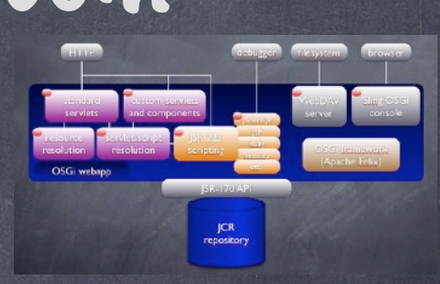
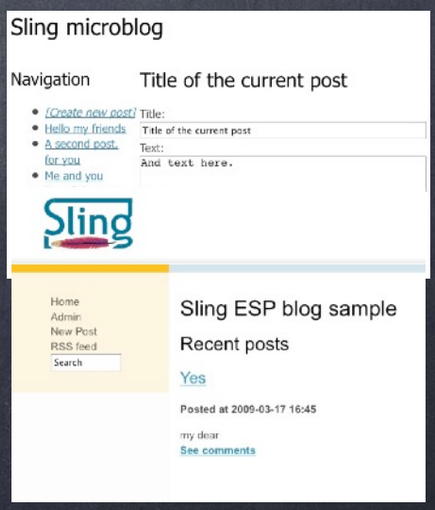
Rapid JCR applications development with Apache Sling

Bertrand Delacrétaz
Senior R&D Developer, Day Software, www.day.com
Member and Director, ASF
bdelacretaz@apache.org
blog at http://grep.codeconsult.ch
twitter: @bdelacretaz





Slides revision: 2009-03-20
Slides theme design: David Nuescheler

Sling

Day

is this slide busy enough?

Intro to JCR

JCR = Java Content Repository API

JSR-170 / JSR-283

Everything Is Content - and JCR manages it as trees of Nodes and Properties, using rich data types.

What's JCR?

JSR-170

Content Repository for JavaTM technology API

Spec-Lead:
Day Software

Status:
Final Release 17-jun-2005

Java Community Process

Expert Group:

FileNet, Hummingbird, SAP, atg, IBM, bea, Apache, Day, Nuxeo, Venetica, macromedia, hp, MediaSurface, Fujitsu, Oracle, EMC documentum, sas, software AG THE XML COMPANY, STELLENT

What's JCR?

JSR-170

JSR-283

Content Repository for JavaTM technology API v2.0

Spec-Lead:
Day Software

Status:
Public Review Closed sep-2007

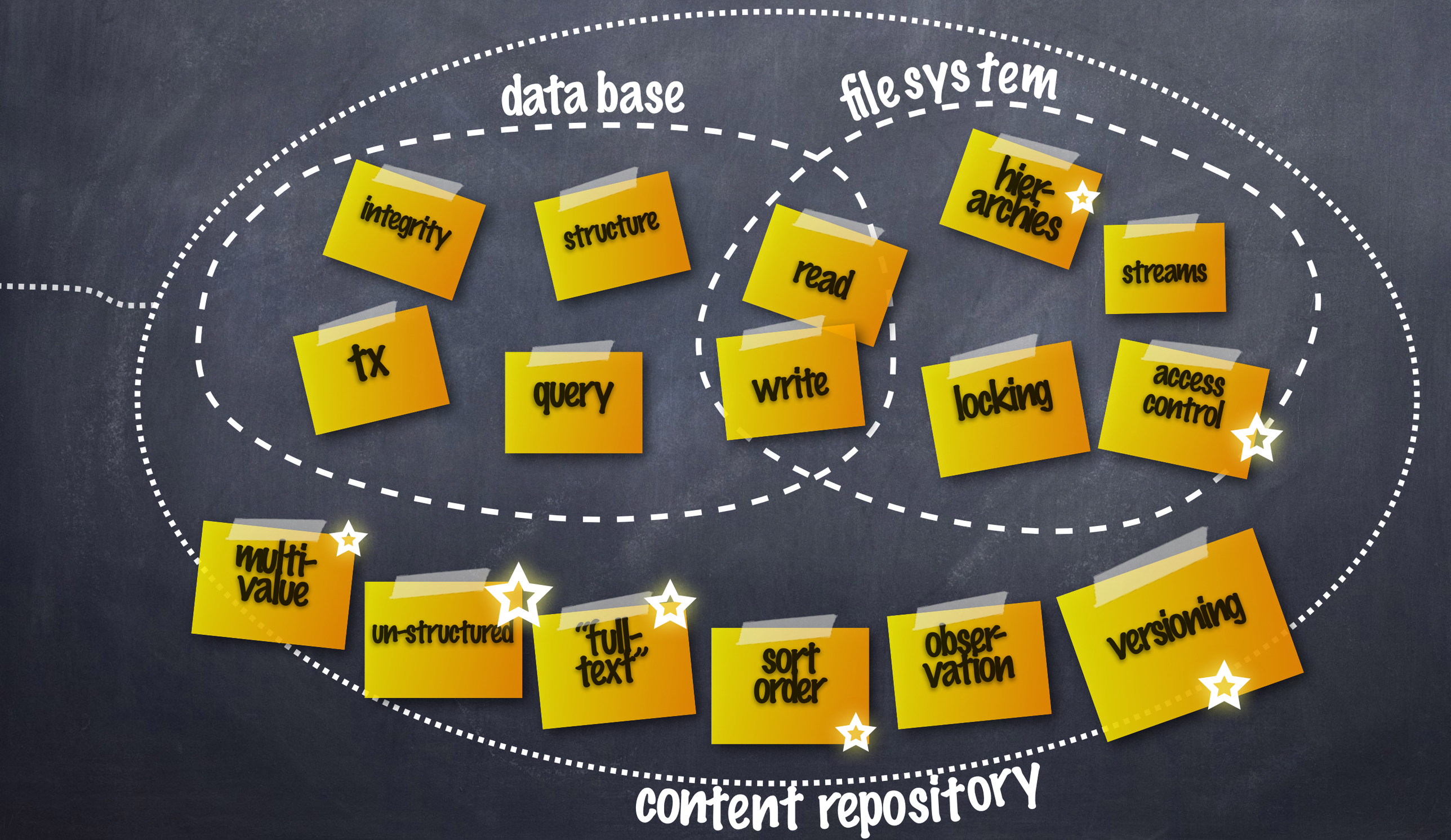
Expert Group:

Day

What's JCR?

"The API should be a standard, implementation independent, way to access content bi-directionally on a granular level to a content repository." ?

Best of both worlds.



Known compliant Repositories

* using third party connector

Apache **Jackrabbit**	Oracle **XML DB**	Exo **ECMS Platform**	Microsoft **Sharepoint** *	OpenText **Livelink** *
Day **CRX**	IBM FileNet **P8**	Xythos **Repository**	Alfresco **ECM**	Vignette **V7** *
Interwoven **Repository** *	IBM **CM**	EMC **Documentum** *		

+hundreds of TCKs **registered**

How many RDBMS vendors do you need?

Some known JCR Applications

BEA **Portal**

Sun **OpenPortal**

JBoss **Portal**

Oracle **Portal**

Enterprise Search

magnolia **WCMS**

Apache **Sling**

Communique DAM

Spring Framework

Alfresco **ECMS**

Mindquarry **Collaboration**

Communique Collab

Apache **Cocoon**

QSLabs **Compliance**

Communiqué WCMS

Apache **Tapestry**

IBM FileNet **WebSiteManager**

Apache **James**

Artifactory **Maven Proxy**

medic-2-medic **mapofmedicine**

GX **WebManager**

Exo **ECMS Platform**

TYPO3 **v5.0 WCM**

Nuxeo **ECM**

Online Community

Hippo **CMS**

Liferay **Enterprise Portal**

Percussion **Rhythmix**

QuickWCM **WCMS**

Jahia **Framework**

Sakai **E-learning**

Lutece **Portal**

Sourcemix **Sourcemix**

JCR code excerpt

```
Repository repository = new TransientRepository();
Session session = repository.login(...);

// Create content
Node root = session.getRootNode();
Node hello = root.addNode("hello");
Node world = hello.addNode("world");

world.setProperty("message", "Hello, World!");
session.save();

// Retrieve content
Node node = root.getNode("hello/world");
print(node.getPath());
print(node.getProperty("message").getString());
```

On to Sling

Day

Sling builds on top of JCR

Scriptable applications layer on top of JCR

OSGi-based industrial-strength framework

Simple, powerful, JCR inside

Runs on Apache Jackrabbit by default

http://incubator.apache.org/sling

join the fun!

Sling == REST over JCR

UNIVERSITY OF CALIFORNIA,
IRVINE

Architectural Styles and the Design of Network-based Software Architec

DISSERTATION

submitted in partial satisfaction of the requirements for the degree

DOCTOR OF PHILOSOPHY

in Information and Computer Science

by

Roy Thomas Fielding

REST -> Roy T. Fielding

JCR: David Nuescheler

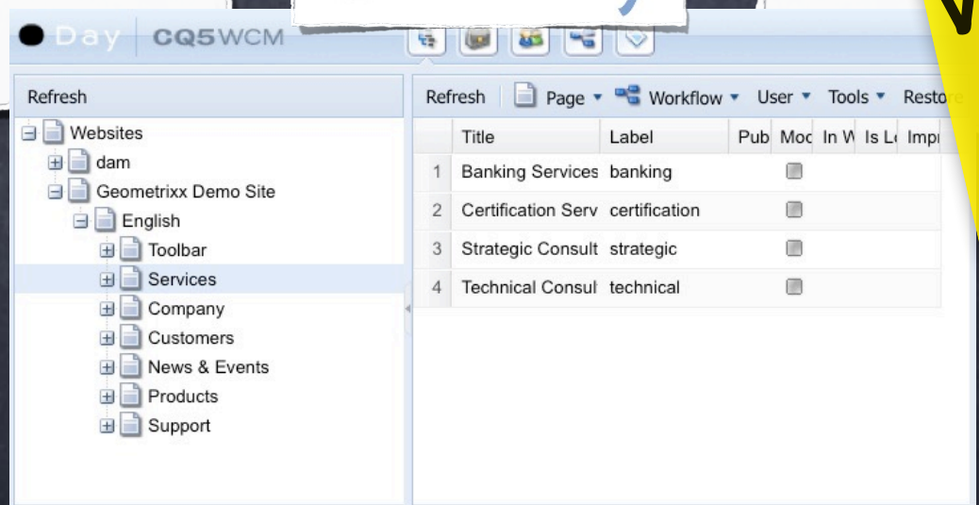
released y2k
162 pages

Content Repository API for Java™ Technology Specification

Java Specification Request 170
version 1.0
11 May 2005

v1.0 released 2005
+300 pages

Day



Day

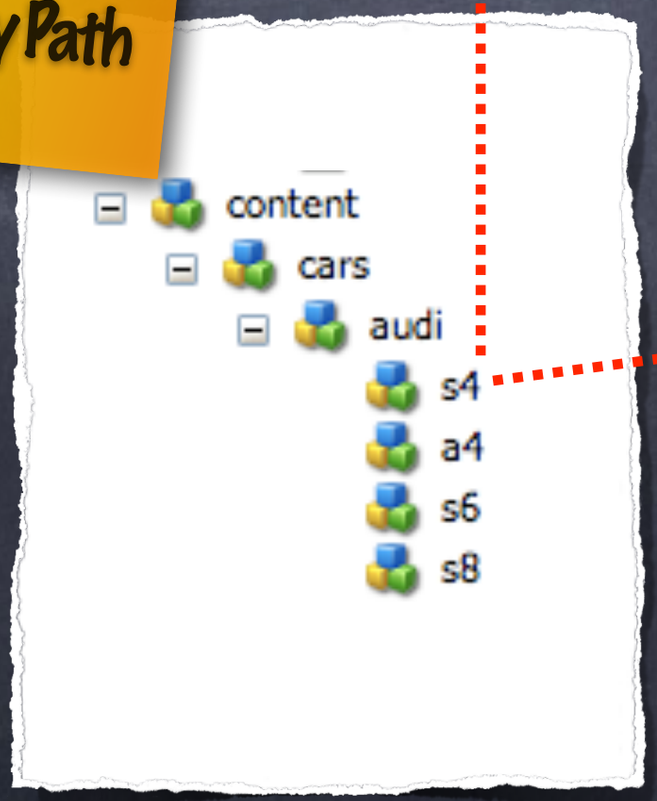
Reclaiming the web.
Sling URL decomposition.

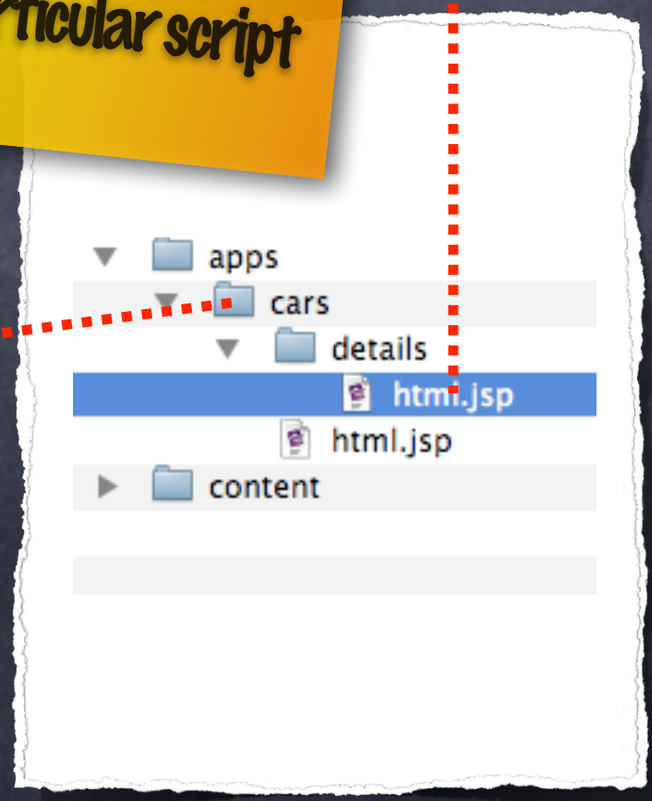
```
/cars/audi/s4.details.html
```

Content Repository Path

...selects a particular script

Repository





Sling architecture

POST, json GET

sling.js utility

HTTP

debugger

filesystem

browser

WebDAV access

standard servlets

custom servlets and components

resource resolution

servlet/script resolution

JSR 223 scripting

javascript

JSP

Ruby

Velocity

etc..

WebDAV server

Sling OSGi console

OSGi framework (Apache Felix)

OSGi webapp

ESP server-side javascript

JSR-170 API

JCR repository

Sling

get the code from
dev.day.com :
http://tinyurl.com/
slingblog46

A minimal Sling blog

Consisting of one .esp script

```
cat /Volumes/localhost/apps/blog/blog.esp | wc -l
54
```

Sling POST servlet

```
# POST to Sling
curl -F title=hello http://localhost:8888/foo
-> 200 OK

# GET created node in json format
curl http://localhost:8888/foo.tidy.json
{
  "jcr:primaryType": "nt:unstructured",
  "title": "hello"
}
```

POST parameters set node properties

Sling

blog step 1: create content

```
<form method="POST">
  Title:
  <input type="text" name="title" style="width:100%"/>

  Text:
  <textarea style="width:100%" name="text"></textarea>

  <input type="submit" value="save"/>

  <input type="hidden" name=":redirect" value="*"/>
</form>
```

Form fields drive the content model

blog step 2: retrieve content

```
<script src="/system/sling.js"></script>
<form method="POST">
...
</form>
```

```
<!-- initialize form fields from current node values -->
<script>Sling.wizard();</script>
```

Instant CRU(D)

blog step 3: navigation

```
<ul>
    <li>
      <a href="/content/blog/*">
      [Create new post]</a>
    </li>
  <script>
    var posts = Sling.getContent("/content/blog", 2);
    for(var post in posts) {
      document.write(
        "<li><a href='" + post + "'>"
        + posts[post].title + "</a></li>");
    }
  </script>
</ul>
```

Navigation

- *[Create new post]*
- Hello my friends
- A second post, for you
- Me and you
- New Orleans at dawn
- Title of the current post

we got a blog!

html form + Sling.wizard() + Sling.getContent()

The ESP blog sample

ESP scripting + java + javascript

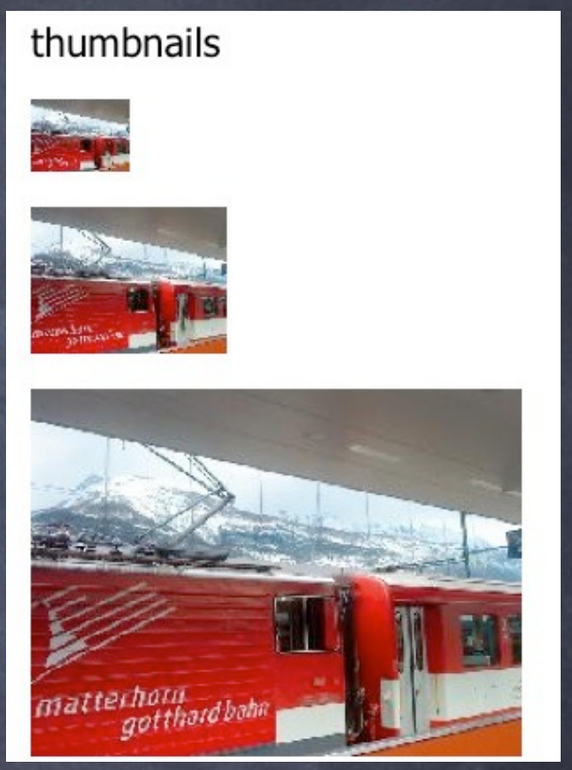
OSGi bundle, initial content, WebDAV, observation, RAD

Code under sling/samples
http://tinyurl.com/slingblogesp

ESP blog in action

create

retrieve



thumbnails

update

ESPblog demo

Day

ESP blog source files

admin.esp
edit.esp
html.esp
list.esp
menu.esp

xml.esp (RSS feed)
constants.esp
header.esp

pom.xml
ThumbnailGeneratorService.java
ThumbnailGeneratorServiceImpl.java

espblog.css
sling-logo.png

espblog resource type points to those scripts

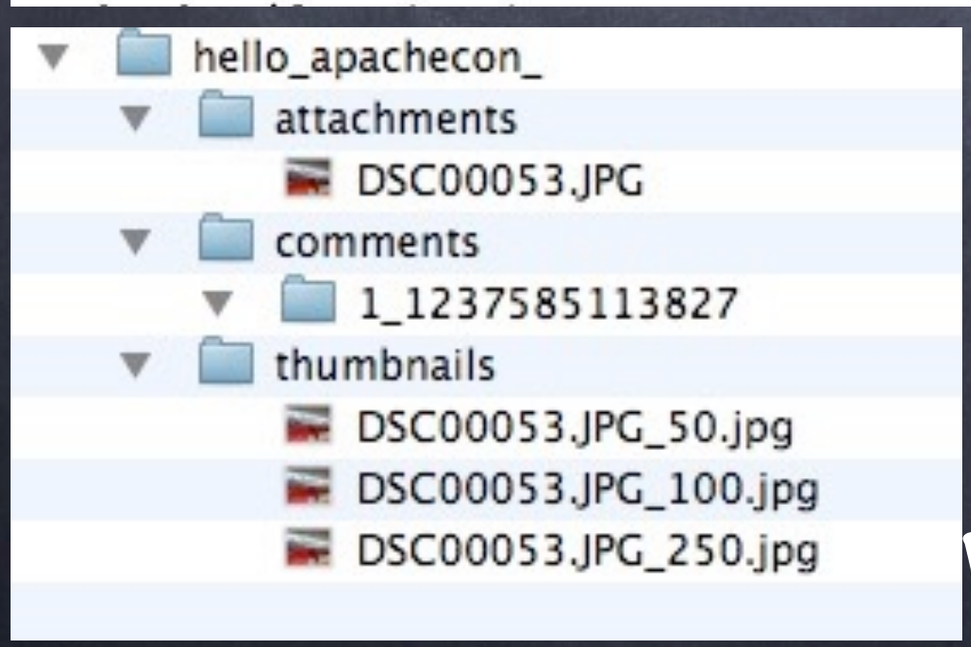
ESP blog content structure

JSON dump

```
  created: "Fri Mar 20 2009 22:38:14 GMT+0100",
  jcr:primaryType: "nt:unstructured",
  posttext: "This is a sample post for my ApacheCon Sling talk",
  title: "Hello, ApacheCon!",
- attachments: {
      jcr:primaryType: "nt:unstructured",
    - DSC00053.JPG: {
          jcr:created: "Fri Mar 20 2009 22:38:14 GM
          jcr:primaryType: "nt:file",
        - jcr:content: {
              jcr:primaryType: "nt:resource",
              :jcr:data: 119212,
              jcr:mimeType: "image/jpeg",
              jcr:uuid: "5ef09129-6e06-449f-9183-194af99f51c6",
              jcr:lastModified: "Fri Mar 20 2009 22:38:14 GMT+0100"
          }
      }
  },
```

```
- thumbnails: {
      jcr:created: "Fri Mar 20 2009
      jcr:primaryType: "nt:folder",
    + DSC00053.JPG_50.jpg: { … },
    + DSC00053.JPG_100.jpg: { … },
    + DSC00053.JPG_250.jpg: { … }
  },
```

```
- comments: {
      jcr:primaryType: "nt:unstructured",
    - 1_1237585113827: {
          commenter: "Bertrand",
          commenttext: "I like it, thanks!",
          jcr:primaryType: "nt:unstructured"
      }
  }
```



WebDAV view

ESP blog edit script

```
<%
pageTitle = currentNode.title; load("header.esp");
%>
<body>
  <form method="POST" action="<%= currentNode %>">
    <p><label>Title</label>
    <input name="title" type="text"
      value="<%= currentNode.title %>"></p>
    ...
    <input type="hidden" name="created"/>
    <input name=":redirect" type="hidden"
      value="/content/espblog/posts.admin.html"/>
    <input type="submit" value="Post" class="button">
  </form>
...
```

ESP blog thumbnails: OSGi service

```
/**
 * Observe the espblog content for changes, and generate
 * thumbnails when images are added.
 *
 * maven-scr-plugin uses annotations to generate the OSGi
 * Declarative Services XML configuration files
 * @scr.service
 * @scr.component immediate="true"
 *
 */
public class ThumbnailGeneratorServiceImpl
  implements ThumbnailGeneratorService, EventListener {
```

ESP blog thumbnails: observation

```
/** @scr.reference (framework injects it automatically) */
private SlingRepository repository;

/** called by framework when service is activated */
protected void activate(ComponentContext context) {
  Session s = repository.loginAdministrative(null);
  // Listen for nt:file NODE_ADDED repository events
  ObservationManager m =
    s.getWorkspace().getObservationManager();
    String[] types = { "nt:file" };
    m.addEventListener(
      this,
      Event.NODE_ADDED,
      contentPath, ...);
```

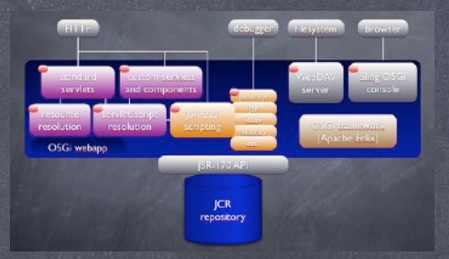
ESP blog thumbnails: NODE_ADDED

```
/** Called by JCR Observation manager for events that this
  *  EventListener registered for
  */
public void onEvent(EventIterator it) {
  while (it.hasNext()) {
    Event event = it.nextEvent();
    if (event.getType() == Event.NODE_ADDED
      && !(event.getPath().contains("thumbnails")))
      {
        String p = event.getPath();
        Node n = session.getRootNode().getNode(p);
        createThumbnails(addedNode);
      }
    }
```

...

We got a typical Sling application!

JCR features: WebDAV, observation, nt:unstructured.



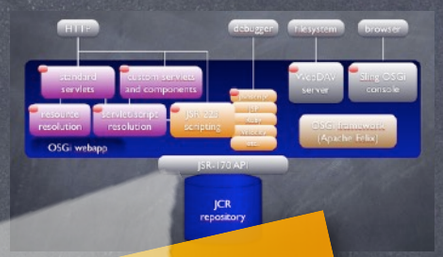
Sling goodies: simple script mappings (BYOL), POST servlet, RESTful interface.

OSGi bundle, code + initial content, Maven plugins.

ESPblog source code is found under sling/samples

Where is Sling going?

Where is Sling going? (aka conclusion)



First web framework designed for JCR.
Embrace the web, act like a very clever web server!
Intelligent HTTP/JSON storage?
OSGi, organic app growth.

Growing community, graduate in 2009?

"Sling is not a web applications framework, it's a **web** framework"

http://incubator.apache.org/sling
http://dev.day.com
http://contentcentric.org/

join the fun!

Sling