

- Create a simple web application that logs requests
 - Clone a web application from github
 - Edit the provided servlet
 - Deploy the application to a servlet container

```
@WebServlet(urlPatterns={"/myservlet"})
public class MyServlet extends HttpServlet
{
    protected void doGet(HttpServletRequest request,
                          HttpServletResponse response)
                          throws ServletException, IOException {
        PrintWriter out = response.getWriter();
        String str = "Hello, world.\n";
        out.append(str)
    }

    //...doPost implementation
}
```

- A *servlet* is a Java class that has methods to handle requests
- Extended the base Servlet class and overrode the `doGet()` method
- URLs are relative to the base URL of your web application

- Notice that the framework methods such as `doGet` receive two arguments
 - `HttpServletRequest`: access information related to the request, url parameters for example
 - `HttpServletResponse`: access functionality related to the response, such as the writer, content headers, and so on
- Also notice that a Servlet is also a Java class
 - For this exercise it is ok to store data in a `static` variable attached to the servlet class

- We have set up a public repository containing a basic Java web application¹
- You can clone this project and import to Eclipse as you have done previously
- Make sure you are in the Java EE perspective (upper right corner)

¹<https://github.com/marks1024/logging-exercise-361>

- Java EE is an open standard with many different implementations to choose from
- *Tomcat*² is probably the easiest to obtain and use
 - Simply download and unzip in a known location on your computer
 - Choose the 8.5 version
- To run the application we have at least two options
 - Tell the IDE where the server is (“run on server” option in eclipse)
 - Package our web application as a .war file and copy to Tomcat’s webapps/ directory

²<http://tomcat.apache.org/>

- Use the web app skeleton from github to create a simple logging application
- Servlet that records every request made to the servlet (time of access, host, path)
- Store a log in a static variable on the servlet class (use an ArrayList) for example (see the documentation for `HttpServletRequest`³)
- Deploy your application either locally and access from your browser at `localhost:8080`

³ <https://tomcat.apache.org/tomcat-5.5-doc/servletapi/javax/servlet/http/HttpServletRequest.html>