

Java Web Fundamentals

Writing Servlets

Kevin Jones
@kevinrjones



pluralsight 
hardcore dev and IT training

Servlet Interface



```
public interface Servlet
{
    public void init(ServletConfig);
    public void service(ServletRequest req,
                       ServletResponse resp);
    public void destroy();
    public ServletConfig getServletConfig();
    public String getServletInfo();
}
```

GenericServlet Class

```
public abstract class GenericServlet
    implements Servlet,
        ServletConfig,
        java.io.Serializable {
}
```

HttpServlet Class

```
public class HttpServlet extends GenericServlet {
    public void service(ServletRequest req, ServletResponse resp) {
        service((HttpServletRequest)req, (HttpServletResponse)resp);
    }

    public void service(HttpServletRequest req, HttpServletResponse resp) {
        String verb = req.getMethod();
        if (verb.equals("GET")) doGet(req, resp);
        else if (verb.equals("POST")) doPost(req, resp);
        else ...
    }

    public void doGet(HttpServletRequest req, HttpServletResponse resp)
    {...}

    public void doPost(HttpServletRequest req, HttpServletResponse resp)
    {...}
    ...
}
```

HttpServlet Class

```
public class HttpServlet extends GenericServlet {
    public void service(ServletRequest req, ServletResponse resp) {
        service((HttpServletRequest)req, (HttpServletResponse)resp);
    }

    public void service(HttpServletRequest req, HttpServletResponse resp) {
        String verb = req.getMethod();
        if (verb.equals("GET")) doGet(req, resp);
        else if (verb.equals("POST")) doPost(req, resp);
        else ...
    }

    public void doGet(HttpServletRequest req, HttpServletResponse resp)
    {...}

    public void doPost(HttpServletRequest req, HttpServletResponse resp)
    {...}
    ...
}
```

HttpServlet Class

```
public class HttpServlet extends GenericServlet {
    public void service(ServletRequest req, ServletResponse resp) {
        service((HttpServletRequest)req, (HttpServletResponse)resp);
    }

    public void service(HttpServletRequest req, HttpServletResponse resp) {
        String verb = req.getMethod();
        if (verb.equals("GET")) doGet(req, resp);
        else if (verb.equals("POST")) doPost(req, resp);
        else ...
    }

    public void doGet(HttpServletRequest req, HttpServletResponse resp)
    {...}

    public void doPost(HttpServletRequest req, HttpServletResponse resp)
    {...}
    ...
}
```

HttpServlet Class

```
public class HttpServlet extends GenericServlet {  
    public void service(ServletRequest req, ServletResponse resp) {  
        service((HttpServletRequest)req, (HttpServletResponse)resp);  
    }  
  
    public void service(HttpServletRequest req, HttpServletResponse resp) {  
        String verb = req.getMethod();  
        if (verb.equals("GET")) doGet(req, resp);  
        else if (verb.equals("POST")) doPost(req, resp);  
        else ...  
    }  
  
    public void doGet(HttpServletRequest req, HttpServletResponse resp)  
    {...}  
  
    public void doPost(HttpServletRequest req, HttpServletResponse resp)  
    {...}  
    ...  
}
```

How to Implement a Servlet

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

public class OurServlet extends HttpServlet {
    public init() {
        // Perform any instance initialisation
    }

    public void doGet(HttpServletRequest req,
        HttpServletResponse resp)
        throws ServletException, IOException {

        // Process the request and generate the response
    }
    ...
}
```


How to Implement a Servlet

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

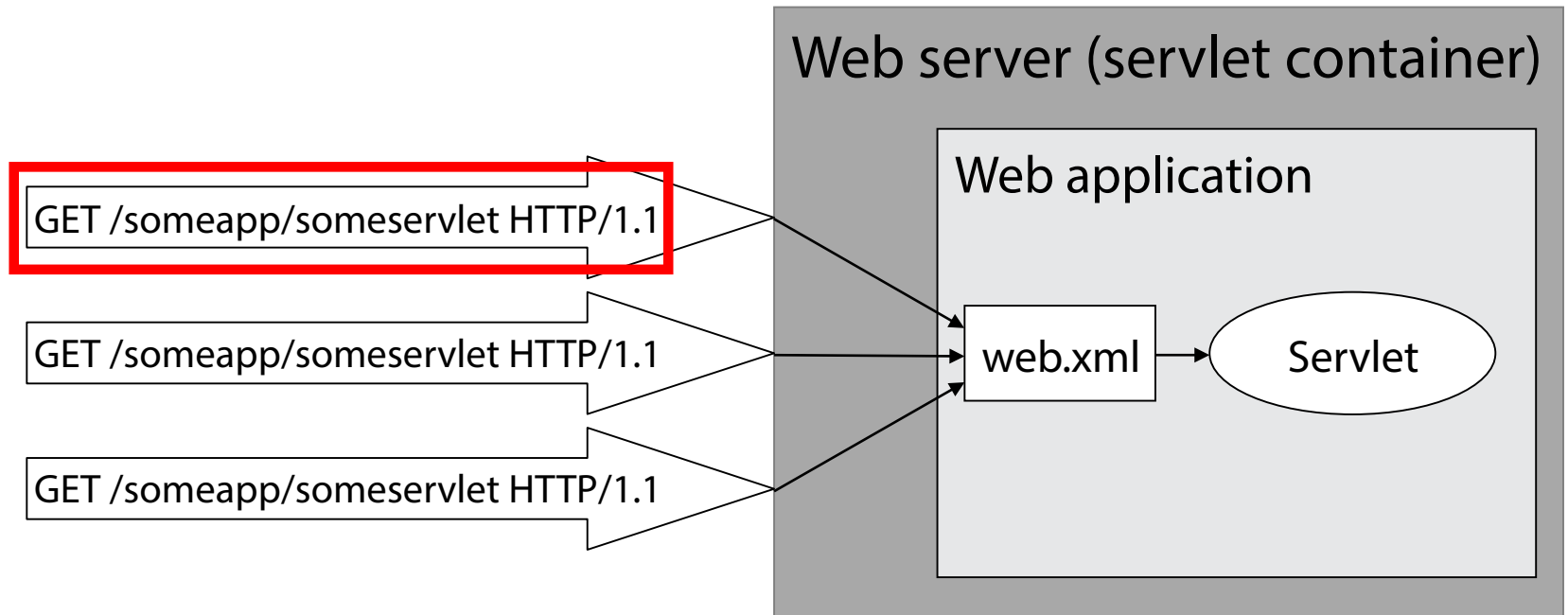
@WebServlet("/home")
public class OurServlet extends HttpServlet {
    public init() {
        // Perform any instance initialisation
    }

    public void doGet(HttpServletRequest req,
        HttpServletResponse resp)
        throws ServletException, IOException {

        // Process the request and generate the response
    }
    ...
}
```

Request Routing

- **Server routes request to servlet using configuration info**
 - Held in web.xml
 - By default, one servlet instance handles all requests to associated URL



Servlet Mapping in web.xml

```
<web-app>
  <servlet>
    <servlet-name>Home</servlet-name>
    <servlet-class>com.mantiso.OurServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>Home</servlet-name>
    <url-pattern>/Bar</url-pattern>
  </servlet-mapping>

  <servlet-mapping>
    <servlet-name>Home</servlet-name>
    <url-pattern>*.abc</url-pattern>
  </servlet-mapping>
</web-app>
```

Servlet Mapping in web.xml

```
<web-app>
  <servlet>
    <servlet-name>Home</servlet-name>
    <servlet-class>com.mantiso.OurServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>Home</servlet-name>
    <url-pattern>/Bar</url-pattern>
  </servlet-mapping>

  <servlet-mapping>
    <servlet-name>Home</servlet-name>
    <url-pattern>*.abc</url-pattern>
  </servlet-mapping>
</web-app>
```

Servlet Mapping in web.xml

```
<web-app>
  <servlet>
    <servlet-name>Home</servlet-name>
    <servlet-class>com.mantiso.OurServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>Home</servlet-name>
    <url-pattern>/Bar</url-pattern>
  </servlet-mapping>

  <servlet-mapping>
    <servlet-name>Home</servlet-name>
    <url-pattern>*.abc</url-pattern>
  </servlet-mapping>
</web-app>
```

Servlet Mapping in web.xml

```
<web-app>
  <servlet>
    <servlet-name>Home</servlet-name>
    <servlet-class>com.mantiso.OurServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>Home</servlet-name>
    <url-pattern>/Bar</url-pattern>
  </servlet-mapping>

  <servlet-mapping>
    <servlet-name>Home</servlet-name>
    <url-pattern>*.abc</url-pattern>
  </servlet-mapping>
</web-app>
```

Servlet Mapping in web.xml

```
<web-app>
  <servlet>
    <servlet-name>Home</servlet-name>
    <servlet-class>com.mantiso.OurServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>Home</servlet-name>
    <url-pattern>/Bar</url-pattern>
  </servlet-mapping>

  <servlet-mapping>
    <servlet-name>Home</servlet-name>
    <url-pattern>*.abc</url-pattern>
  </servlet-mapping>
</web-app>
```

How to Implement a Servlet

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

@WebServlet(urlPatterns = {"/home", "/*.do"})
public class OurServlet extends HttpServlet {
    public init() {
        // Perform any instance initialisation
    }

    public void doGet(HttpServletRequest req,
        HttpServletResponse resp)
        throws ServletException, IOException {

        // Process the request and generate the response
    }
    ...
}
```


HTTP Request Processing

- **HttpServletRequest** wraps an HTTP request

```
public void doGet(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException {

    // Read HTTP headers
    String strHost = req.getHeader("Host");

    // Read Content-Type of request body
    String strContentType = req.getContentType();

    // Read parameters
    // e.g. http://someserver/someapp/someservlet?uid=bob
    String strName = req.getParameter("uid");

    // Access request body
    BufferedReader = req.getReader();
    ...
}
```

Generate HTTP Response

- **HttpServletResponse** wraps the potential HTTP response



```
public void doGet(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException {
    try {
        if (notLoggedIn()) {
            resp.sendRedirect("/logon");
            return;
        }
        resp.setContentType("text/xml");
        resp.setHeader("X-Custom-Header", new Date());
        PrintWriter out = resp.getWriter();
        out.write("<message>Hello world</message>");
    } catch (Exception e) {
        resp.sendError(response.SC_INTERNAL_SERVER_ERROR);
    }
}
```

Providing Initialization Parameters

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.IO.*;

@WebServlet(urlPatterns = {"/home", "*.do"})
public class OurServlet extends HttpServlet {
    public init() {
        connstr = getInitParameter("connstr");
    }

    ...
}
```

Providing Initialization Parameters

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

@WebServlet(urlPatterns = {"/home", "*.do"},
           initParams =
               {@WebInitParam(name="connstr",
                             value="server=...")})
public class OurServlet extends HttpServlet {
    public init() {
        connstr = getInitParameter("connstr");
    }

    ...
}
```

Providing Initialization Parameters

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.IO.*;


@WebServlet(urlPatterns = {"/home", "*.do"},
            initParams =
                {@WebInitParam(name = "connstr",
                               value="server=...")})
public class OurServlet extends HttpServlet {
    public init() {
        connstr = getInitParameter("connstr");
    }

    ...
}
```

Using initialisation parameters




```
<web-app>
  <servlet>
    <servlet-name>Foo</servlet-name>
    <servlet-class>com.mantiso.OurServlet</servlet-class>
    <init-param>
      <param-name>connstr</param-name>
      <param-value>server=homer;catalog=pubs;uid=sa;pwd=;</param-value>
    </init-param>
  </servlet>
</web-app>
```




```
public class OurServlet extends HttpServlet {
    String connstr;
    public void init() {
        connstr = getInitParameter("connstr");
    }
}
```

ServletContext



```
<web-app>
  <context-param>
    <param-name>connstr</param-name>
    <param-value>server=homer;catalog=pubs;uid=sa;pwd=;</param-value>
  </context-param>
</web-app>
```



```
public class OurServlet extends HttpServlet {
    String connstr;
    public void init() {
        connstr = getServletContext().getInitParameter("connstr");
    }
}
```