

adaptTo()

Apache Sling & FRIENDS TECH MEETUP
BERLIN, 28-30 SEPTEMBER 2015

Apache Sling Rookie Session
Sebastian Schlick, pro!vision GmbH

About the Speaker

- CQ5/AEM6 Developer
- Apache Sling User
- Lead dev pro!vision GmbH



PRO!VISION
SOFTWARE CRAFTSMANSHIP

<http://www.pro-vision.de>



- Apache Sling ?
- Apache Sling Basics
- `org.adaptto.rookie.demo`

Apache Sling ?

What is Apache Sling?

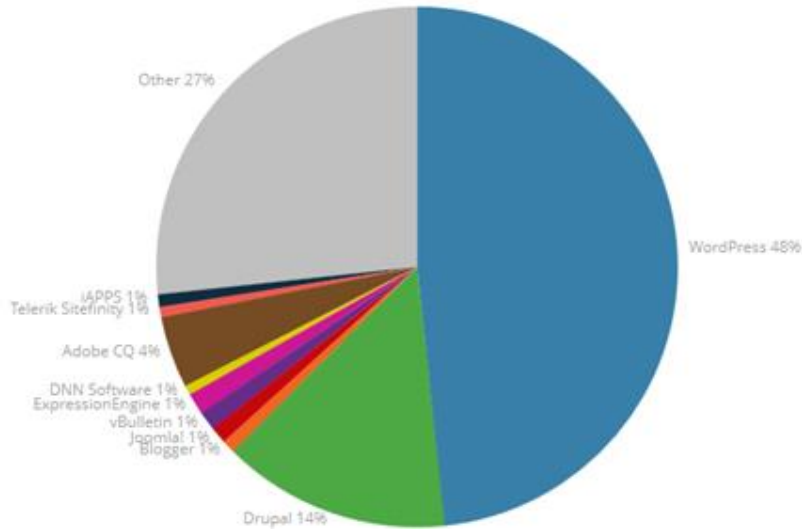
- is a web framework
- a team player
- plays well with Java Content Repository
- delivers the content REST ful

Who is using Apache Sling?

- Only one major product (at least I know of)
 - Adobe CQ / Adobe Experience Manager (AEM)

CMS Usage Statistics

Statistics for websites using CMS technologies



Top in Content Management System

- Week beginning Aug 10th 2015

- Top 10k sites:

- WordPress
- Drupal
- **Adobe CQ**
- WordPress VIP
- **Adobe Experience Manager**
- Sitecore CMS

<http://trends.builtwith.com/cms>

What is Adobe AEM / CQ ?

- Adobe AEM / CQ is leading Enterprise Content Management (2015) according to
 - Gartner's Magic Quadrant
 - The Forrester Wave
 - Our perception

Technical Reasons to look at Apache Sling

- **Open Source**
 - Top level project of the Apache Software Foundation since 2009
- **RESTful**
- **OSGi ready**
- **Geek friendly, mature framework**

- You need only Git, Java and Maven to build it

```
git clone https://github.com/apache/sling.git  
cd sling  
mvn -Dmaven.test.skip=true clean install
```

```
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----
```

Apache Sling Basics

- Building and starting the launchpad

```
cd launchpad/builder
```

```
mvn package
```

```
java -jar target/org.apache.sling.launchpad-8-SNAPSHOT.jar
```

- Browse Apache Sling at <http://localhost:8080>
- Always read the README.txt
 - Consider a work directory outside of the Apache Sling project

What do you get?

- **HTTP** for user interaction
 - <http://localhost:8080>
- **OSGi** to run apps on
 - <http://localhost:8080/system/console>
- **JCR** to persist everything (apps and content)
 - <http://localhost:8080/.explorer.html>

Apache Sling per cURL: POST

HTTP POST: Create a content node (nodes are a [JCR](#) concept, a unit of storage) with cURL

curl is a tool to transfer data from or to a server

- -F, --form <name=content>
- (HTTP) This lets curl emulate a filled-in form in which a user has pressed the submit button. This causes curl to POST data using the Content-Type multipart/form-data according to [RFC 2388](#).

```
$ curl -u admin:admin  
-F"sling:resourceType=foo/bar"  
-F"title=some title" http://localhost:8080/content/mynode
```

<http://Apache Sling.apache.org/documentation/getting-started/discover-Apache Sling-in-15-minutes.html>

Apache Sling per cURL: POST

[+ Query](#)

- /
- |- jcr:system
- |- rep:policy
- |- var
- |- apps
- |- gradient.png
- |- index.html
- |- signup.html
- |- sling-logo.png
- |- sling-test
- |- sling.css
- |- etc
- |- sightly
- |- libs
- |- content
 - |- mynode
- |- system

mynode

[delete this node](#)

[/content/mynode](#) (JCR identifier: f03a4081-19e7-402c-8845-6f2d2e91eaed)

add new child node

Name hint sling:resourceType jcr:primaryType [new child node](#)

general

jcr:primaryType

mixin types

jcr:mixinTypes [add](#)

properties

sling:resourceType [String] [update property](#) [delete property](#)

title [String] [update property](#) [delete property](#)

add property

Name Multi? Type Value [add property](#)

Apache Sling per: cURL GET

HTTP GET: The resulting node can be seen also with cURL:

```
$ curl -u admin:admin http://localhost:8080/content/mynode.json
```

```
{"title":"some title","sling:resourceType":"foo/bar","jcr:primaryType":"nt:unstructured"}
```

<http://sling.apache.org/documentation/getting-started/discover-sling-in-15-minutes.html>

Apache Sling per cURL: PUT a render script

HTTP POST: Create two Apache Sling folders with cURL:

```
curl -u admin:admin -F"jcr:primaryType=sling:Folder" http://localhost:8080/apps/foo
```

```
curl -u admin:admin -F"jcr:primaryType=sling:Folder" http://localhost:8080/apps/foo/bar
```

HTTP PUT: upload the script

```
curl -u admin:admin -T html.html http://localhost:8080/apps/foo/bar/html.html
```

Any **http get** with protocol **html** pointing to a **node** with `sling:resourceType` of `foo/bar` is now rendered with the **html.html**

<http://sling.apache.org/documentation/getting-started/discover-sling-in-15-minutes.html>

HTML.HTML renders HTML requests using the sightly HTML

```
<h1 data-sly-test="{properties.title}">
  <a href="{properties.path}.html">
    {properties.title}
  </a>
</h1>
```

XML.esp renders HTML requests using esp script

...

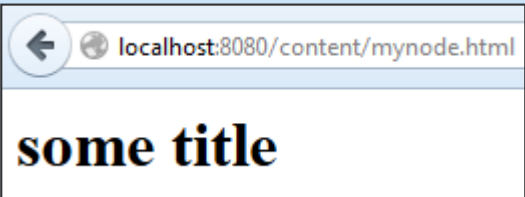


- Expressions, Operators
- Context
- `data-sly-test`
- `data-sly-use`
- `<!--/* A Sightly Comment */-->`
- `data-sly-list`
- `data-sly-resource`
- `data-sly-call`
- `data-sly-unwrap`
- ...

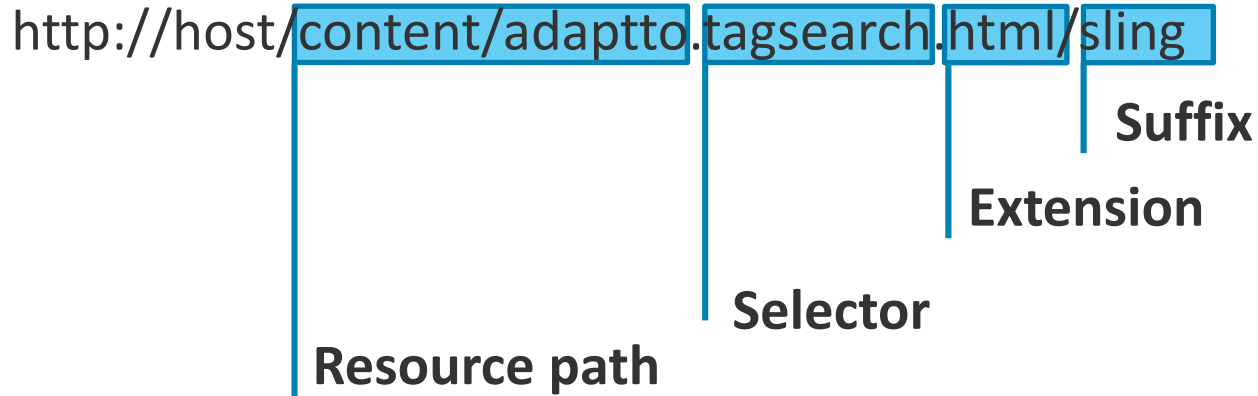


http://dev.day.com/content/ddc/en/gems/introduction-to-sightly/_jcr_content/par/download/file.res/Sightly%20Component%20Development.pdf

Apache Sling per cURL: PUT a render script

HTTP	Apache Sling	JCR
<p>http://localhost:8080/content/mynode.html</p> 	<p>magic</p>	<pre> /content/mynode sling:resourceType [String] : foo/bar title [String] : some title /apps/foo/bar/html.html <h1 data-sly-test="\${currentPage.title}"> \${currentPage.title} </h1> </pre>

Apache Sling URL decomposition



<https://sling.apache.org/documentation/the-sling-engine/url-decomposition.html>

Apache Sling script and servlet rendering

- Java code, using OSGi services and servlets
- Server side:
 - JSP, JavaScript, Freemarker, Groovy, Java, JST, Python, Ruby, Scala, Velocity, Xproc, Thymeleaf, **Sightly**, ...
- Client side rendering
 - using Apache Sling's default JSON rendering
- Or any combination of the stated methods

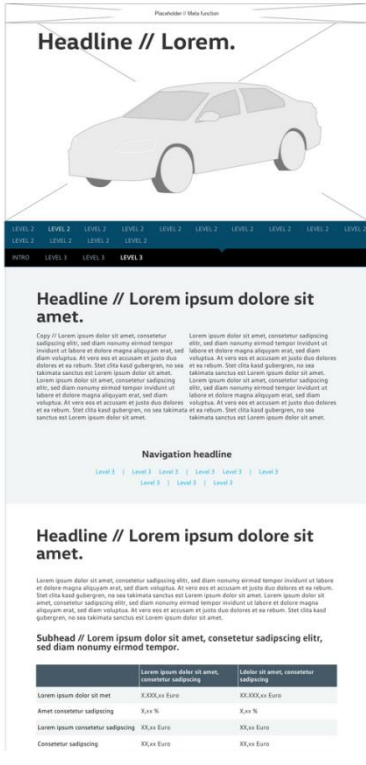
Apache Sling script and servlet rendering

HTTP	Apache Sling	JCR /app
http://localhost:8080/content/mynode.json (xml/txt)	default	-
http://localhost:8080/content/mynode.html	custom	/apps/foo/bar/html.esp
http://localhost:8080/content/mynode.printview.html	custom	/apps/foo/bar/printview.html.html
http://localhost:8080/content/mynode.title.png	custom	probably servlet
http://localhost:8080/content/mynode.result.zip/search?param=true	custom	probably servlet

more Apache Sling features



- JCR Nodes as Resources
- Resource Tree (SuperTypes)
- Apache Sling Models + Sightly
- Security
- eventing and scheduling
- et cetera



- [m310](#) - Editorial stage (1)
- [m503](#) - Quick Access (1)

Navigation Area

- [m510](#) - Tab bar (1)

Intro area (1-n)

- [m101](#) - Introcopy (1)
 - [m107](#) - Link list (0-n)
- [m511](#) - Anchor bar (0-1)

Content Section (1-n)

- [m102](#) - Copy (0-n)
 - [m107](#) - Link list (0-n)
- [m103](#) - Copy Media (0-n)
 - [m107](#) - Link list (0-n)
- [m104](#) - Table (0-n)
- [m105](#) - Media Gallery (0-n)
- [m108](#) - Accordion (0-n)
- [m109](#) - Tab Box (0-n)
- [m211](#) - Next Step Teaser (0-1)
- [m212](#) - Shopping Teaser (0-1)
- [m201](#) - Big Image Teaser (0-n)

[m520](#) - Footer (1)

- Templating (CI/CD)
- Components
 - Reusability
- Integration
- Architecture
 - Separation of Design, Content, Functionality

org.adaptto.rookie.demo

Join us on Day 2

■ 15:15 - 18:00 Rookie-Hands-On

12:30 - 13:30	Lunch	
13:30 - 14:00	AWS Implementation	Adam Pazik
14:00 - 14:15	Break	
14:15 - 15:00	Building AEM applications with wcm.io	Stefan Seifert
15:00 - 15:15	Break	
15:15 - 18:00	Playground Session: Hacking, Testing & Rookie-Hands-On	
18:00 - 23:00	Evening Event - Buffet, Hacking and Socializing	

- With running Apache Sling Launchpad:

```
git clone https://github.com/adapitto-conf/2015-sling-rookie-session.git  
cd 2015-sling-rookie-session  
mvn -Dsling.url=http://localhost:8080 clean install sling:install
```

```
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----
```

- <http://localhost:8080/content/adapitto.html>

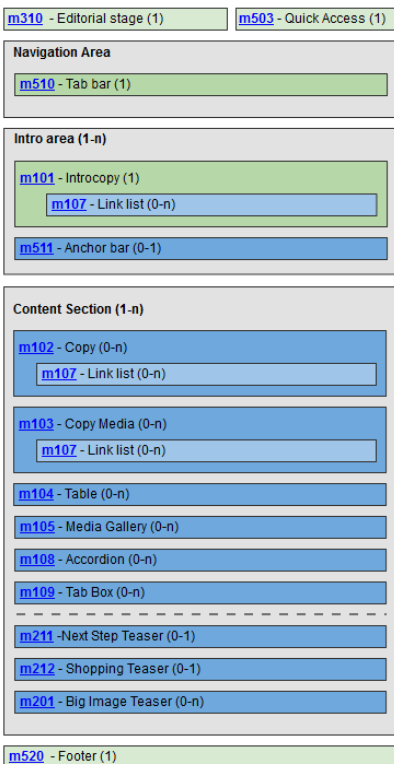
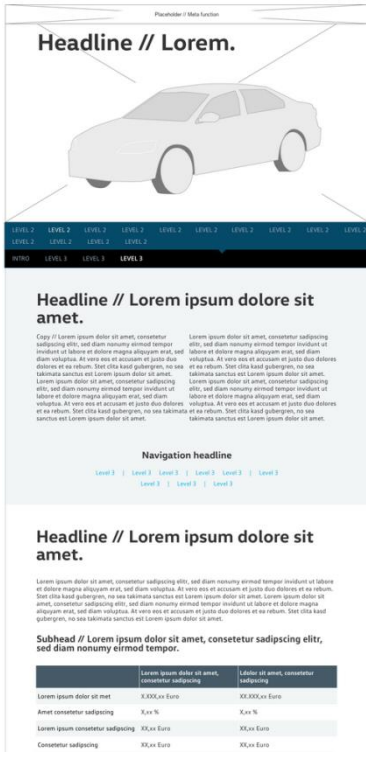
rookie.demo overview

User	CMS	File	Apache Sling / Sightly Feature
Homepage	Template	components/index/html.html	data-sly-include, data-sly-resource
Search	Function	components/index/tagsearch.html	Apache Sling models xpathQuery
Talk	Template	components/talk/vcs.html	data-sly-use.contentType, Apache Sling models
Comment list	Component	components/social/commentList/html.html	data-sly-include
New comment	Function	components/social/newComment/html.html	JCR Post
Like	Function	LikeMe.java	Apache Sling Servlet
	Background Job	CommentCleanUpCronJob.java	OSGi Component

```
<!doctype html>
<html>
  <!--/* Include html_head script inherited from super component "common"
  */-->
  <head data-sly-include="html_head.html"></head>

  <body>
    <h1>adaptTo() Rookie Slightly Demo</h1>

    <!--/* Include childlist via selector view inherited from super
    component "common" */-->
    <h2>Overview</h2>
    <div data-sly-resource="${'.' @ selectors='childlist'}"></div>
```



- Templates (CI/CD)
- Components
 - Reusability / Caching
- Integration
- Architecture
 - Design, Content, Functionality

```
<!doctype html>
<html>
  <!--/* Include html_head script inherited from super component "common"
  */-->
  <head data-sly-include="html_head.html"></head>

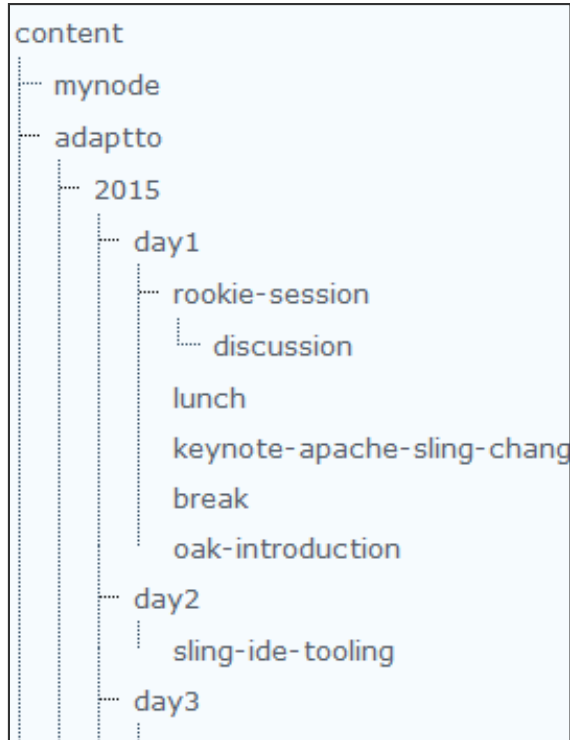
  <body>
    <h1>adaptTo() Rookie Slightly Demo</h1>

    <!--/* Include childlist via selector view inherited from super
    component "common" */-->
    <h2>Overview</h2>
    <div data-sly-resource="${'.' @ selectors='childlist'}"></div>
```



```
<!--/* Iterate over all child resources from current resource */-->
<ul data-sly-list.child="{resource.listChildren}">
  <li>
    <a href="{child.path}.html">{child.valueMap.jcr:title}</a>
  </li>
</ul>
```

Content is king



- Information architecture
- Component hierarchy
- Templates
- Functions

Next steps (final slide)

- Play with the demo
- Talk to me
- Join us on Day 2 - Rookie-Hands-On
- Build your own Apache Sling app