

Sling-based JCR explorer

Implementations

Two implementations exist:

- [jQuery based. https://issues.apache.org/jira/browse/SLING-993](https://issues.apache.org/jira/browse/SLING-993) contains a screenshot.
- [GWT based. https://issues.apache.org/jira/browse/SLING-990](https://issues.apache.org/jira/browse/SLING-990) contains a screenshot.

While I was able to build the jQuery based bundle using Maven but I got errors for the GWT based one.

There also exists [JCR-Explorer](#) but that software seems to be somewhat dead. To use it with Sling see: [Using the JCR-Explorer with Sling](#) and [How to run the JCR Explorer with Sling?](#).

Requirements

Here's a quick and dirty spec for a JCR Explorer, feel free to expand/refine!

More or less in priority order:

Visualize the nodes of the JCR repository in a tree view, and/or in a search view with a search box at the top and a list of matching Nodes below.

Display all the details of a Node or Property: type, mixins, value, and later locking status, versions, etc.

Provide RESTful URLs to access each Node and Property.

Allow Nodes and Properties to be created, updated, deleted, and moved around the tree.

When creating a Node, allow its nodetype to be specified.

Allow mixins to be added to a Node later on.

Handle the JCR Session save/abort concept, where a number of changes to Nodes and Properties are only saved when explicitly actioning a "save" command.

Later, use OSGi-based plugins to edit Property values.

The JCR explorer has to be usable for pure JCR repositories (non-Sling), even though the explorer itself would run under Sling (use cases : display the list of available repositories, add/delete/update a new JCR repo, ...).

Technical Issues

Select a good UI framework : JQuery, YUI or GWT ?

I would vote for Dojo. It has a `dojo.data` API (similar purpose to JDO/JPA+beansbinding), that can be used to connect to Sling using HTTP, and to the widgets using data binding. Hopefully Sling can also expose a "standard protocol" that can be used with Dojo's `JsonRestStore`.

Example of a Dojo app, using `dojo.data` API, backend by a REST store, using SOFEA approach: <http://www.persvr.org/explorer.html>

~ Hendy Irawan