### OSGI R6 annotations in AEM 6.3

For all code examples, go to the following github URL

<https://github.com/nateyolles/aem-osgi-annotation-demo/tree/master/core/src/main/java/com/nateyolles/aem/osgiannotationdemo/core>

From AEM 6.2, Apache Felix Maven SCR Plugin has been deprecated and in place of this, Maven Bundle Plugin (version 6.2 or greater) has been introduced.

**How OSGi plugin has replaced all SCR annotations in AEM?**

Now the package “**org.apache.felix.scr.annotations.\***” will be replaced with “**org.osgi.service.component.annotations.\***” and “**org.osgi.service.metatype.annotations.\***”

While using Maven SCR plugin, we can find the Declarative Service(DS) output under /target/classes/OSGi-INF but using OSGi Bundle Plugin our DS output are found in /OSGI-INF, packaged under the compiled JAR File.

**Note**: The [Apache Felix SCR annotations](http://felix.apache.org/documentation/subprojects/apache-felix-maven-scr-plugin/scr-annotations.html) are replaced in AEM 6.3 projects with OSGi R6 annotations

**How To start with OSGi Annotations?**

**1**. Add a plugin in Parent Pom.xml

|  |
| --- |
| <plugin>  <groupId>org.apache.felix</groupId>  <artifactId>maven-bundle-plugin</artifactId>  <version>3.2.0</version>  <inherited>true</inherited> </plugin> |

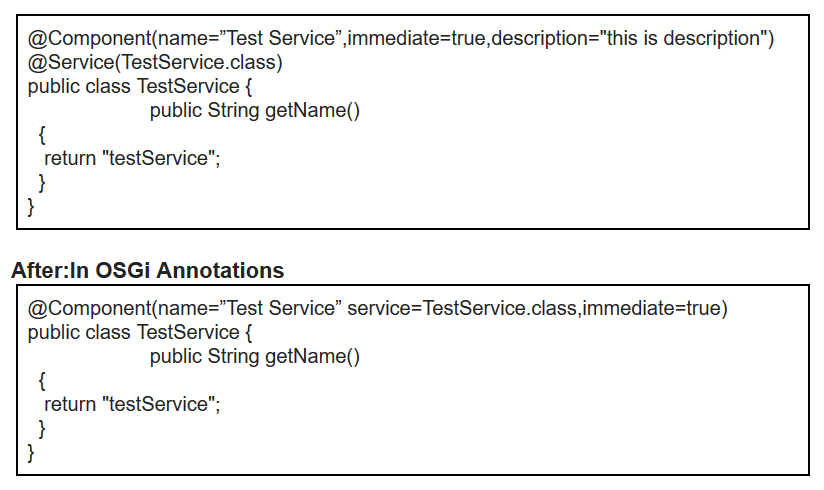
**2**. Add the following maven dependencies.

|  |
| --- |
| <dependency>  <groupId>org.osgi</groupId>  <artifactId>org.osgi.service.component.annotations</artifactId>  <version>1.3.0</version> </dependency>  <dependency>  <groupId>org.osgi</groupId>  <artifactId>org.osgi.annotation</artifactId>  <version>6.0.0</version> </dependency>  <dependency>  <groupId>org.osgi</groupId>  <artifactId>org.osgi.service.metatype.annotations</artifactId>  <version>1.3.0</version> </dependency>  <dependency>  <groupId>org.osgi</groupId>  <artifactId>org.osgi.compendium</artifactId>  <version>4.2.0</version> <scope>provided</scope> </dependency> |

**Note**: Make sure that while building the code, plugin is working fine.

**Migration of OSGi Components and Service**

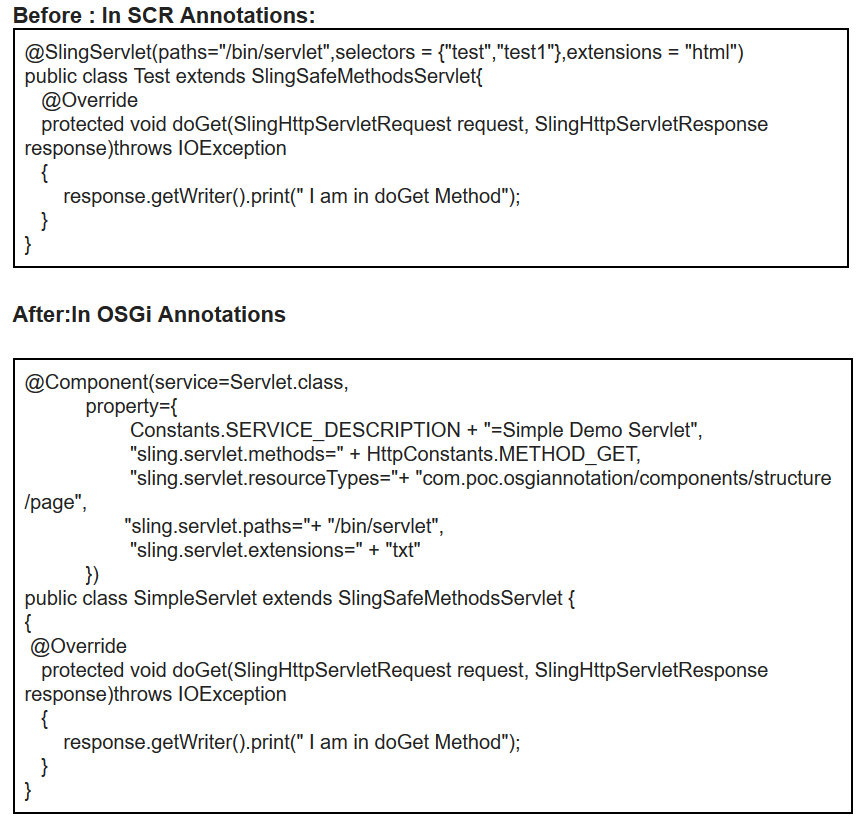
**Before: In SCR Annotations:**



**In the Sling Servlets:**

We used to use @Component, @Service @SlingServlet @Properties in SCR Annotations.

OSGi annotation just have @Component with the collaboration of all these annotations.

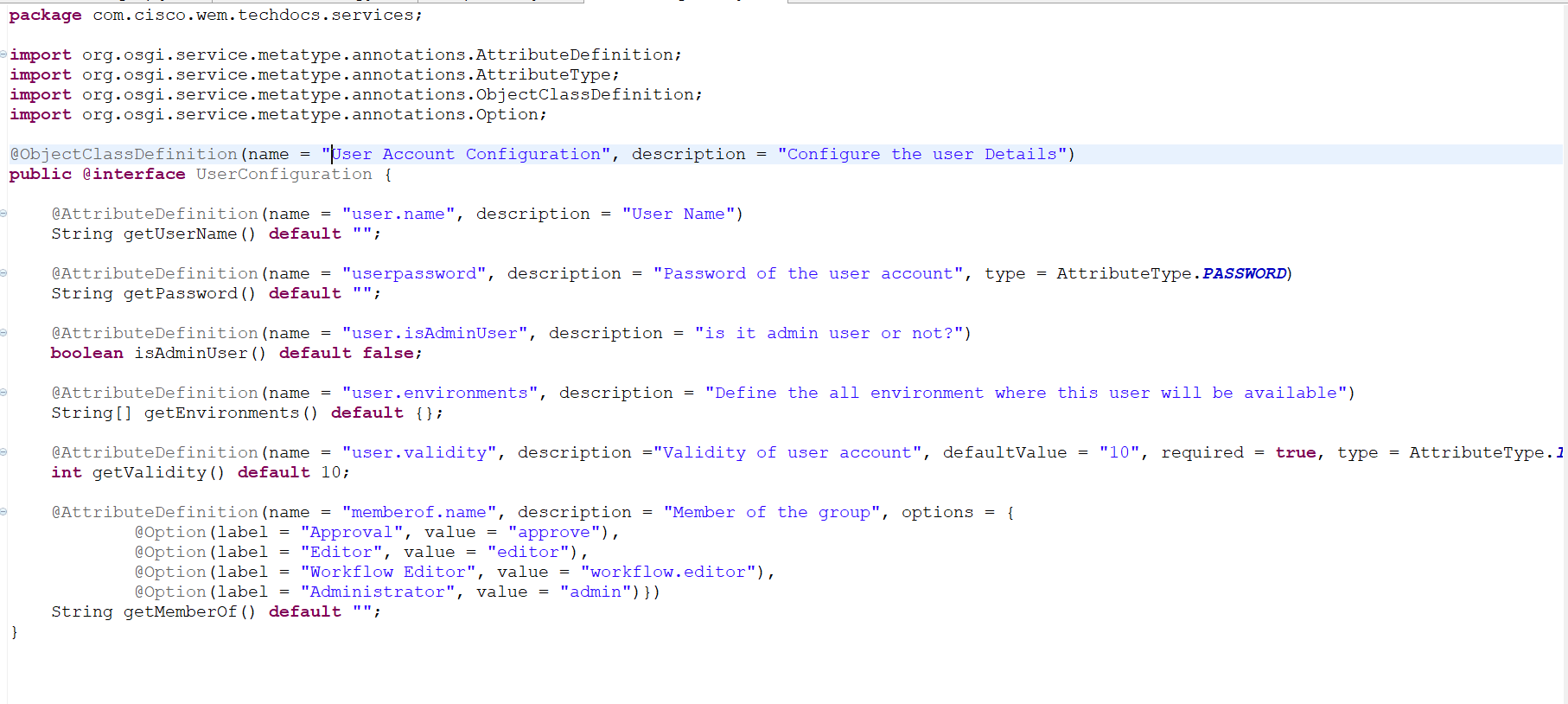


**Custom OSGi Configuration in AEM**

OSGi annotations provided a flexibility to create configuration in a separate interface and we can use it in any place.

The OSGi Configurations generates metatype specification that provides a way to describe configuration of the component. Such a metatype information can be used at runtime to generate forms to edit/create the configuration of a component. For example the Apache Felix web console does exactly that.

**UserConfiguration.java**



**UserAccountDetail.java**

**package** com.cisco.wem.techdocs.serviceimpl;

**import** org.osgi.service.component.annotations.Activate;

**import** org.osgi.service.component.annotations.Component;

**import** org.osgi.service.component.annotations.Modified;

**import** org.osgi.service.metatype.annotations.Designate;

**import** com.cisco.wem.techdocs.services.UserConfiguration;

@Component(service=UserAccountDetail.**class**,immediate=**true**)

@Designate(ocd = UserConfiguration.**class**)

**public** **class** UserAccountDetail {

**private** String userName;

**private** String password;

**private** Boolean isAdminUser;

**private** String memberOf;

**private** **int** validity;

**private** String[] environments;

@Activate @Modified

**protected** **void** activate(**final** UserConfiguration userConfiguration) {

userName = userConfiguration.getUserName();

password = userConfiguration.getPassword();

isAdminUser = userConfiguration.isAdminUser();

memberOf = userConfiguration.getMemberOf();

validity = userConfiguration.getValidity();

environments = userConfiguration.getEnvironments();

}

/\*\*

\* **@return** the userName

\*/

**public** String getUserName() {

**return** userName;

}

/\*\*

\* **@return** the password

\*/

**public** String getPassword() {

**return** password;

}

/\*\*

\* **@return** the isAdminUser

\*/

**public** Boolean getIsAdminUser() {

**return** isAdminUser;

}

/\*\*

\* **@return** the memberOf

\*/

**public** String getMemberOf() {

**return** memberOf;

}

/\*\*

\* **@return** the validity

\*/

**public** **int** getValidity() {

**return** validity;

}

/\*\*

\* **@return** the environments

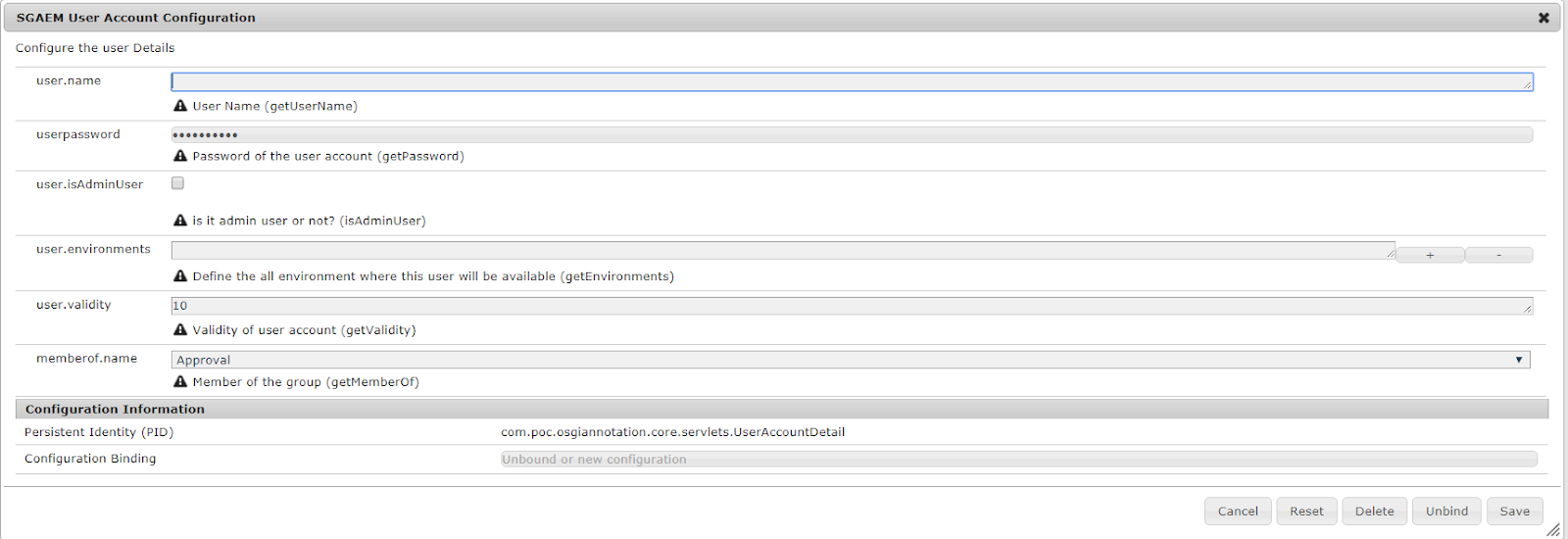
\*/

**public** String[] getEnvironments() {

**return** environments;

}

}



**Migration of Custom Workflow Process Step**

