PRO!VISION
SOFTWARE CRAFTSMANSHIP

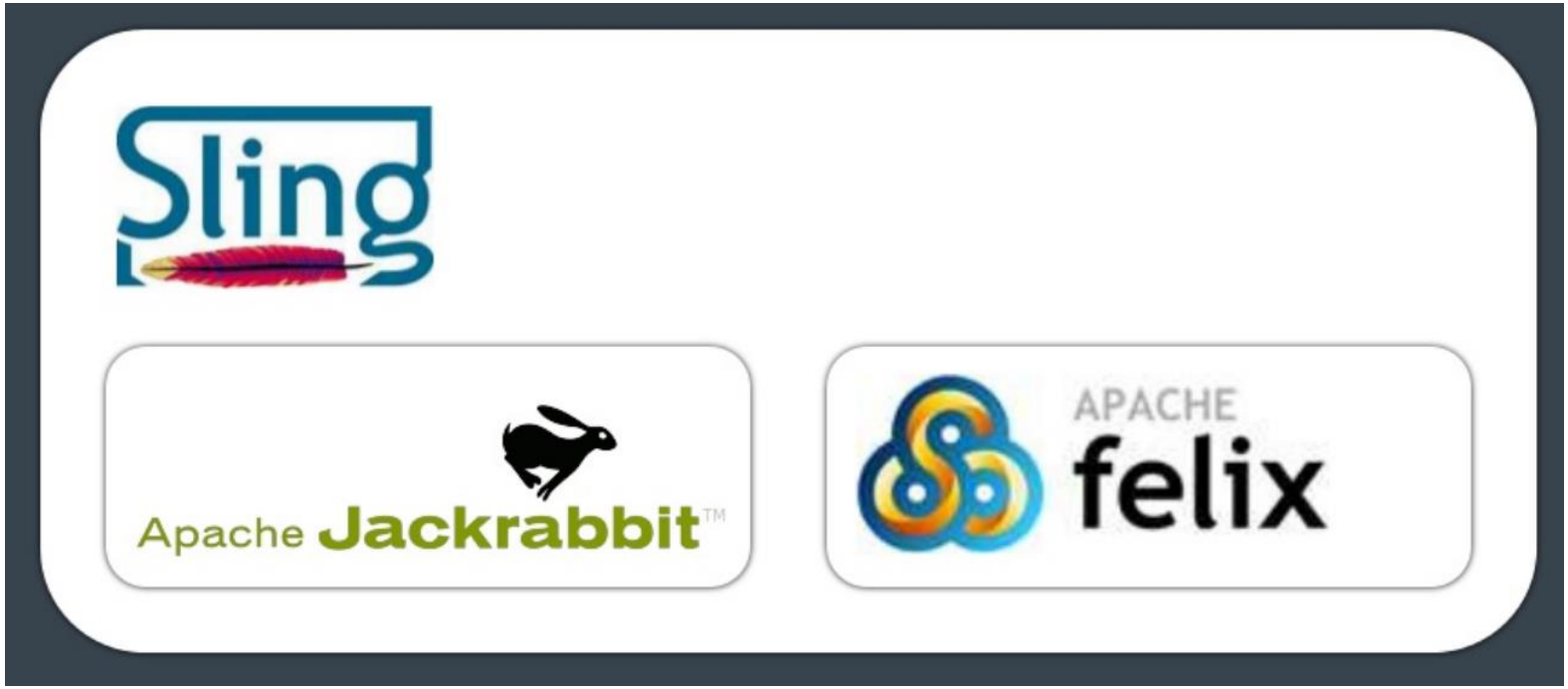
Webanwendungen RESTful mit Apache Sling

Sebastian Schlick, Dominik Süß

Präsentiert auf der BED Con

http://www.bed-con.de

Inhalt



REST und Webapplication Frameworks

Serverseitige, zustandsverwaltende Systeme vs. „einfache Lösungen“

Webapplikationen (generell)

- Webapplikation
  - Anwendung oder Dienst im Webbrowser
- **Vorteile**
  - Keine Client-Installation
  - Viele Endgeräte
- **Merkmale**
  - HTTP als etabliertes Protokoll
  - Browser als existente Nutzerschnittstelle
    - Darstellung
    - Basisfunktionen

Warum eigentlich REST?

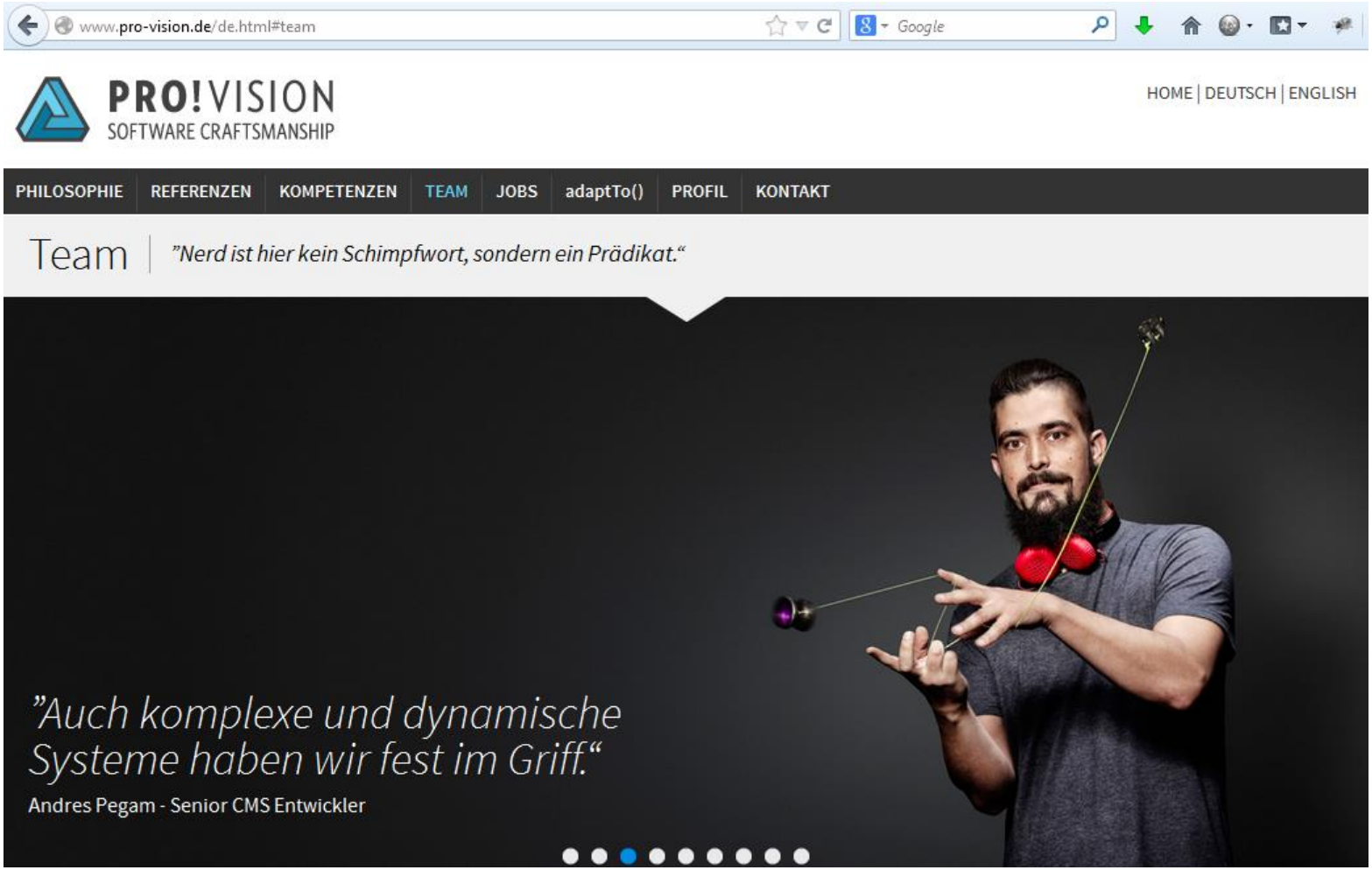
- JSF, .Net, Wicket, Tapestry, etc.
- Nehmen dem Entwickler viel Arbeit ab. Lösen bestimmte „Probleme“

- Wicket ist wirklich cool
- Aber:

Die Realität

Warum ist das oft so? Muss das so sein?

Besser: Zustand in der URL



Webframeworks & REST

- **Grundprobleme**
  - Klassisches MVC Frameworks adressieren oft Controler
  - Transaktionale Funkitionalität basiert häufig auf Sessions
  - (Relationales Datenmodell als Ausgangspunkt von Applikationsdesign)
  - Requestparameter als Standard Parametrisierung in (Java) Servlet-API

- **Lösungsansatz der meisten Frameworks**
  - REST durch Vanity-URLs (Mapping)
    - Persistierter State in einer (manuell) definierten URL

UnRESTful URL

http://localhost/schedule.jsp?event=1&year=2013&id=1&format=JSON

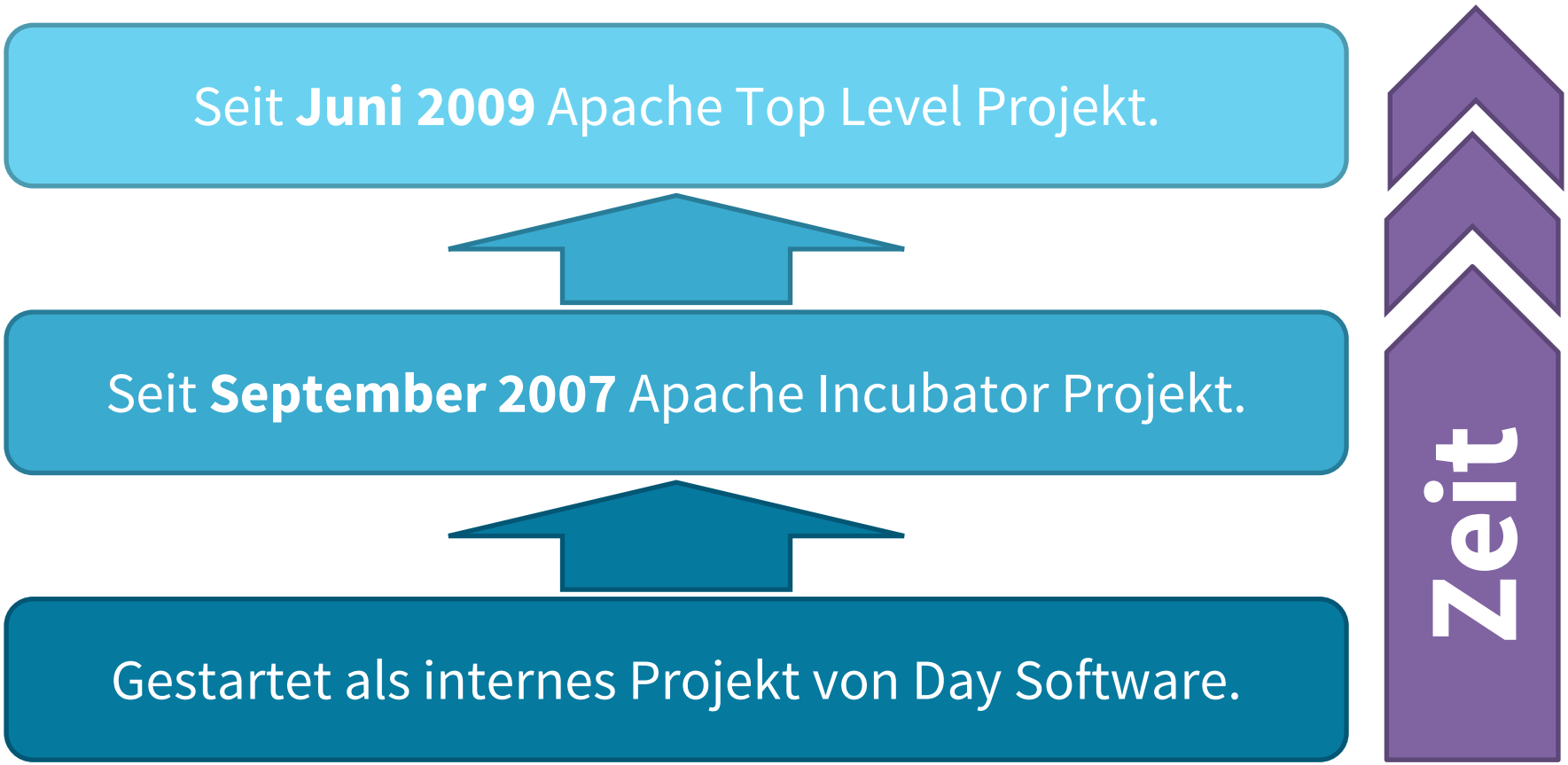
RESTful URL

http://localhost:4502/content/adaptto/2013/day1/rookie-session.json

Apache Sling

RESTful Webapplication Framework

Sling - Geschichte



Apache Sling

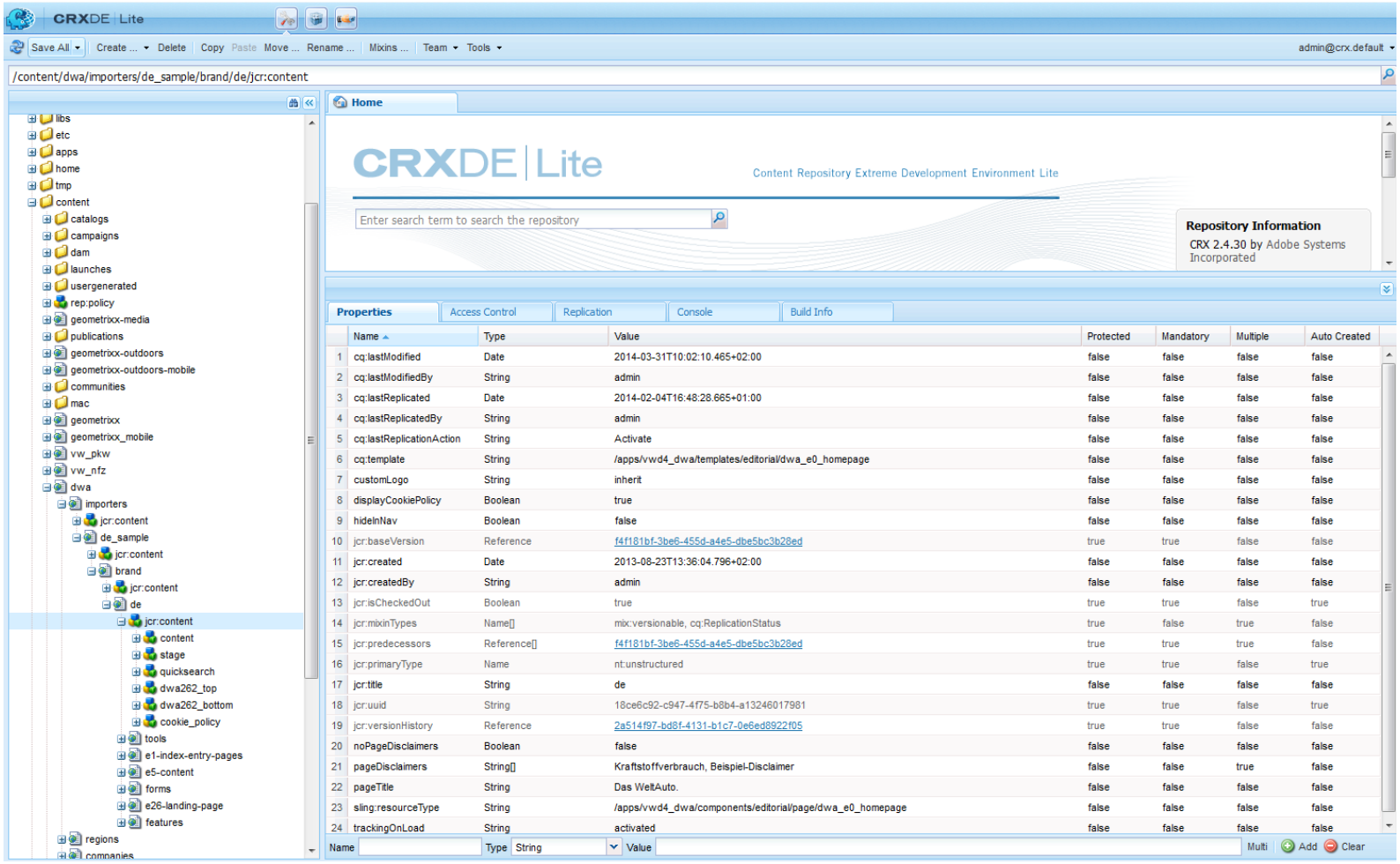
- Web Application Framework
- HTTP Requests RESTful verarbeiten
- Verschiedene Resource Provider für Datenablage
  - darunter als Hauptpersistenz: Java Content Repository (JCR-283)
- Scripts oder Servlets zum Verarbeiten von Content
- OSGi zum Deployment von Modulen zur Laufzeit

Content Repository (kurz)

Java Content Repository (JCR-283)

Content Repository

Content Repository (Beispiel)



Content Repository (Beispiel)

jcr:baseVersion	Reference	f4f181bf-3be6-455d-a4e5-dbe5bc3b28ed
jcr:created	Date	2013-08-23T13:36:04.796+02:00
jcr:createdBy	String	admin
jcr:isCheckedOut	Boolean	true
jcr:mixinTypes	Name[]	mix:versionable, cq:ReplicationStatus
jcr:predecessors	Reference[]	f4f181bf-3be6-455d-a4e5-dbe5bc3b28ed
jcr:primaryType	Name	nt:unstructured
jcr:title	String	de
jcr:uuid	String	18ce6c92-c947-4f75-b8b4-a13246017981
jcr:versionHistory	Reference	2a514f97-bd8f-4131-b1c7-0e6ed8922f05
noPageDisclaimers	Boolean	false
pageDisclaimers	String[]	Kraftstoffverbrauch, Beispiel-Disclaimer
pageTitle	String	Das WeltAuto.
sling:resourceType	String	/apps/vwd4_dwa/components/editorial/page/dwa_e0_homepage
trackingOnLoad	String	activated

Resourcen-orientiertes Denken

Sling

Motto:

Alles
ist eine
Resource

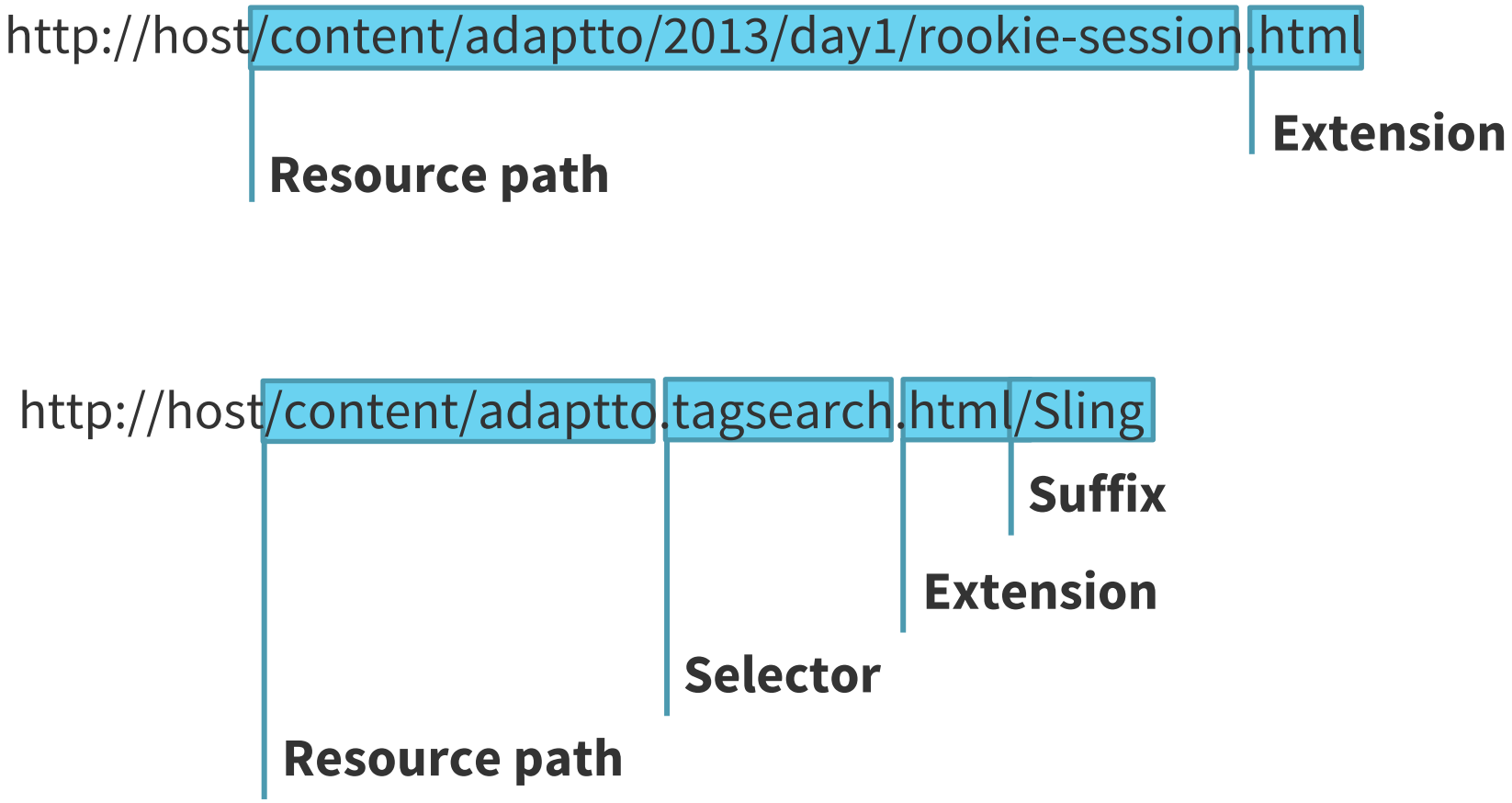
Apache Sling und URLs

Representational State Transfer (REST)

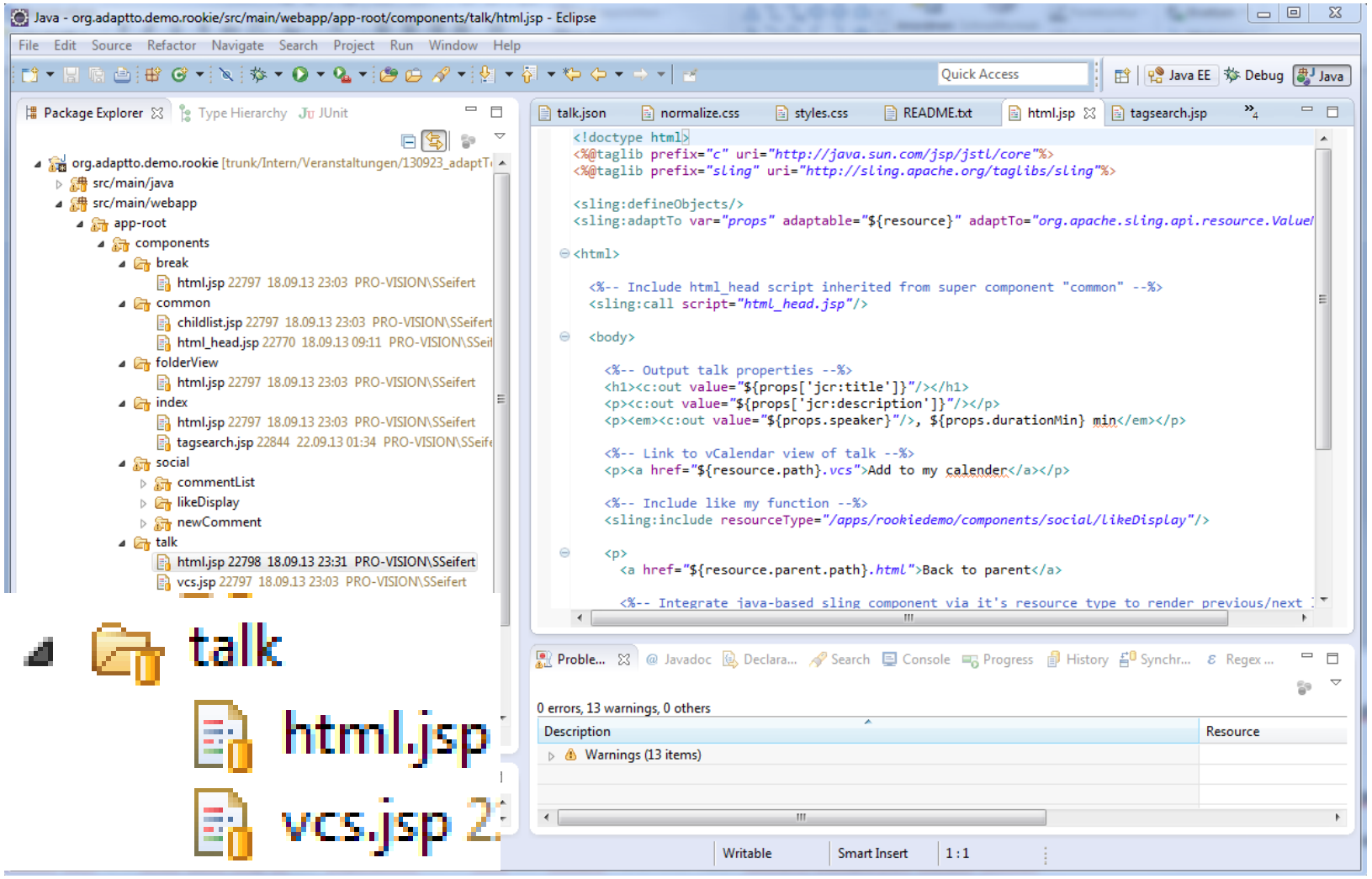
Sling basiert auf dem REST Paradigma

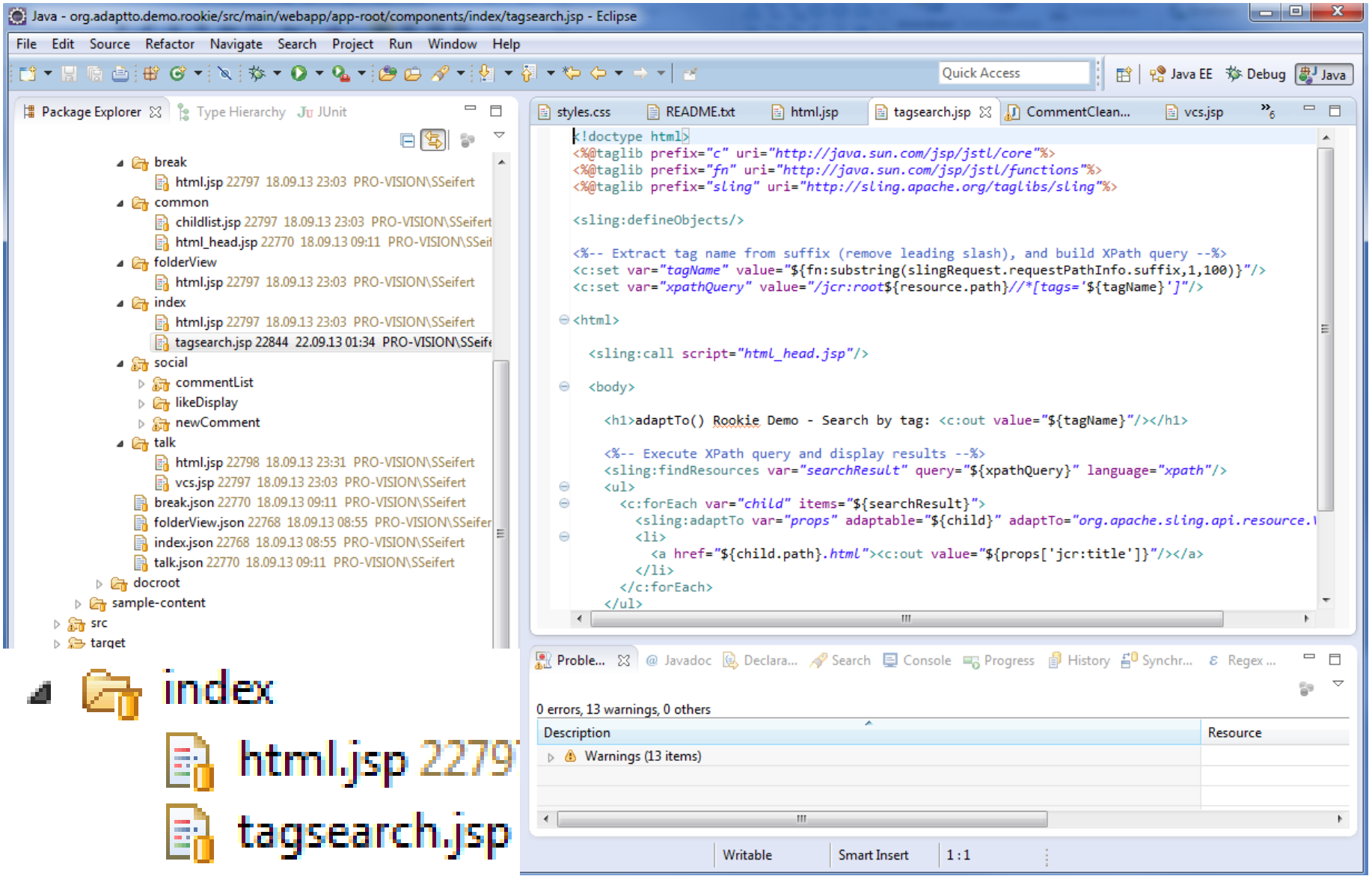
- Denken in adressierbaren Resourcen
- Auswahl der Representation (HTML, JSON, PDF...) über URI
- HTTP-Request-Methoden:
  GET + POST + PUT + DELETE
- State als Bestandteil des Request

Beispiele zur URL Dekomposition



org.adaptto.demo.rookie




Java - org.adaptto.demo.rookie/src/main/webapp/app-root/components/index/tagsearch.jsp - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java EE
Debug
Java
Package Explorer
Type Hierarchy
JUnit
break
html.jsp 22797 18.09.13 23:03 PRO-VISION\SSeifert
common
childlist.jsp 22797 18.09.13 23:03 PRO-VISION\SSeifert
html_head.jsp 22770 18.09.13 09:11 PRO-VISION\SSeifert
folderView
html.jsp 22797 18.09.13 23:03 PRO-VISION\SSeifert
index
html.jsp 22797 18.09.13 23:03 PRO-VISION\SSeifert
tagsearch.jsp 22844 22.09.13 01:34 PRO-VISION\SSeifert
social
commentList
likeDisplay
newComment
talk
html.jsp 22798 18.09.13 23:31 PRO-VISION\SSeifert
vcs.jsp 22797 18.09.13 23:03 PRO-VISION\SSeifert
break.json 22770 18.09.13 09:11 PRO-VISION\SSeifert
folderView.json 22768 18.09.13 08:55 PRO-VISION\SSeifert
index.json 22768 18.09.13 08:55 PRO-VISION\SSeifert
talk.json 22770 18.09.13 09:11 PRO-VISION\SSeifert
docroot
sample-content
src
target
styles.css
README.txt
html.jsp
tagsearch.jsp
CommentClean...
vcs.jsp
<!doctype html>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions"%>
<%@taglib prefix="sling" uri="http://sling.apache.org/taglibs/sling"%>
<sling:defineObjects/>
<%-- Extract tag name from suffix (remove leading slash), and build XPath query --%>
<c:set var="tagName" value="${fn:substring(slingRequest.requestPathInfo.suffix,1,100)}"/>
<c:set var="xpathQuery" value="/jcr:root${resource.path}//*[tags='${tagName}']"/>
<html>
<sling:call script="html_head.jsp"/>
<body>
<h1>adaptTo() Rookie Demo - Search by tag: <c:out value="${tagName}"/></h1>
<%-- Execute XPath query and display results --%>
<sling:findResources var="searchResult" query="${xpathQuery}" language="xpath"/>

<c:forEach var="child" items="${searchResult}">
<sling:adaptTo var="props" adaptable="${child}" adaptTo="org.apache.sling.api.resource.

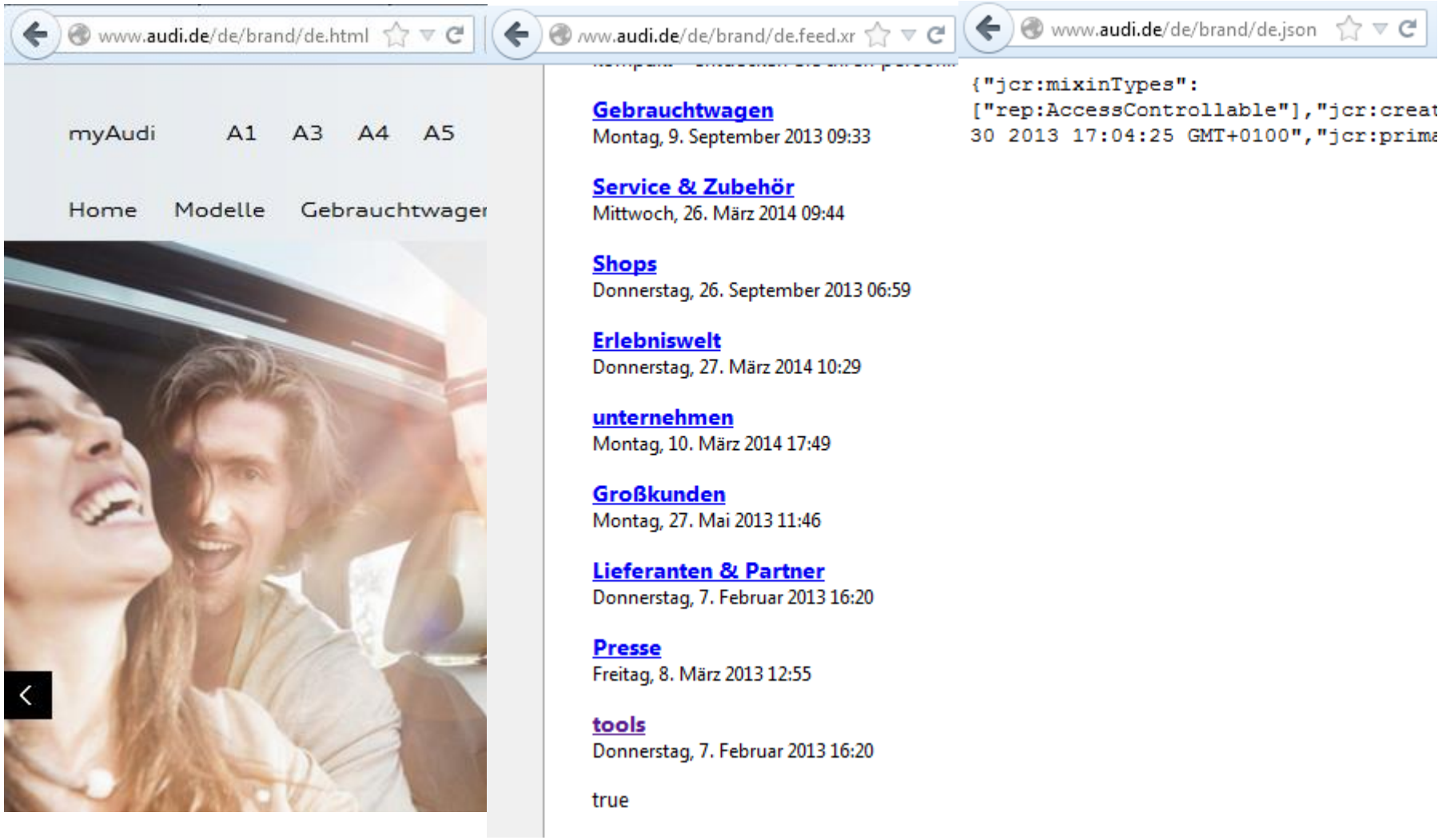
<c:out value="${props['jcr:title']}"/>

</c:forEach>

Proble...
Javadoc
Declara...
Search
Console
Progress
History
Synchr...
Regex ...
0 errors, 13 warnings, 0 others
Description
Resource
Warnings (13 items)
Writable
Smart Insert
1 : 1

index
html.jsp 2279
tagsearch.jsp

Einfaches Werkzeug, Inhalte schnell auszuliefern…



Sling Default JSON/XML Mapping

http://localhost:4502/**content/adaptto/2013/day1/rookie-session.json**


```
{
    durationMin: 135,
    jcr:description: "Basic introduction to JCR and Sling",
    speaker: "Andres Pegam, Stefan Seifert",
    sling:resourceType: "/apps/rookiedemo/components/talk",
    jcr:createdBy: "admin",
    jcr:title: "Rookie-Session: JCR & Sling",
  - tags: [
        "Sling",
        "JCR"
    ],
    startDate: "20130923T081500Z",
    jcr:created: "Sat Sep 21 2013 23:23:16 GMT+0200",
    endDate: "20130923T103000Z",
    jcr:primaryType: "sling:OrderedFolder"
}
```


http://localhost:4502/**content/adaptto/2013/day1/rookie-session.xml**


```
- <rookie-session jcr:primaryType="sling:OrderedFolder" durationMin="135" endDate="20130923T103000Z"
  jcr:created="2013-09-21T23:23:16.837+02:00" jcr:createdBy="admin" jcr:description="Basic introduction to JCR and Sling"
  jcr:title="Rookie-Session: JCR & Sling" sling:resourceType="/apps/rookiedemo/components/talk" speaker="Andres Pegam, Stefan Seifert"
  startDate="20130923T081500Z">
   + <discussion jcr:primaryType="sling:OrderedFolder" jcr:created="2013-09-21T23:23:16.837+02:00" jcr:createdBy="admin">
     </discussion>
  </rookie-session>
```

Sonstige Features von Sling

- Open Source
- Nutzt OSGi Services
- Resource Events (auch verteilt)
- Sling Job Distribution
- Diverse JCR Features

- Fazit: Generell gut für massive Content Distribution

Wie anfangen?

Der Weg zur Beispielapplikation org.adaptto.demo.rookie

Sling setup

Java 7, Maven 2

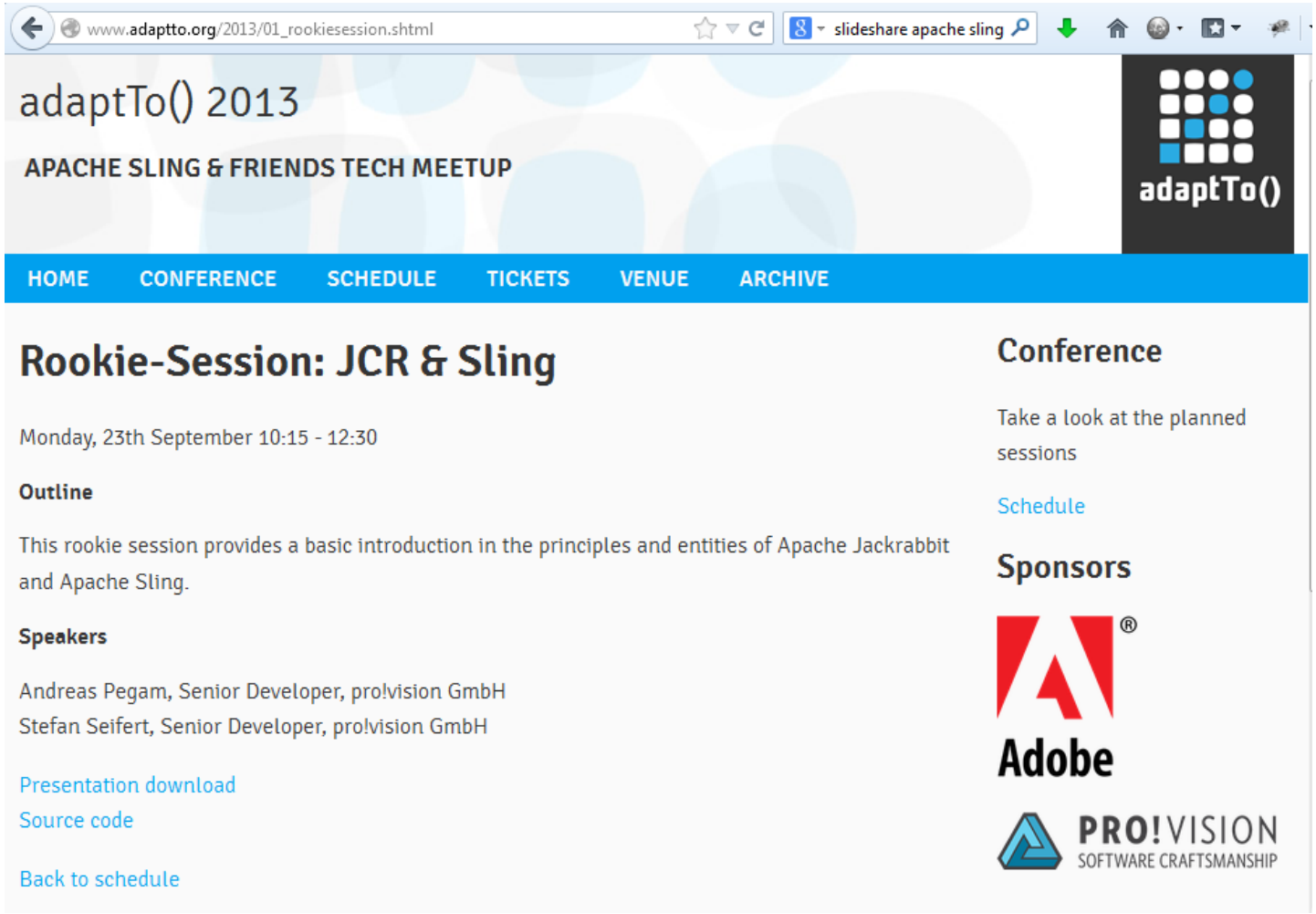
```
$ svn checkout http://svn.apache.org/repos/asf/sling/trunk sling
oder
$ git clone https://github.com/apache/sling.git

cd sling
mvn clean install   // kann dauern

cd launchpad\builder
mvn clean install
// das standalone jar aus dem target irgendwohin bewegen, dann
java -jar org.apache.sling.launchpad-7-SNAPSHOT-standalone.jar
```

Upload commons-lang3:
http://localhost:8080/system/console/bundles
http://mvnrepository.com/artifact/org.apache.commons/commons-lang3/3.0

http://www.adaptto.org/2013/01_rookiesession.shtml

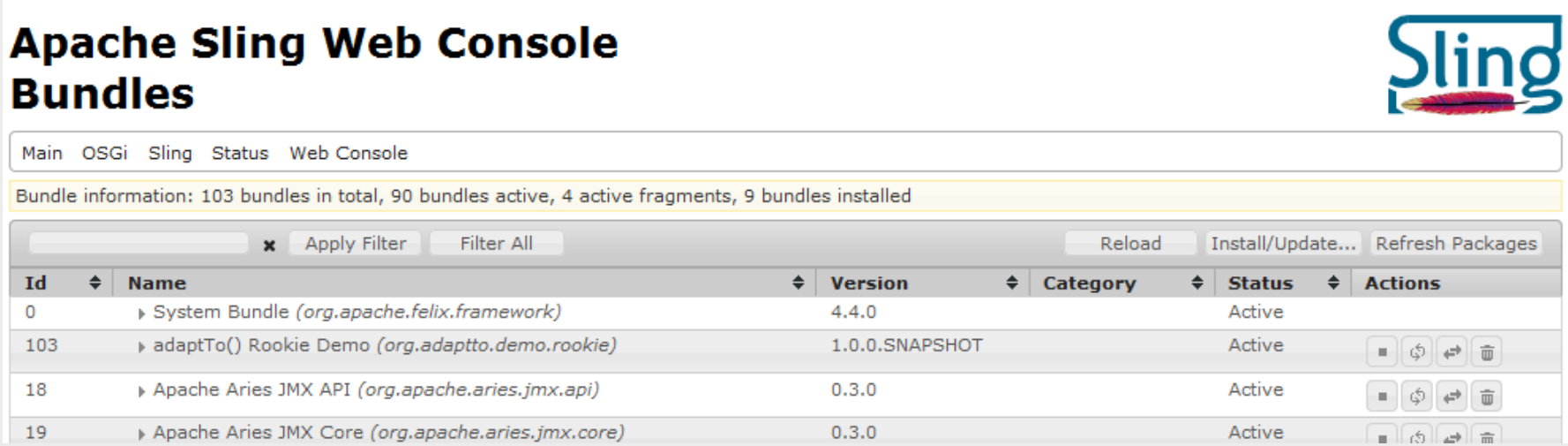


Installation der Beispielapplikation

Sourcen entpacken,

In der pom.xml den sling port setzen

```
<sling.url>http://localhost:8080
mvn clean install
mvn sling:install
```



Noch mehr Beispiele

org.adaptto.demo.rookie

Writing data to repository with Sling

- SlingPostServlet
  - Supports writing back data to repository (resources) without custom code
  - Maps POST parameters to property names
  - Supports additional logic that can be triggered by special parameter names
  - Make sure POST is only possible with proper authentication (delegates to repository authentication by default)

JSP Example: Post new discussion entry

JSP Script in JCR: **/apps/rookiedemo/components/social/newComment/html.jsp**

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@taglib prefix="sling" uri="http://sling.apache.org/taglibs/sling"%>
<sling:defineObjects/>

<%-- Post to "*" which means create a new resource with unique name --%>
<form action="${resource.path}/discussion/*" method="POST" enctype="multipart/form-data">

  <%-- Define resource type for new node --%>
  <input type="hidden" name="sling:resourceType" value="/apps/rookiedemo/components/social/comment"/>
  <%-- Ensure proper charset encoding --%>
  <input type="hidden" name="_charset_" value="UTF-8"/>
  <%-- Redirect to main view after writing content --%>
  <input type="hidden" name=":redirect" value="${resource.path}.html"/>

  <%-- Post to properties "author" and "text" in repository --%>
  <table>
    <tr>
      <td>Your name:</td>
      <td><input type="text" name="author"/></td>
    </tr>
    <tr>
      <td>Comment:</td>
      <td><textarea name="text"/></textarea></td>
    </tr>
    <tr>
      <td></td>
      <td><input type="submit" value="Add comment"/></td>
    </tr>
  </table>

</form>
```

Servlet Example: Render discussion entry

Servlet in OSGi Bundle: **org.adaptto.demo.rookie.components.DiscussionComment**

```
@SlingServlet(resourceTypes="/apps/rookiedemo/components/social/comment")
public class DiscussionComment extends SlingSafeMethodsServlet {

  protected void doGet(SlingHttpServletRequest pRequest, SlingHttpServletResponse pResponse)
      throws ServletException, IOException {
    Writer out = pResponse.getWriter();

    // read properties via Sling API
    ValueMap props = ResourceUtil.getValueMap(pRequest.getResource());
    String author = props.get("author", "Anonymous");
    Date created = props.get("jcr:created", Date.class);
    String text = props.get("text", "");

    // output comment as HTML
    out.write("<p>");
    out.write("<em>" + StringEscapeUtils.escapeHtml4(author)
        + " (" + DateFormat.getDateTimeInstance().format(created) + ")</em></br>");
    out.write(StringEscapeUtils.escapeHtml4(text));
    out.write("</p>");

  }

}
```

The example shows the usage of Sling API for reading content from resources. Less verbose and more convenient than directly using JCR API.

Servlet Example: Custom POST, Sling CRUD

Servlet in OSGi Bundle: **org.adaptto.demo.rookie.components.LikeMe**

```
@SlingServlet(resourceTypes="/apps/rookiedemo/components/talk", selectors="likeme", methods="POST")
public class LikeMe extends SlingAllMethodsServlet {

  protected void doPost(SlingHttpServletRequest pRequest, SlingHttpServletResponse pResponse)
      throws ServletException, IOException {
    updateLikeCounter(pRequest);
    // return to main view
    pResponse.sendRedirect(pRequest.getResource().getPath() + ".html");
  }

  private void updateLikeCounter(SlingHttpServletRequest pRequest) throws PersistenceException {
    ValueMap props = ResourceUtil.getValueMap(pRequest.getResource());

    // check if a user with this ip address has already liked this
    String ipAddress = pRequest.getRemoteAddr();
    String[] likedAddresses = props.get("likedAddresses", new String[0]);
    if (ArrayUtils.contains(likedAddresses, ipAddress)) {
      return;
    }

    // increment like counter and store ip address
    ValueMap writeProps = pRequest.getResource().adaptTo(ModifiableValueMap.class);
    writeProps.put("likes", writeProps.get("likes", 0L) + 1);

    List<String> updatedLikedAddresses = new ArrayList<>(Arrays.asList(likedAddresses));
    updatedLikedAddresses.add(ipAddress);
    writeProps.put("likedAddresses", updatedLikedAddresses.toArray());

    // save to repository
    pRequest.getResourceResolver().commit();
  }

}
```

OSGi example: Scheduled background job

Component in OSGi Bundle: **org.adaptto.demo.rookie.services.CommentCleanUp**

```
@Component(immediate = true, metatype = true, label = "adaptTo() Rookie Demo Comment Cleanup Service")
@Service(value = Runnable.class)
public class CommentCleanUp implements Runnable {
  @Property(value = "0 0/15 * * * ?", label = "Scheduler Expression")
  private static final String PROPERTY_CRON_EXPRESSION = "scheduler.expression";
  @Reference
  ResourceResolverFactory mResourceResolverFactory;

  public void run() {
    ResourceResolver adminResolver = mResourceResolverFactory.getAdministrativeResourceResolver(null);

    // fire query to get all comment nodes
    Iterator<Resource> comments = adminResolver.findResources("SELECT * "
        + "FROM [sling:OrderedFolder] "
        + "WHERE ISDESCENDANTNODE([/content/adaptto]) "
        + "AND [sling:resourceType]='/apps/rookiedemo/components/social/comment'", Query.JCR_SQL2);

    // iterate over all comments and remove those that have empty text
    while (comments.hasNext()) {
      Resource comment = comments.next();
      ValueMap props = ResourceUtil.getValueMap(comment);
      if (StringUtils.isEmpty(props.get("text", String.class))) {
        adminResolver.delete(comment);
      }
    }

    // save changes to repository
    if (adminResolver.hasChanges()) {
      adminResolver.commit();
    }
    adminResolver.close();
  }
}
```

- Apache Sling ist noch wesentlich umfangreicher
- Und wird fortlaufend ergänzt

- I18n
- Eventing & Jobs
- Chron Jobs & Scheduling
- User & Groupmanagement
- Caching
- Model Objects
- Tenants
- Cluster
- Replication
- Healthcheck

Links

- http://sling.apache.org
- Sling Mailinglisten
  - http://sling.apache.org/project-information.html#mailing-lists
- JCR Browser
  - http://svn.apache.org/repos/asf/sling/trunk/contrib/explorers/gwt
- ***adaptTo() 2014 (inkl. Archiv)***
  - http://www.adaptto.org
- Samplecode
  - http://www.adaptto.org/2013/01_rookiesession.shtml

PRO!VISION
SOFTWARE CRAFTSMANSHIP

adaptTo()

APACHE SLING & FRIENDS TECH MEETUP

Kulturbrauerei Berlin, 22-24 SEPTEMBER 2014