the current label. If this argument is not given, the text cell will be inserted at the current vertical insertion point position (depending on your last insertion).	
halign	string	**Horizontal alignment** within the label. Valid values are • L: Left • C: Center • R: Right	L
valign	string	**Vertical alignment** within the label. Valid values are • T: Top • C: Center • R: Right Not sure if this argument works right now ...	T
fontfamily	string	The **font family** the text is to be rendered with. This has to be either a custom defined font family (see odl:tcpdf.font view helper) or one of the predefined fonts: • *courier* : Courier • *courierB* : Courier Bold • *courierBI* : Courier Bold Italic • *courierI* : Courier Italic • *helvetica* : Helvetica • *helveticaB* : Helvetica Bold • *helveticaBI* : Helvetica Bold Italic • *helveticaI* : Helvetica Italic • *symbol* : Symbol • *times* : Times New Roman • *timesB* : Times New Roman Bold • *timesBI* : Times New Roman Bold Italic • *timesI* : Times New Roman Italic • *zapfdingbats* : Zapf Dingbats	times

Attribute:	Data type:	Description:	Default:
fontstyle	string	The **font style** used for the text cell. May be one of • B (for bold) • I (for italic) • U (for underline) • D (for line through) • O (for overline) or an arbitrary combination / concatenation of these letters. An empty string means "regular" (default).	
fontsize	float	**Font size** (in points) to be used.	See **fontsize** argument of the odl:tcpdf view helper
fontcolor	array	**Font color**, given as fluid array of RGB values.	{0: 0, 1: 0, 2:0} (= black)
cursor	integer	**Cursor position** after the insertion of the text cell: • 0: Top / right to the text cell • 1: Beginning of next line • 2: Below the text cell	0

Known problems

There are no known problems right now. Please report any problems to the author.

To-Do list

Future improvements could include:

- Implementation of a JSON API format in conformance with the opendevicelab.com specifications (as soon as available)

If there's something else you think that might be useful for ODL webmasters then please let us know!

ChangeLog

Version:	Changes:
0.20	Language label and example template additions; Added backend fields for browser features and debugging options; Manual corrections
0.1.2	Maintenance release; Added “connectivity” options to device records; manual corrections
0.1.1	Manual corrections
0.1.0	Initial public release to the TYPO3 Extension Repository