



# Adobe@Adobe

How Adobe uses its own technology to run its business

[HOME](#) CQ Tips and Tricks #1 – How to define a SlingServlet [CQ5.5-5.6]

## CQ Tips and Tricks #1 – How to define a SlingServlet [CQ5.5-5.6]

There are many ways to define an OSGI service.

This is the preferred coding style for CQ version 5.5-5.6 that we follow on our team.

Registering the servlet by path

```

1  import org.apache.felix.scr.annotations.Properties;
2  import org.apache.felix.scr.annotations.Property;
3  import org.apache.felix.scr.annotations.sling.SlingServlet;
4  import org.apache.sling.api.SlingHttpServletRequest;
5  import org.apache.sling.api.SlingHttpServletResponse;
6  import org.apache.sling.api.servlets.SlingAllMethodsServlet;
7
8  import javax.servlet.ServletException;
9  import java.io.IOException;
10
11  @SlingServlet(
12      paths={"/services/unicom/v1/"}
13  )
14  @Properties({
15      @Property(name="service.pid", value="com.adobe.unicom.v1.servlets.OmnitureLoggingServlet",propertyPrivate=true),
16      @Property(name="service.description",value="Omniture service call logging servlet", propertyPrivate=false),
17      @Property(name="service.vendor",value="Adobe Systems Incorporated - Adobe@Adobe Team", propertyPrivate=true)
18  })
19  public class OmnitureLoggingServlet extends SlingAllMethodsServlet
20  {
21      @Override
22      protected void doGet(SlingHttpServletRequest request, SlingHttpServletResponse response) throws ServletException
23      {
24          //Do something fun here
25      }
26
27      @Override
28      protected void doPost(SlingHttpServletRequest request, SlingHttpServletResponse response) throws ServletException
29      {
30          //Do something fun here
31      }
32  }

```

OR registering the servlet by resource type and extension

```

1  import org.apache.felix.scr.annotations.Properties;
2  import org.apache.felix.scr.annotations.Property;
3  import org.apache.felix.scr.annotations.sling.SlingServlet;
4  import org.apache.sling.api.SlingHttpServletRequest;
5  import org.apache.sling.api.SlingHttpServletResponse;
6  import org.apache.sling.api.servlets.SlingAllMethodsServlet;
7
8  import javax.servlet.ServletException;
9  import java.io.IOException;
10
11  @SlingServlet(

```

```

11 @SlingServlet(
12     resourceTypes = {"rep:User"},
13     methods = {"GET", "POST"}
14 )
15 @Properties({
16     @Property(name="service.pid", value="com.adobe.unicom.v1.servlets.OmnitureLoggingServlet",propertyPrivate=true),
17     @Property(name="service.description",value="Omniture service call logging servlet", propertyPrivate=false),
18     @Property(name="service.vendor",value="Adobe Systems Incorporated - Adobe@Adobe Team", propertyPrivate=false)
19 })
20 public class OmnitureLoggingServlet extends SlingAllMethodsServlet
21 {
22     @Override
23     protected void doGet(SlingHttpServletRequest request, SlingHttpServletResponse response) throws ServletException
24     {
25         //Do something fun here
26     }
27
28     @Override
29     protected void doPost(SlingHttpServletRequest request, SlingHttpServletResponse response) throws ServletException
30     {
31         //Do something fun here
32     }
33 }

```

To handle all requests and filter on selector you can change the example above to read something like this.

```

1 import org.apache.felix.scr.annotations.Properties;
2 import org.apache.felix.scr.annotations.Property;
3 import org.apache.felix.scr.annotations.sling.SlingServlet;
4 import org.apache.sling.api.SlingHttpServletRequest;
5
6 import org.apache.sling.api.SlingHttpServletResponse;
7 import org.apache.sling.api.servlets.SlingAllMethodsServlet;
8
9 import javax.servlet.ServletException;
10 import java.io.IOException;
11
12 @SlingServlet(
13     resourceTypes = {"sling/servlet/default"},
14     methods = {"GET"},
15     selectors = {"report"},
16     extensions = {"json"}
17 )
18 @Properties({
19     @Property(name="service.pid", value="com.adobe.unicom.v1.servlets.OmnitureLoggingServlet",propertyPrivate=true),
20     @Property(name="service.description",value="Omniture service call logging servlet", propertyPrivate=false),
21     @Property(name="service.vendor",value="Adobe Systems Incorporated - Adobe@Adobe Team", propertyPrivate=false)
22 })
23 public class OmnitureLoggingServlet extends SlingAllMethodsServlet
24 {
25     .....
26 }

```

For more details on the Felix Annotations see

## 2 Responses to CQ Tips and Tricks #1 – How to define a SlingServlet [CQ5.5-5.6]



Scott Brodersen says:

January 16, 2013 at 3:52 pm

It seems to me that the paths property of the @SlingServlet annotation overrides the resourceTypes and methods properties:

The javadoc for the SlingServlet annotation [0] says that resourceTypes and paths are converted to sling.servlet.resourceTypes and sling.servlet.paths properties, respectively.

The Sling documentation [1] states that if a value for sling.servlet.paths is specified, the resourceTypes property and the methods property are ignored.

regards,  
scott

[0] <http://svn.apache.org/repos/asf/felix/trunk/src/plugin/annotations/src/main/java/org/apache/felix/sr/annotations/sling/SlingServlet.java>

[1] <http://sling.apache.org/site/servlets.html>



**dbenge** says:

January 17, 2013 at 4:41 am

Thanks for the correction. I missed that. I have updated the examples with the corrections you outlined.

By dbenge

CQ (9) Felix (1) OSGI (1)

Sling (2)



Comments (2)

Created

September 14, 2012

[Careers](#)

[Permissions & Trademarks](#)

[EULAs](#)

[Report Piracy](#)

[Contact Adobe](#)

[Security](#)

Copyright © 2014 Adobe Systems Incorporated. All rights reserved.

[Terms of Use](#) | [Privacy Policy and Cookies](#) (Updated)