CQ Community
Meet, ask, and follow the best in the CQ community

CQ coding patterns: Sling vs JCR (part 2)

November 22, 2012

- Jörg Hoh

In the last posting I showed the benefits of Sling regarding resource access over plain JCR. But not only in resource access both frameworks offer similar functionality, but also in the important area of listening to changes in the repository. So today I want to compare JCR observation to Sling eventing.

JCR observation is a part of the JCR specification and is a very easy way to listen for changes in the repository.

```
@component (immediate=true, metatype=false)
@service
class Listener implements ObservationListener {

  @Reference
  SlingRepository repo;

  Session session;
  Logger log = LoggerFactory.getLogger (Listener.class);

  @activate
  protected void activate () {
    try {
      Session adminSession = repo.loginAdministrative(null);
      session = adminSession.impersonate (new SimpleCredentials("author",new char[0]));
      adminSession.logout();
      adminSession = null;
      session.getObservationManager.addEventListener( this, // listener
        NODE_CREATED|NODE_DELETED|NODE_MOVED, // eventTypes
        "/", // absPath
        true, // isDeep
        null, // uuid
        null, //nodeTypeNames
        true // noLocal
      );
    } catch (RepositoryException e) {
      log.error ("Error while registering observation", e);
    }
  }

  @deactivate
  protected void deactivate() {
    session.getObservationManager.removeListener(this);
    session.logout();
    session = null:
  }

  private handleEvents (Events events) {
    while (events.hasNext()) {
      Event e = events.next();
      … // do here your event handling
     }
  }
}
```

In JCR the creation of an observation listener is straight forward, also the event listening. The observation process is tied to the lifetime of the session, which is started and stopped at activation/deactivation of this sling service. This kind of implementation is a common pattern.

…

Read the complete blog post at http://cqdump.wordpress.com/2012/11/13/cq-coding-patterns-sling-vs-jcr-part-2/

Adobe CQ Social Post

Tags

Adobe Adobe AEM Adobe CQ Amazon Apache AWS backup best practices Cloud Manager community CQ CRX custom widget DAM Dan Klco David McMahon Day Day. CQ Day Software developer development Digital Marketing disk space Dispatcher error feature pack FileJournal framework hotfix integration issue JCR Jörg Hoh OSGi performance programming Rackspace replication Samartha Vashishtha Scott Macdonald Sensei Martin SLING troubleshooting using WEM

Helpful CQ resources

- Adobe CQ Documentation
- The *Experience Delivers* blog
- Adobe CQ forum

Categories

- Announcement
- Best practices
- Cloud Manager
- Developer
- FAQ
- How To
- Performance
- Samples
- Tips and Tricks
- Top Questions
- Troubleshooting
- Uncategorized
- Workflows

Recent Posts

- Launch the new Digital Marketing Support Community
- Submitting Mobile Data to Adobe CQ using a Sling Post Servlet
- Developing a Spring MVC project using Maven, JSTL tags, and data persistence
- AEM5.6: Socoadmin: reply button and refresh issues in moderation UI

Custom CQ5 workflow step that integrates Twitter and Jabber

Tags: CQ, JCR, Jörg Hoh, SLING

Comments are closed.

« Previous Post

Next Post »