Table of Contents:

1. Java Servlets - Overview
2. Java Servlets
3. Servlet Life Cycle
4. Environment Setup and Java Servlet Example
5. HTTP Servlet
6. Interfaces of Servlet

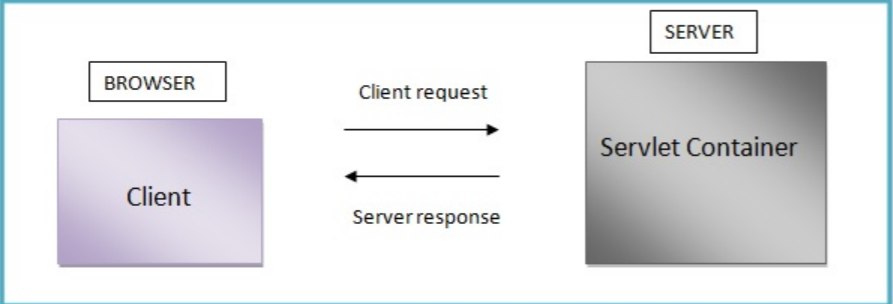
**Servlets Overview**

Servlet is a Java programming language class, part of Java Enterprise Edition (Java EE). Sun Microsystems developed its first version 1.0 in the year 1997. Its current Version is Servlet 3.1. Servlets are used for creating dynamic web applications in java by extending the capability of a server. It can run on any web server integrated with a Servlet container. In fact, servlets have access to the entire family of Java APIs, including the JDBC API to access enterprise databases. (Exelixis Media (P.C., 2015)

Before Servlets, CGI (Common Gateway Interface) programming was used to create web applications. It is an interface that allows HTTP clients, such as Web browsers and other user agents, to pass information back to a server for processing (David Reily, Michael Reily,2012). They come up with certain disadvantages such as having a high response time because CGI programs execute in their own OS shell, not scalable not always secure or object-oriented. So, they need to find a solution. By this, Sun Microsystems developed **Servlet** which serves as their solution over traditional CGI technology.

Servlets Process

When client is requesting, the request is sent to a servlet container. Servlets are run inside the servlet container. No matter how many requests is send by the client. The time the request is arrived, the web server searches for the servlet and initiates it. The request of the client will be processed by the servlet and forwarded the response back to the server. Then the Server response is then forwarded to the client.



Servlets advantages according to Exelixis Media (P.C., 2015):

-Servlets are platform independent as they can run on any platform.

- The Servlet API inherits all the features of the Java platform. It means that, Servlets have access to the entire family of Java APIs, including the JDBC API to access enterprise databases

- It builds and modifies the security logic for server-side extensions.

- Servlets inherit the security provided by the Web Server.

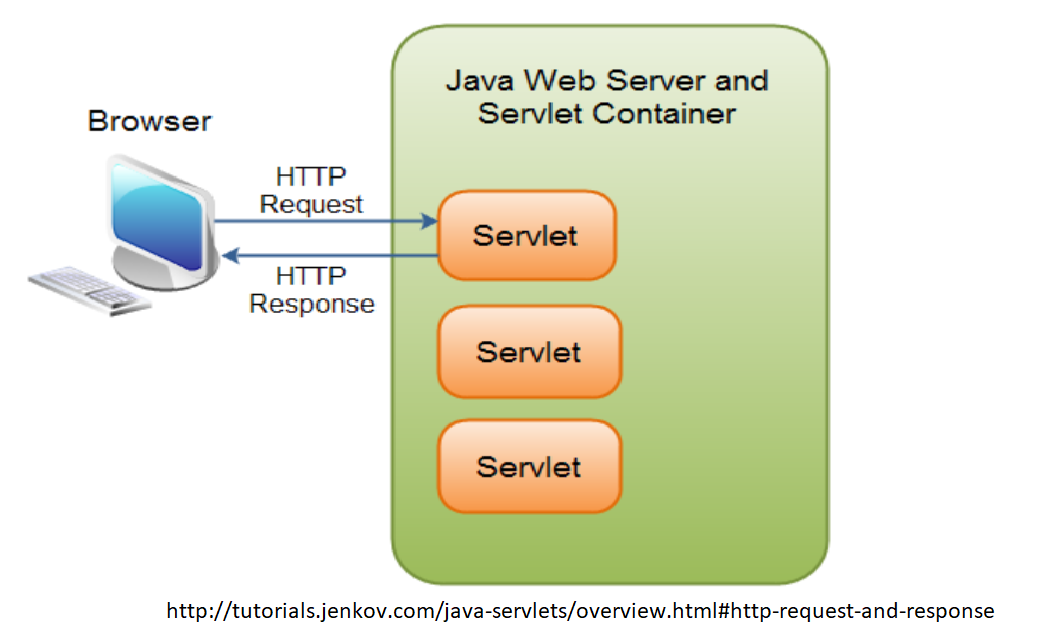
- In Servlet, only a single instance of the requests runs concurrently. It does not run in a separate process. So, it saves the memory by removing the overhead of creating a new process for each request.

- It collects input from users through web page forms, present records from a database or another source, and create web pages dynamically. It means that Servlet container may run multiple and execute web applications at the same time, each having multiple servlets running inside.

**JAVA SERVLETS**

Java Servlets are server-side programming applications that execute similar to CGI scripts,but without a separate process for each request(David Reily, Michael Reily,2012).It is described as a java object that are intended to play the role server components in client-server communications and a java class extending HTTPServlet class. It handles client requests by generating responses to such requests. A java servlets class needs to be compiled prior to using it; it must use servley-api.jar. Java Servlets are part of the Java Enterprise Edition (Java EE). It is run inside a Servlet compatible "Servlet Container" (e.g. web server) in order for them to work such as Jelly, Apache Tomcat that are free Java web servers.

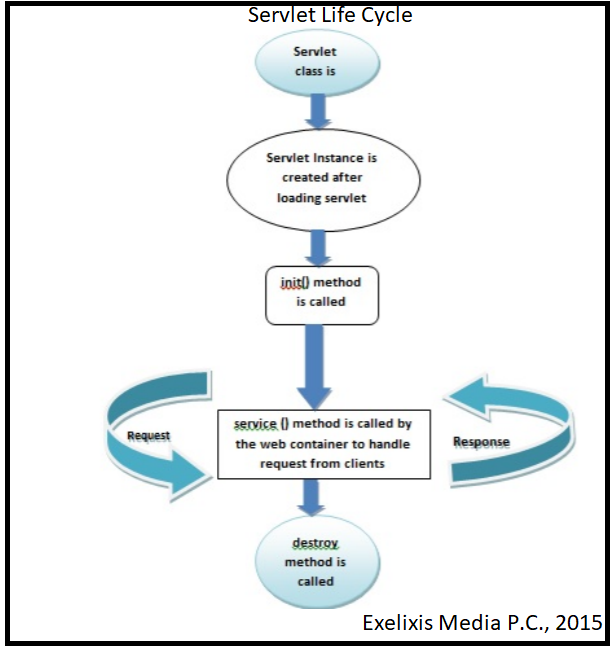
Servlets is hosted in a ‘servlet container’ that provides the environment in which the server run as the ‘servlet life cycle’.



Servlet Container is known as servlet engine which manages Java Servlet components on top of a web server to the request send by the client.

**Servlet Life Cycle**

Servlet lifecycle describes how the servlet container manages the servlet object.



When a client sends a request to a servlet-enabled server that invokes a server first checks to see of the servlet is loaded. If it is not, the servlet class is loaded and a new instance is created. And the cycle will start with the:

First phase: Instantiation

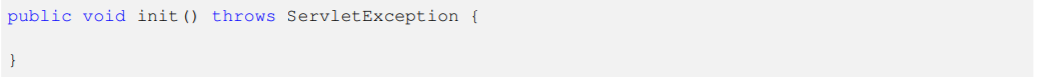
- ‘constructor all…’

- Creates an instance of the servlet to service client request

-Invoked implicitly by the servlet container when the servlet is called then it creates instance to create servlet.

Second phase: Initialization

- ‘init() method’ : This is called only once when the servlet is created.

- Invoked once and is intended for any startup initialization code for the servlet.

Third phase: Request Handling

- ‘service method’: It is called by the web container to handle request from client

-Invoked repeatedly for every client request

-Servlet performs whatever logic necessary in order to serve the request and response.

-Multi-client request typically handled by single multi-threaded

- Must have ‘thread safe’

Fourth phase: Destruction

- ‘destroy method’: It is used to clean resources and called before removing the servlet instance.

- Invoked before the servlet instance is ‘unloaded’ and is intended for ‘housekeeping’.

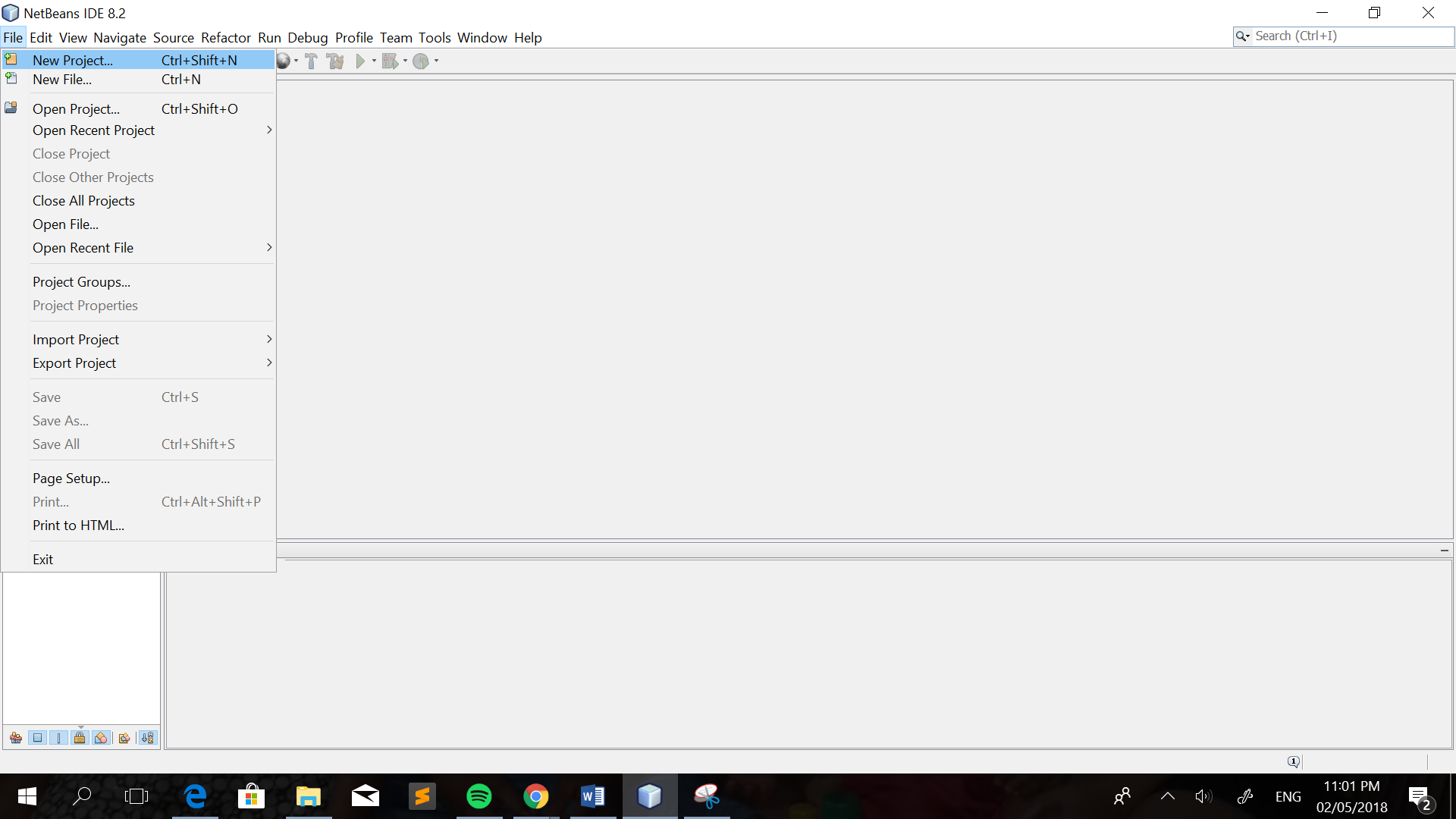


**Environment Setup and Java Servlet Example**

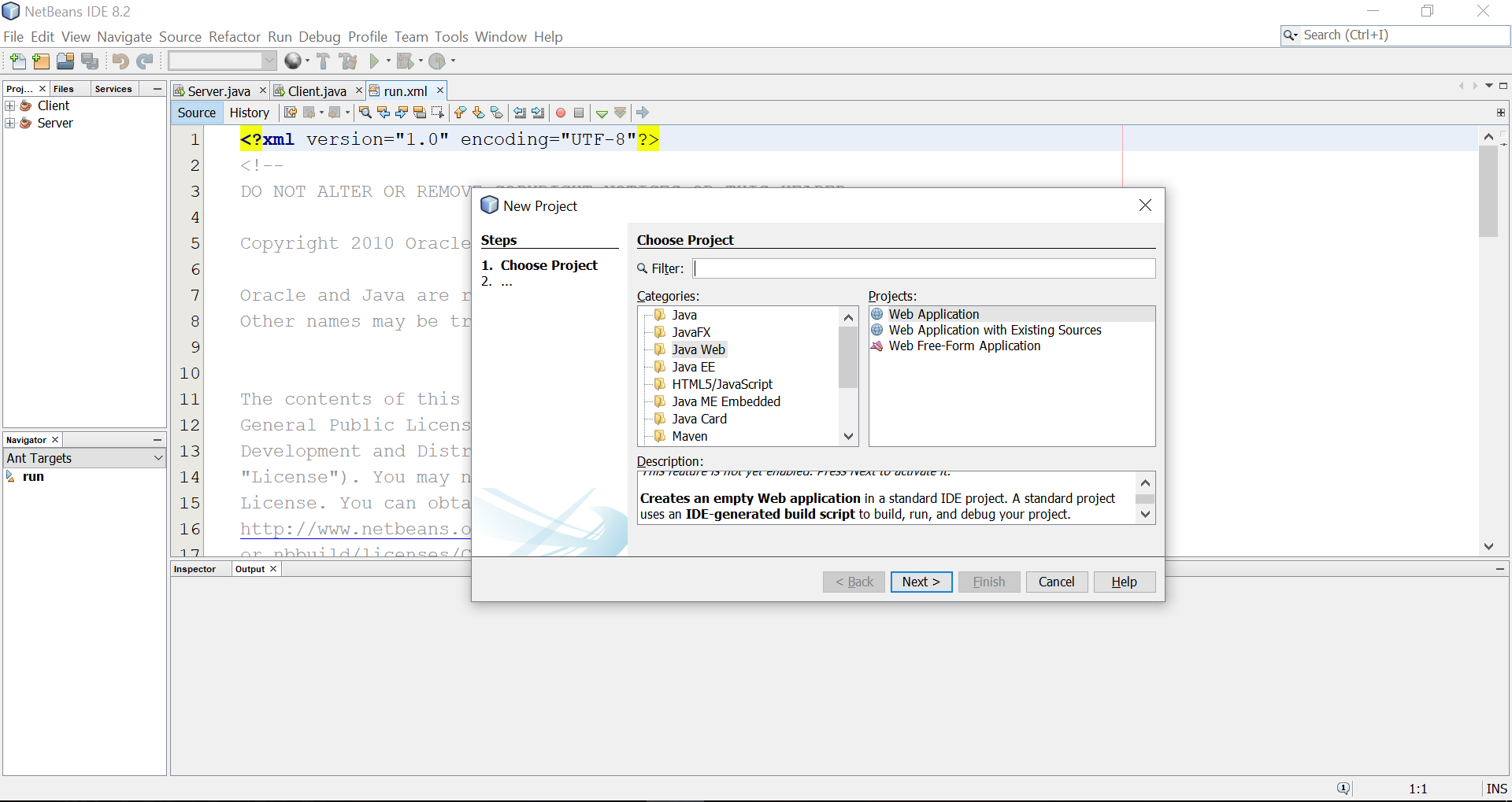
NetBeans IDE is one of the tool commonly used by a developer that provides facilities to computer programmers for software development. So here, we will be demonstrating in setting up the your first servlet application.

### Creating First Servlet Application using NetBeans IDE

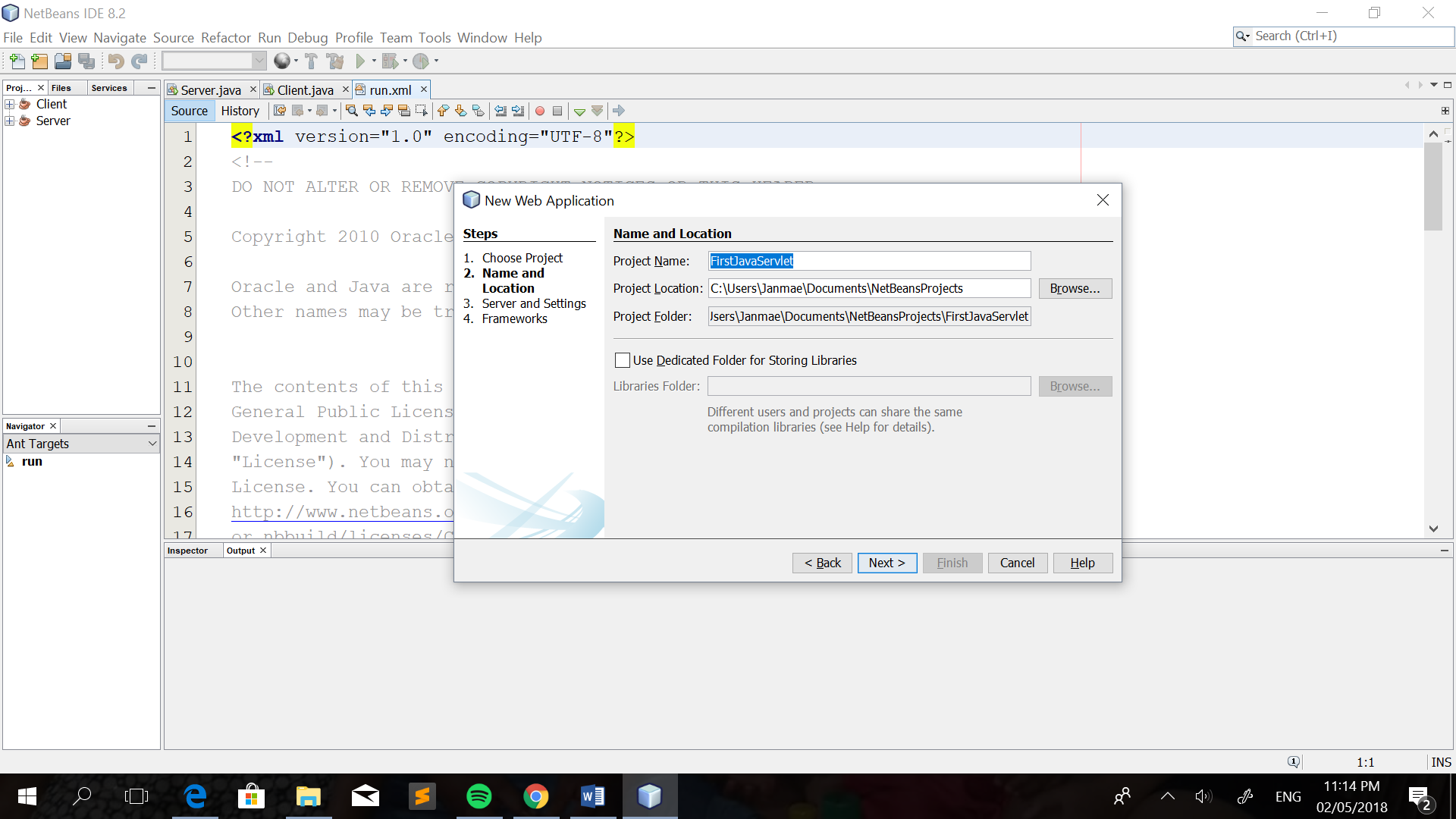
1. We will setup the Servlet first in using Integrated Development Environment(IDE) on NetBeans. Developers take it as the easiest way to create Servlet Applications. So, the first step is, open the **NetBeans IDE**, under the nav bar select **File** and click the **New Project** to create a project.



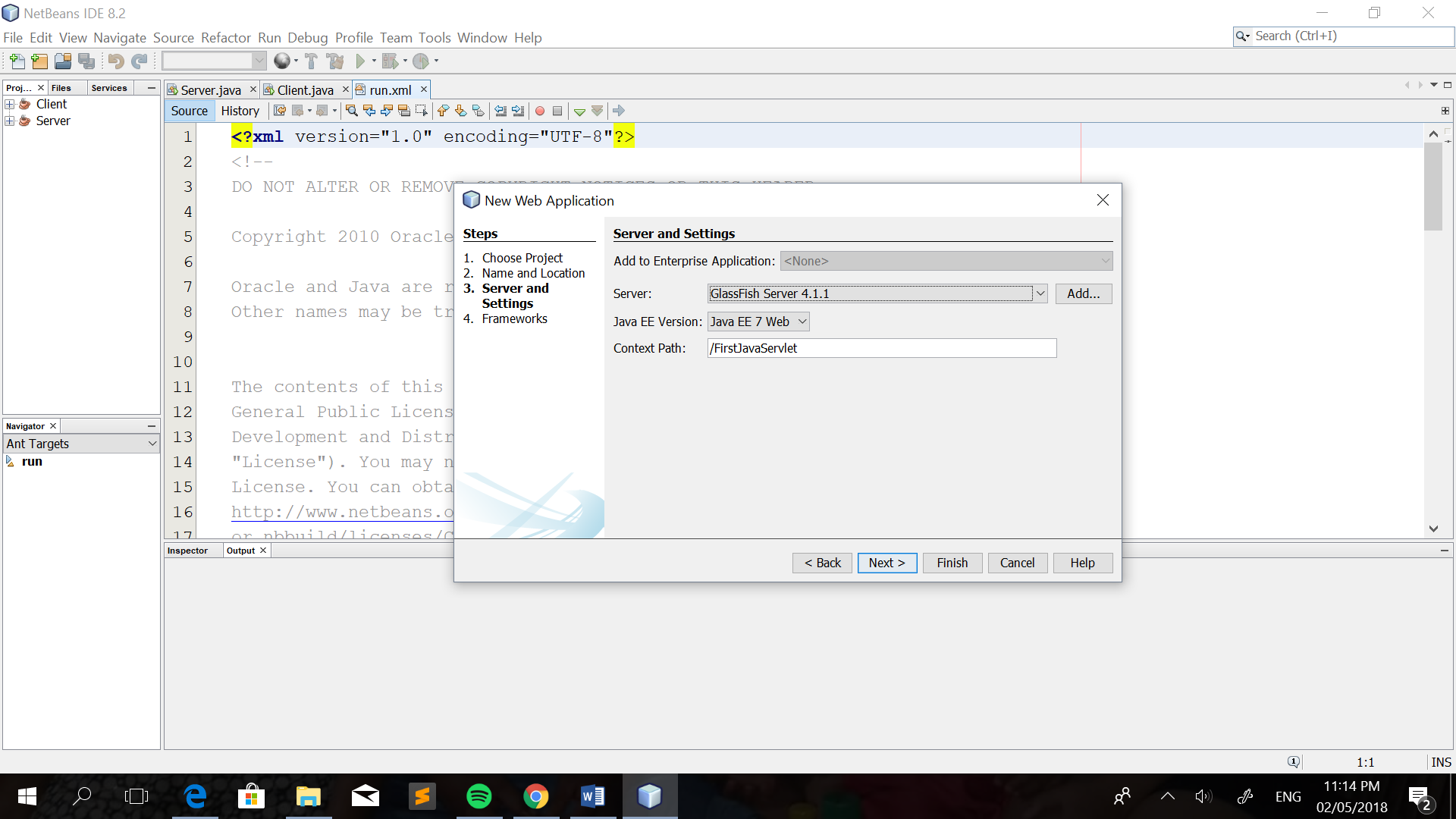
1. After that, you will navigate to a page wherein you will choose the category (e.g. Java, Java Web, Java EE, Maven, NetBeans Module, Samples). Click the **Java Web**. This will create an empty Web application in a standard IDE project.



3.Set the **project name.** If you want to change the location of the file then edit on the **Project Location**. Then click **Next.**



1. After that, you will be directed to the next step wherein you will specify the server that you will use, for this tutorial, we will use GlassFish Server. It is already built-in on the Java NetBeans IDE. But you can add a server like the most popular server, the Apache Tomcat, just click on the ‘**Add**’ button. Then click ‘**Finish**’.

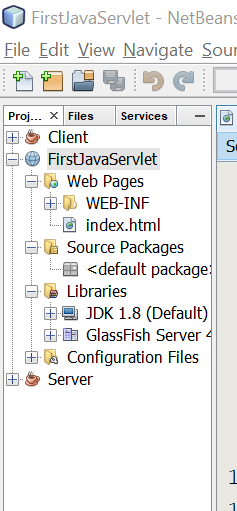


The context path will be added on the URL address if you want to view the file created on the web.

Example:

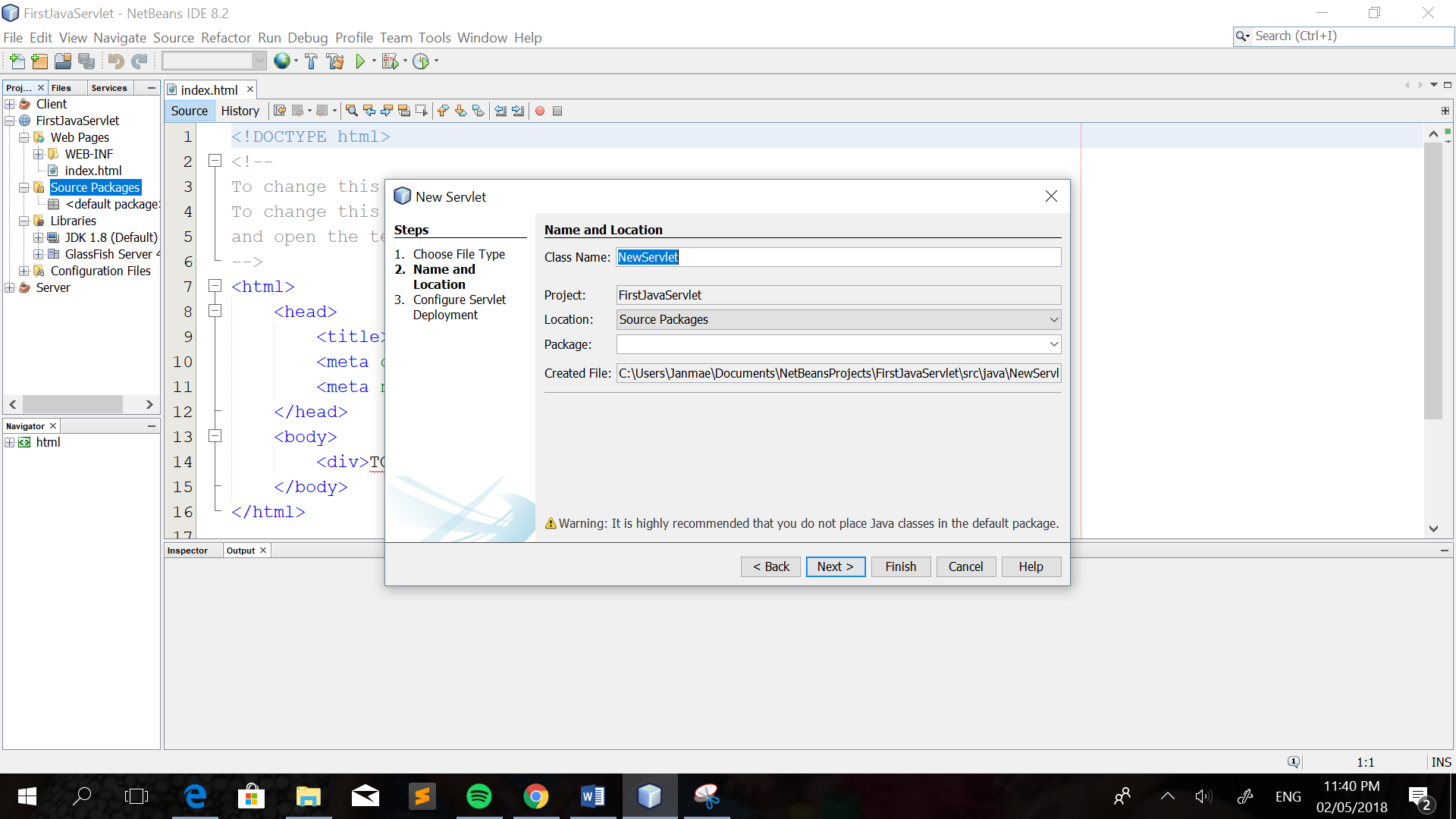


5. After theproject is created, the complete directory structure required for the Servlet Application will be created automatically by the IDE.

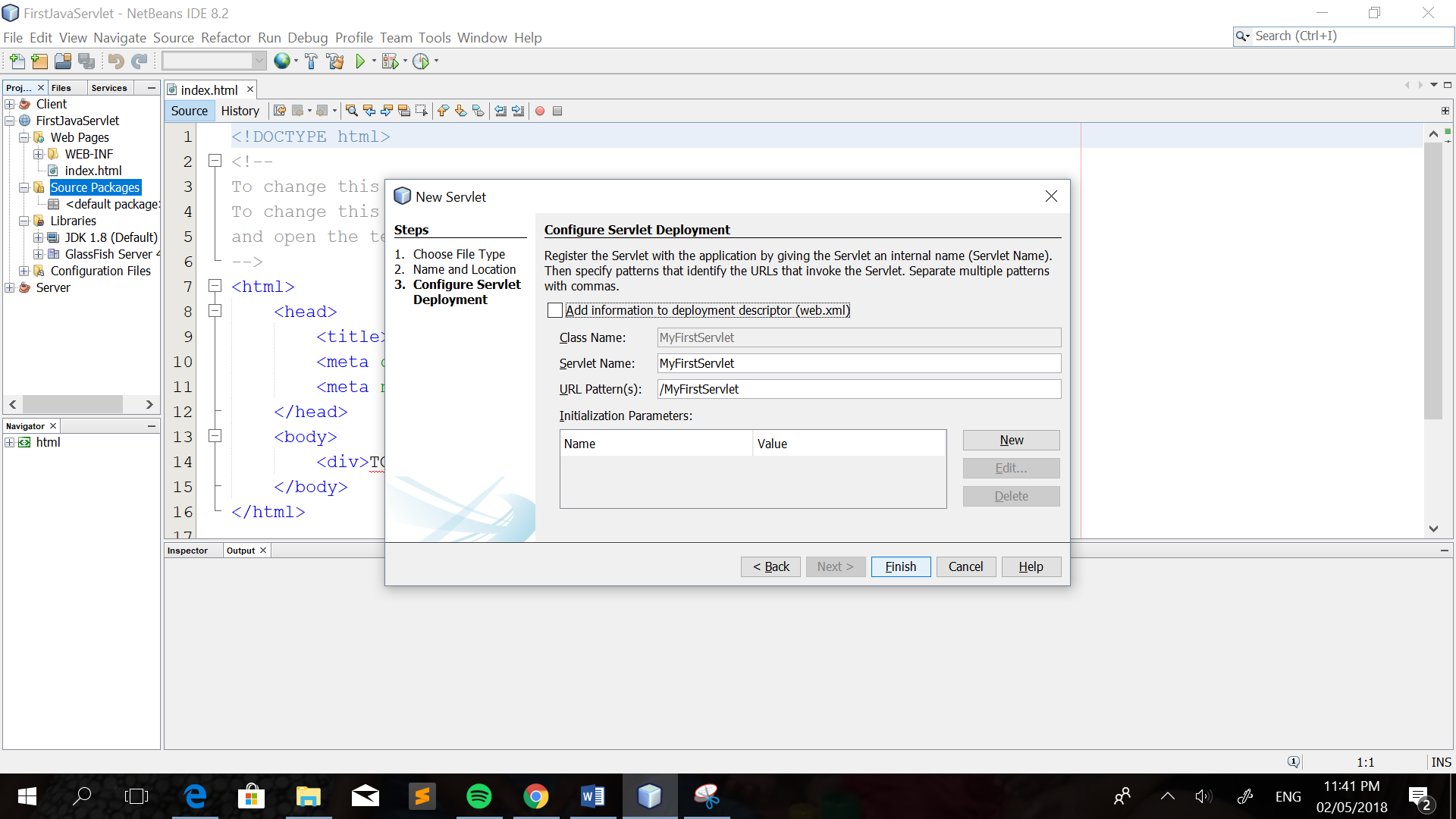


As you can see on the figure, the **WEB-INF** folder contains the **index.htm**l file and the libraries like the JDK 1.8 and the server library is found and stored on the Libraries folder.

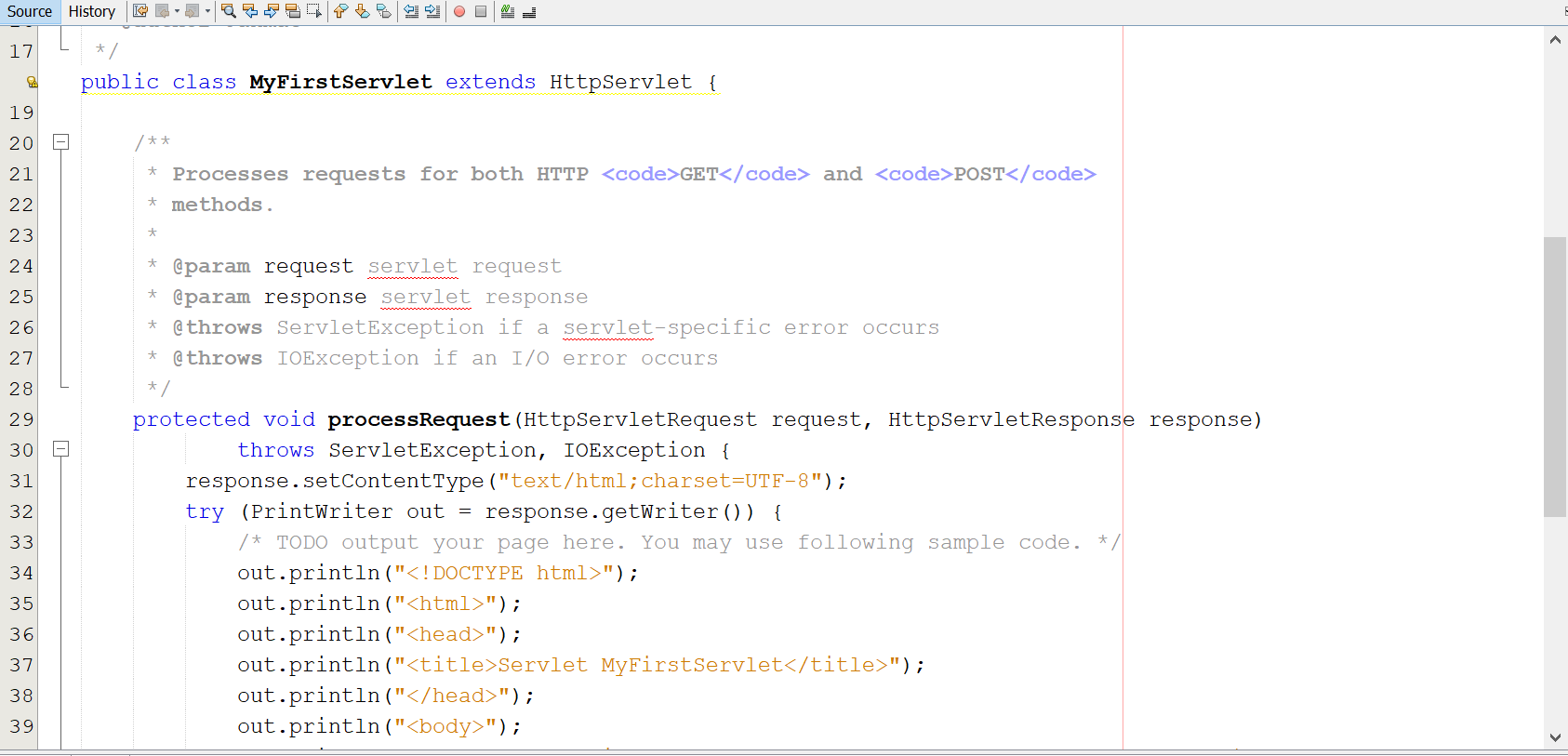
6. In creating a servlet, open **Source Package**, right click on **default packages** and then click **New** and finally click **Servlet**. Provide the name of your class, for this tutorial, I will have named it ‘MyFirstServlet’. Then click **‘Next’**.



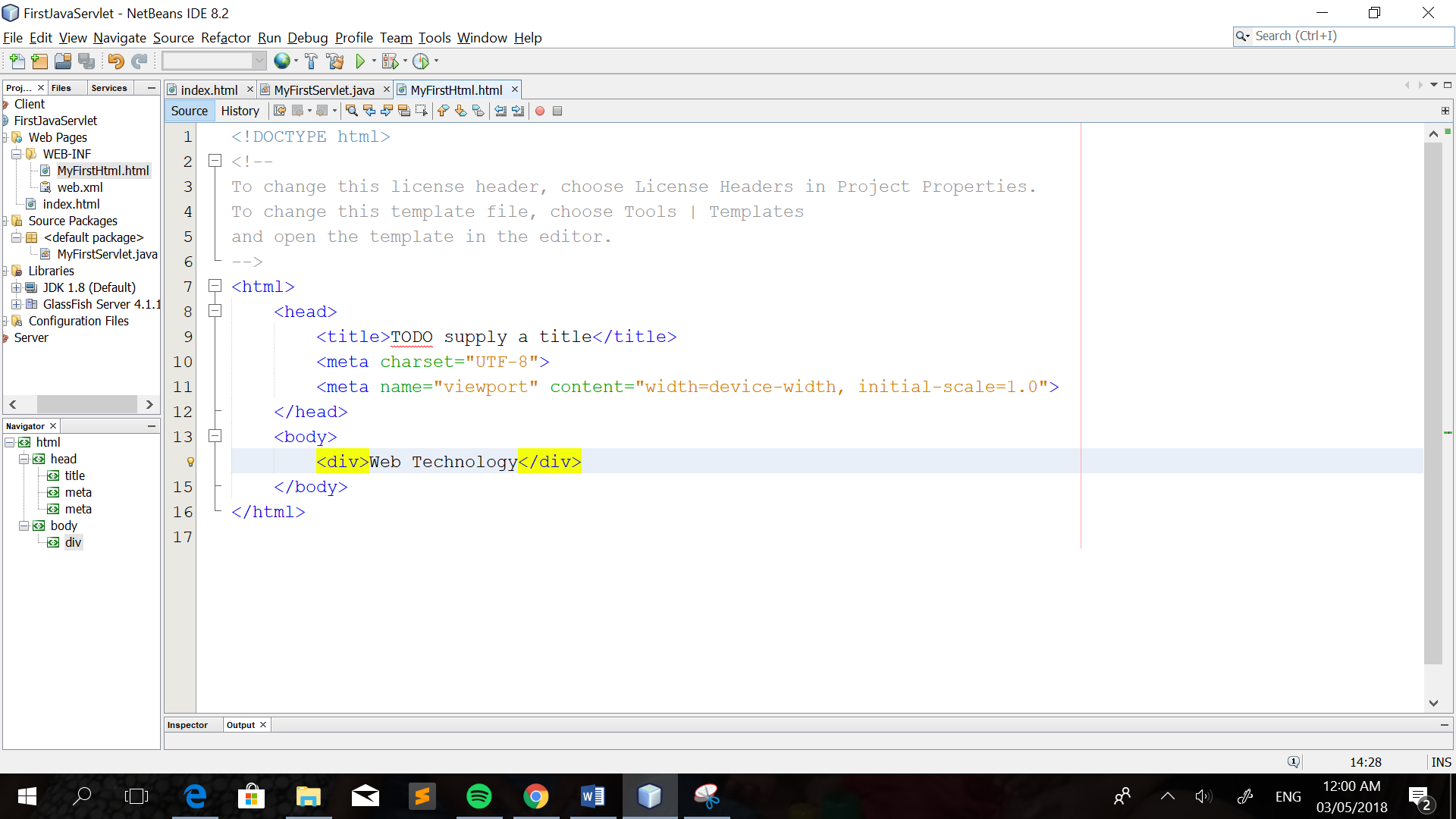
1. Next will be, **tick** the box indicating ‘Add information……’. This will add servlet information in web.xml file, it will auto generate the codes that are needed. Then click ‘**Finish**’.



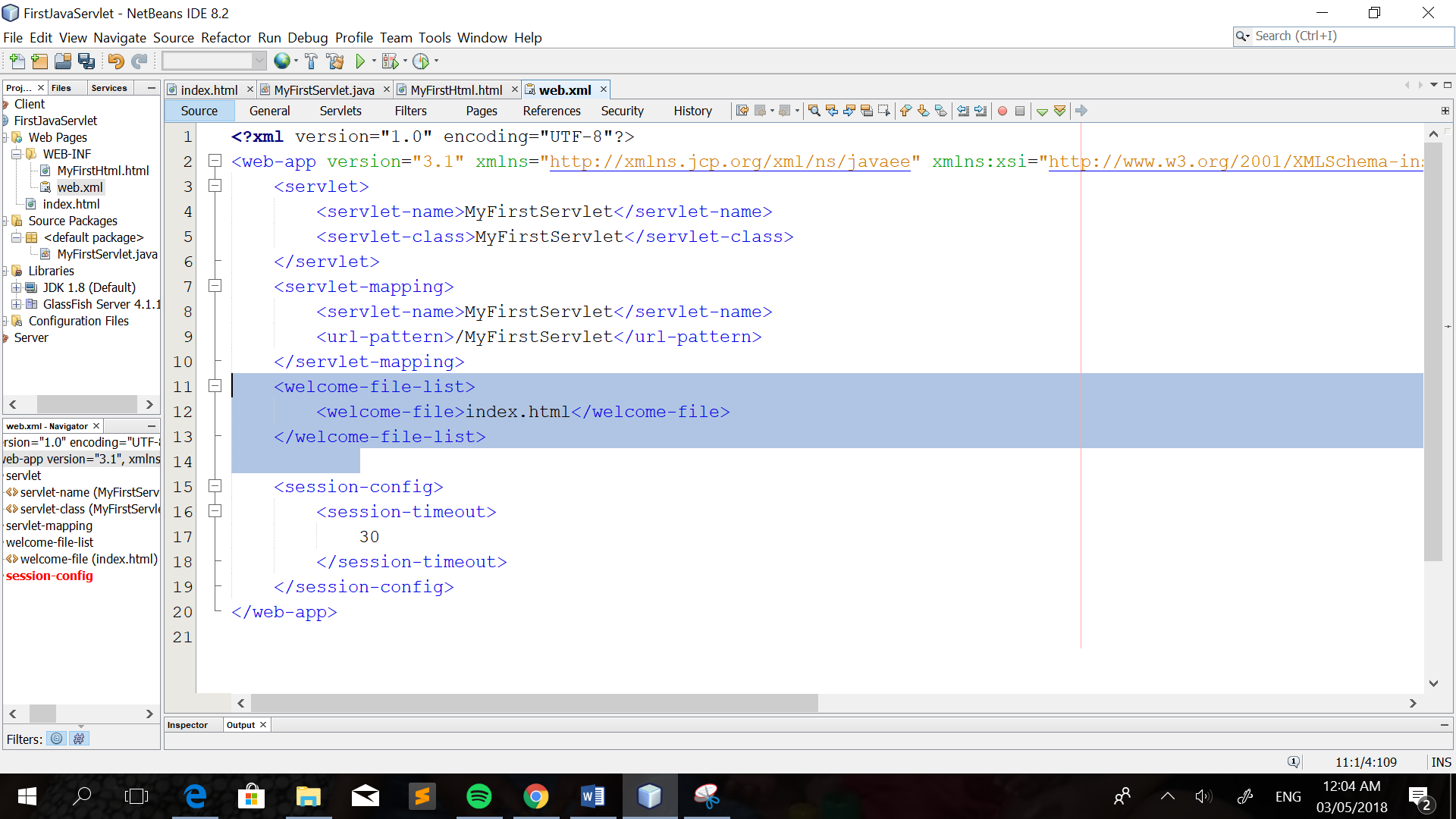
1. After that, the servlet class is already been created.You can edit and add codes if you want.



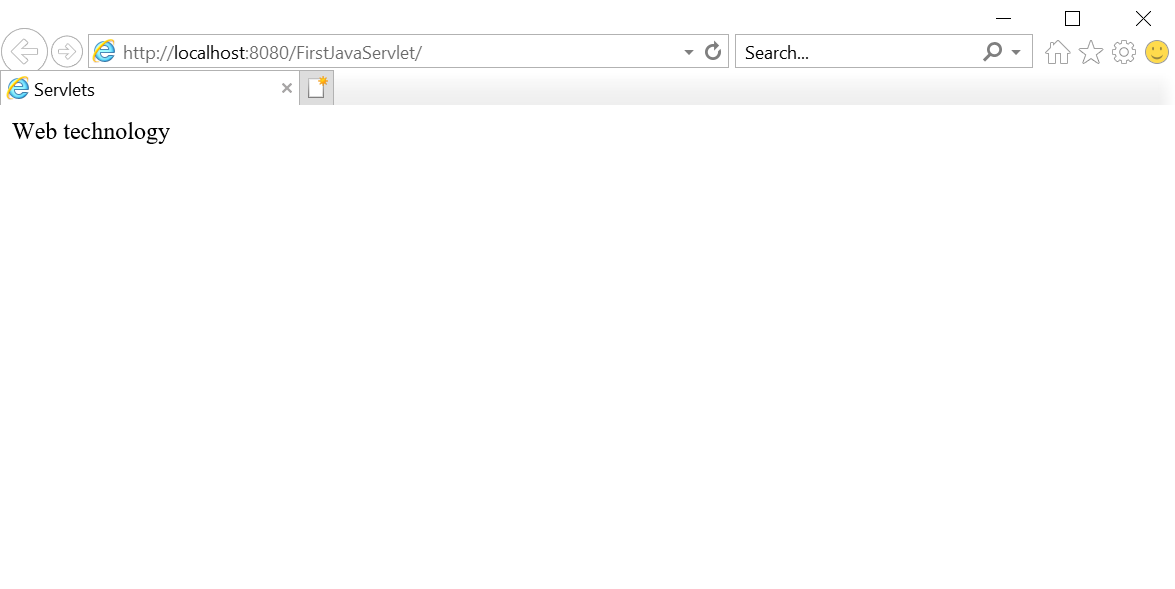
1. When creating a HTML file, right click on the **WEB-INF** folder located at the left pane under the FirstServlet Project, click new and then HTML. After creating the HTML file. Put the name of the HTML file, it is recommended to name it **index,** because browser will always pick up the **index.html** file automatically from a directory.



1. Next, edit web.xml file. In the web.xml file, add the following code highlighted. And the save it. Then, right click on your Project and select **Run**.



1. Finally, you have been successfully created your first-class servlets.



**HTTP Servlet**

Java Servlet use to handle HTTP request and generate HTTP response. It is hosted in ‘web-container’ which is component in Java ‘application-server’. Application server like WildFly, GlassFish, Apache Tomcat, IBM WebSphere and more.

What happens on the Request handling on the third phase of the servlets life cycle. The service() method call is routed to a d0XXX() call, depending on the HTTP request method , e.g. doGet(), doPost(), doHead(), doPut(), doDelete().

|  |  |
| --- | --- |
| doGet () | if the servlet supports HTTP GET requests |
| doPost() | for HTTP POST requests |
| doPut() | for HTTP PUT requests |
| doDelete() | for HTTP DELETE requests |
| Init()and destroy() method | to manage resources that are held for the life of the servlet |
| getServletInfo() | which the servlet uses to provide information about itself |

The doXXX() methods passed two arguments: the object representation of the HTTP request sent by the client and received by servlet. It is use to accessed information from the request message (e.g. HTTP request, request URI, query string, message readers, message payload).

**Interfaces of Servlet**

ServletRequest and HttpServletRequest

The ServletRequest and HttpServletRequest classes hold all of the accessible information about the client and the server. HttpServletRequest is a subclass of ServletRequest and is passed to each servlet handler method (such as doGet(..), doPut(..), etc.). (David Reily, Michael Reily,2012)

Example:

getContentLength(), getContentType(),getServerName(),getAuthType(),getServletPath(),getMethod()

ServletResponse and HttpResponse

For servlets, the ServletResponse and its subclass HttpResponse are two of the most vital classes. These classes perform the function of sending the MIME-encoded data back to the client. Without this, there would be no servlets at all. This class supports both writers (for communicating text-based characters) and output streams (for sending back binary data), as well as enabling the setting of content length and type parameters. (David Reily, Michael Reily,2012)

Example:

GetWriter, GetCharacterEncoding, SetContentType, AddCookie ,ContainsHeader ,SetHeader ,SetStatus Sets the HTTP status

References:

(n.d.). Retrieved from <https://www.studytonight.com/servlet/creating-servlet-in-netbeans.php>

Tutorials Point. (2018, February 13). Servlets Tutorial. Retrieved from https://www.tutorialspoint.com/servlets/index.htm