



# Webanwendungen RESTful mit Apache Sling

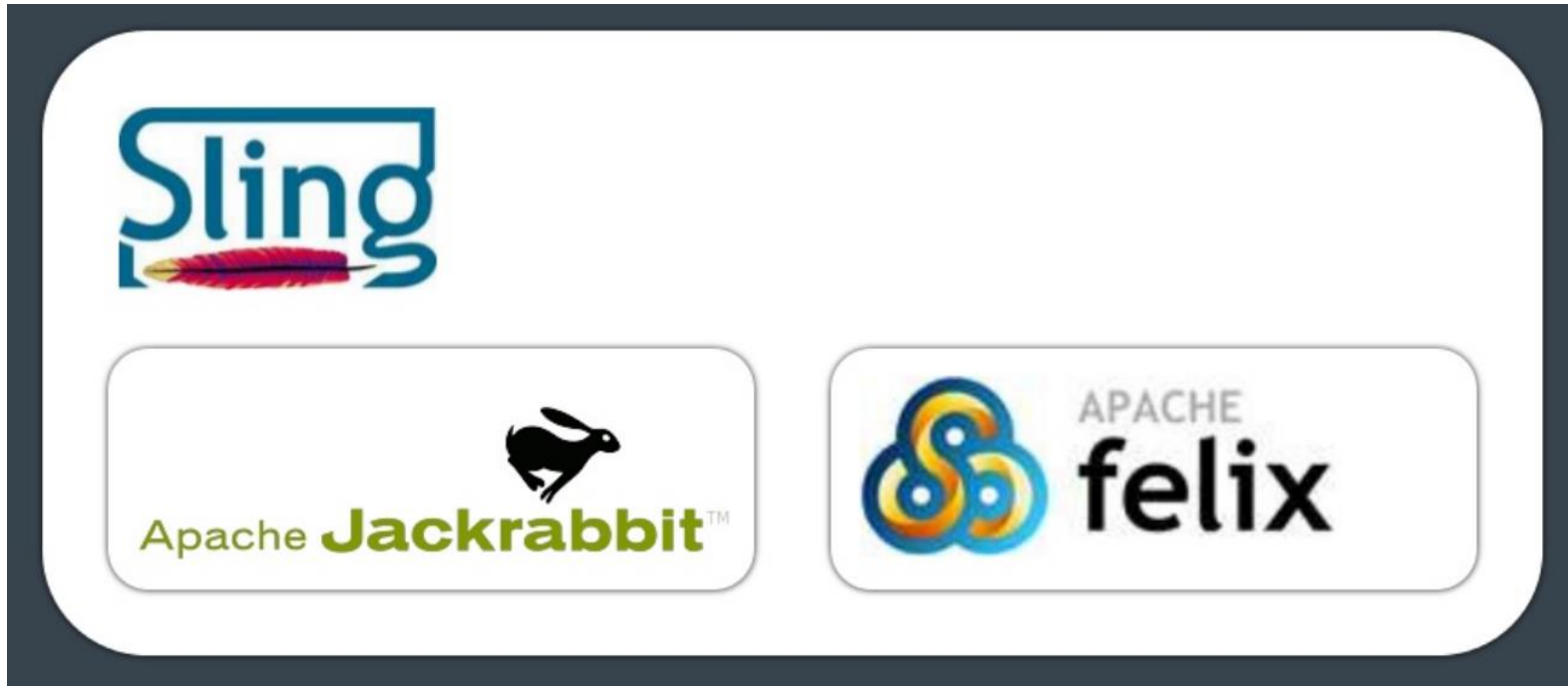
Sebastian Schlick, Dominik Süß

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?

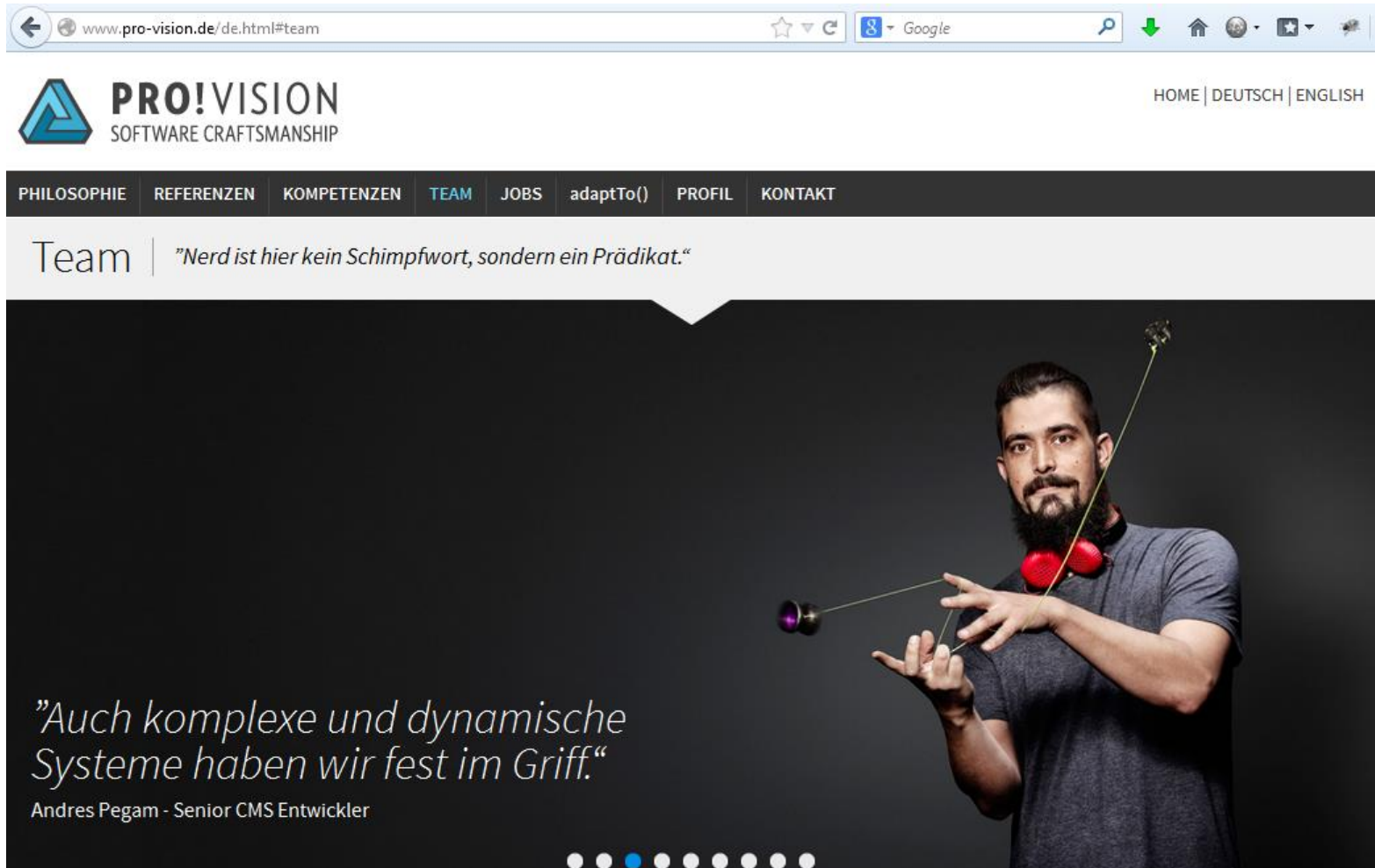
- JSF, .Net, Wicket, Tapestry, etc.
- Nehmen dem Entwickler viel Arbeit ab. Lösen bestimmte „Probleme“
- Wicket ist wirklich cool
- Aber:

# Die Realität

The screenshot shows a web browser window with the URL `https://www.interamt.de/koop/app/trefferliste?5-1.JLinkListener-table-body-rows-1-cells-1-cell-editLink` highlighted in the address bar. Below the address bar, the Interamt logo and navigation menu are visible. A callout box with a purple background and white text points to the URL, stating: "Selber Link zeigt ohne Session auf Übersicht". Below the navigation menu, there is a button labeled "< zurück zur Übersicht". The main content area contains a notice: "Auf dieses Stellenangebot können Sie sich nicht online über interamt.de bewerben, sondern nur „klassisch“ per Brief oder - falls angegeben - per E-Mail an den eingetragenen Kontakt-Ansprechpartner." Below this, there are two sections: "Details" and "Stellenbeschreibung". The "Details" section shows "Interamt Angebots-ID" as "222656". The "Stellenbeschreibung" section shows the text "Das Sekretariat der Ständigen Konferenz der Kultusminister c".

## Warum ist das oft so? Muss das so sein?

# Besser: Zustand in der URL





# Webframeworks & REST

- **Grundprobleme**

- Klassisches MVC Frameworks adressieren oft Controller
- Transaktionale Funktionalität 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

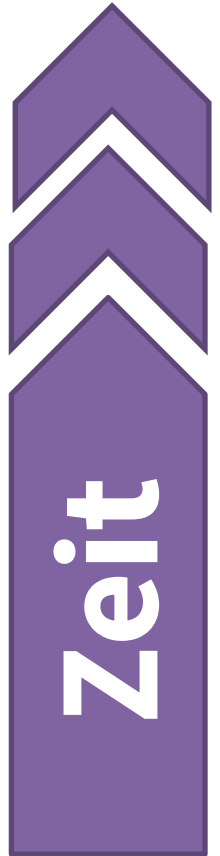
---

# Sling - Geschichte

Seit **Juni 2009** Apache Top Level Projekt.

Seit **September 2007** Apache Incubator Projekt.

Gestartet als internes Projekt von Day Software.



# 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

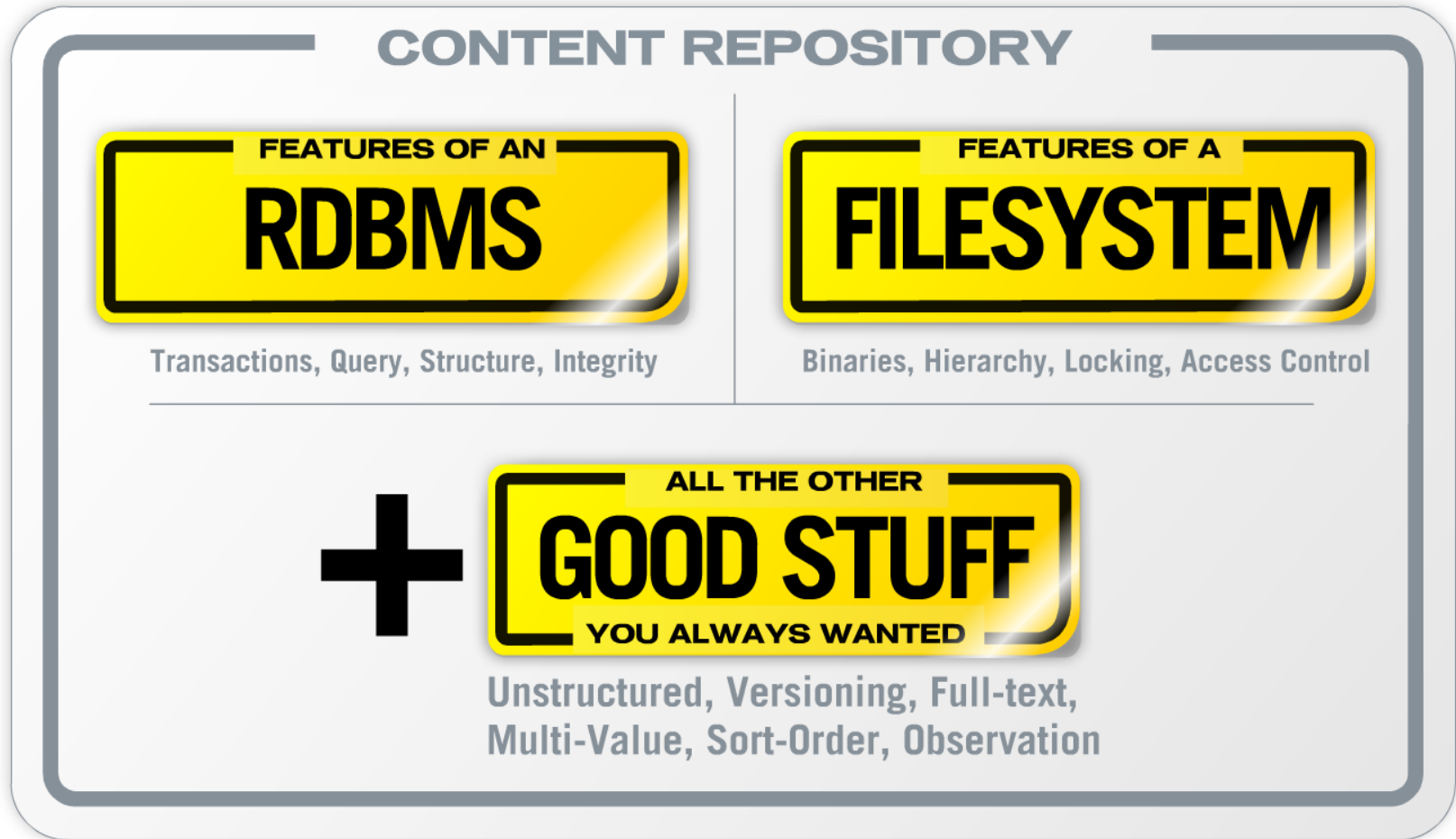
---

# Content Repository (kurz)

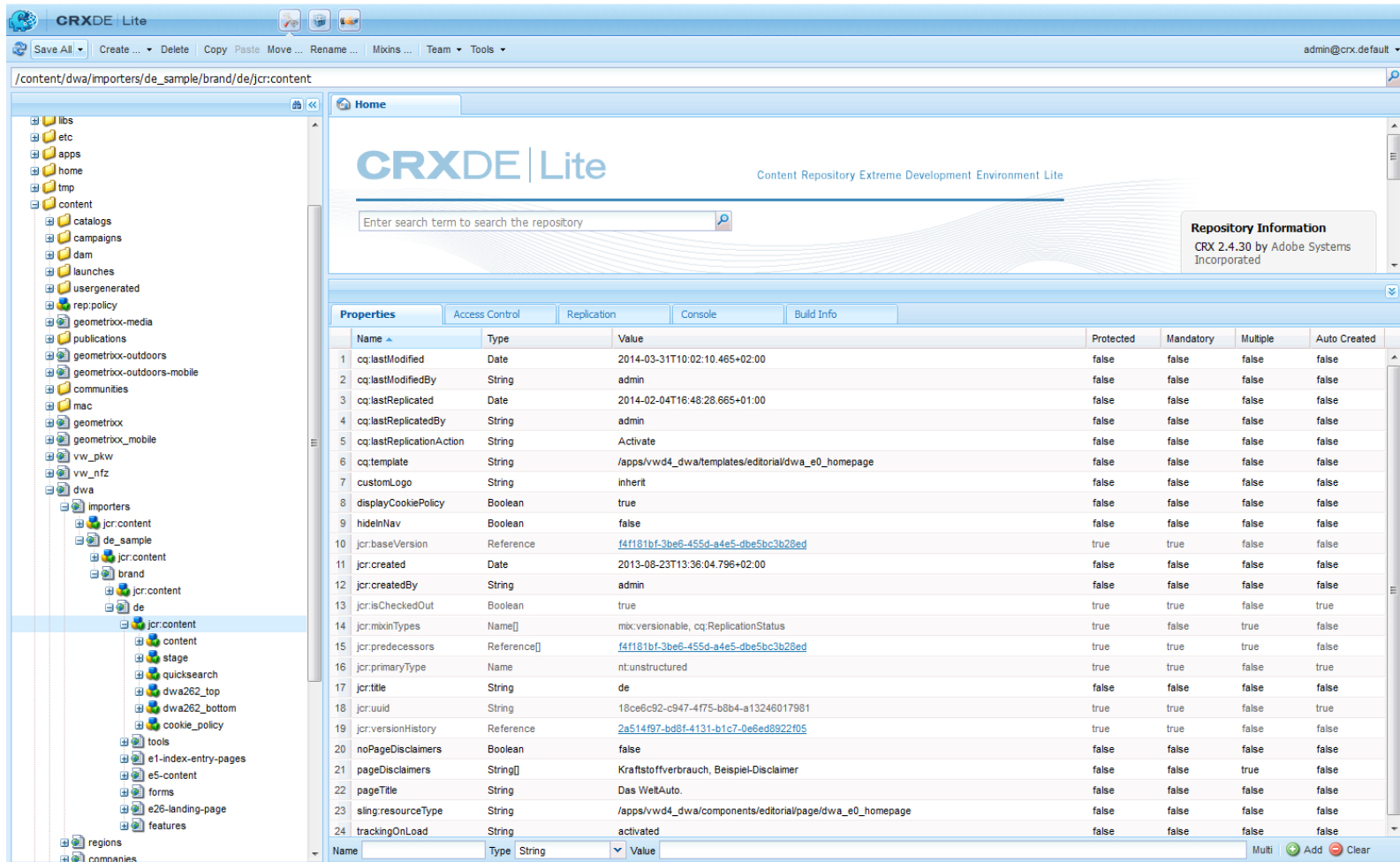
Java Content Repository (JCR-283)

---

# Content Repository



# Content Repository (Beispiel)



The screenshot shows the CRXDE Lite interface. On the left is a tree view of the content repository structure. The main area displays the 'Properties' tab for the selected content item. The table below shows the properties of the content item.

Name	Type	Value	Protected	Mandatory	Multiple	Auto Created	
1	cq:lastModified	Date	2014-03-31T10:02:10.465+02:00	false	false	false	false
2	cq:lastModifiedBy	String	admin	false	false	false	false
3	cq:lastReplicated	Date	2014-02-04T16:48:28.665+01:00	false	false	false	false
4	cq:lastReplicatedBy	String	admin	false	false	false	false
5	cq:lastReplicationAction	String	Activate	false	false	false	false
6	cq:template	String	/apps/vw_d4_dwa/templates/editorial/dwa_e0_homepage	false	false	false	false
7	customLogo	String	inherit	false	false	false	false
8	displayCookiePolicy	Boolean	true	false	false	false	false
9	hideInNav	Boolean	false	false	false	false	false
10	jcr:baseVersion	Reference	<a href="#">f4f181bf-3be6-455d-a4e5-dbe5bc3b28ed</a>	true	true	false	false
11	jcr:created	Date	2013-08-23T13:36:04.796+02:00	false	false	false	false
12	jcr:createdBy	String	admin	false	false	false	false
13	jcr:isCheckedOut	Boolean	true	true	true	false	true
14	jcr:mixinTypes	Name[]	mix:versionable, cq:ReplicationStatus	true	false	true	false
15	jcr:predecessors	Reference[]	<a href="#">f4f181bf-3be6-455d-a4e5-dbe5bc3b28ed</a>	true	true	true	false
16	jcr:primaryType	Name	nt:unstructured	true	true	false	true
17	jcr:title	String	de	false	false	false	false
18	jcr:uuid	String	18ce6c92-c947-4f75-b8b4-a13246017981	true	true	false	true
19	jcr:versionHistory	Reference	<a href="#">2a514f97-bd8f-4131-b1c7-0e6ed8922f05</a>	true	true	false	false
20	noPageDisclaimers	Boolean	false	false	false	false	false
21	pageDisclaimers	String[]	Kraftstoffverbrauch, Beispiel-Disclaimer	false	false	true	false
22	pageTitle	String	Das WeltAuto.	false	false	false	false
23	sling:resourceType	String	/apps/vw_d4_dwa/components/editorial/page/dwa_e0_homepage	false	false	false	false
24	trackingOnLoad	String	activated	false	false	false	false



# Content Repository (Beispiel)

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

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

## Beispiele zur URL Dekomposition

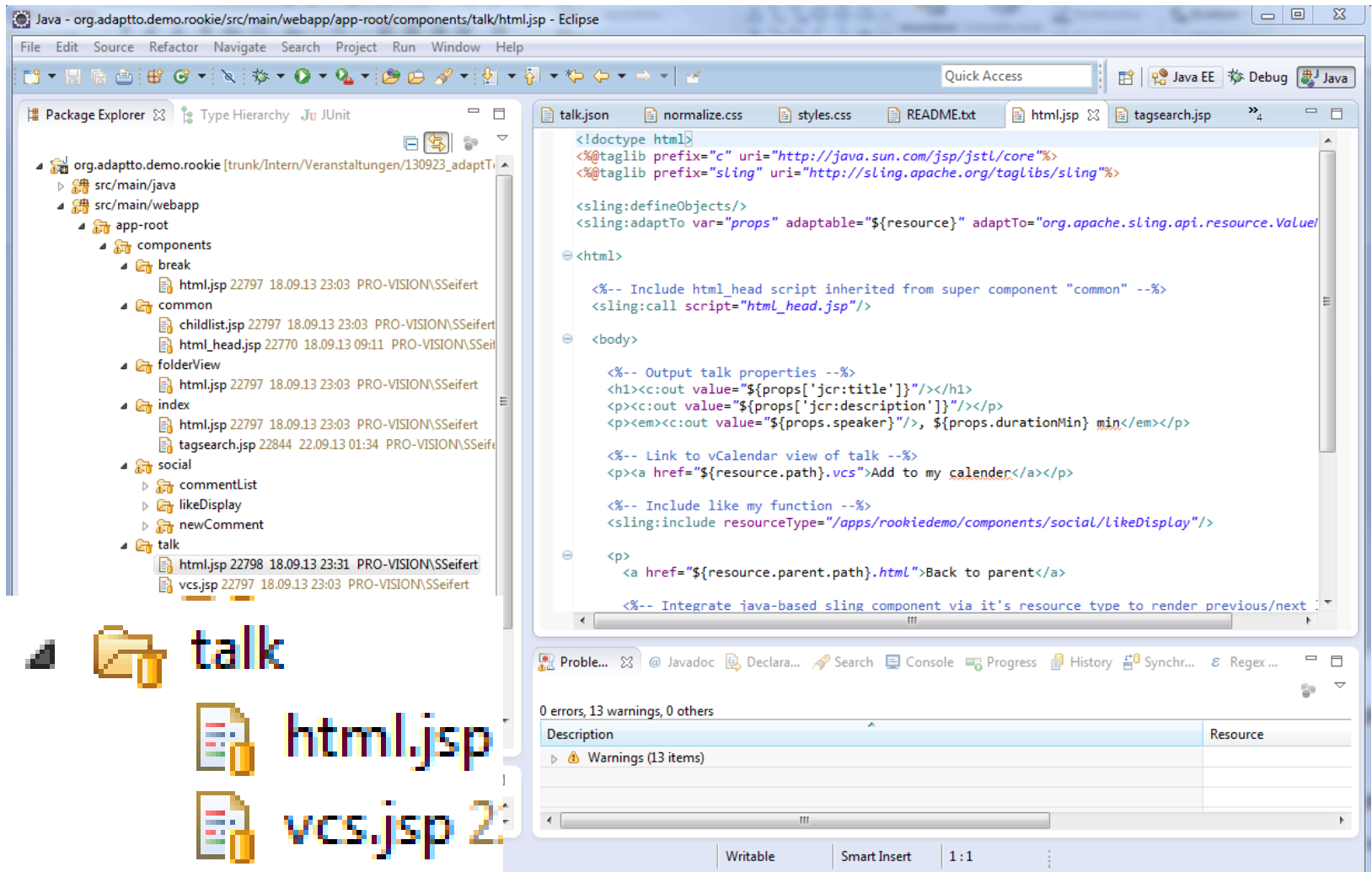
http://host/content/adaptto/2013/day1/rookie-session.html

**Resource path**      **Extension**

http://host/content/adaptto.tagsearch.html/Sling

**Resource path**      **Selector**      **Extension**      **Suffix**

# org.adappto.demo.rookie



The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays the project structure for `org.adappto.demo.rookie`, with the `talk` component selected. The main editor shows the `html.jsp` file, which contains Sling tags for rendering a page. The code includes a Sling component definition, a call to `html_head.jsp`, and a body with dynamic content and links.

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

<sling:defineObjects/>
<sling:adaptTo var="props" adaptable="${resource}" adaptTo="org.apache.sling.api.resource.Value" />

<html>

  <!-- Include html_head script inherited from super component "common" -->
  <sling:call script="html_head.jsp"/>

  <body>

    <!-- Output talk properties -->
    <h1><c:out value="${props['jcr:title']}" /></h1>
    <p><c:out value="${props['jcr:description']}" /></p>
    <p><em><c:out value="${props.speaker}" />, <c:out value="${props.durationMin}" min</em></p>

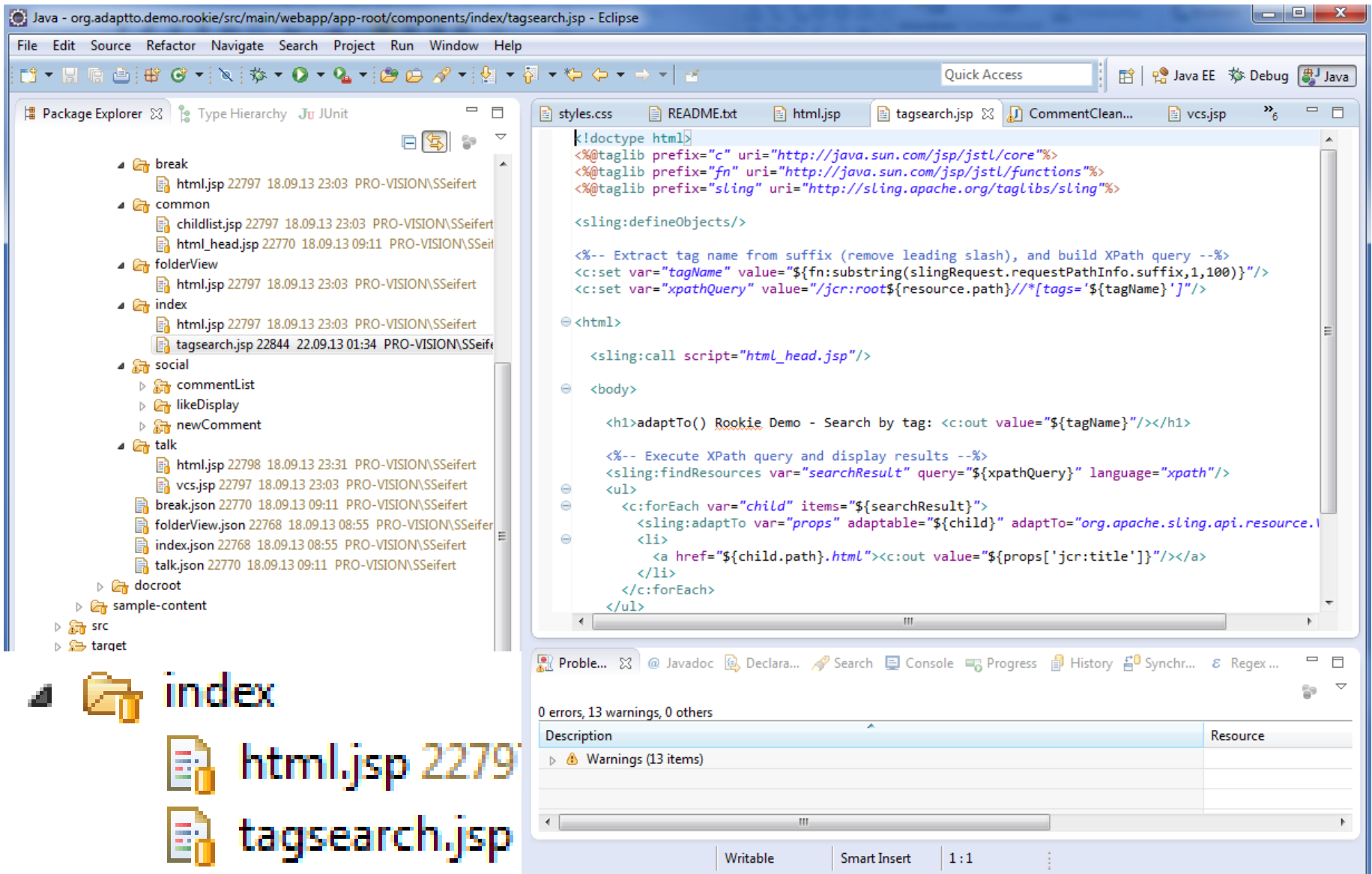
    <!-- Link to vCalendar view of talk -->
    <p><a href="${resource.path}.vcs">Add to my calendar</a></p>

    <!-- Include like my function -->
    <sling:include resourceType="/apps/rookiedemo/components/social/LikeDisplay"/>

    <p>
      <a href="${resource.parent.path}.html">Back to parent</a>

    <!-- Integrate java-based sling component via it's resource type to render previous/next -->
```

Below the IDE screenshot, there are three stylized icons representing files: a folder icon labeled `talk`, a document icon labeled `html.jsp`, and another document icon labeled `vcs.jsp 2`.



The screenshot shows the Eclipse IDE interface. The Package Explorer on the left shows a project structure with folders like 'break', 'common', 'index', 'social', and 'talk'. The main editor displays the content of 'tagsearch.jsp', which is an HTML JSP page. The code includes a doctype declaration, taglib declarations for 'c', 'fn', and 'sling', and a scriptlet that defines an XPath query to search for tags in the request path. The page content includes a header, a search prompt, and a list of search results.

```

<!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>

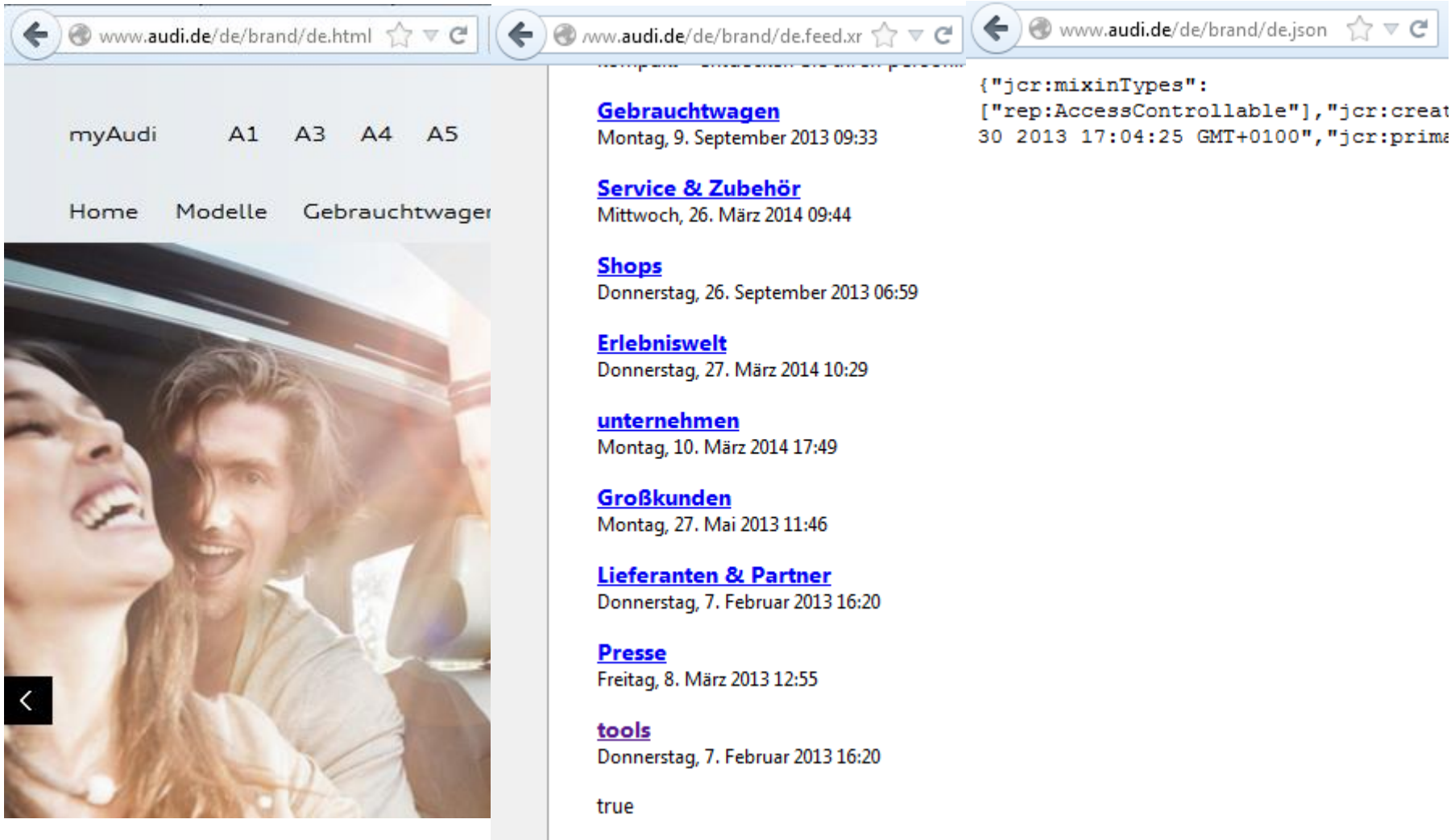
    <!-- Execute XPath query and display results -->
    <sling:findResources var="searchResult" query="\${xpathQuery}" language="xpath"/>
    <ul>
      <c:forEach var="child" items="\${searchResult}">
        <sling:adaptTo var="props" adaptable="\${child}" adaptTo="org.apache.sling.api.resource.IResource"/>
        <li>
          <a href="\${child.path}.html"><c:out value="\${props['jcr:title']}"/></a>
        </li>
      </c:forEach>
    </ul>
  
```

The bottom panel shows the 'Problems' view, which displays 0 errors and 13 warnings. The table below shows the structure of the warnings:

Description	Resource
Warnings (13 items)	

index  
 html.jsp 2279  
 tagsearch.jsp

# Einfaches Werkzeug, Inhalte schnell auszuliefern...



The screenshot displays a web browser with three tabs: `www.audi.de/de/brand/de.html`, `www.audi.de/de/brand/de.feed.xr`, and `www.audi.de/de/brand/de.json`. The main content area is divided into two columns. The left column features a navigation menu with links for 'myAudi', 'A1', 'A3', 'A4', 'A5', 'Home', 'Modelle', and 'Gebrauchtwagen'. Below the menu is a large image of a smiling couple in a car. The right column lists several content categories with their dates:

- Gebrauchtwagen** (Monday, 9. September 2013 09:33)
- Service & Zubehör** (Wednesday, 26. März 2014 09:44)
- Shops** (Thursday, 26. September 2013 06:59)
- Erlebniswelt** (Thursday, 27. März 2014 10:29)
- unternehmen** (Monday, 10. März 2014 17:49)
- Großkunden** (Monday, 27. Mai 2013 11:46)
- Lieferanten & Partner** (Thursday, 7. Februar 2013 16:20)
- Presse** (Friday, 8. März 2013 12:55)
- tools** (Thursday, 7. Februar 2013 16:20)

At the bottom of the right column, the text 'true' is visible. To the right of the browser tabs, a snippet of JSON code is shown:

```
{"jcr:mixinTypes":  
["rep:AccessControllable"],"jcr:creat  
30 2013 17:04:25 GMT+0100","jcr:prim
```



# 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.adappto.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
```

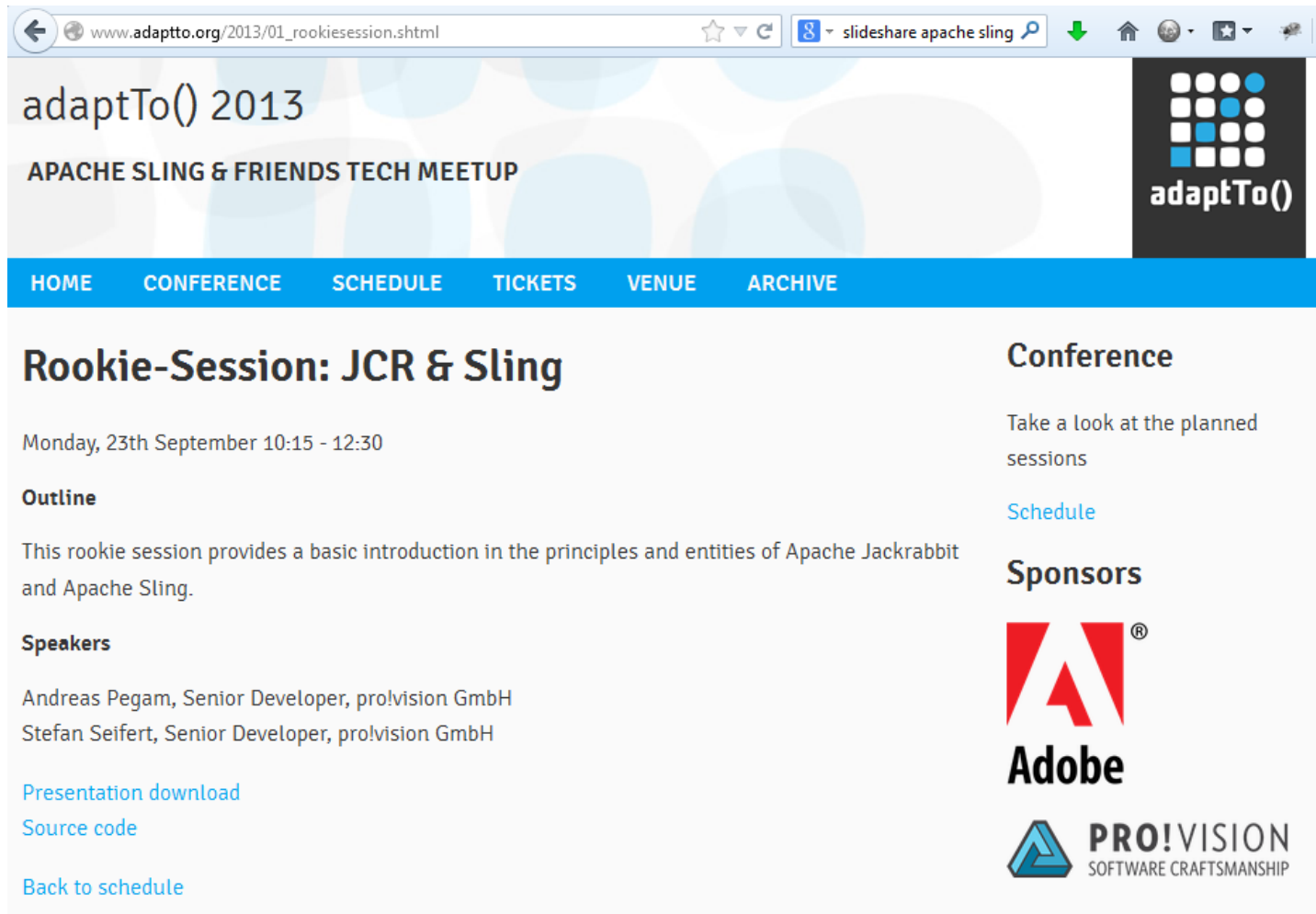
```
java -jar org.apache.sling.launchpad-7-SNAPSHOT-standalone.jar
```

## **Upload commons-lang3:**

```
http://localhost:8080/system/console/bundles
```

```
http://mvnrepository.com/artifact/org.apache.commons/commons-lang3/3.0
```

http://www.adapto.org/2013/01\_rookiesession.shtml



The screenshot shows a web browser window with the URL `www.adapto.org/2013/01_rookiesession.shtml`. The page title is "adaptTo() 2013" and the subtitle is "APACHE SLING & FRIENDS TECH MEETUP". The browser's address bar shows a search for "slideshare apache sling". The page features a blue navigation bar with links for HOME, CONFERENCE, SCHEDULE, TICKETS, VENUE, and ARCHIVE. The main content area is titled "Rookie-Session: JCR & Sling" and includes the date and time "Monday, 23th September 10:15 - 12:30". Under the "Outline" section, it states: "This rookie session provides a basic introduction in the principles and entities of Apache Jackrabbit and Apache Sling." The "Speakers" section lists "Andreas Pegam, Senior Developer, pro!vision GmbH" and "Stefan Seifert, Senior Developer, pro!vision GmbH". There are links for "Presentation download", "Source code", and "Back to schedule". On the right side, there is a "Conference" section with a link to "Schedule" and a "Sponsors" section featuring logos for Adobe and PRO!VISION SOFTWARE CRAFTSMANSHIP.

# Installation der Beispielapplikation

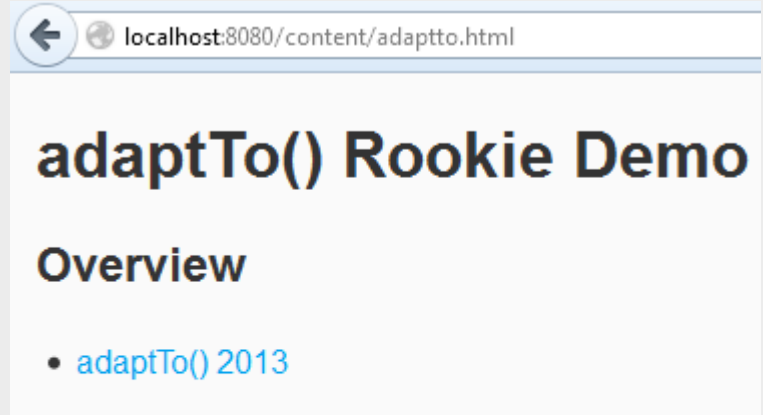
**Sourcen entpacken,**

**In der pom.xml den sling port setzen**

```
<sling.url>http://localhost:8080
```

```
mvn clean install
```

```
mvn sling:install
```



## Apache Sling Web Console Bundles



Main OSGi Sling Status Web Console

Bundle information: 103 bundles in total, 90 bundles active, 4 active fragments, 9 bundles installed

<input type="text"/> <input type="button" value="x"/> <input type="button" value="Apply Filter"/> <input type="button" value="Filter All"/> <input type="button" value="Reload"/> <input type="button" value="Install/Update..."/> <input type="button" value="Refresh Packages"/>							
Id	Name	Version	Category	Status	Actions		
0	System Bundle ( <i>org.apache.felix.framework</i> )	4.4.0		Active			
103	adaptTo() Rookie Demo ( <i>org.adappto.demo.rookie</i> )	1.0.0.SNAPSHOT		Active	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
18	Apache Aries JMX API ( <i>org.apache.aries.jmx.api</i> )	0.3.0		Active	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
19	Apache Aries JMX Core ( <i>org.apache.aries.jmx.core</i> )	0.3.0		Active	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

---

# Noch mehr Beispiele

[org.adaptto.demo.rookie](http://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.adapto.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();
    }
}
```

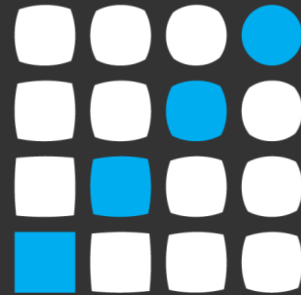
- Apache Sling ist noch wesentlich umfangreicher
- Und wird fortlaufend ergänzt
  - I18n
  - Eventing & Jobs
  - Chron Jobs & Scheduling
  - User & Groupmanagement
  - Caching
  - Model Objects
  - Tenants
  - Cluster
  - Replication
  - Healthcheck

# 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](http://www.adaptto.org/2013/01_rookiesession.shtml)



**PRO!VISION**  
SOFTWARE CRAFTSMANSHIP



**adaptTo()**

**APACHE SLING & FRIENDS TECH MEETUP**

**Kulturbrauerei Berlin, 22-24 SEPTEMBER 2014**