Introduction to Servlets

After which you will doGet it

Servlet technology

A Java Servlet is a class which can be called via a web server. With Servlets, we can offer programmable dynamic web pages (or web services) using Java, similar to what you can do using PHP or .NET

We can write applications in Java which are accessed over HTTP (as a web server or web service).

Download Winstone

We need a Servlet container (a program which can run Servlets).

```
$ wget 'https://sourceforge.net/projects/winstone/files/latest/download?source=typ_redirect' -O winstone.jar
```

Link: https://sourceforge.net/projects/winstone/files/latest/download?source=typ_redirect

Winstone is an executable JAR which can run Servlets.

Set up the Servlet's environment

In order for the Servlet container to find your Servlet (we haven't written one yet but will), it needs some predefined structure and information.

Create some directories:

```
$ mkdir -p www/WEB-INF
```

Set up the Servlet's environment

Inside WEB-inf we'll put a configuration file, web.xml, and a dirctory for the Servlet class files.

```
$ mkdir www/WEB-INF/classes
```

Create the web.xml file inside the WEB-INF directory.

You can copy the one on the next slide, just save it as web.xml in www/WEB-INF/

web.xml

```
<?xml version="1.0" encoding="utf-8"?>

<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
  <servlet>
    <servlet-name>hello</servlet-name>
    <servlet-class>se.itu.web.HelloServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>hello</servlet-name>
    <url-pattern>/hello</url-pattern>
  </servlet-mapping>
</web-app>
```

web.xml - Define a servlet

```
<?xml version="1.0" encoding="utf-8"?>

<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
  <servlet>
    <servlet-name>hello</servlet-name>
    <servlet-class>se.itu.web.HelloServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>hello</servlet-name>
    <url-pattern>/hello</url-pattern>
  </servlet-mapping>
</web-app>
```

web.xml - Define a servlet

```
<servlet>
  <servlet-name>hello</servlet-name>
  <servlet-class>se.itu.web.HelloServlet</servlet-class>
</servlet>
```

The above connects the servlet we named hello to the class se.itu.web.HelloServlet (which we will write and put in www/WEB-INF/classes/se/itu/web/).

web.xml - map to URL pattern

```
<?xml version="1.0" encoding="utf-8"?>

<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
  <servlet>
    <servlet-name>hello</servlet-name>
    <servlet-class>se.itu.web.HelloServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>hello</servlet-name>
    <url-pattern>/hello</url-pattern>
  </servlet-mapping>
</web-app>
```

web.xml - map to URL pattern

```
  <servlet-mapping>
    <servlet-name>hello</servlet-name>
    <url-pattern>/hello</url-pattern>
  </servlet-mapping>
```

The above maps the servlet we named hello to the URL

http://localhost:8080/hello

Write the servlet

```
$ mkdir -p www/WEB-INF/classes/se/itu/web/

# Create the file www/WEB-INF/classes/se/itu/web/HelloServlet.java
# Import statements:

package se.itu.web;

import java.io.IOException;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import javax.servlet.*;
import javax.servlet.http.*;
import static java.nio.charset.StandardCharsets.UTF_8;
```

Write the servlet

```
public class HelloServlet extends HttpServlet {
    @Override
    public void doGet(HttpServletRequest request, HttpServletResponse response)
                      throws ServletException, IOException {
      request.setCharacterEncoding(UTF_8.name());
      response.setContentType("text/html;charset="+UTF_8.name());
      PrintWriter out =
        new PrintWriter(new OutputStreamWriter(response.getOutputStream(),
                                               UTF_8), true);
      out.println("<!DOCTYPE html>");
      out.println("<html lang=\"en\">");
      out.println("<head><title>Hello Servlet!</title></head>");
      out.println("<body>");
      out.println("<h1>Hello Servlet!</h1>");
      out.println("</body>");
      out.println("</html>");
    }
}
```

Compile the servlet

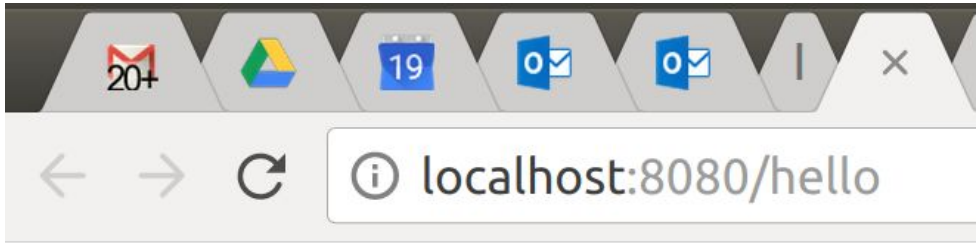
```
$ javac -cp winstone.jar www/WEB-INF/classes/se/itu/web/HelloServlet.java

# We need winstone.jar on the classpath, because it contains the Servlet
specific classes we are importing, extending etc.
```

Start the servlet engine Winstone

```
$ java -jar winstone.jar --webroot=www
[Winstone 2017/12/20 15:50:19] - No webapp lib folder found -
/home/rikard/opt/progund/web-misc/servlets/www/WEB-INF/lib
[Winstone 2017/12/20 15:50:19] - HTTP Listener started: port=8080
[Winstone 2017/12/20 15:50:19] - Winstone Servlet Engine v0.9.10 running:
controlPort=disabled
[Winstone 2017/12/20 15:50:19] - AJP13 Listener started: port=8009
```

Point your browser to the Servlet

http://localhost:8080/hello



Hello Servlet!

Investigate the source code of the page

Ctrl-U or right-click, View page source

```
<!DOCTYPE html>
<html lang="en">
<head><title>Hello Servlet!</title></head>
<body>
<h1>Hello Servlet!</h1>
</body>
</html>
```

Voilà! You have created a Servlet!

You have now configured, written, compiled and deployed a Servlet!

Let’s look at some small details.

Some small details

```
public class HelloServlet extends HttpServlet
```

- Your class HelloServlet *is an HttpServet (by inheritance)*

```
@Override
public void doGet(HttpServletRequest request, HttpServletResponse response)
```

- This method is overridden and handles the HTTP request and response (communicating with your browser, for instance)

```
request.setCharacterEncoding(UTF_8.name());
response.setContentType("text/html;charset="+UTF_8.name());
```

- These lines sets the content-type of the response - so that your browser knows it's HTML and can render the page for you.

```
PrintWriter out = new PrintWriter(new OutputStreamWriter(response.getOutputStream(),
                                                         UTF_8), true);
```

- Used for writing the actual response. Goes to your browser (for instance).

Further reading

- Source code https://github.com/progund/web-misc/tree/master/servlets
- Wikipedia https://en.wikipedia.org/wiki/Java_servlet