

# cviewer-javaee6 - Programmatically register servlets and filters

## Application Overview

The cviewer-javaee6 sample includes a JSP file that calls a servlet. A form allows the user to enter a java class name. Then a filter will catch the class name and turn null or invalid class name into fully qualified class name. The servlet uses java reflection to extract information about the class which, in turn, is returned to the user.

This sample will demonstrate the programmatically adding and configuring servlets and filters features in Servlet 3.0.

The ability to programmatically add a servlet to a context is useful for framework developers. A framework could declare a controller servlet using this method. The return value of this method is a `ServletRegistration.Dynamic` object further allows you to set up the other parameters of the servlet like init-params, url-mappings during `ServletContext` initialization.

## Application Content

The cviewer application consists of following list of packages and classes.

### `org.apache.geronimo.samples.javaee6.cviewer`

- **CviewerFilter.java** It works as a class name filter. The filter will get the class name string inputted by user from console and then get the `Class` object associated with the class with the given string name. If the user inputted nothing, this filter will change class name to "java.lang.Integer". If the class name is invalid, this filter will change class name parameter to "java.lang.String" and give prompting message.
- **CviewerServlet.java** It displays the detail information of the class you are querying, including classname, super classes, implemented interfaces, methods and so on.
- **CviewerServletListener.java** It works as a web listener and will programmatically register a servlet and filter.

The list of web application files in the application is depicted in the following.

```
|- WEB_INF
    +- web.xml
    +- geronimo-web.xml
|- index.html
|- header.html
|- sample-docu.jsp
```

## Application Implementation

`geronimo-web.xml` specifies the module's information and the url for the web application

Information about the project such as module's unique identification, dependencies is described inside the `<sys:environment/>` tags. It is a good practise to give a module an unique identification, so that it can later be referenced by some other deployable application. This module is in the group `org.apache.geronimo.samples`. The path specified in the `<context-root>` tag will be the entry point of this web application. Therefore you can access this web application at `http://<hostname>:<port>/cviewer-javaee6`.

`xmlgeronimo-web.xml`

`CviewerServletListener.java` is a web listener which implements `ServletContextListener`, it will programmatically register a servlet and filter.

- **@WebListener**  
The `WebListener` annotation is used to annotate a listener to get events for various operations on the particular web application context. The `CviewerServletListener` class annotated with `@WebListener` implements the interface `javax.servlet.ServletContextListener`.
- **addServlet("ClassViewer", "org.apache.geronimo.samples.javaee6.cviewer.CviewerServlet")**  
The first parameter `ClassViewer` is the servlet name, the second parameter `org.apache.geronimo.samples.javaee6.cviewer.CviewerServlet` is a servlet class. The `addServlet` method allows the application to declare the `ClassViewer` servlet programmatically. It adds the servlet with the given name and servlet instance to the servlet context.
- **addFilter("CviewerFilter", "org.apache.geronimo.samples.javaee6.cviewer.CviewerFilter");**  
The first parameter `CviewerFilter` is the filter name and the second parameter `org.apache.geronimo.samples.javaee6.cviewer.CviewerFilter` is the corresponding filter class name. The `addFilter` method allows the application to declare a filter programmatically. It adds the filter with the given name and class name to the web application.

`javaCviewerServletListener.java`

## Get Source Code

Please reference [Samples General Information](#) for information on obtaining and building the source for this and other samples.

## Build the web application

Once all the sources get checked out the next step is to build **cvviewer-javaee6**. It requires Maven 2 for building the binaries.

From the **<cvviewer-javaee6\_home>** directory run the following command.

```
mvn clean install
```

This process will take a couple of minutes. The binaries will be generated in the corresponding **target** directory .

## Deploy the web application

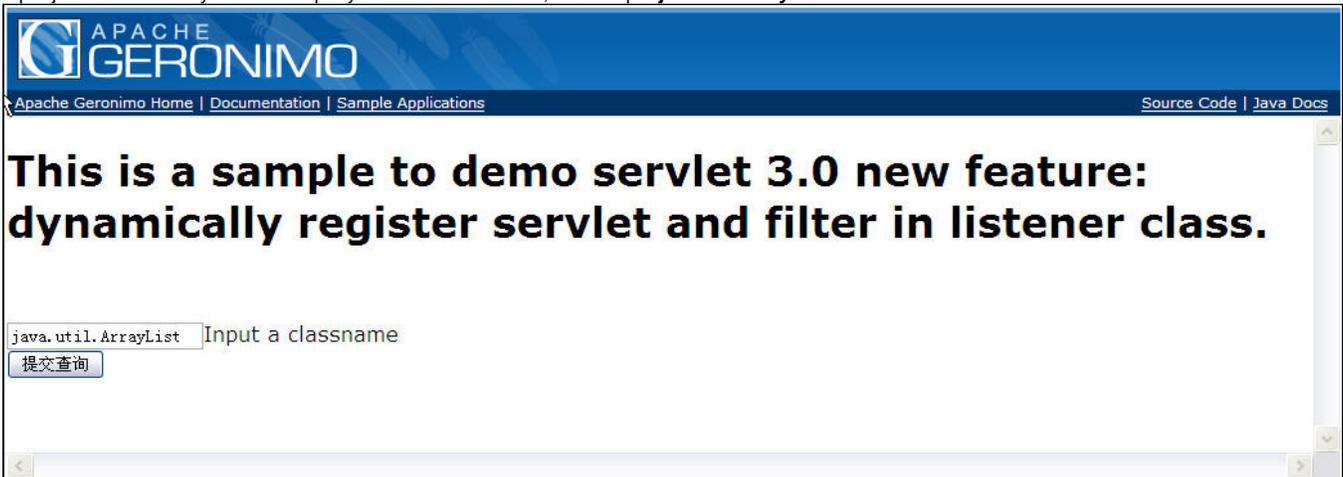
Deploying sample application is pretty straight forward as we are going to use the Geronimo Console.

1. Scroll down to **Deploy New** from the Console Navigation panel.
2. Load **cvviewer-javaee6-war-3.0-M1.war** from **cvviewer-javaee6-war/target** folder in to the **Archive** input box.
3. Press **Install** button to deploy application in the server.

## Test the web application

The app is visible at <http://localhost:8080/cvviewer-javaee6/>

Input java class name you want to query for detail information,for example **java.util.ArrayList**.



If the class name is null or in bad format,filter will change it to **java.lang.Integer** or **java.lang.String** by default.The servlet will display detail information of the class.

