How to Create a new JAX-RS resource

This is a brief introduction to wiring in additional JAX-RS resources into Fedora.

Boilerplate

```
@Component
@Scope("prototype")
@Path("/{path: .*}/fcr:text")
public class MyCustomTextExtractionEndpoint extends AbstractResource {

    @InjectedSession
    protected Session session;

    private final Logger logger = getLogger(FedoraContent.class);
}
```

We make extensive use of component scanning in Fedora 4, in order to make it easy to drop in new resources. While you could wire your component into Spring manually, consider marking it as a "@Component".

Our REST API resources are marked as "@Scope("prototype")" in order to inject the current JCR session (within a transaction, workspace, or otherwise) as a field.

Fedora 4 uses a "globbing" style path. The path to the node (object or datastream) comes first, and the action to perform is last. The "@Path" annotation will collect the best-match path up to your static part ("fcr:text").

- Note, for consistency with other Path suffixes, use all lowercase strings (e.g. use: "fcr:nodetype" instead of: "fcr:nodeType")

Try to make extensive use of TRACE-level logging (in addition to inline comments and javadoc comments, of course) to help expose the internal workings of your class in a sensible way to downstream users.

An Action

Here is an example API action:

```
 /**
     * Create an anonymous DS with a newly minted name
     * and content from request body
     * @param pathList
     * @throws RepositoryException
     */
    @GET
    public String extractText(@PathParam("path")
    final List<PathSegment> pathList) {
       String path = toPath(pathList);

       try {
         final FedoraResource resource =
                    nodeService.getObject(session, path);

         // do stuff
         return "";

       } finally {
         session.logout();
       }

    }
```

The "pathList" contains the full REST path up to the static part of the request. Fedora's AbstractResource provides the helper method "toPath" to turn that into a node path. This path can then be used in any of Fedora's Services (or even direct JCR access, although we discourage reaching that far down the stack from a REST endpoint, and instead break the functionality into the high-level REST API and some low-level services).

It is important that the JCR session is logged out after we're done with it. To do this, we wrap the guts of the action in a try/finally block.