

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
Java™ technology API

Spec-Lead:
Day Software

Status:
Final Release 17-jun-2005



Expert Group:



What's JCR?

JSR-170

JSR-283

Content Repository for
Java™ technology API v2.0

Spec-Lead:
Day Software

Status:
Public Review Closed sep-2007



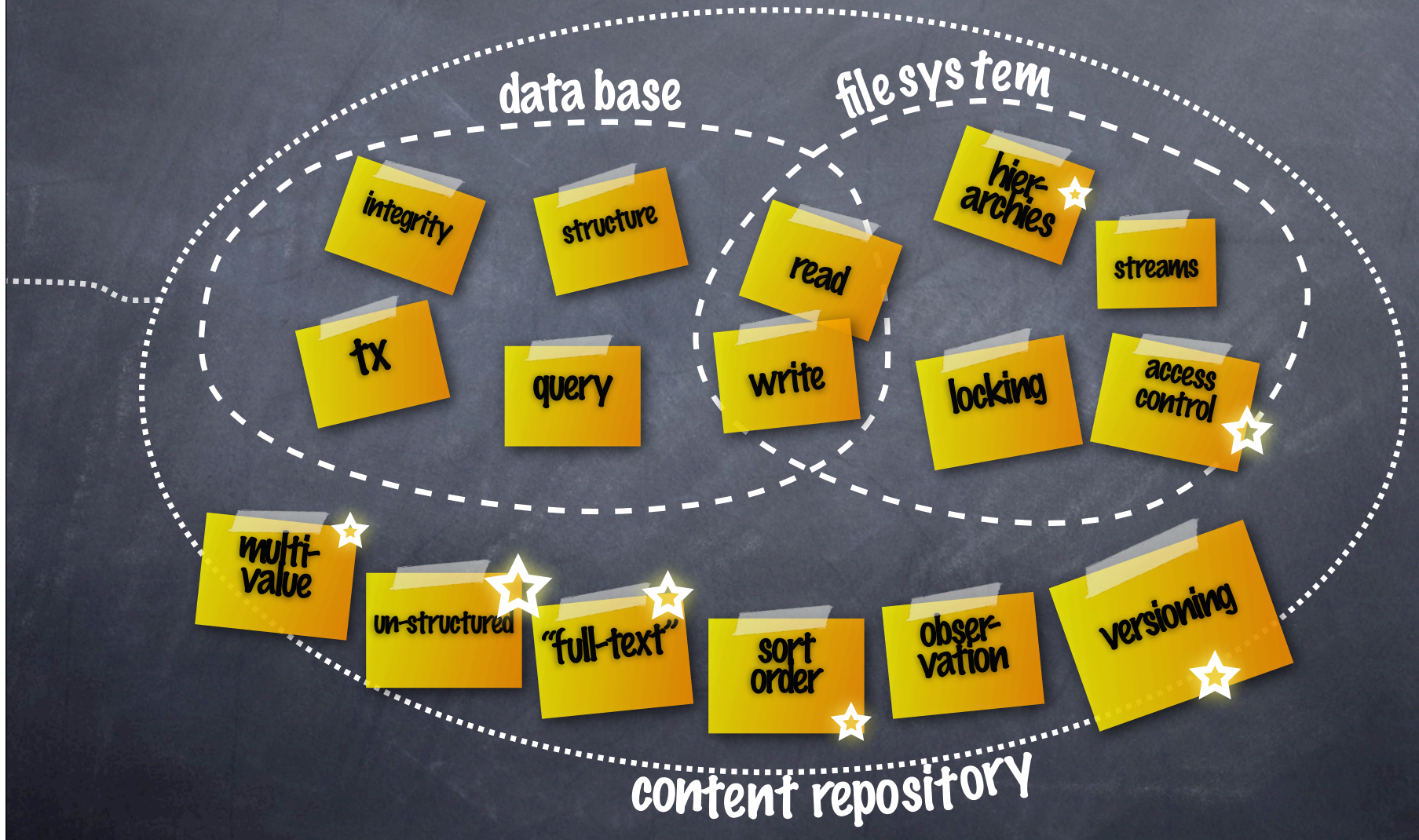
Expert Group:



What's JCR?










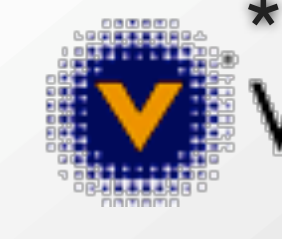


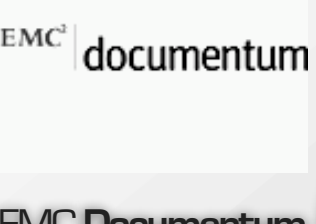
“The API should be a standard, implementation independent, way to access content bi-directionally on a granular level to a content repository.” ?

Best of both worlds.

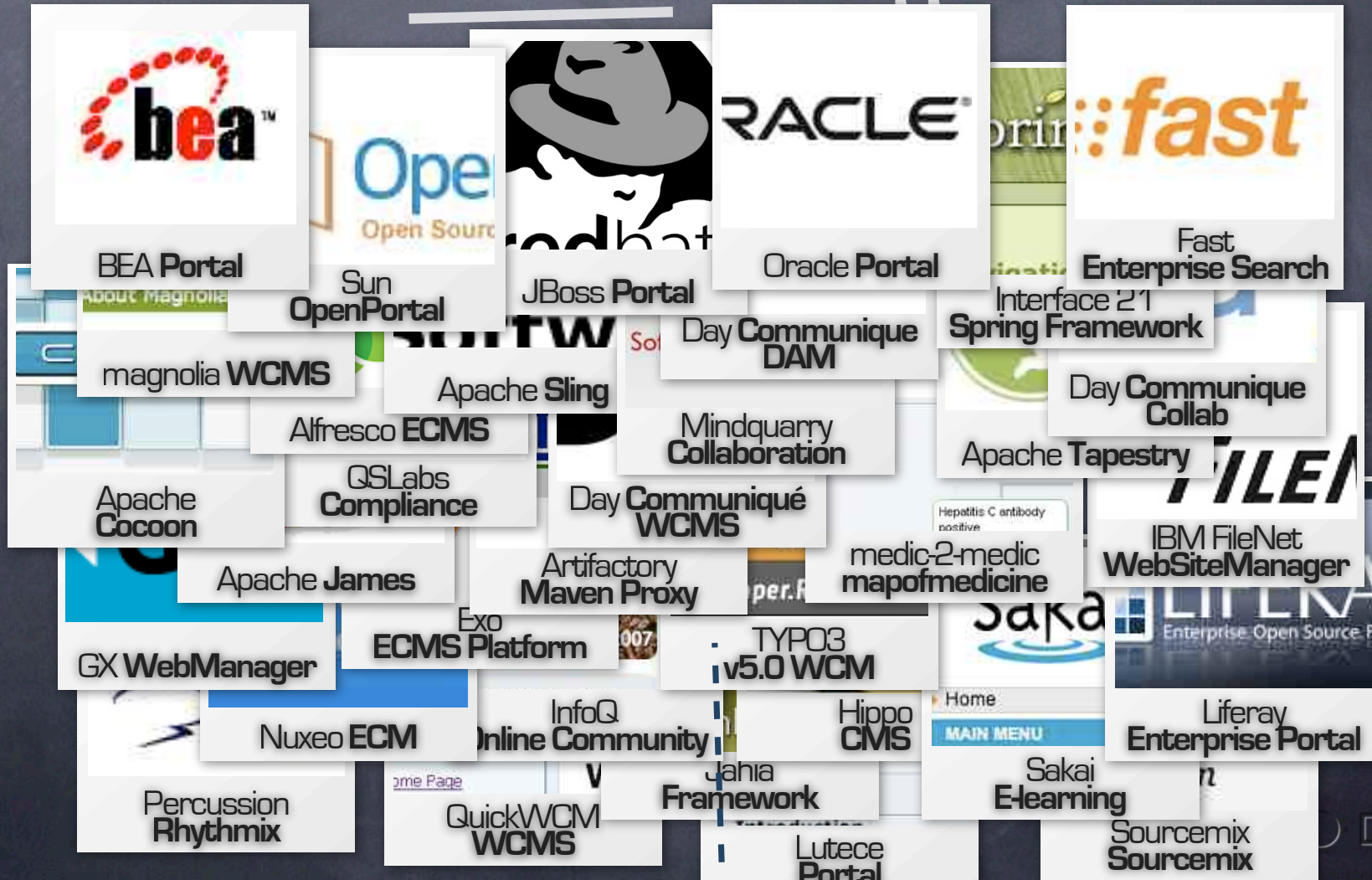


Known compliant Repositories

* using third party connector

 <p>Apache Jackrabbit</p>	 <p>Oracle XML DB</p>	 <p>Exo ECMS Platform</p>	 <p>Microsoft Sharepoint</p>	 <p>OpenText Livelink</p>
 <p>Day CRX</p>	 <p>IBM FileNet P8</p>	 <p>Xythos Repository</p>	 <p>Alfresco ECM</p>	 <p>Vignette V7</p>
 <p>Interwoven Repository</p>	 <p>IBM CM</p>	 <p>EMC Documentum</p>	<div data-bbox="1254 1133 1635 1468" style="background-color: white; padding: 5px;"> <p>+hundreds of TCKs registered</p> </div> <div data-bbox="1612 1165 2016 1468" style="background-color: yellow; padding: 5px; transform: rotate(-10deg);"> <p>How many RDBMS vendors do you need?</p> </div>	

Some known JCR Applications



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

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

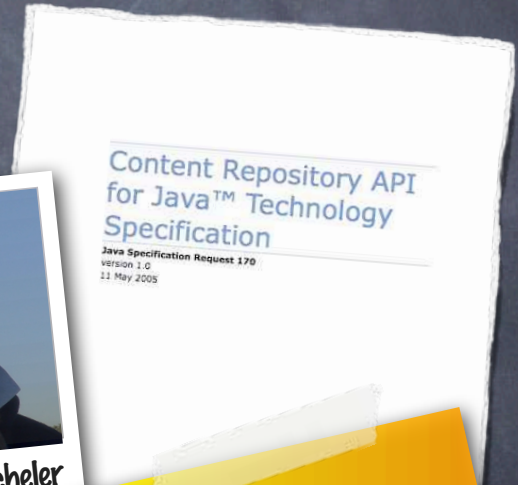
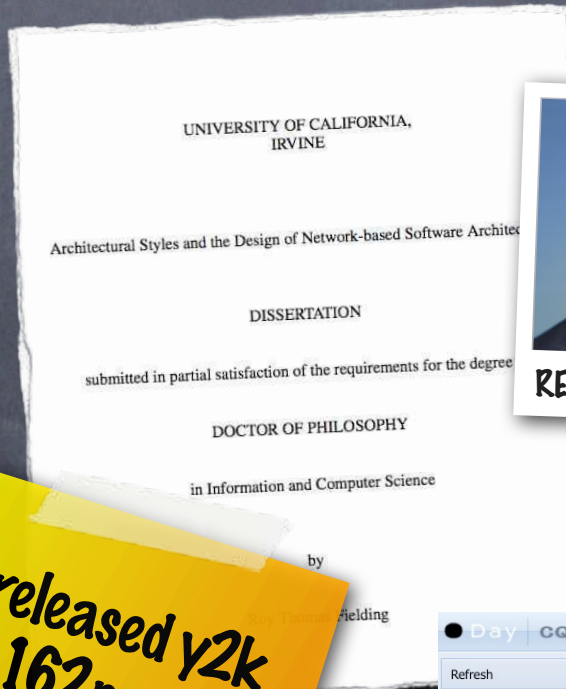


<http://incubator.apache.org/sling>

Join the
fun!

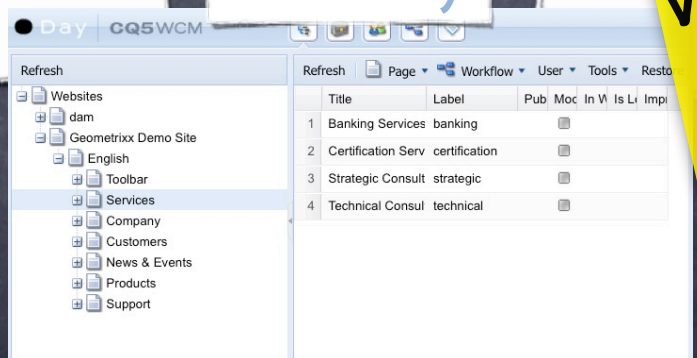


Sling == REST over JCR



released y2k
162 pages

v1.0 released 2005
+300 pages

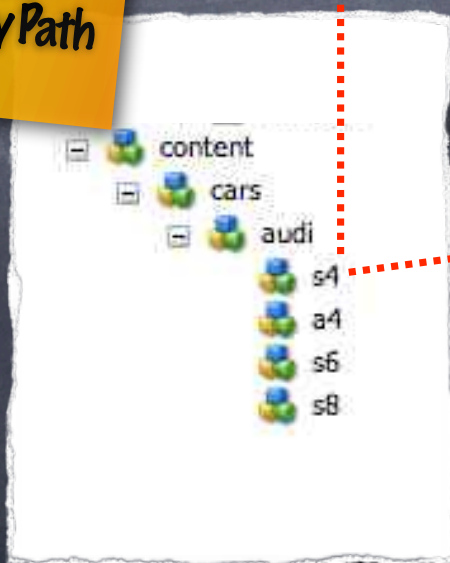


Reclaiming the web. Sling URL decomposition.

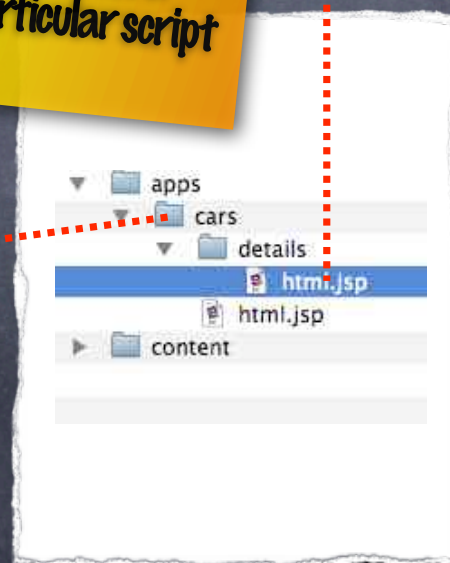
/cars/audi/s4.details.html

Repository

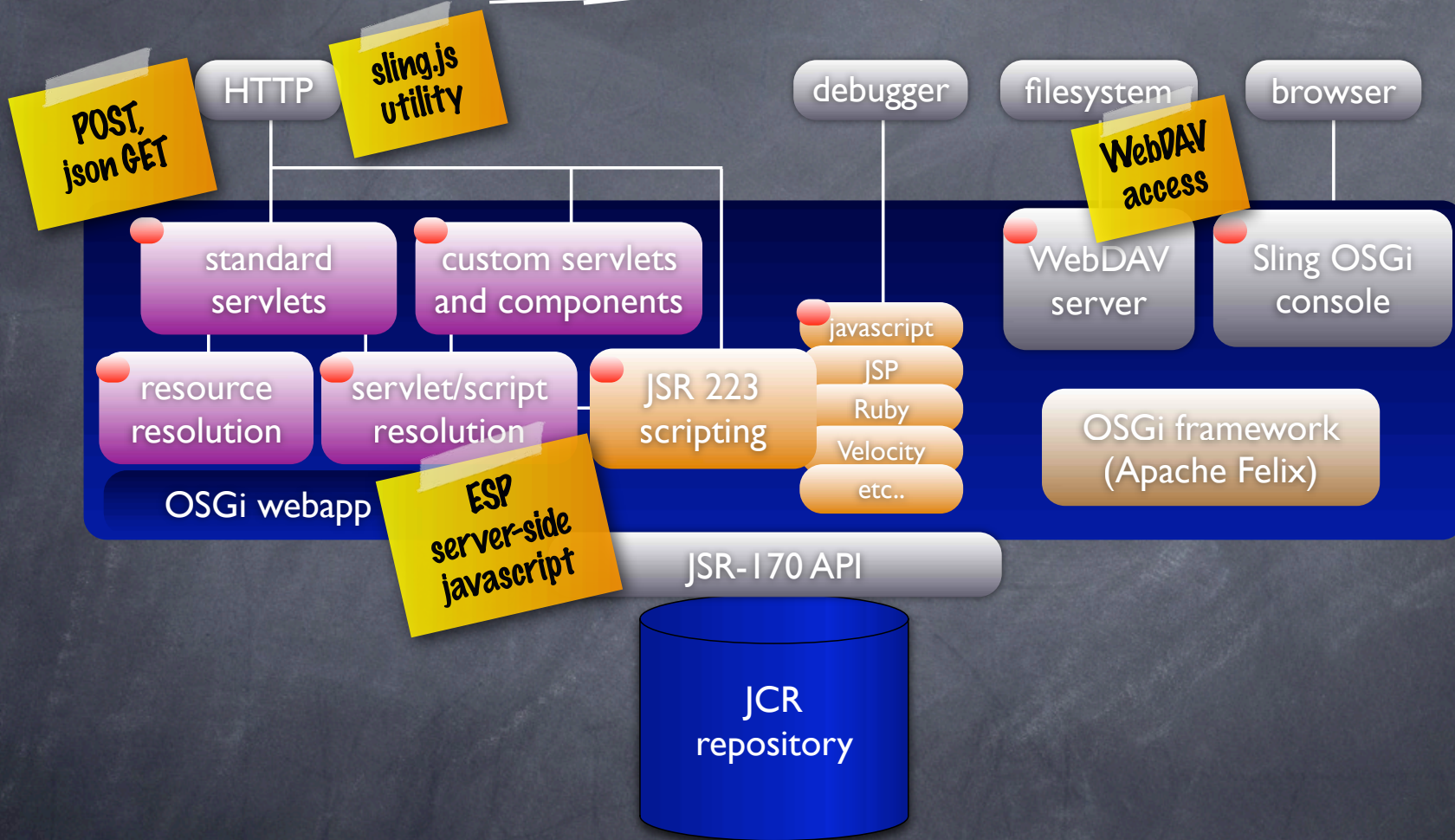
Content
Repository Path



...selects a
particular script



Sling architecture



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

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>
```

Title of the current post

Title:

Text:



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 `slingsamples`
<http://tinyurl.com/slingblogesp>

ESP blog in action

New post

Title

Body

File:

create

Sling ESP blog sample

Hello, ApacheCon!

This is a sample post for my ApacheCon Sling talk

Comments

Bertrand said:

I like it, thanks!

retrieve

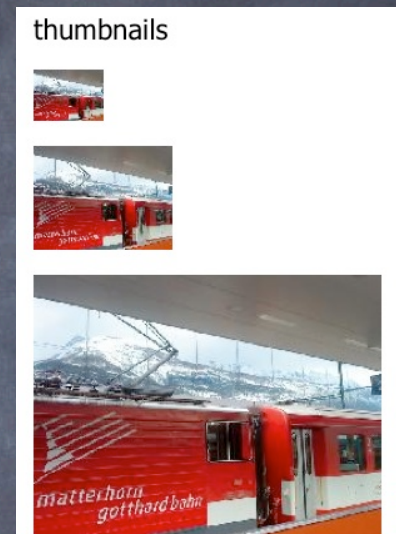
Hello, ApacheCon!

Title

Body

File:

update



thumbnails

Blog admin

Title	Date	Controls
Hello, ApacheCon!	2009-03-20 22:38	View Edit <input type="button" value="Delete"/>

ESPblog demo

ESP blog source files

admin.esp

edit.esp

html.esp

list.esp

menu.esp

xml.esp (RSS feed)

constants.esp

header.esp

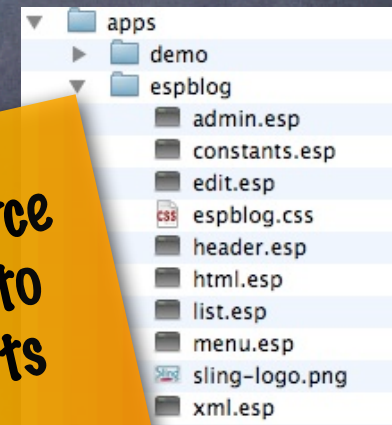
pom.xml

ThumbnailGeneratorService.java

ThumbnailGeneratorServiceImpl.java

esblog.css

sling-logo.png



**esblog resource
type points to
those scripts**

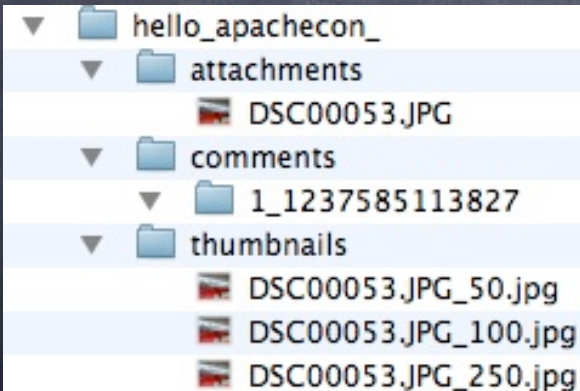
ESP blog content structure

```
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"
    }
  }
},
```

JSON dump

```
- 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"
  }
}
```



A WebDAV view showing a directory tree for a blog post. The root is 'hello_apachecon_'. It contains several sub-directories: 'attachments', 'comments', and 'thumbnails'. The 'attachments' directory contains a file 'DSC00053.JPG'. The 'comments' directory contains a sub-directory '1_1237585113827'. The 'thumbnails' directory contains three files: 'DSC00053.JPG_50.jpg', 'DSC00053.JPG_100.jpg', and 'DSC00053.JPG_250.jpg'.

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);  
        }  
    }  
}
```

...

Where is Sling going?

