AIT-LAB-02 (Exercise 3- 6) By Amir Ebrahimi

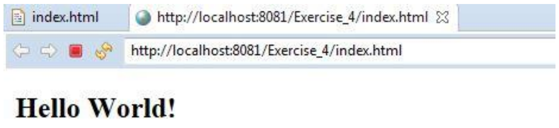
When Eclipse JEE Neon 3 is tried to install, "Could not create the Java Virtual Machine." Error is got. Xmx variable in eclipse.ini is changed; though, the error remains. Hence all of the java versions is removed (uninstalling, and removing the directories, and removing the variables in environment variables on the system, and removing all of related things in the registry editor. Afterwards, the new JDK (JDK 9.0.4) is installed. Java Runtime Environment (JRE) is basically the Java Virtual Machine where the Java programs run on, and JDK is the full featured software development kit for java, including JRE, and the compilers and tools (like JavaDoc, and Java Debugger) to create and compile programs. After searching on the internet and changing something, Eclipse does not work. So JDK 8u161 with Netbeans 8.2 is installed, and Eclipse with that version of Java can work properly.

Apache Tomcat 9.0.6 is downloaded, and two new environment variables, named JAVA_HOME with the value of the jdk directory, and CATALINA_HOME with the value of the Tomcat directory are added on the system. Also "chmod 777 catalina.bat" is necessary for the permission if it is needed.

With running startup.bat and shutdown.bat, starting and shut downing Tomcat is available. In the server.xml file in the conf directory, it is possible to change the port number.

Also in the conf directory of the Tomcat, there is a file, named tomcat.users.xml. "<user username="admin" password="admin123" roles="manager-gui" />" code is added in the file. Afterwards, it is needed to restart. In the webapps directory, docs, examples, host-manager, manager, and also root are available.

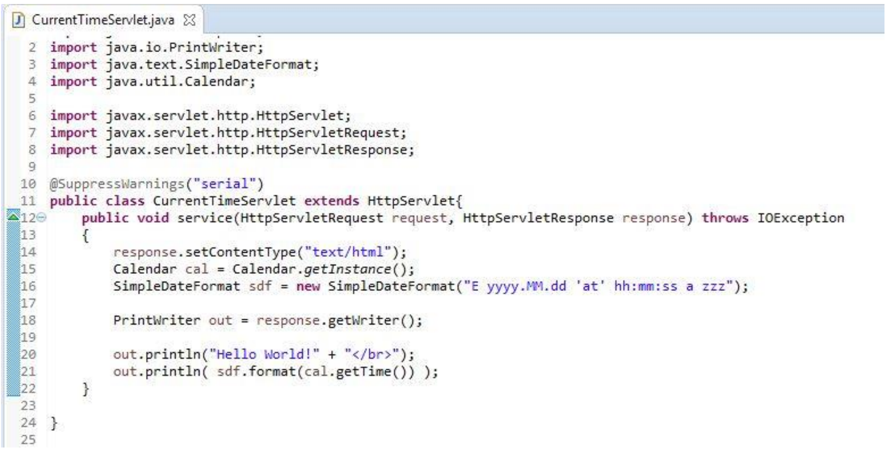
In the Eclipse, the webapps directory of the Tomcat is used as a workspace. The first step is to create new server. Tomcat 9 is used, and the directory of the Tomcat installation should be added. The server in the previous part was started. Hence it is necessary to stop it, and start it from Eclipse, and it is possible to change port from here. Now it is time to create a new dynamic project, named Exercise_4. "Generate web.xml deployment descriptor" should be enabled in the wizard. At first, one html file is created, and one simple <h2> tag is added. The head part is deleted.



http://localhost:8081/Exercise_4/index.html

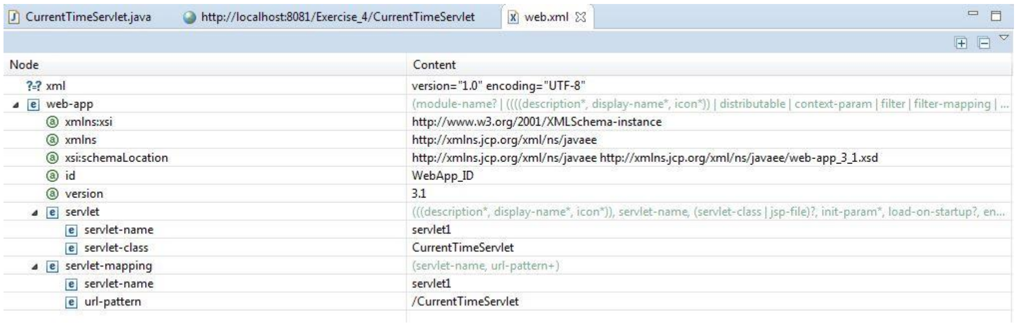
Hello World!

Secondly, one class, named CurrentTimeServlet, is added, and HttpServlet should be extended since server can recognize CurrentTimeServlet as a servlet. In that class a method, named service, with request and response parameters is used. Current time can be showed in the service. "response.setContentType("text/html");" is added in order to show a br tag in "out.println("Hello World!" + "</br>");".

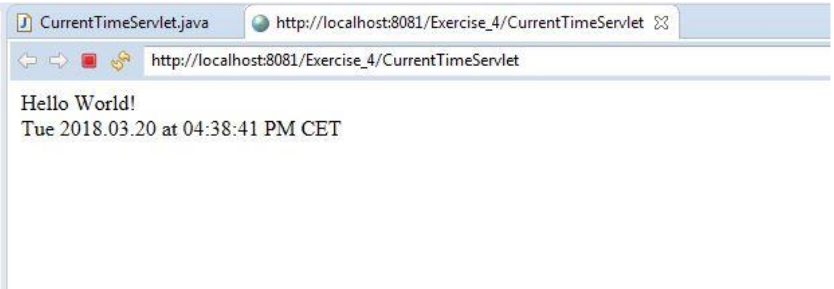


```
import java.io.PrintWriter;
import java.text.SimpleDateFormat;
import java.util.Calendar;

import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@SuppressWarnings("serial")
public class CurrentTimeServlet extends HttpServlet{
    public void service(HttpServletRequest request, HttpServletResponse response) throws IOException
    {
        response.setContentType("text/html");
        Calendar cal = Calendar.getInstance();
        SimpleDateFormat sdf = new SimpleDateFormat("E yyyy.MM.dd 'at' hh:mm:ss a zzz");

        PrintWriter out = response.getWriter();

        out.println("Hello World!" + "</br>");
        out.println( sdf.format(cal.getTime()) );
    }

}
```

Web.xml is also called as a deployment descriptor. All of tags in web-app tag are removed. it is time to configure the server. So servlet and servlet-mapping tags are used here. Actually, for every servlet, we need to create servlet and servlet-mapping tags. There are servlet-name and servlet-class tags in the servlet tags, and also there are servlet-name and url-pattern tags in the servlet-mapping tags in order to prevent the confusion that which servlet-mapping tag is linked with which servlet tag.

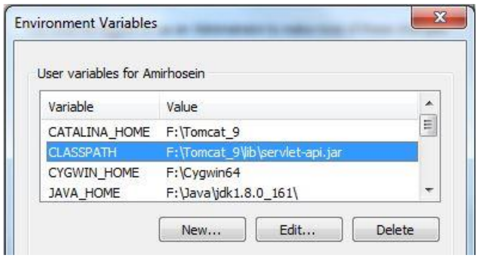


Node	Content
?-? xml	version="1.0" encoding="UTF-8"
web-app	(module-name? | ((((description*, display-name*, icon*)) | distributable | context-param | filter | filter-mapping | ...
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xmlns	http://xmlns.jcp.org/xml/ns/javaee
xsi:schemaLocation	http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd
id	WebApp_ID
version	3.1
servlet	(((description*, display-name*, icon*)), servlet-name, (servlet-class | jsp-file)?, init-param*, load-on-startup?, en...
servlet-name	servlet1
servlet-class	CurrentTimeServlet
servlet-mapping	(servlet-name, url-pattern+)
servlet-name	servlet1
url-pattern	/CurrentTimeServlet


CurrentTimeServlet.java
http://localhost:8081/Exercise_4/CurrentTimeServlet
http://localhost:8081/Exercise_4/CurrentTimeServlet

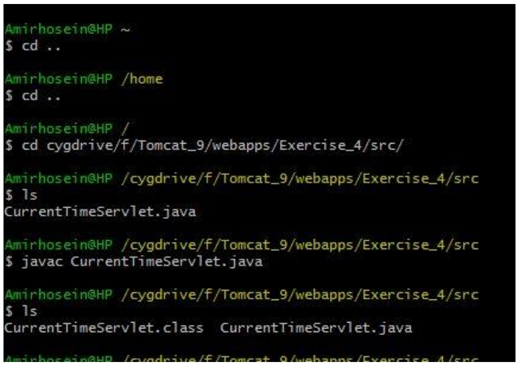
Hello World!
Tue 2018.03.20 at 04:38:41 PM CET

In order to work with javac, the path of the bin directory of the JDK should be add to the environment variable of the PATH for the user in the system. As well, CLASSPATH environment variable with the value of the "F:\Tomcat_9\lib\servlet-api.jar" is added. So there is no need to add –cp or –classpath.



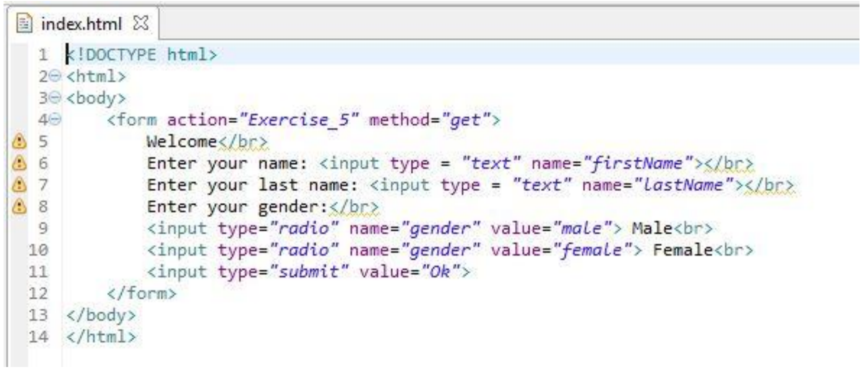
Variable	Value
CATALINA_HOME	F:\Tomcat_9
CLASSPATH	F:\Tomcat_9\lib\servlet-api.jar
CYGWIN_HOME	F:\Cygwin64
JAVA_HOME	F:\Java\jdk1.8.0_161\

Now it is time to compile the java file in the src directory of the project directory.



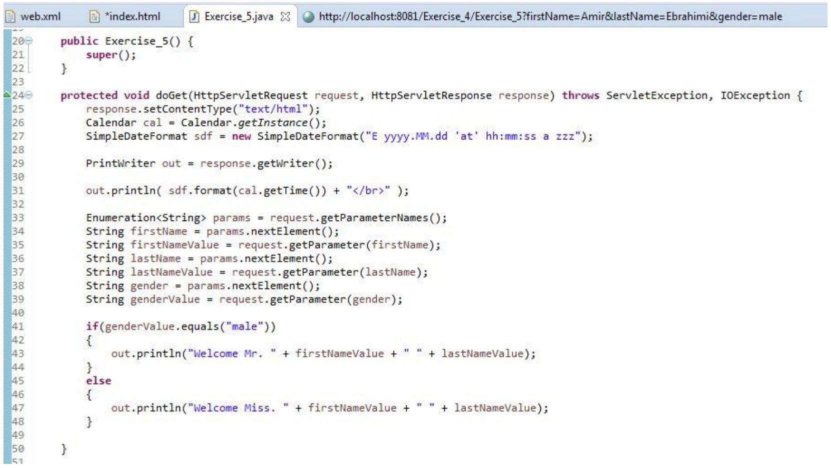
```
Amirhosein@HP ~
$ cd ..

Amirhosein@HP /home
$ cd ..

Amirhosein@HP /
$ cd cygdrive/f/Tomcat_9/webapps/Exercise_4/src/

Amirhosein@HP /cygdrive/f/Tomcat_9/webapps/Exercise_4/src
$ ls
CurrentTimeServlet.java

Amirhosein@HP /cygdrive/f/Tomcat_9/webapps/Exercise_4/src
$ javac CurrentTimeServlet.java

Amirhosein@HP /cygdrive/f/Tomcat_9/webapps/Exercise_4/src
$ ls
CurrentTimeServlet.class  CurrentTimeServlet.java

Amirhosein@HP /cygdrive/f/Tomcat_9/webapps/Exercise_4/src
```

<h2> tag in the html file should be changed to the <form> tag with "get" method and "Exercise_5" action. In the form, user should enter name, last name, and gender.



```
<!DOCTYPE html>
<html>
<body>
    <form action="Exercise_5" method="get">
        Welcome</br>
        Enter your name: <input type = "text" name="firstName"></br>
        Enter your last name: <input type = "text" name="lastName"></br>
        Enter your gender:</br>
        <input type="radio" name="gender" value="male"> Male<br>
        <input type="radio" name="gender" value="female"> Female<br>
        <input type="submit" value="Ok">
    </form>
</body>
</html>
```

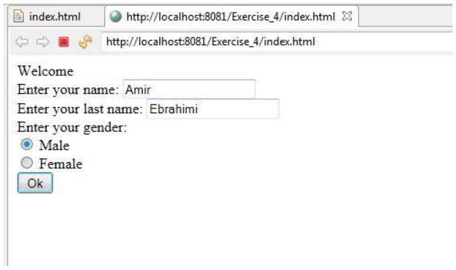
A new servlet, named Exercise_5, is added. It is possible to receive the data from the URL in the servlet. The doGet function is used.


web.xml
*index.html
Exercise_5.java
http://localhost:8081/Exercise_4/Exercise_5?firstName=Amir&lastName=Ebrahimi&gender=male

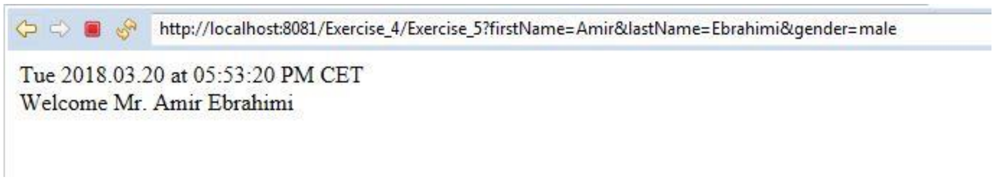
```
    public Exercise_5() {
        super();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html");
        Calendar cal = Calendar.getInstance();
        SimpleDateFormat sdf = new SimpleDateFormat("E yyyy.MM.dd 'at' hh:mm:ss a zzz");

        PrintWriter out = response.getWriter();

        out.println( sdf.format(cal.getTime()) + "</br>" );

        Enumeration<String> params = request.getParameterNames();
        String firstName = params.nextElement();
        String firstNameValue = request.getParameter(firstName);
        String lastName = params.nextElement();
        String lastNameValue = request.getParameter(lastName);
        String gender = params.nextElement();
        String genderValue = request.getParameter(gender);

        if(genderValue.equals("male"))
        {
            out.println("Welcome Mr. " + firstNameValue + " " + lastNameValue);
        }
        else
        {
            out.println("Welcome Miss. " + firstNameValue + " " + lastNameValue);
        }

    }
```

If web.xml is not changed, "Server Tomcat v9.0 Server at localhost failed to start." Will be got. Hence servlet and servlet and servlet-mapping tags should be deleted. With Java EE annotations, the standard web.xml deployment descriptor is optional.

Now it is possible to run the html file, and see the result.


index.html
http://localhost:8081/Exercise_4/index.html
http://localhost:8081/Exercise_4/index.html

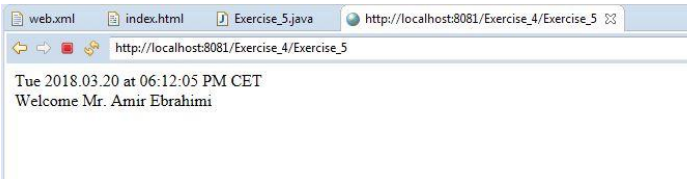
Welcome
Enter your name: Amir
Enter your last name: Ebrahimi
Enter your gender:
● Male
○ Female
Ok



http://localhost:8081/Exercise_4/Exercise_5?firstName=Amir&lastName=Ebrahimi&gender=male

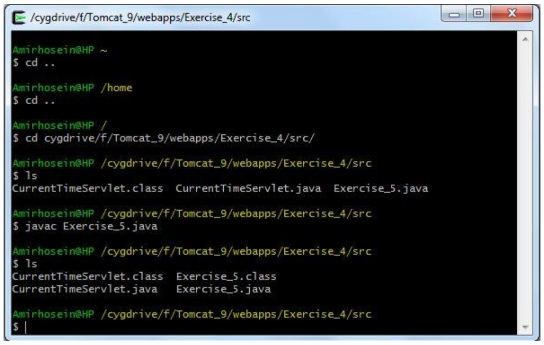
Tue 2018.03.20 at 05:53:20 PM CET
Welcome Mr. Amir Ebrahimi

As it is seen, the data is observable in the URL. In order to avoid this, we can use doPost function in the servlet. Also, the method in the form should be changed to post.



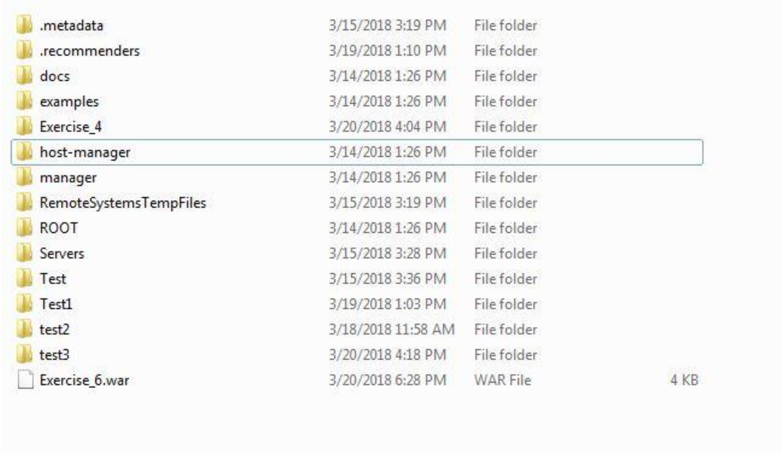
http://localhost:8081/Exercise_4/Exercise_5

Tue 2018.03.20 at 06:12:05 PM CET
Welcome Mr. Amir Ebrahimi

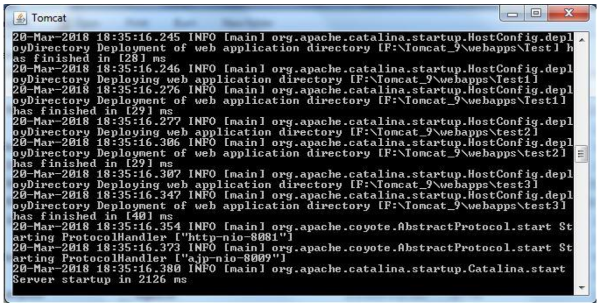


```
Amirhosein@HP ~
$ cd ..

Amirhosein@HP /home
$ cd ..

Amirhosein@HP /
$ cd cygdrive/f/Tomcat_9/webapps/Exercise_4/src/

Amirhosein@HP /cygdrive/f/Tomcat_9/webapps/Exercise_4/src
$ ls
CurrentTimeServlet.class  CurrentTimeServlet.java  Exercise_5.java

Amirhosein@HP /cygdrive/f/Tomcat_9/webapps/Exercise_4/src
$ javac Exercise_5.java

Amirhosein@HP /cygdrive/f/Tomcat_9/webapps/Exercise_4/src
$ ls
CurrentTimeServlet.class  Exercise_5.class
CurrentTimeServlet.java   Exercise_5.java

Amirhosein@HP /cygdrive/f/Tomcat_9/webapps/Exercise_4/src
$ |
```

Now it is time to use war file. A war file, named Exercise_6, is exported. If the file is in the webapps directory of the Tomcat, after some seconds a new folder, named Exercise_6, is created in the webapps. This creation is done if the Tomcat server is running.

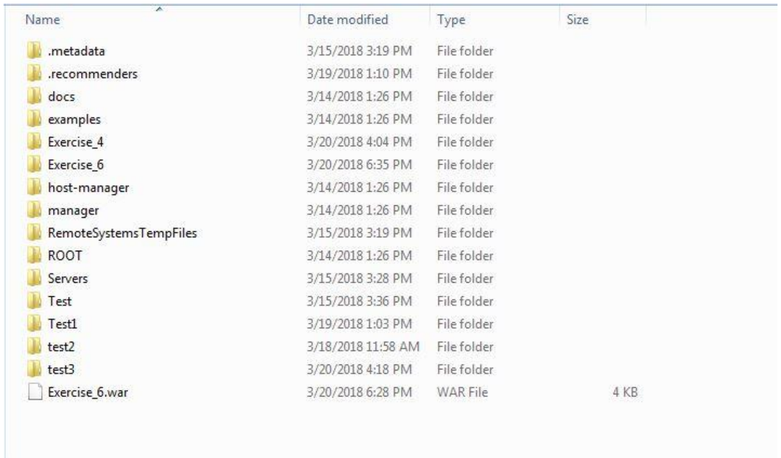


.metadata	3/15/2018 3:19 PM	File folder	
.recommenders	3/19/2018 1:10 PM	File folder	
docs	3/14/2018 1:26 PM	File folder	
examples	3/14/2018 1:26 PM	File folder	
Exercise_4	3/20/2018 4:04 PM	File folder	
host-manager	3/14/2018 1:26 PM	File folder	
manager	3/14/2018 1:26 PM	File folder	
RemoteSystemsTempFiles	3/15/2018 3:19 PM	File folder	
ROOT	3/14/2018 1:26 PM	File folder	
Servers	3/15/2018 3:28 PM	File folder	
Test	3/15/2018 3:36 PM	File folder	
Test1	3/19/2018 1:03 PM	File folder	
test2	3/18/2018 11:58 AM	File folder	
test3	3/20/2018 4:18 PM	File folder	
Exercise_6.war	3/20/2018 6:28 PM	WAR File	4 KB



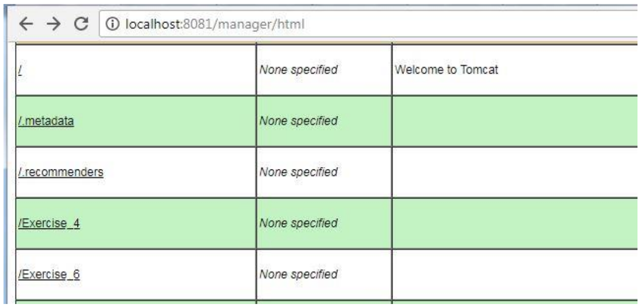
Tomcat

```
20-Mar-2018 18:35:16.245 INFO [main] org.apache.catalina.startup.HostConfig.depl
oyDirectory Deployment of web application directory [F:\Tomcat_9\webapps\Test] h
as finished in [28] ms
20-Mar-2018 18:35:16.246 INFO [main] org.apache.catalina.startup.HostConfig.depl
oyDirectory Deploying web application directory [F:\Tomcat_9\webapps\Test1]
20-Mar-2018 18:35:16.276 INFO [main] org.apache.catalina.startup.HostConfig.depl
oyDirectory Deployment of web application directory [F:\Tomcat_9\webapps\Test1]
has finished in [29] ms
20-Mar-2018 18:35:16.277 INFO [main] org.apache.catalina.startup.HostConfig.depl
oyDirectory Deploying web application directory [F:\Tomcat_9\webapps\test2]
20-Mar-2018 18:35:16.306 INFO [main] org.apache.catalina.startup.HostConfig.depl
oyDirectory Deployment of web application directory [F:\Tomcat_9\webapps\test2]
has finished in [29] ms
20-Mar-2018 18:35:16.307 INFO [main] org.apache.catalina.startup.HostConfig.depl
oyDirectory Deploying web application directory [F:\Tomcat_9\webapps\test3]
20-Mar-2018 18:35:16.347 INFO [main] org.apache.catalina.startup.HostConfig.depl
oyDirectory Deployment of web application directory [F:\Tomcat_9\webapps\test3]
has finished in [40] ms
20-Mar-2018 18:35:16.354 INFO [main] org.apache.coyote.AbstractProtocol.start St
arting ProtocolHandler ["http-nio-8081"]
20-Mar-2018 18:35:16.373 INFO [main] org.apache.coyote.AbstractProtocol.start St
arting ProtocolHandler ["ajp-nio-8009"]
20-Mar-2018 18:35:16.380 INFO [main] org.apache.catalina.startup.Catalina.start
Server startup in 2126 ms
```



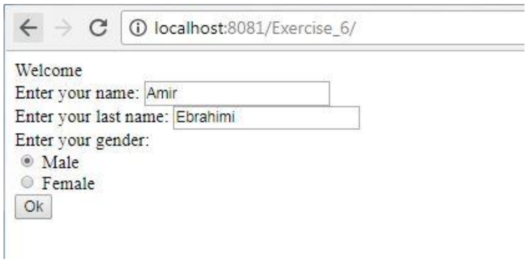
Name	Date modified	Type	Size
.metadata	3/15/2018 3:19 PM	File folder	
.recommenders	3/19/2018 1:10 PM	File folder	
docs	3/14/2018 1:26 PM	File folder	
examples	3/14/2018 1:26 PM	File folder	
Exercise_4	3/20/2018 4:04 PM	File folder	
Exercise_6	3/20/2018 6:35 PM	File folder	
host-manager	3/14/2018 1:26 PM	File folder	
manager	3/14/2018 1:26 PM	File folder	
RemoteSystemsTempFiles	3/15/2018 3:19 PM	File folder	
ROOT	3/14/2018 1:26 PM	File folder	
Servers	3/15/2018 3:28 PM	File folder	
Test	3/15/2018 3:36 PM	File folder	
Test1	3/19/2018 1:03 PM	File folder	
test2	3/18/2018 11:58 AM	File folder	
test3	3/20/2018 4:18 PM	File folder	
Exercise_6.war	3/20/2018 6:28 PM	WAR File	4 KB

Actually, deploying the war file in the manager section is unavailable, since it is already in the webapps folder.



localhost:8081/manager/html

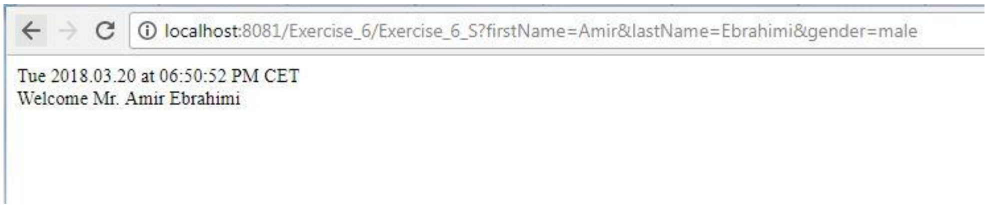
/	None specified	Welcome to Tomcat
/.metadata	None specified	
/.recommenders	None specified	
/Exercise_4	None specified	
/Exercise_6	None specified	



localhost:8081/Exercise_6/

Welcome
Enter your name: Amir
Enter your last name: Ebrahimi
Enter your gender:
● Male
○ Female

Second Lab- by Amir(aebrahimi@unibz.it)



localhost:8081/Exercise_6/Exercise_6_S?firstName=Amir&lastName=Ebrahimi&gender=male

```
Tue 2018.03.20 at 06:50:52 PM CET
Welcome Mr. Amir Ebrahimi
```