SCR Annotations for Fun and Profit

Lightningtalks.adaptTo(Berlin, 2012)

Mike Pfaff
Freelance CQ Consultant

mikepfaff83@gmail.com

Huh?

- Short intro on how to use SCR annotations to easily define servlets and services, use references to access services, define configurable components, etc.

- Show ease of "new style" annotations compared to "old style" JavaDoc comments

How to: Use annotations

```
...
<dependencies>
    ...
    <dependency>
        <groupId>org.apache.felix</groupId>
        <artifactId>org.apache.felix.scr.annotations</artifactId>
        <version>1.7.0</version>
        <scope>provided</scope>
    </dependency>
    ...
</dependencies>
...
```

How to: Maven plugin

```
...
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.felix</groupId>
            <artifactId>maven-scr-plugin</artifactId>
            <version>1.8.0</version>
        </plugin>
        ...
    </plugins>
</build>
...
```

Reference: Old Style

```
...

/**
* @scr.reference
*/
private SlingRepository repo;

...
```

Reference: New Style

```
...

@Reference
private SlingRepository repo;

...
```

Servlet: Old Style

```
/**
 * Foo Bar Servlet
 *
 * @scr.component metatype="false"
 * @scr.service interface="javax.servlet.Servlet"
 * @scr.property name="sling.servlet.methods" value="POST"
 * @scr.property name="sling.servlet.selectors" value="foobar"
 * @scr.property name="sling.servlet.resourceTypes" value="adaptto2012/
components/page"
 **/
public class FooBarServlet extends SlingAllMethodsServlet {

    ...

}
```

Servlet: New Style

```java
/**
 * Foo Bar Servlet
 **/

@SlingServlet(
        resourceTypes = {"adaptto2012/components/page"},
        methods       = {"POST"},
        selectors     = {"foobar"}
)
public class FooBarServlet extends SlingAllMethodsServlet {

    ...

}
```

SlingServlet options

- Based on resourceType (recommended!)
  - resourceTypes - Required
  - selectors - Optional
  - extensions - Optional
  - methods - Optional
- Based on path (caution - no access control!)
  - paths - Required

Configurable Component: Define

```java
@Component(
    label = "adaptTo() - Sample Configurable Component",
    description = "This is a sample for a configurable component",
    metatype = true
)
public class SampleConfigurableComponent {
    @Property(
        label = "Hostname",
        description = "Hostname of some external service"
    )
    static final String PROP_HOSTNAME = "hostname";
}
```

Configurable Component: Configure

Configuration: Retrieve

```
protected void activate(ComponentContext context) {
    Dictionary configProperties = context.getProperties();
    String hostName = (String)configProperties.get(PROP_HOSTNAME);
}
```

Other annotations

Define a service (which you can then @Reference)
    @Service

Sling filters:
    @SlingFilter
    @SlingFilterScope

Override default method names:
    @Activate
    @Deactivate
    @Modified

Some best practices

- Only set metatype to true if you want your component to be configurable through the Apache Felix web console. Defaults are false for annotations (so better than JavaDoc tags)

- Always set nice labels and descriptions to avoid cryptic entries in the Apache Felix web console (also applies to non configurable components!)

Documentation

- SCR Annotations
  http://felix.apache.org/site/scr-annotations.html

- Sling SCR Annotations
  http://felix.apache.org/site/extending-scr-annotations.html

- Maven Plugin
  http://felix.apache.org/site/apache-felix-maven-scr-plugin.html

Thank you!