Sling Architecture

Felix Meschberger
Day Management AG
8. April 2008
Topics

• Modules and Extension Points
• Request Processing
• Resource and ResourceResolver
• Runtime Framework
• Get Sling
• Questions
### Modules

<table>
<thead>
<tr>
<th>Sling API</th>
<th>jcr/api</th>
<th>sling/core</th>
<th>sling/event</th>
</tr>
</thead>
<tbody>
<tr>
<td>jcr/resource</td>
<td>sling/servlet-resolver</td>
<td>sling/scheduler</td>
<td>sling/threads</td>
</tr>
<tr>
<td>jcr/jackrabbit</td>
<td>sling/servlets-*</td>
<td>sling/threads</td>
<td></td>
</tr>
<tr>
<td>scripting/api</td>
<td>sling/adapter</td>
<td>sling/bundleresource</td>
<td></td>
</tr>
<tr>
<td>scripting/*</td>
<td>sling/i18n</td>
<td></td>
<td></td>
</tr>
<tr>
<td>commons/*</td>
<td>Maven 2 Plugins</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Commons/**

**Maven 2 Plugins**

**osgi/**
Extension Points

- Servlets and Scripts
- Servlet Filters
- ScriptEngine[Factory]
- ResourceProvider
- JcrDefaultResourceTypeProvider
- AuthenticationHandler
- LocaleResolver
- etc.
Main Components

• SlingMainServlet
  – Outermost Request Handler
  – Starts Request Processing

• ResourceResolver
  – Resolves the URL to a Resource

• ServletResolver
  – Resolver the Resource Type to a Servlet/Script
Basic Request Processing Steps

• Resolve the Resource
  – Source: Request URI

• Resolve Servlet or Script
  – Source: Resource Type

• Call Servlet Filters

• Call Servlet or Script
Resource

• Resource is Sling's abstraction of the thing addressed by the request URI

• Properties of Resources
  – Path, e.g. JCR Item path
  – Type, e.g. sling:resourceType
  – Super Type, e.g. sling:resourceSuperType
  – Adapters
  – Metadata, e.g. last modification date
ResourceResolver

- Accesses Resources
- Abstracts the path resolution
- Abstracts access to the Persistence
- Currently there is a 1:1 mapping between the ResourceResolver and a single JCR Session

Tasks:
- Finding Resources
- Getting Resources
- Simplification of Query Execution
Kinds of Resources

- JCR Items (Node, Property)
- Servlets (Registered as OSGi Services)
- Synthetic Resources
- Provided Resources (ResourceProvider)
Servlet Resolution

• Servlets and Scripts are Equal
• Resolution Steps
  – Turn Type of Request Resource to path
    (e.g. sling:redirect ==> sling/redirect)
  – Apply search path
    (e.g. [ ,,/libs“, ,,/apps“ ])
  – Servlet Name from Extension or Method
    (e.g. html.jsp, POST.esp)
Servlet Resolution (Example)

- Search Path : [ „/libs“, „/apps“ ]
- Resource Type : „myapp:sample“
- Request Extension: „html“
- Request Method: „GET“
Servlet Resolution (Example)

/apps/myapp/sample/html[.]*
/libs/myapp/sample/html[.]*
/apps/myapp/sample/GET[.]*
/libs/myapp/sample/GET[.]*
-- above for resource super type
/apps/sling/servlet/default/html[.]*
/libs/sling/servlet/default/html[.]*
/apps/sling/servlet/default/GET[.]*
/libs/sling/servlet/default/GET[.]*
Runtime Framework: Heritage

- Sling stems from Communiqué 4
- Communiqué 4 status
  - Some modularisation
  - Incomplete Lifecycle Support
  - Problematic Quick Fixing
  - Restarts required often
Runtime Framework: Requirement

• Modularization
• Dependency Management
  – Code
  – Services
• Lifecycle Management
• Dynamic System Changes
• Configuration Management
Runtime Framework: OSGi

• Core
  – Modularization – Capabilities & Requirements
  – Lifecycle – Install, Start, Stop, Update, Uninstall
  – Services – Get, Use, Unget
  – (Security – JAAS based, not used by Sling)

• Compendium
  – Configuration Admin Service
  – Declarative Services
How is Sling Delivered?

- OSGi Bundles
- Executable JAR File
- Web Application Archive
But: Parts of Sling are Static!

- The Launcher (5 simple classes)
- The OSGi Framework implementation
- The OSGi core and compendium libraries

- Total: ca. 720KB

- Everything else is a Bundle
Does Sling require the Launcher?

- No
- Sling can be deployed in any compliant OSGi R4 framework.
  - e.g. Integration of Sling into the ServiceMix 4 Framework instance
  - e.g. Equinox