Generic Use Case: Extensions implemented as JAX-RS resources

Title (Goal)	As a developer, I would like to implement an extension as a JAX-RS resource that the API Extension Architecture manages as part of deploying my extension
Primary Actor	Developer
Scope	
Level	
Author	Aaron Birkland
Story (A paragraph or two describing what happens)	As a developer, I have created a simple API extension that can be implemented as a JAX-RS resource and a few additional support classes. While the API extension architecture provides a means to wire in a web service to implement an extension, I don't want to go through the pain of developing en entire webapp, nor deploying and maintaining a live webapp it when the extension is live. Instead, I'd like the API Extension architecture to manage the lifecycle of my JAX-RS resource, and direct appropriate requests to it. When packaging my extension for distribution, I'd like to include my classes as part of the packaged extension. This will make it extremely simple for users to evaluate and deploy my extension; they won't have to separately deploy an implementing webapp. The API Extension architecture itself will take care of running my JAX-RS resource automatically, without effort on the part of the person deploying my extension.

API Extension Architecture's role:

- Deploy an extension's JAX-RS resource, and associated classes
- Direct user requests from the public URI (/path/to/object/**ext:segment**) to the JAX-RS resource
- Provide a specification or guidelines for distributing code as part of a packaged extension.

Developer's role:

- Write the JAX-RS resource which implements an extension
- Package it for distribution