next →

# Servlets | Servlet Tutorial


**Servlet** technology is used to create a web application (resides at server side and generates a dynamic web page).

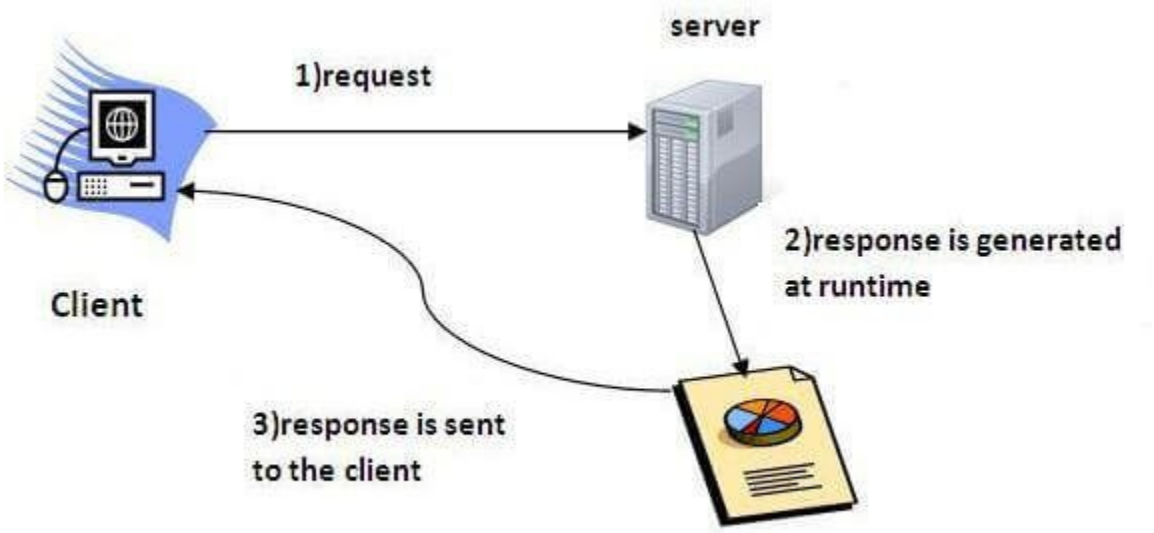
**Servlet** technology is robust and scalable because of java language. Before Servlet, CGI (Common Gateway Interface) scripting language was common as a server-side programming language. However, there were many disadvantages to this technology. We have discussed these disadvantages below.

There are many interfaces and classes in the Servlet API such as Servlet, GenericServlet, HttpServlet, ServletRequest, ServletResponse, etc.

## What is a Servlet?

Servlet can be described in many ways, depending on the context.

- Servlet is a technology which is used to create a web application.
- Servlet is an API that provides many interfaces and classes including documentation.
- Servlet is an interface that must be implemented for creating any Servlet.
- Servlet is a class that extends the capabilities of the servers and responds to the incoming requests. It can respond to any requests.
- Servlet is a web component that is deployed on the server to create a dynamic web page.




## Do You Know?

- What is the web application and what is the difference between Get and Post request?
- What information is received by the web server if we request for a Servlet?
- How to run servlet in Eclipse, MyEclipse and Netbeans IDE?
- What are the ways for servlet collaboration and what is the difference between RequestDispatcher and sendRedirect() method?
- What is the difference between ServletConfig and ServletContext interface?
- How many ways can we maintain the state of a user? Which approach is mostly used in web development?
- How to count the total number of visitors and whole response time for a request using Filter?
- How to run servlet with annotation?
- How to create registration form using Servlet and Oracle database?
- How can we upload and download the file from the server?

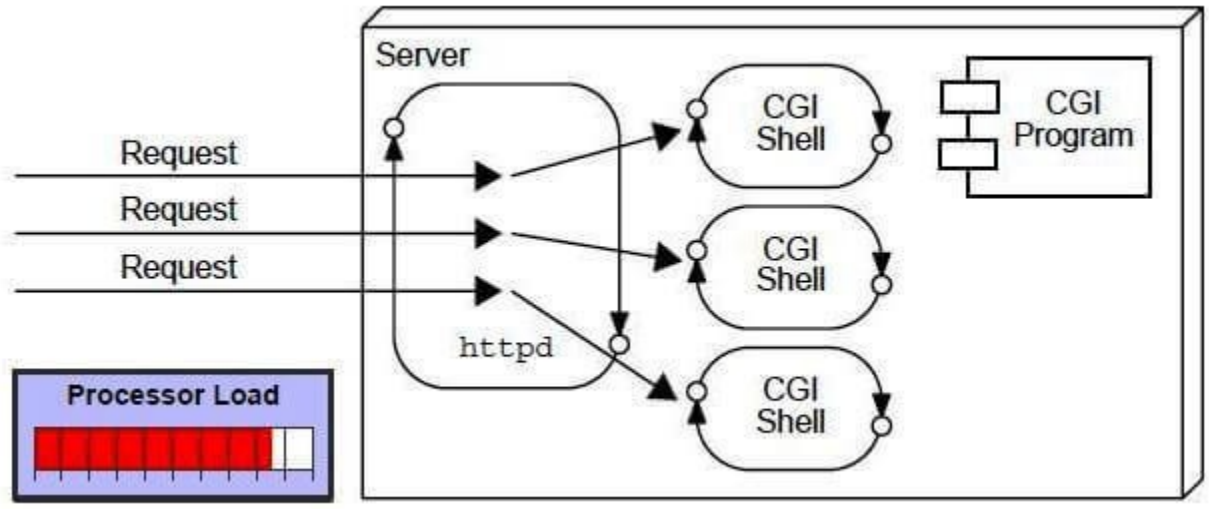
### What is a web application?

A web application is an application accessible from the web. A web application is composed of web components like Servlet, JSP, Filter, etc. and other elements such as HTML, CSS, and JavaScript. The web components typically execute in Web Server and respond to the HTTP request.

---

## CGI (Common Gateway Interface)

CGI technology enables the web server to call an external program and pass HTTP request information to the external program to process the request. For each request, it starts a new process.



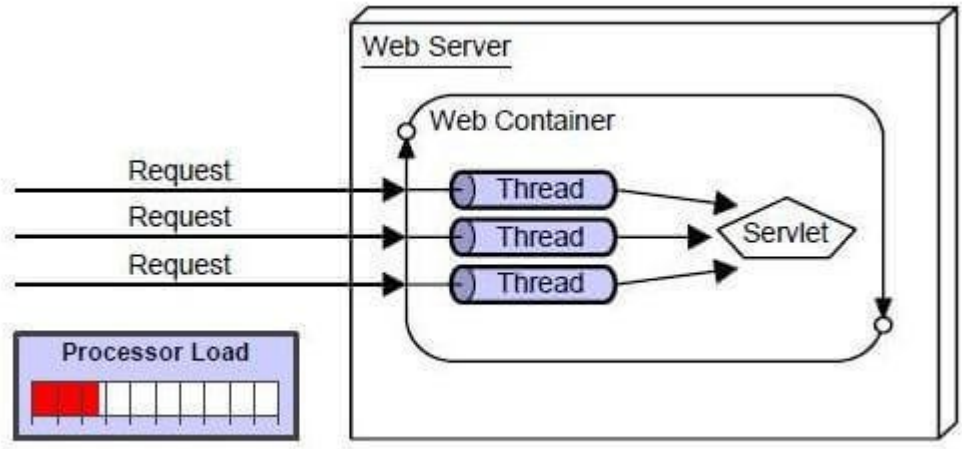

## Disadvantages of CGI

There are many problems in CGI technology:

1. If the number of clients increases, it takes more time for sending the response.
2. For each request, it starts a process, and the web server is limited to start processes.
3. It uses platform dependent language e.g. C, C++, perl.

---

## Advantages of Servlet



There are many advantages of Servlet over CGI. The web container creates threads for handling the multiple requests to the Servlet. Threads have many benefits over the Processes such as they share a common memory area, lightweight, cost of communication between the threads are low. The advantages of Servlet are as follows:

1. **Better performance:** because it creates a thread for each request, not process.
2. **Portability:** because it uses Java language.
3. **Robust:** JVM manages Servlets, so we don't need to worry about the memory leak, garbage collection, etc.
4. **Secure:** because it uses java language.

next →

## Servlet API

The javax.servlet and javax.servlet.http packages represent interfaces and classes for servlet api.

The **javax.servlet** package contains many interfaces and classes that are used by the servlet or web container. These are not specific to any protocol.

The **javax.servlet.http** package contains interfaces and classes that are responsible for http requests only.

Let's see what are the interfaces of javax.servlet package.

## Interfaces in javax.servlet package

There are many interfaces in javax.servlet package. They are as follows:

1. Servlet
2. ServletRequest
3. ServletResponse
4. RequestDispatcher
5. ServletConfig
6. ServletContext
7. SingleThreadModel
8. Filter
9. FilterConfig
10. FilterChain
11. ServletRequestListener
12. ServletRequestAttributeListener
13. ServletContextListener
14. ServletContextAttributeListener

## Classes in javax.servlet package

There are many classes in javax.servlet package. They are as follows:

1. GenericServlet
2. ServletInputStream
3. ServletOutputStream
4. ServletRequestWrapper
5. ServletResponseWrapper
6. ServletRequestEvent
7. ServletContextEvent
8. ServletRequestAttributeEvent

9. ServletContextAttributeEvent
10. ServletException
11. UnavailableException

---

## Interfaces in javax.servlet.http package

There are many interfaces in javax.servlet.http package. They are as follows:

1. HttpServletRequest
2. HttpServletResponse
3. HttpSession
4. HttpSessionListener
5. HttpSessionAttributeListener
6. HttpSessionBindingListener
7. HttpSessionActivationListener
8. HttpSessionContext (deprecated now)

## Classes in javax.servlet.http package

There are many classes in javax.servlet.http package. They are as follows:

1. HttpServlet
2. Cookie
3. HttpServletRequestWrapper
4. HttpServletResponseWrapper
5. HttpSessionEvent
6. HttpSessionBindingEvent
7. HttpUtils (deprecated now)

<<prev next>>

# Servlet Interface



next>> <<prev

**Servlet interface provides** commonbehaviorto all the servlets.Servlet interface defines methods that all servlets must implement.

Servlet interface needs to be implemented for creating any servlet (either directly or indirectly). It provides 3 life cycle methods that are used to initialize the servlet, to service the requests, and to destroy the servlet and 2 non-life cycle methods.

## Methods of Servlet interface

There are 5 methods in Servlet interface. The init, service and destroy are the life cycle methods of servlet. These are invoked by the web container.

| Method | Description |
|---|---|
| **public void init(ServletConfig config)** | initializes the servlet. It is the life cycle method of servlet and invoked by the web container only once. |
| **public void service(ServletRequest request,ServletResponse response)** | provides response for the incoming request. It is invoked at each request by the web container. |
| **public void destroy()** | is invoked only once and indicates that servlet is being destroyed. |
| **public ServletConfig getServletConfig()** | returns the object of ServletConfig. |
| **public String getServletInfo()** | returns information about servlet such as writer, copyright, version etc. |

## Servlet Example by implementing Servlet interface

Let's see the simple example of servlet by implementing the servlet interface.

<<prev next>>

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

public class First implements Servlet{
ServletConfig config=null;

public void init(ServletConfig config){
this.config=config;
System.out.println("servlet is initialized");
}

public void service(ServletRequest req,ServletResponse res)
throws IOException,ServletException{

res.setContentType("text/html");

PrintWriter out=res.getWriter();
out.print("<html><body>");
out.print("<b>hello simple servlet</b>");
out.print("</body></html>");

}
public void destroy(){System.out.println("servlet is destroyed");}
public ServletConfig getServletConfig(){return config;}
public String getServletInfo(){return "copyright 2007-1010";}

}
```

# Creating Servlet Example in Eclipse

Eclipse is an open-source ide for developing JavaSE and JavaEE (J2EE) applications. You can download the eclipse ide from the eclipse website http://www.eclipse.org/downloads/.

You need to download the eclipse ide for JavaEE developers.

Creating **servlet example in eclipse ide**, saves a lot of work to be done. It is easy and simple to create a servlet example. Let's see the steps, you need to follow to create the first servlet example.

- Create a Dynamic web project
- create a servlet
- add servlet-api.jar file
- Run the servlet

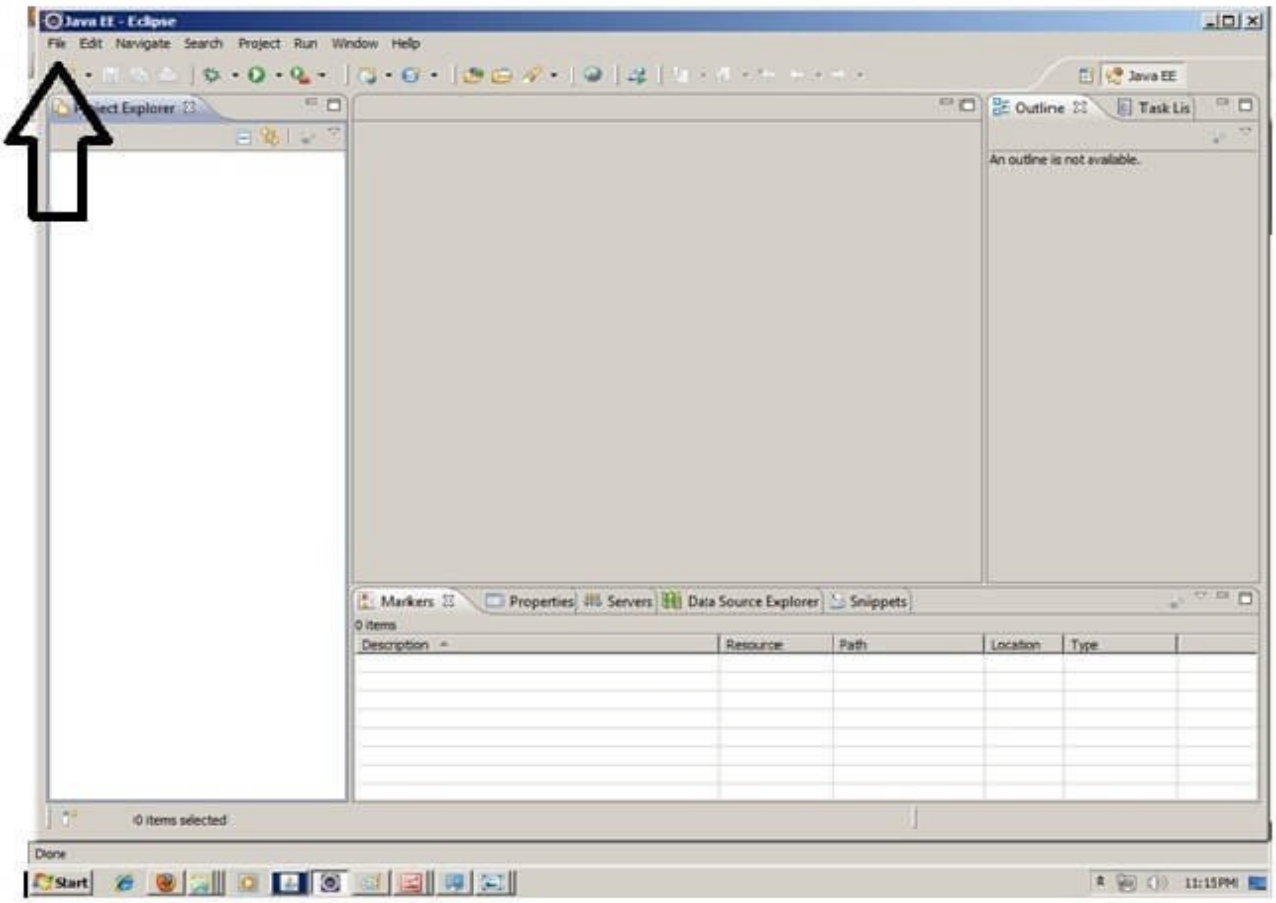
---

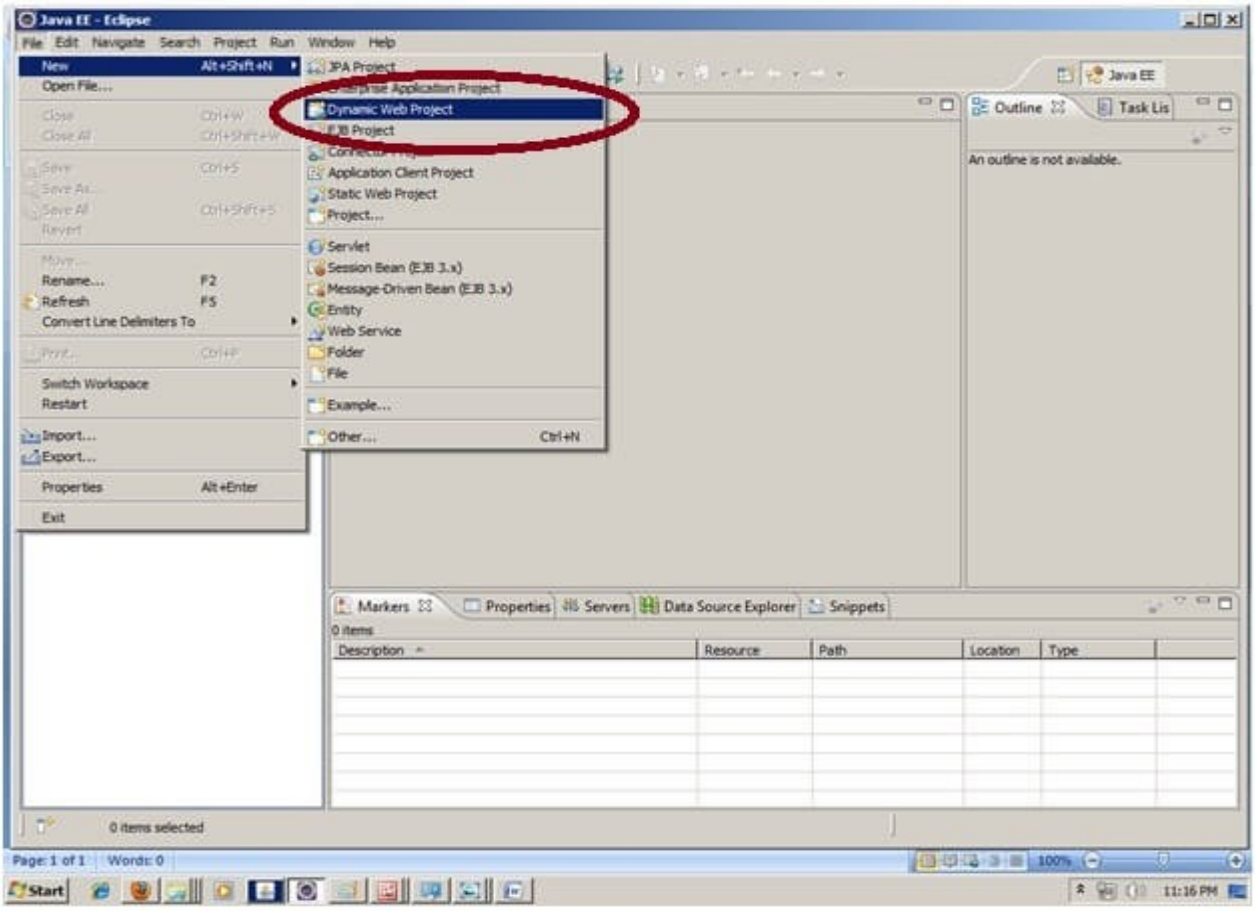
download this example (developed in eclipse)

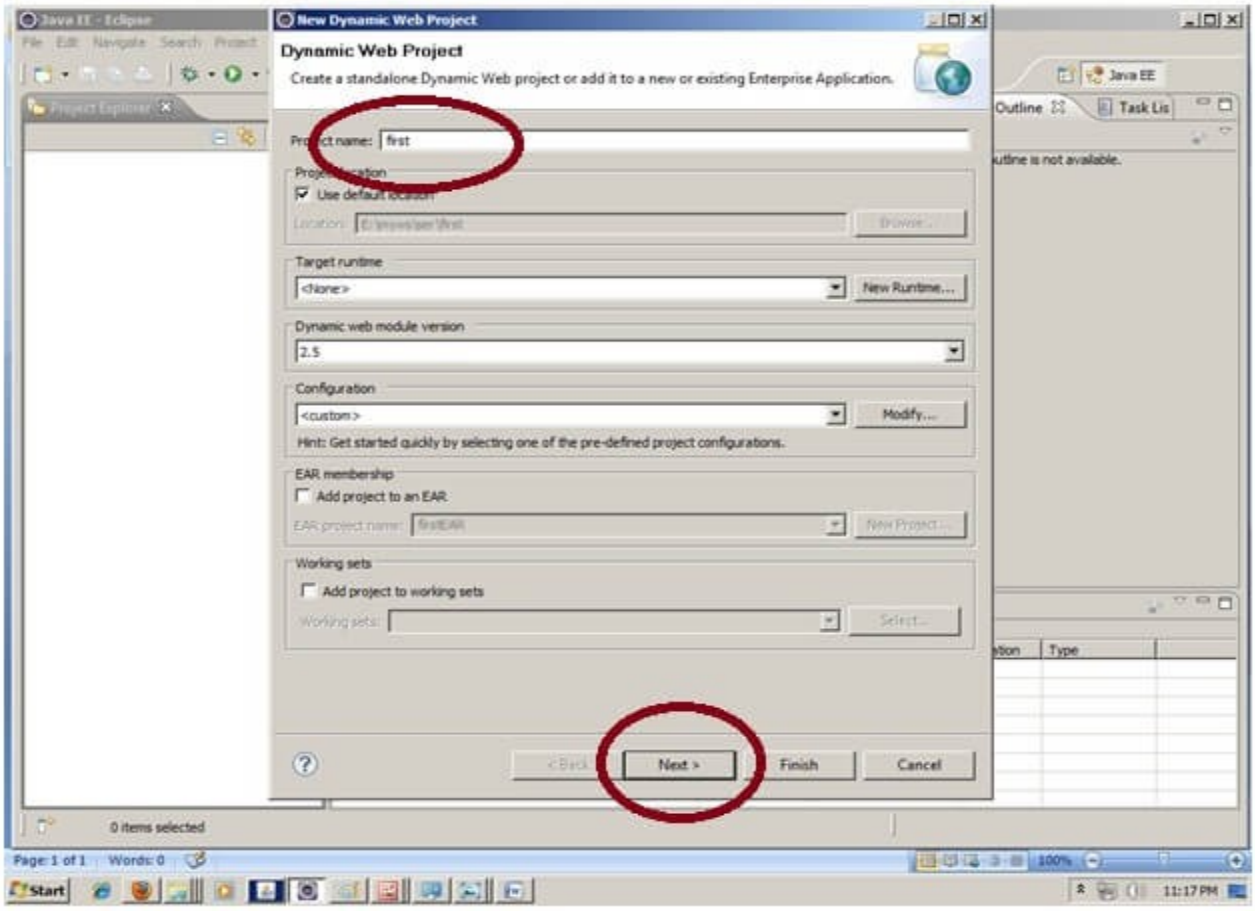
---

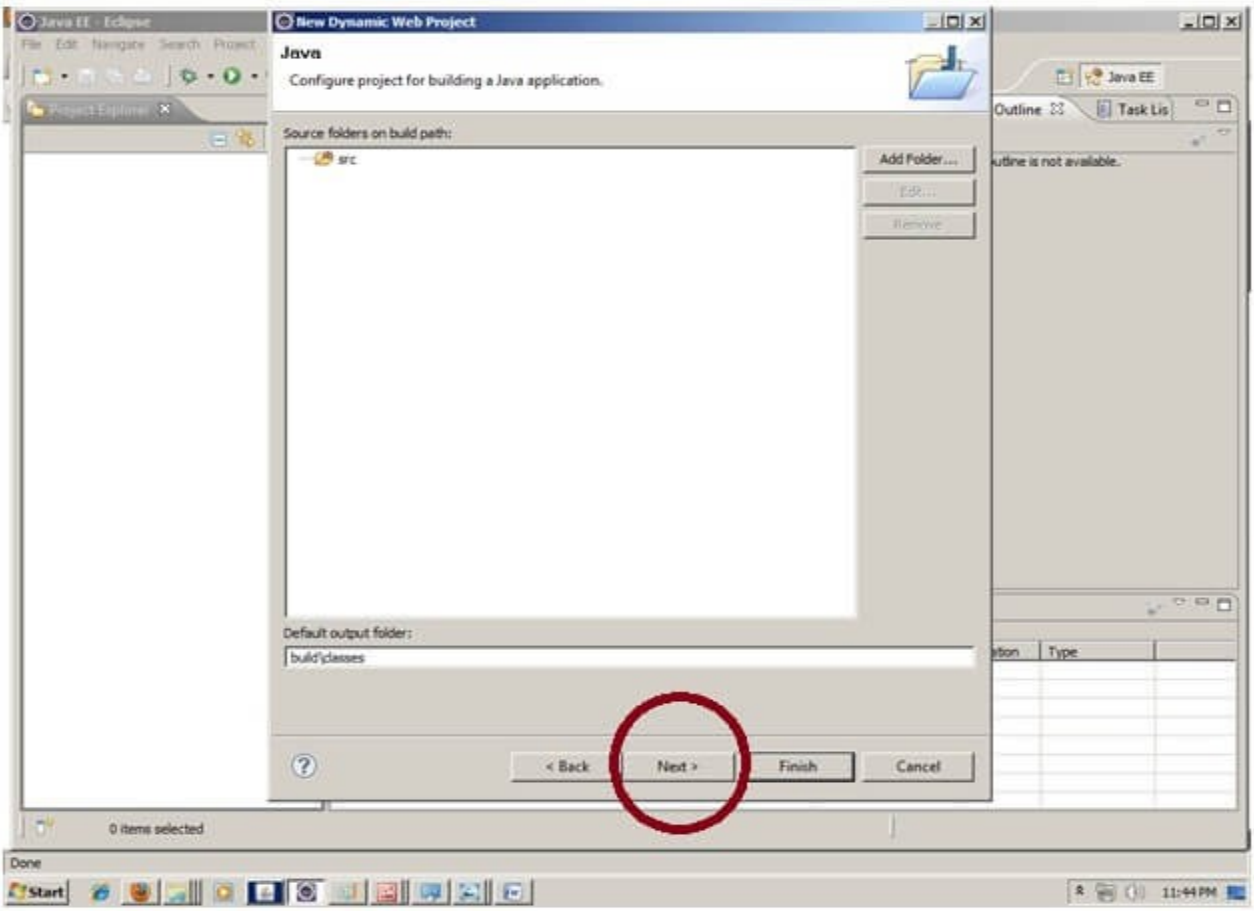
## 1) Create the dynamic web project:

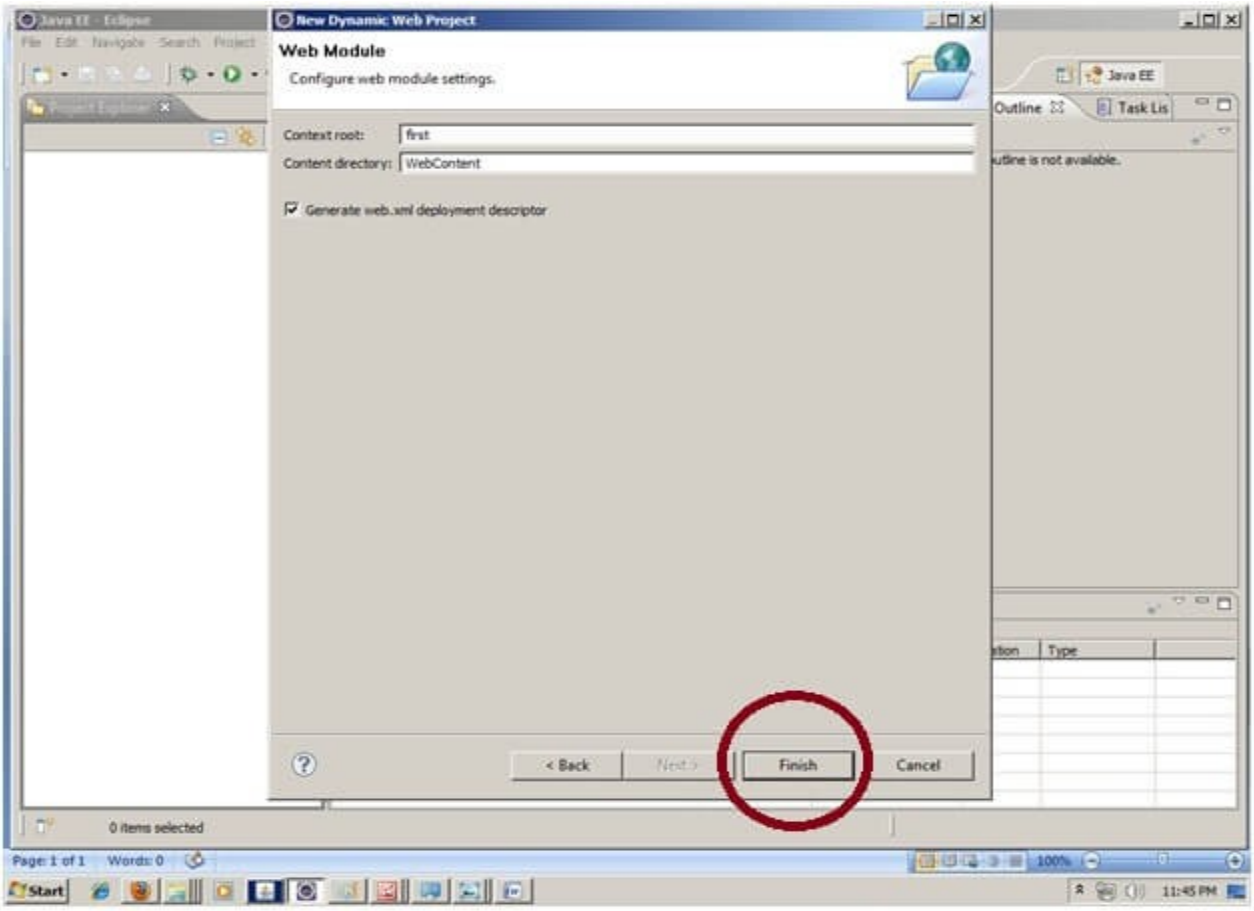
For creating a dynamic web project **click on File Menu -> New -> Project..-> Web -> dynamic web project -> write your project name e.g. first -> Finish**.

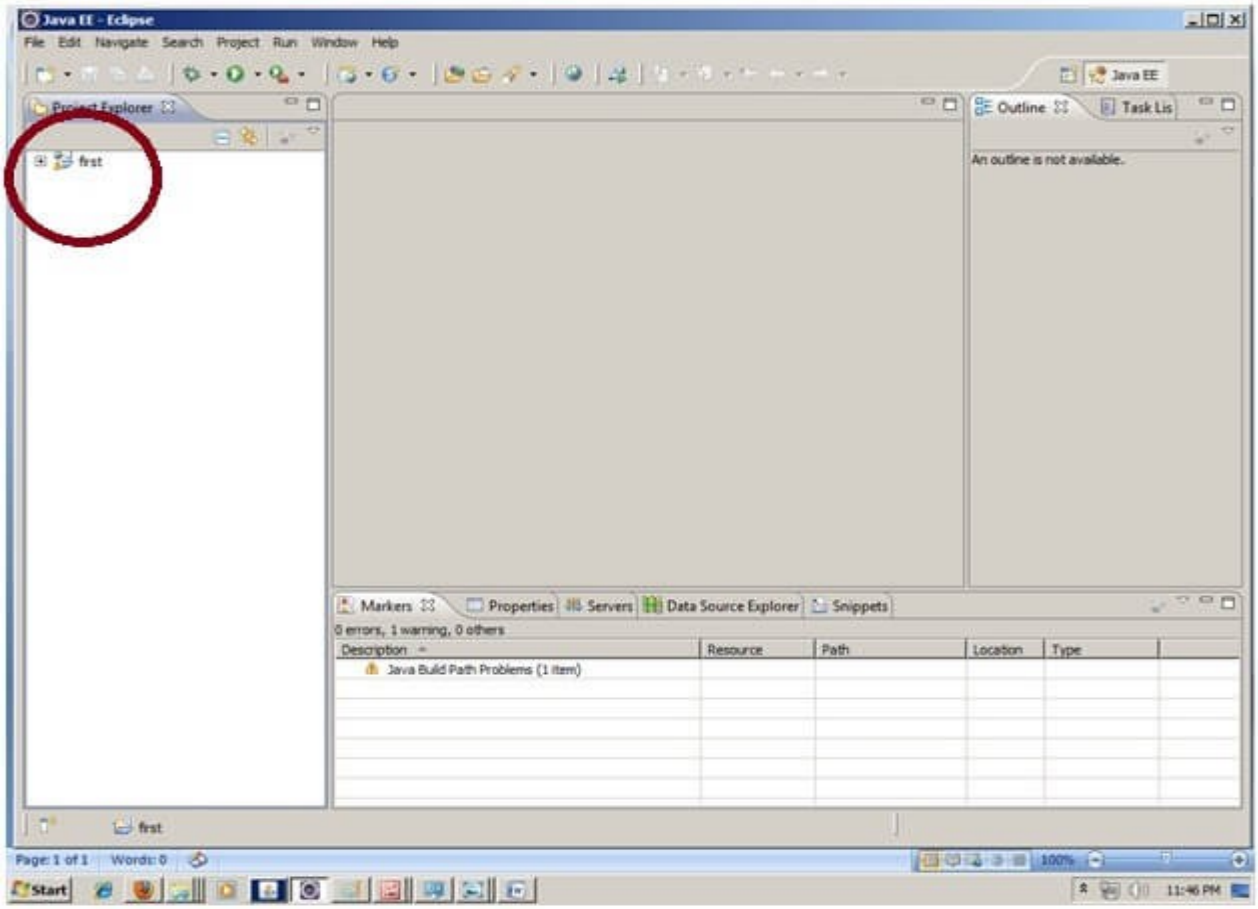

Java EE - Eclipse
File Edit Navigate Search Project Run Window Help
Java EE
Outline
Task Lis
An outline is not available.
Markers
Properties
Servers
Data Source Explorer
Snippets
0 items
Description
Resource
Path
Location
Type
0 items selected
Done
Start
11:15 PM


Java EE - Eclipse
File Edit Navigate Search Project Run Window Help
New Alt+Shift+N
Open File...
Close Ctrl+W
Close All Ctrl+Shift+W
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Revert
Move...
Rename... F2
Refresh F5
Convert Line Delimiters To
Print... Ctrl+P
Switch Workspace
Restart
Import...
Export...
Properties Alt+Enter
Exit
JPA Project
Enterprise Application Project
Dynamic Web Project
EJB Project
Connector Project
Application Client Project
Static Web Project
Project...
Servlet
Session Bean (EJB 3.x)
Message-Driven Bean (EJB 3.x)
Entity
Web Service
Folder
File
Example...
Other... Ctrl+N
Java EE
Outline
Task List
An outline is not available.
Markers
Properties
Servers
Data Source Explorer
Snippets
0 items
Description
Resource
Path
Location
Type
0 items selected
Page: 1 of 1
Words: 0
100%
Start
11:16 PM


New Dynamic Web Project
Dynamic Web Project
Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.
Project name: first
Use default location
Target runtime
<None>
New Runtime...
Dynamic web module version
2.5
Configuration
<custom>
Modify...
Hint: Get started quickly by selecting one of the pre-defined project configurations.
EAR membership
Add project to an EAR
Working sets
Add project to working sets
Select...
< Back
Next >
Finish
Cancel
Outline
Task List
Java EE
Type
0 items selected

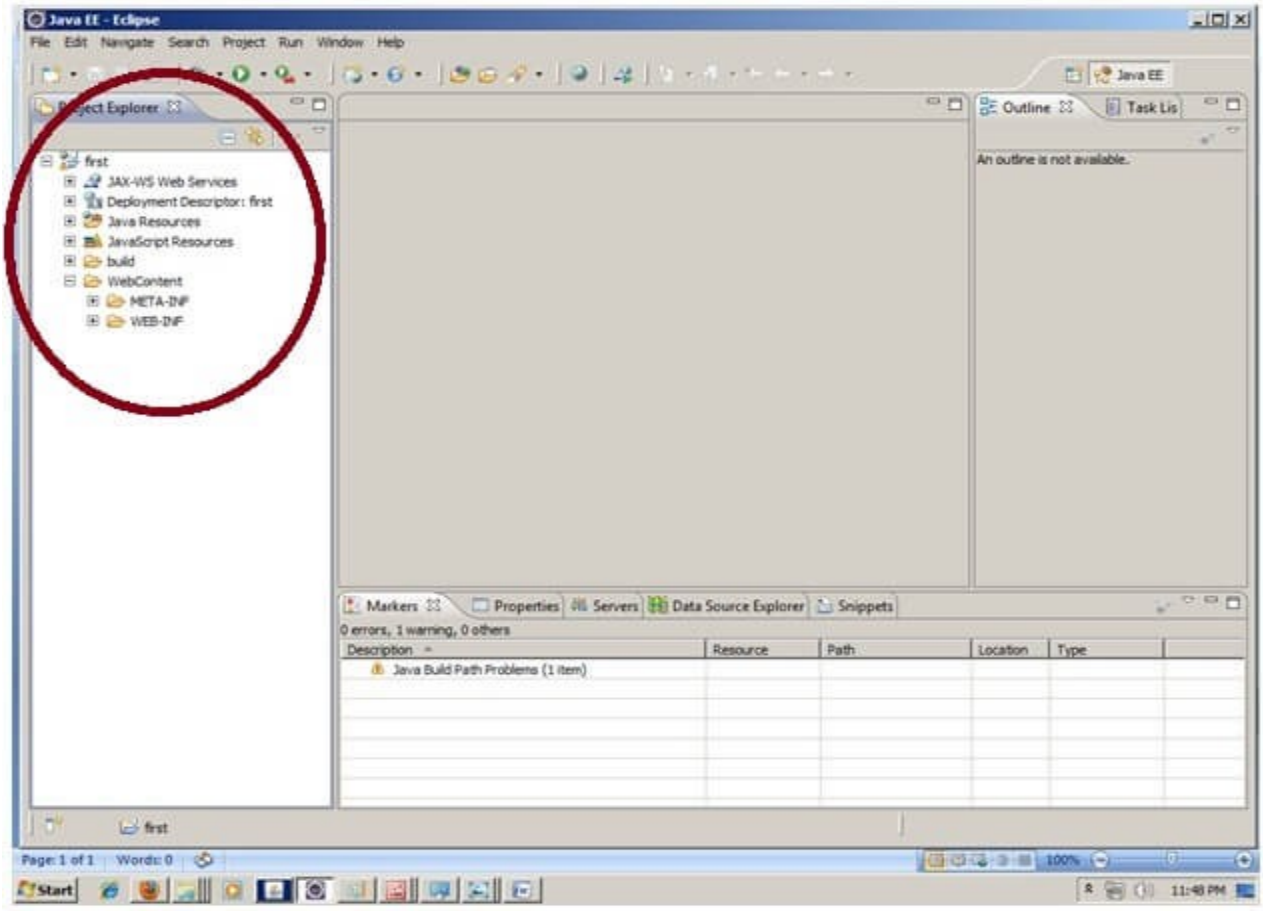

New Dynamic Web Project
Java
Configure project for building a Java application.
Source folders on build path:
src
Add Folder...
Edit...
Remove
Default output folder:
build\classes
< Back
Next >
Finish
Cancel
Java EE
Outline
Task Lis
utline is not available.
Type
0 items selected
Done
Start
11:44 PM

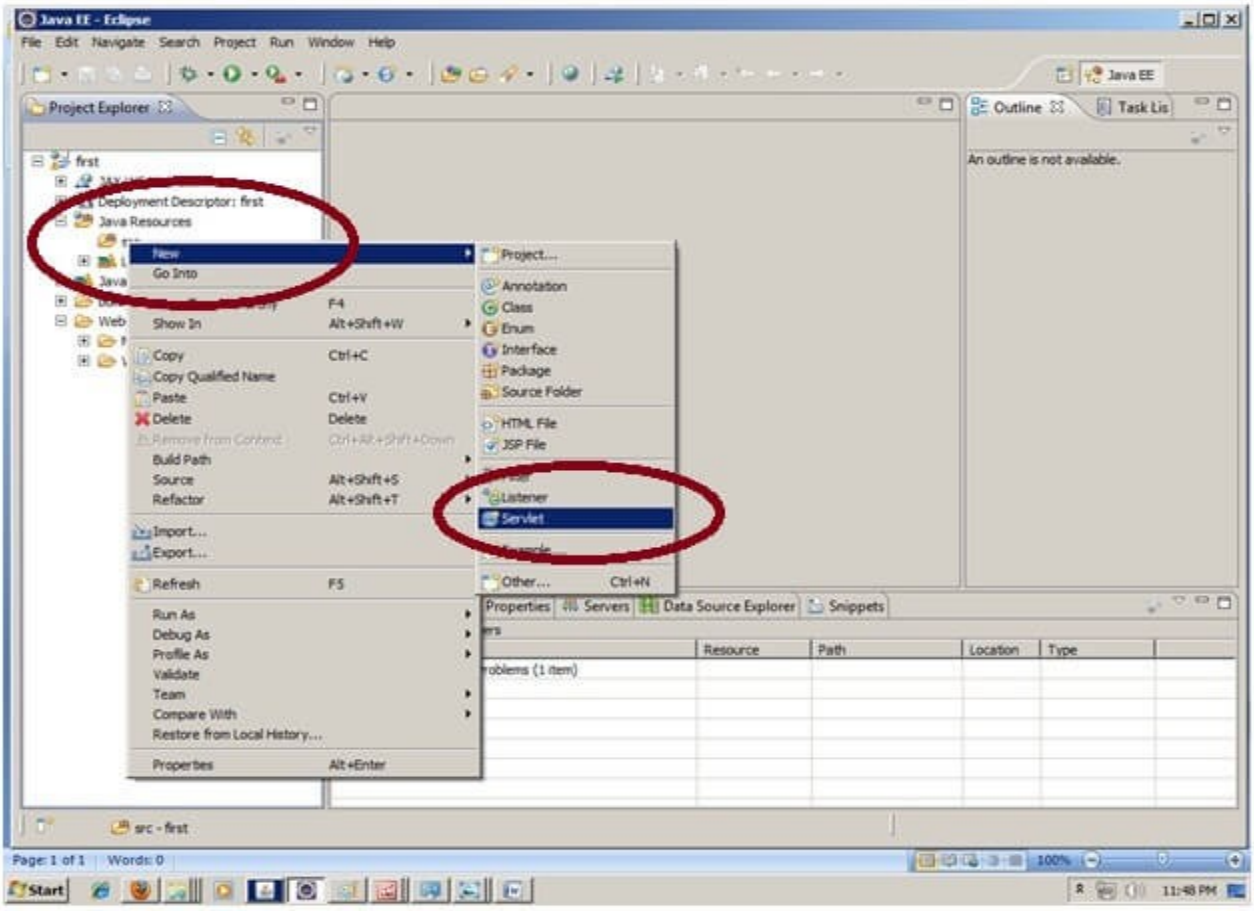

New Dynamic Web Project
Web Module
Configure web module settings.
Context root:
first
Content directory:
WebContent
Generate web.xml deployment descriptor
< Back
Next >
Finish
Cancel
Java EE
Outline
Task Lis
Type
0 items selected
Page: 1 of 1
Words: 0
100%
Start
11:45 PM

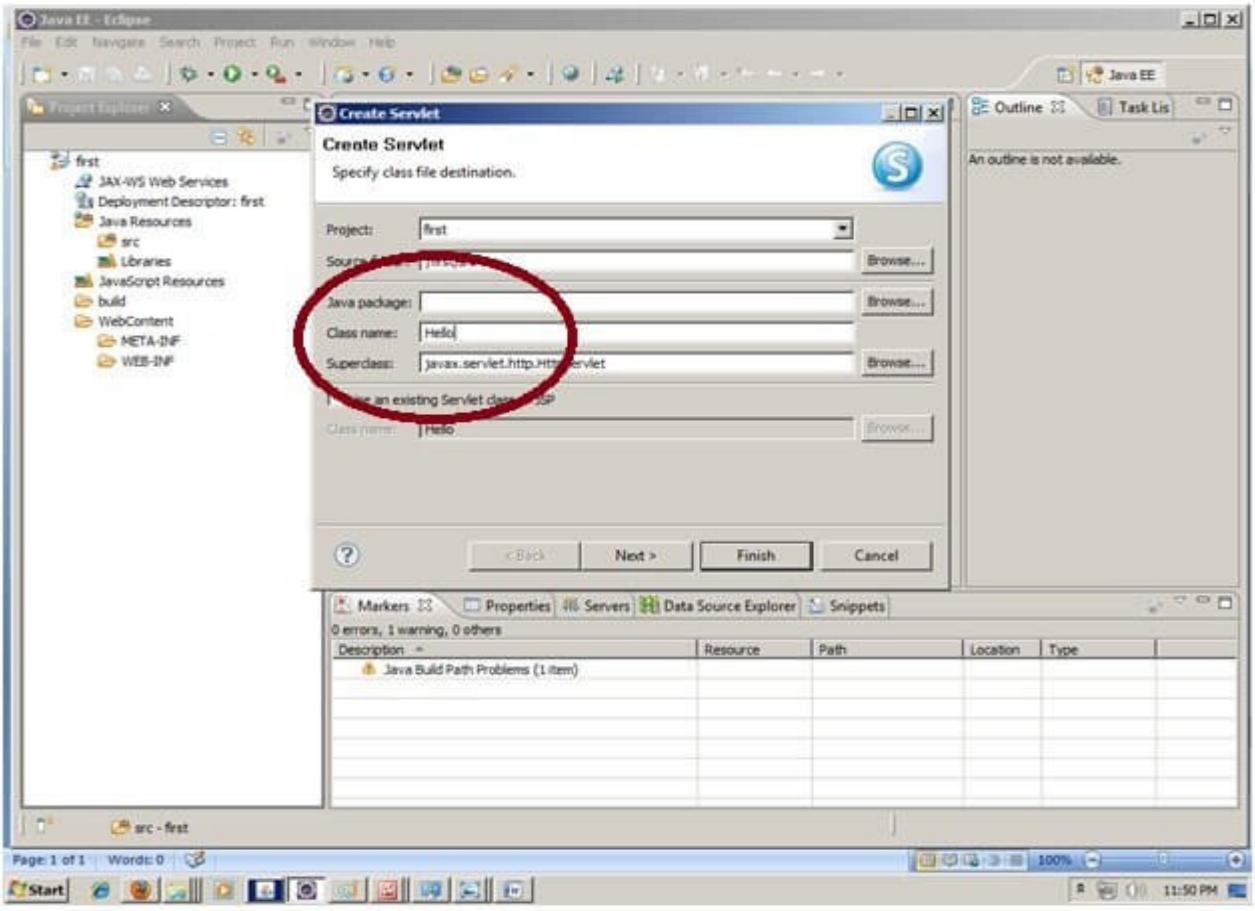


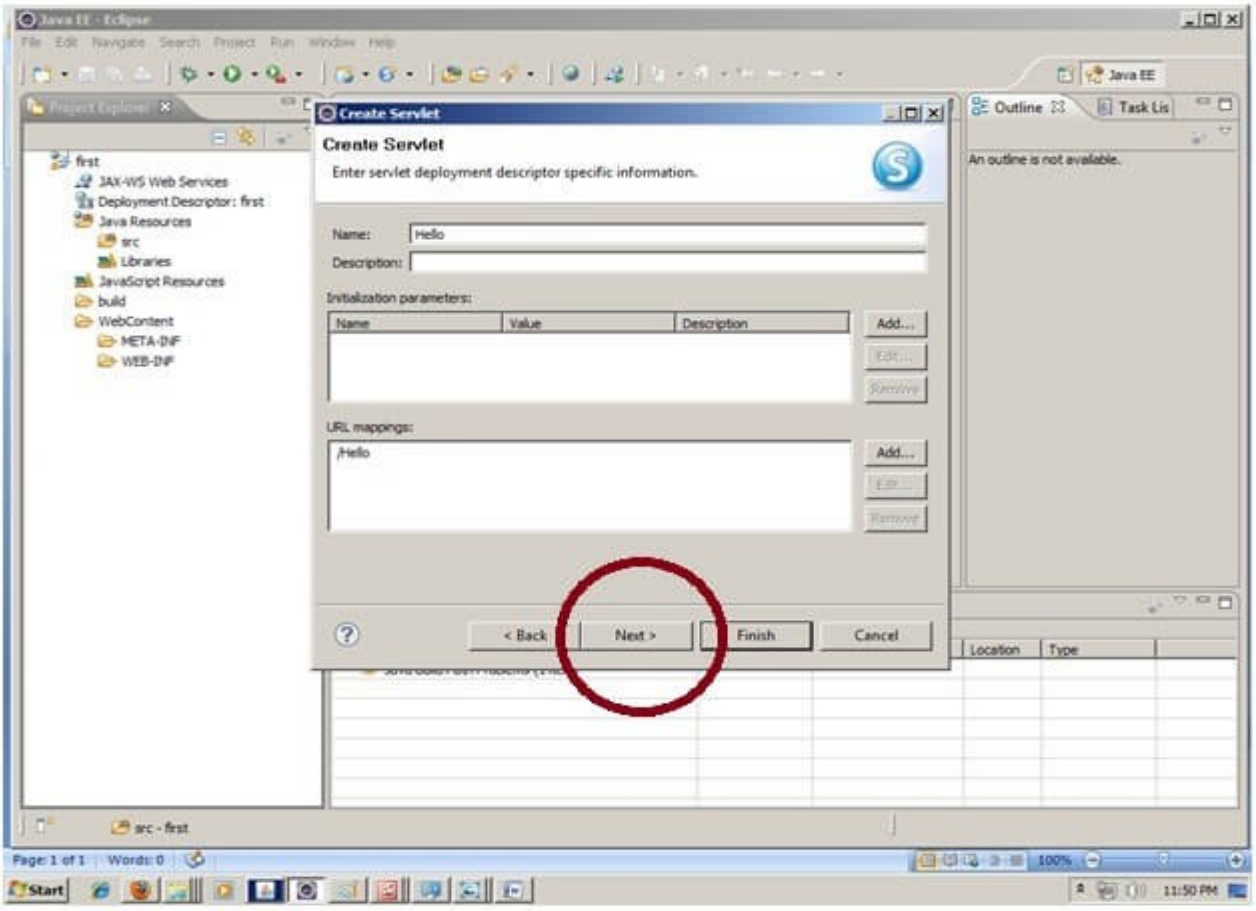

## 2) Create the servlet in eclipse IDE:

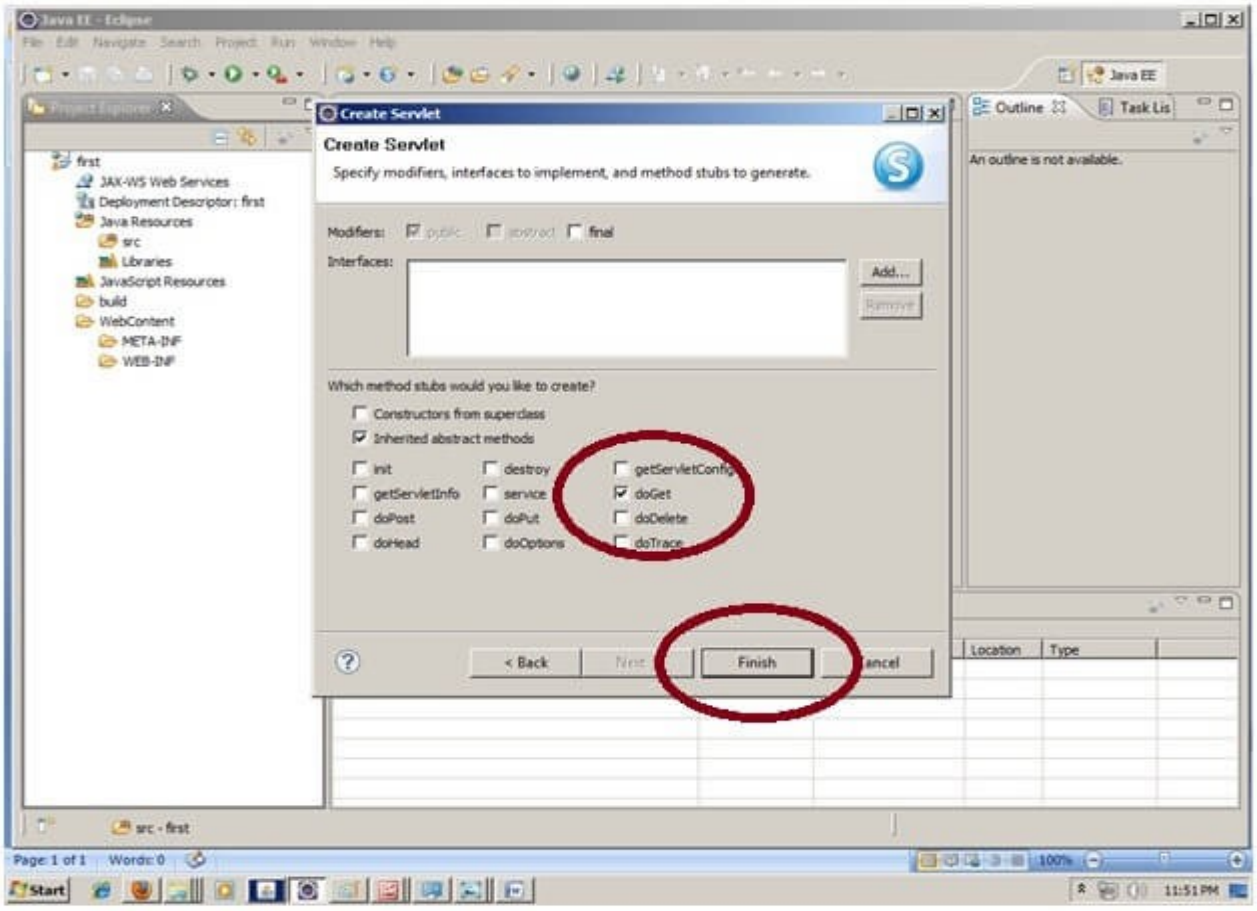
For creating a servlet, **explore the project by clicking the + icon -> explore the Java Resources -> right click on src -> New -> servlet -> write your servlet name e.g. Hello -> uncheck all the checkboxes except doGet() -> next -> Finish**.

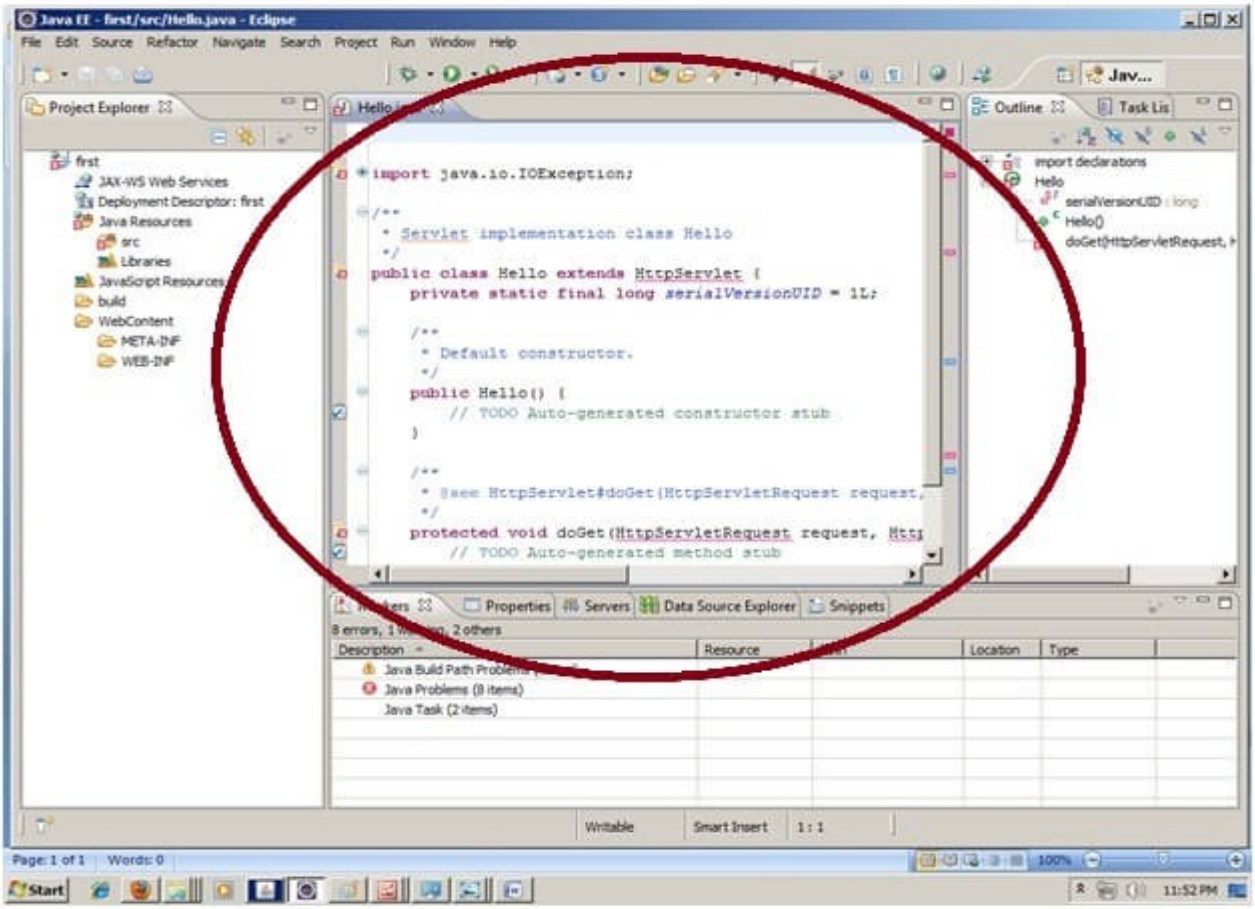

Java EE - Eclipse
File Edit Navigate Search Project Run Window Help
Java EE
Project Explorer
first
JAX-WS Web Services
Deployment Descriptor: first
Java Resources
JavaScript Resources
build
WebContent
META-INF
WEB-INF
Outline
Task Lis
An outline is not available.
Markers
Properties
Servers
Data Source Explorer
Snippets
0 errors, 1 warning, 0 others
Description
Resource
Path
Location
Type
Java Build Path Problems (1 item)
first
Page: 1 of 1
Words: 0
100%
Start
11:48 PM


Java EE - Eclipse
File Edit Navigate Search Project Run Window Help
Java EE
Project Explorer
Outline
Task Lis
An outline is not available.
first
Deployment Descriptor: first
Java Resources
New
Go Into
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Run As
Debug As
Profile As
Validate
Team
Compare With
Restore from Local History...
Properties Alt+Enter
Project...
Annotation
Class
Enum
Interface
Package
Source Folder
HTML File
JSP File
Listener
Servlet
Other... Ctrl+N
Properties
Servers
Data Source Explorer
Snippets
Resource
Path
Location
Type
Problems (1 item)
src - first
Page: 1 of 1
Words: 0
100%
Start
11:48 PM


Create Servlet
Create Servlet
Specify class file destination.
Project:
first
Java package:
Class name:
Hello
Superclass:
javax.servlet.http.HttpServlet
Browse...
Next >
Finish
Cancel
Outline
An outline is not available.
Markers
Properties
Servers
Data Source Explorer
Snippets
0 errors, 1 warning, 0 others
Description
Resource
Path
Location
Type
Java Build Path Problems (1 item)
first
JAX-WS Web Services
Deployment Descriptor: first
Java Resources
src
Libraries
JavaScript Resources
build
WebContent
META-INF
WEB-INF

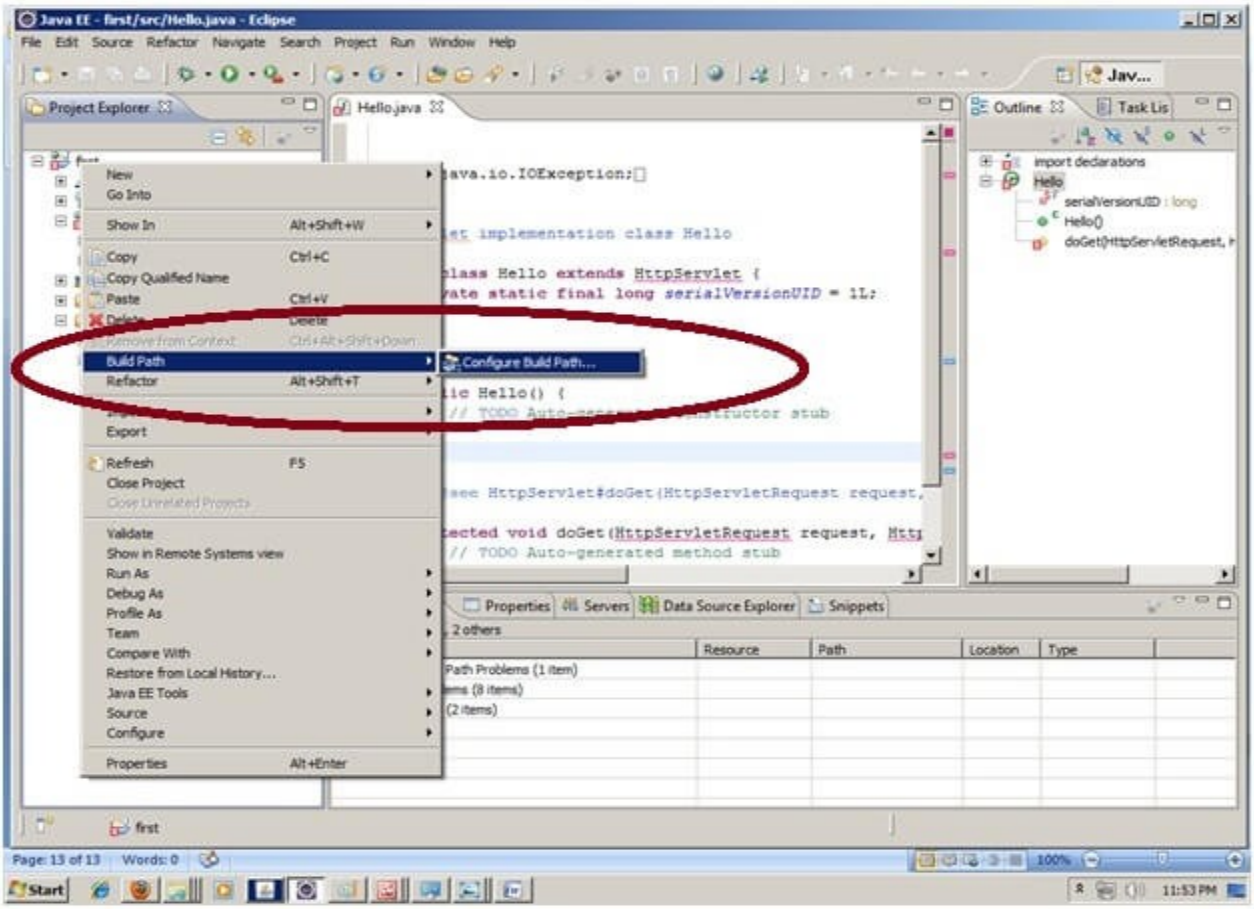

Java EE - Eclipse
File Edit Navigate Search Project Run Window Help
Java EE
Project Explorer
first
JAX-WS Web Services
Deployment Descriptor: first
Java Resources
src
Libraries
JavaScript Resources
build
WebContent
META-INF
WEB-INF
Create Servlet
Create Servlet
Enter servlet deployment descriptor specific information.
Name: Hello
Description:
Initialization parameters:
Name
Value
Description
Add...
Edit...
Remove
URL mappings:
/Hello
Add...
Edit...
Remove
< Back
Next >
Finish
Cancel
Outline
Task List
An outline is not available.
Location
Type
src - first
Page: 1 of 1
Words: 0
100%
Start
11:50 PM

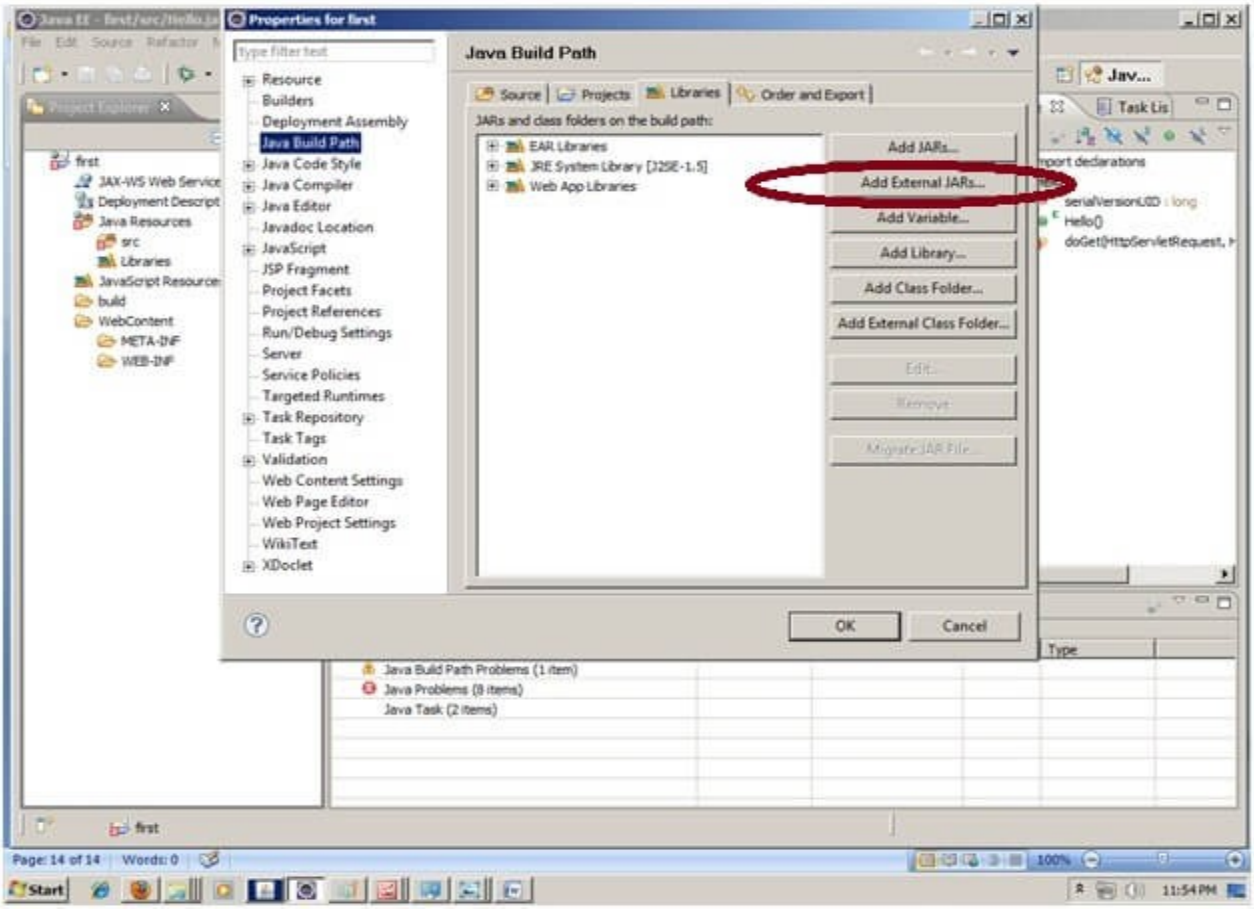

Java EE - Eclipse
Create Servlet
Create Servlet
Specify modifiers, interfaces to implement, and method stubs to generate.
Modifiers:
public
abstract
final
Interfaces:
Add...
Remove
Which method stubs would you like to create?
Constructors from superclass
Inherited abstract methods
init
destroy
getServletConfig
getServletInfo
service
doGet
doPost
doPut
doDelete
doHead
doOptions
doTrace
< Back
Next
Finish
Cancel
Outline
Task List
An outline is not available.
Location
Type
first
JAX-WS Web Services
Deployment Descriptor: first
Java Resources
src
Libraries
JavaScript Resources
build
WebContent
META-INF
WEB-INF
src - first
Page: 1 of 1
Words: 0
Start
11:51 PM

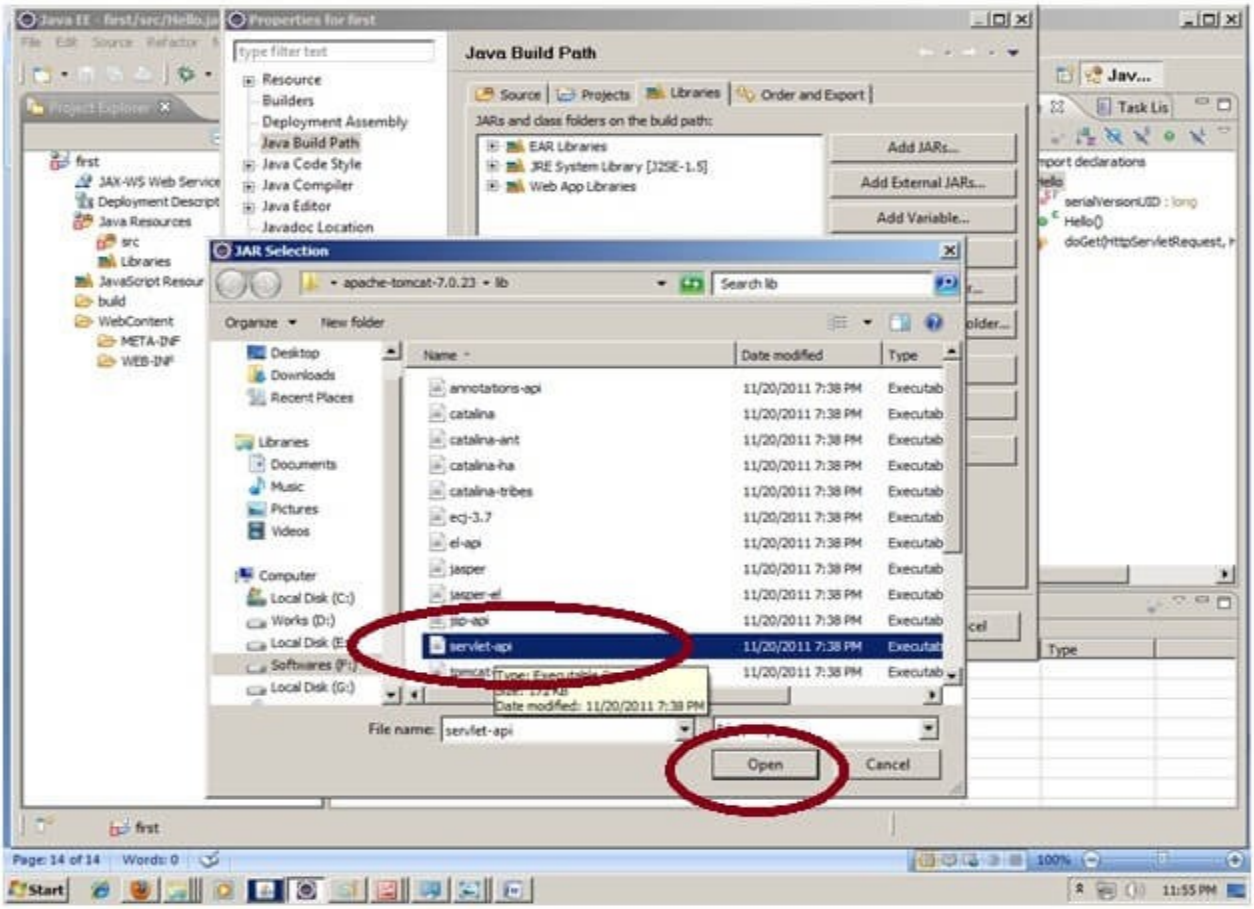


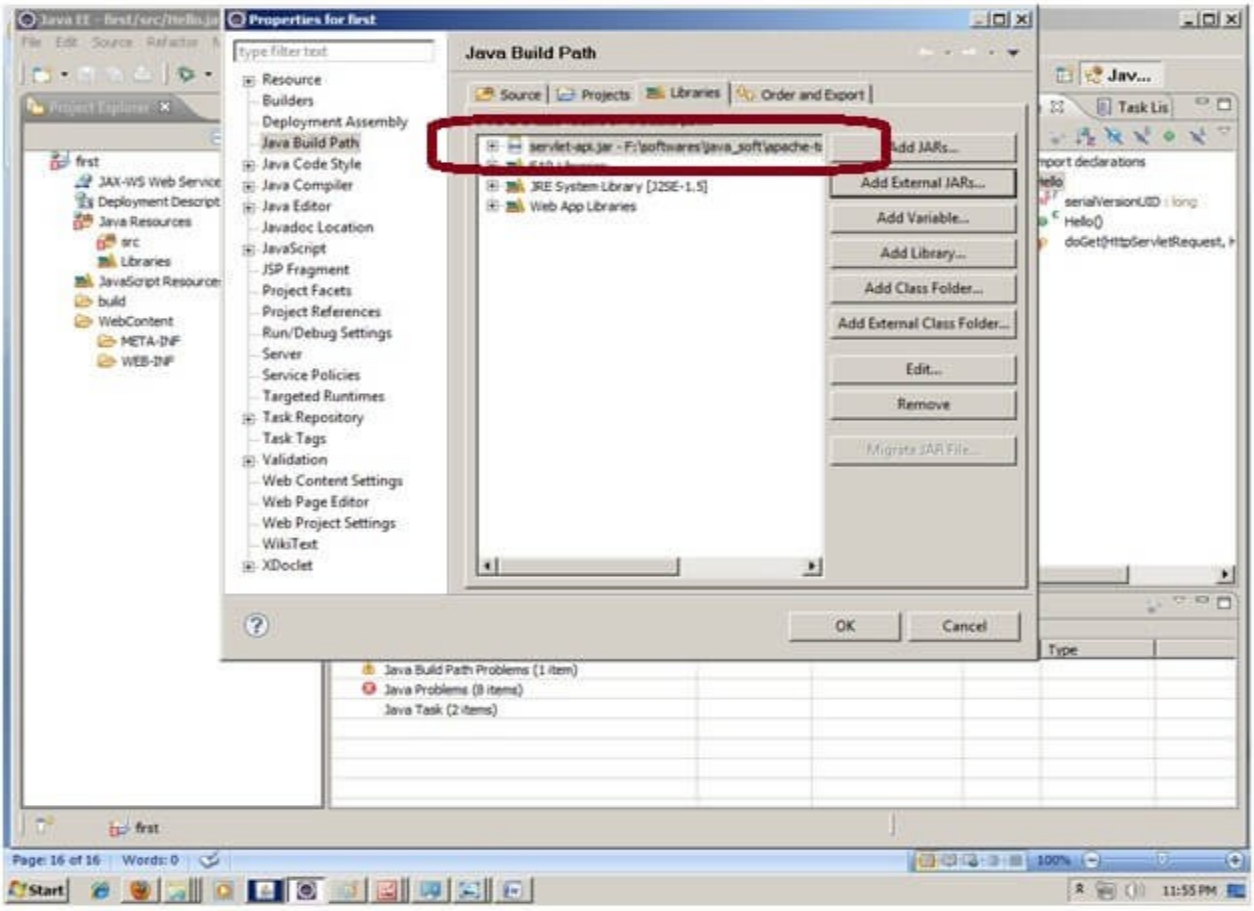

## 3) add jar file in eclipse IDE:

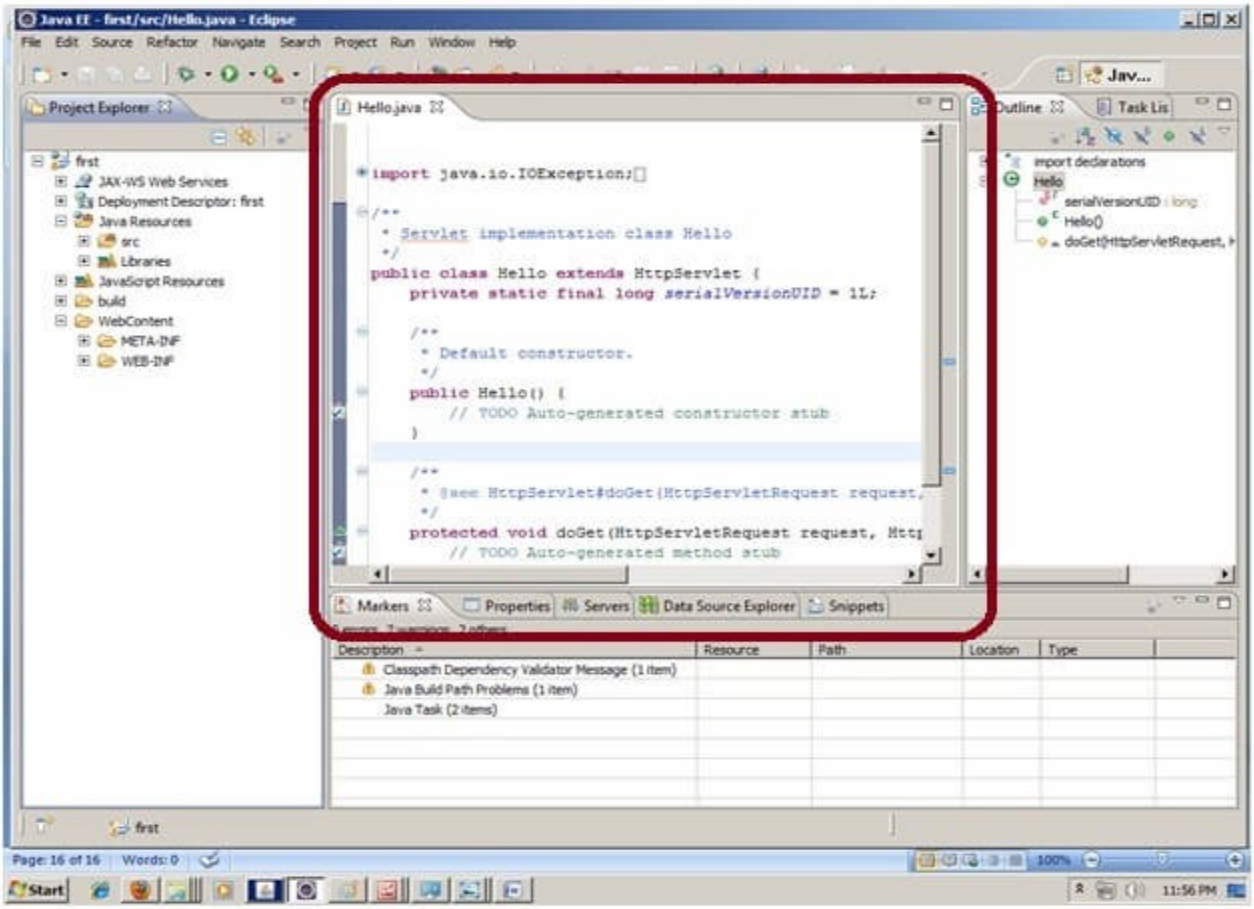
For adding a jar file, **right click on your project -> Build Path -> Configure Build Path -> click on Libraries tab in Java Build Path -> click on Add External JARs button -> select the servlet-api.jar file under tomcat/lib -> ok.**


Java EE - first/src/Hello.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
Hello.java
Outline
Task Lis
import declarations
Hello
serialVersionUID : long
Hello()
doGet(HttpServletRequest, h
New
Go Into
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Refactor Alt+Shift+T
Export
Refresh F5
Close Project
Close Unrelated Projects
Validate
Show in Remote Systems view
Run As
Debug As
Profile As
Team
Compare With
Restore from Local History...
Java EE Tools
Source
Configure
Properties Alt+Enter
Configure Build Path...
Properties
Servers
Data Source Explorer
Snippets
Resource
Path
Location
Type
Path Problems (1 item)
(2 items)
first
Page: 13 of 13
Words: 0
100%
Start
11:53 PM

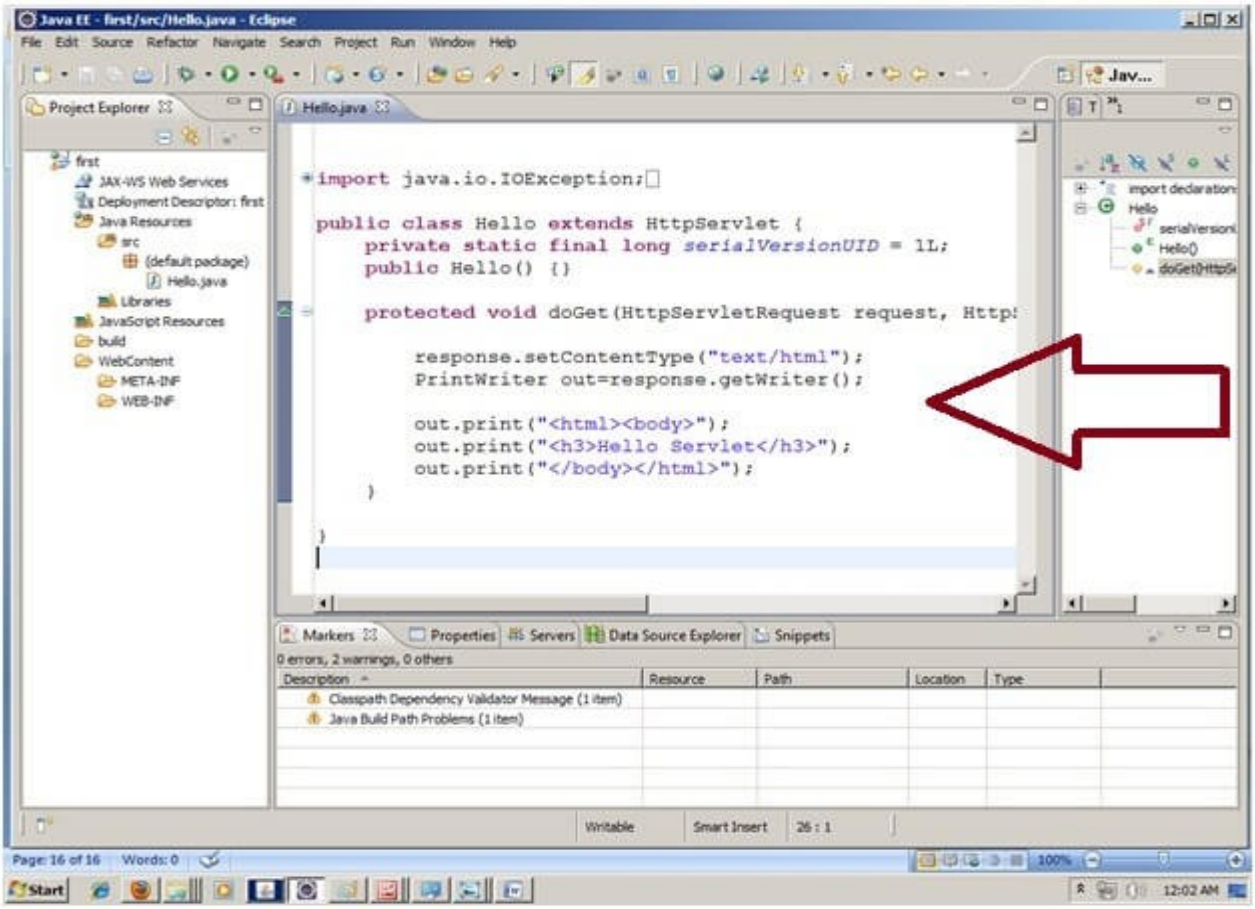

Properties for first
Java Build Path
Source
Projects
Libraries
Order and Export
JARs and class folders on the build path:
EAR Libraries
JRE System Library [J2SE-1.5]
Web App Libraries
Add JARs...
Add External JARs...
Add Variable...
Add Library...
Add Class Folder...
Add External Class Folder...
Edit...
Remove
Migrate JAR File...
Resource
Builders
Deployment Assembly
Java Build Path
Java Code Style
Java Compiler
Java Editor
Javadoc Location
JavaScript
JSP Fragment
Project Facets
Project References
Run/Debug Settings
Server
Service Policies
Targeted Runtimes
Task Repository
Task Tags
Validation
Web Content Settings
Web Page Editor
Web Project Settings
WikiText
XDoclet
OK
Cancel
Java Build Path Problems (1 item)
Java Problems (8 items)
Java Task (2 items)
first
Page: 14 of 14
Words: 0
100%
11:54 PM


Java Build Path
Source
Projects
Libraries
Order and Export
JARs and class folders on the build path:
EAR Libraries
JRE System Library [J2SE-1.5]
Web App Libraries
Add JARs...
Add External JARs...
Add Variable...
JAR Selection
apache-tomcat-7.0.23 ▸ lib
Search lib
Organize
New folder
Desktop
Downloads
Recent Places
Libraries
Documents
Music
Pictures
Videos
Computer
Local Disk (C:)
Works (D:)
Softwares (F:)
Local Disk (G:)
Name
Date modified
Type
annotations-api 11/20/2011 7:38 PM Executab
catalina 11/20/2011 7:38 PM Executab
catalina-ant 11/20/2011 7:38 PM Executab
catalina-ha 11/20/2011 7:38 PM Executab
catalina-tribes 11/20/2011 7:38 PM Executab
ecj-3.7 11/20/2011 7:38 PM Executab
el-api 11/20/2011 7:38 PM Executab
jasper 11/20/2011 7:38 PM Executab
jasper-el 11/20/2011 7:38 PM Executab
jsp-api 11/20/2011 7:38 PM Executab
servlet-api 11/20/2011 7:38 PM Executab
Date modified: 11/20/2011 7:38 PM
File name: servlet-api
Open
Cancel
first
Page: 14 of 14
Words: 0
100%
Start
11:55 PM


Properties for first
Java Build Path
Source
Projects
Libraries
Order and Export
Add JARs...
Add External JARs...
Add Variable...
Add Library...
Add Class Folder...
Add External Class Folder...
Edit...
Remove
Migrate JAR File...
JRE System Library [J2SE-1.5]
Web App Libraries
OK
Cancel
Resource
Builders
Deployment Assembly
Java Build Path
Java Code Style
Java Compiler
Java Editor
Javadoc Location
JavaScript
JSP Fragment
Project Facets
Project References
Run/Debug Settings
Server
Service Policies
Targeted Runtimes
Task Repository
Task Tags
Validation
Web Content Settings
Web Page Editor
Web Project Settings
WikiText
XDoclet
Java Build Path Problems (1 item)
Java Problems (8 items)
Java Task (2 items)
Type
Page: 16 of 16
Words: 0
Start
11:55 PM

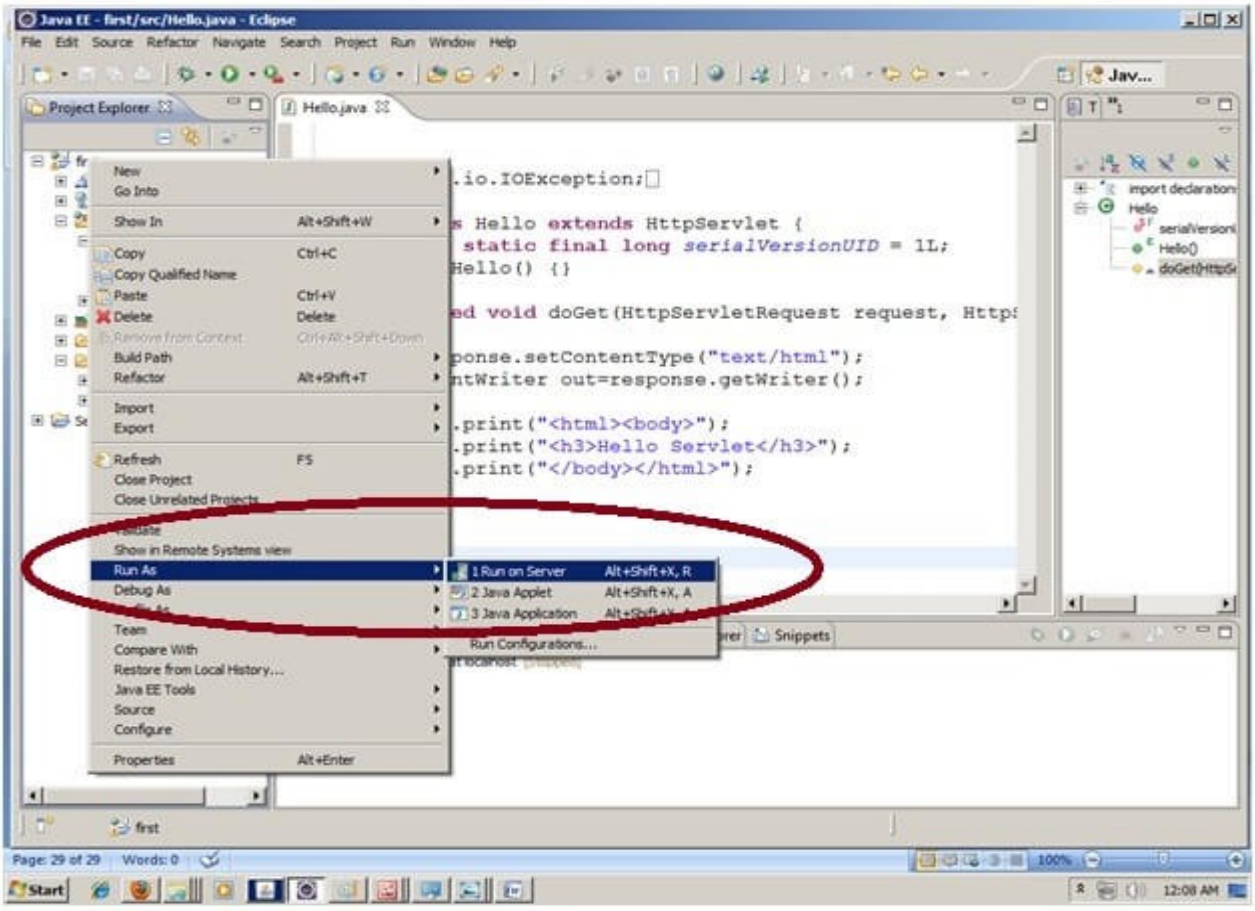


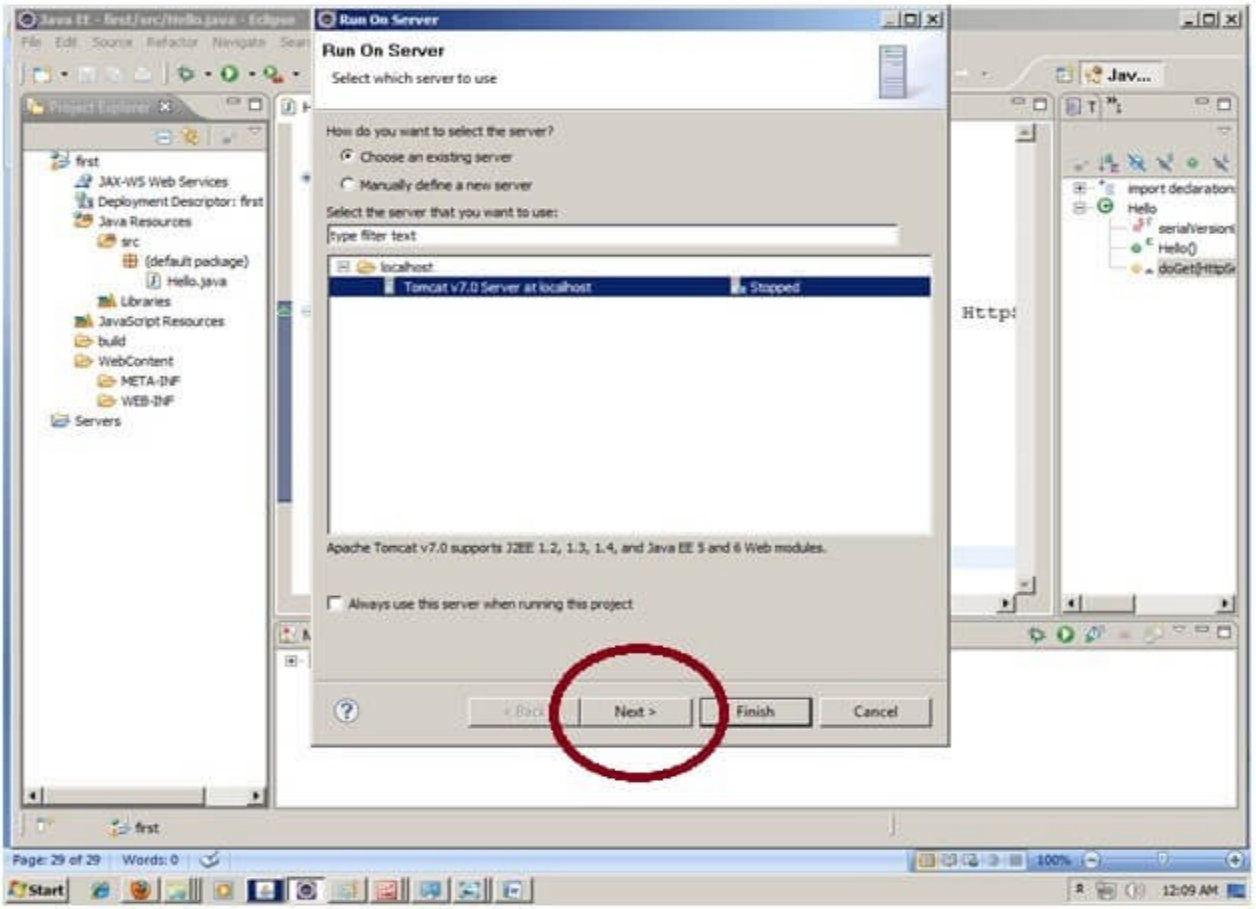

Now servlet has been created, Let's write the first servlet code.

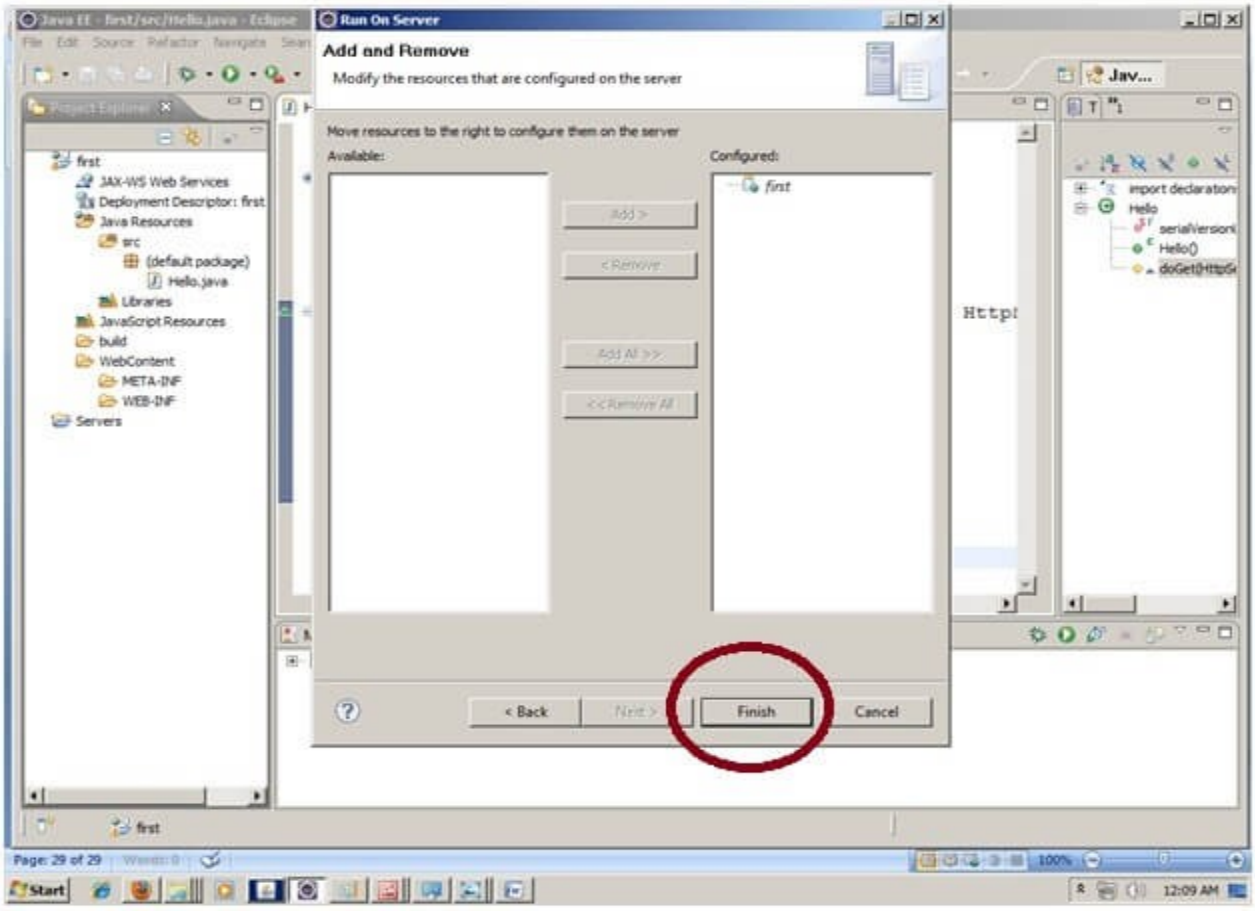


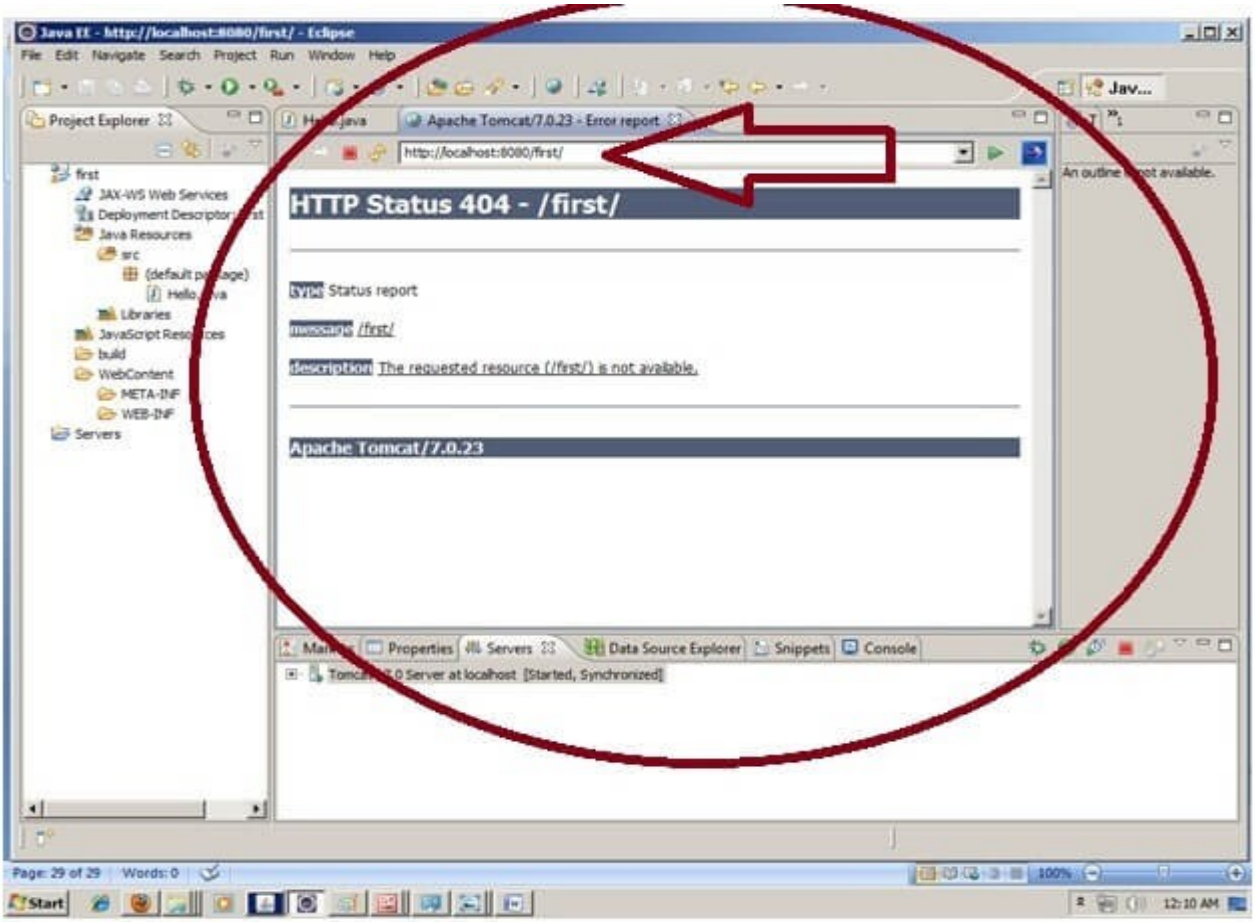

## 4) Start the server and deploy the project:

For starting the server and deploying the project in one step, **Right click on your project -> Run As -> Run on Server -> choose tomcat server -> next -> addAll -> finish.**

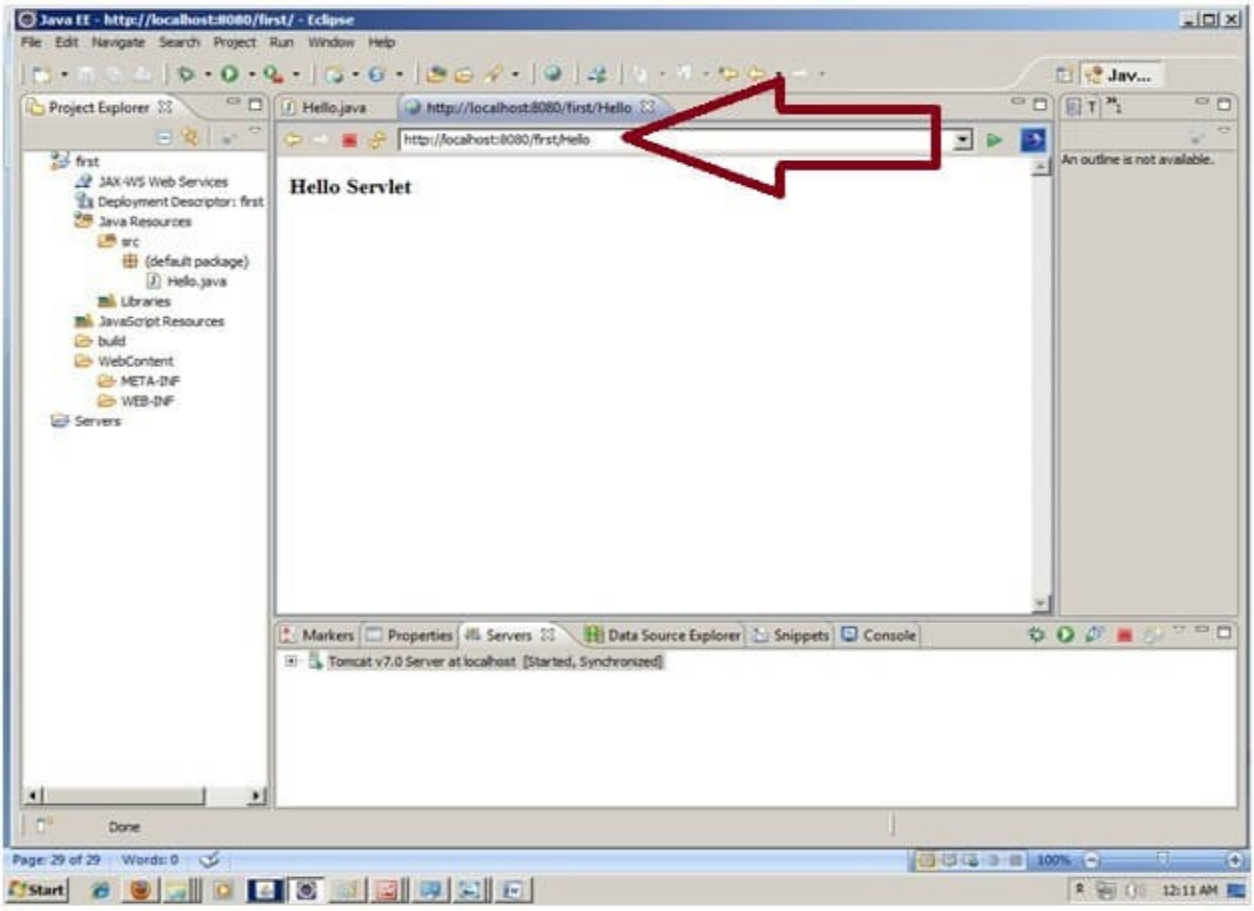

Java EE - first/src/Hello.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
Hello.java
.io.IOException;
s Hello extends HttpServlet {
static final long serialVersionUID = 1L;
Hello() {}
ed void doGet(HttpServletRequest request, Https
ponse.setContentType("text/html");
ntWriter out=response.getWriter();
.print("<html><body>");
.print("<h3>Hello Servlet</h3>");
.print("</body></html>");
New
Go Into
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Refactor Alt+Shift+T
Import
Export
Refresh F5
Close Project
Close Unrelated Projects
Validate
Show in Remote Systems view
Run As
Debug As
Team
Compare With
Restore from Local History...
Java EE Tools
Source
Configure
Properties Alt+Enter
1 Run on Server Alt+Shift+X, R
2 Java Applet Alt+Shift+X, A
3 Java Application
Run Configurations...
Snippets
import declaration
Hello
serialVersion
Hello()
doGet(HttpS
first
Page: 29 of 29 Words: 0
100%
Start
12:08 AM


Run On Server
Run On Server
Select which server to use
How do you want to select the server?
Choose an existing server
Manually define a new server
Select the server that you want to use:
type filter text
localhost
Tomcat v7.0 Server at localhost
Stopped
Apache Tomcat v7.0 supports J2EE 1.2, 1.3, 1.4, and Java EE 5 and 6 Web modules.
Always use this server when running this project
< Back
Next >
Finish
Cancel
first
JAX-WS Web Services
Deployment Descriptor: first
Java Resources
src
(default package)
Hello.java
Libraries
JavaScript Resources
build
WebContent
META-INF
WEB-INF
Servers
Page: 29 of 29
Words: 0
Start
12:09 AM


Run On Server
Add and Remove
Modify the resources that are configured on the server
Move resources to the right to configure them on the server
Available:
Configured:
first
Add >
< Remove
Add All >>
<< Remove All
< Back
Next >
Finish
Cancel
Project Explorer
first
JAX-WS Web Services
Deployment Descriptor: first
Java Resources
src
(default package)
Hello.java
Libraries
JavaScript Resources
build
WebContent
META-INF
WEB-INF
Servers
Http
Hello
Hello()
Start
12:09 AM




Now tomcat server has been started and project is deployed. To access the servlet write the url pattern name in the URL bar of the browser. In this case Hello then enter.

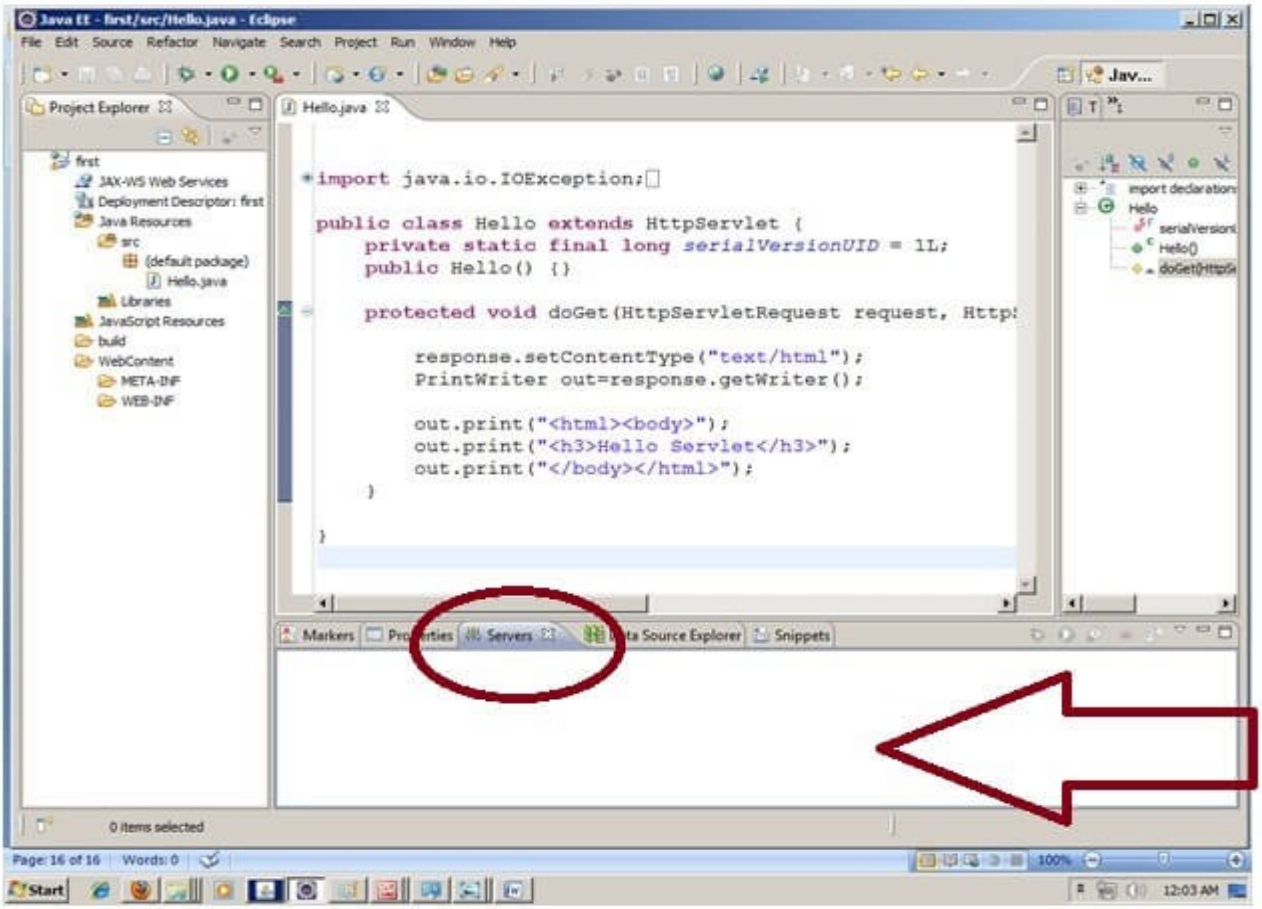


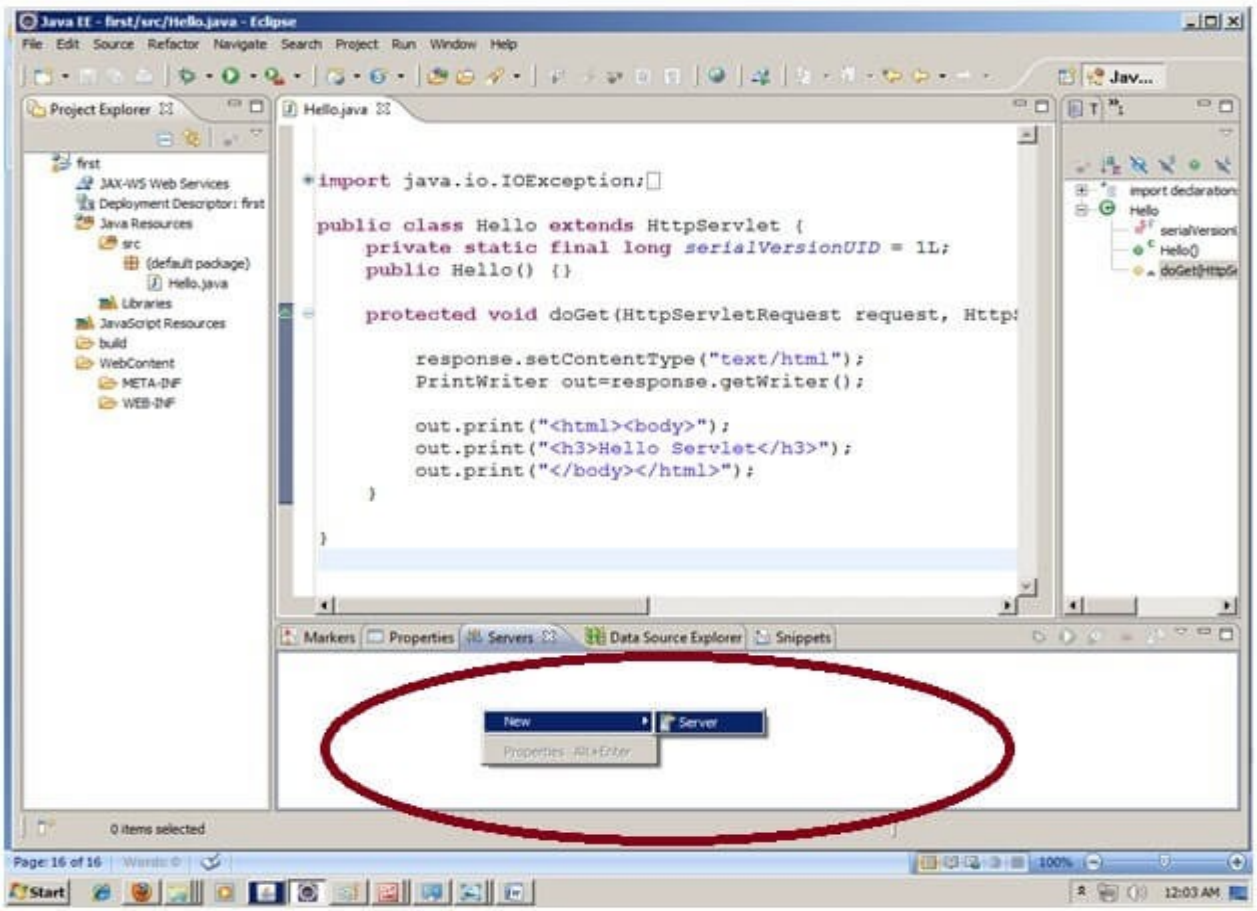

download this example

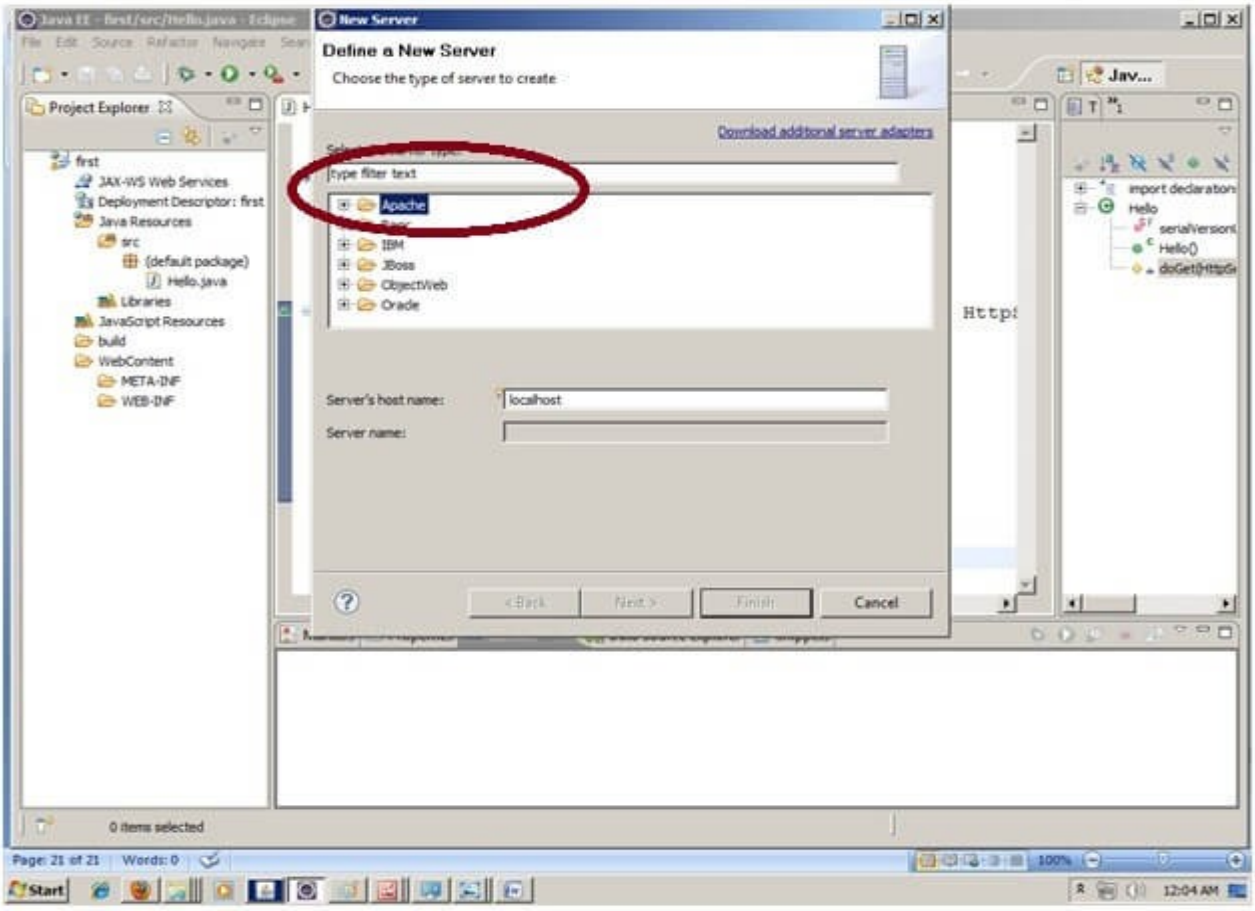
## How to configure tomcat server in Eclipse ? (One time Requirement)

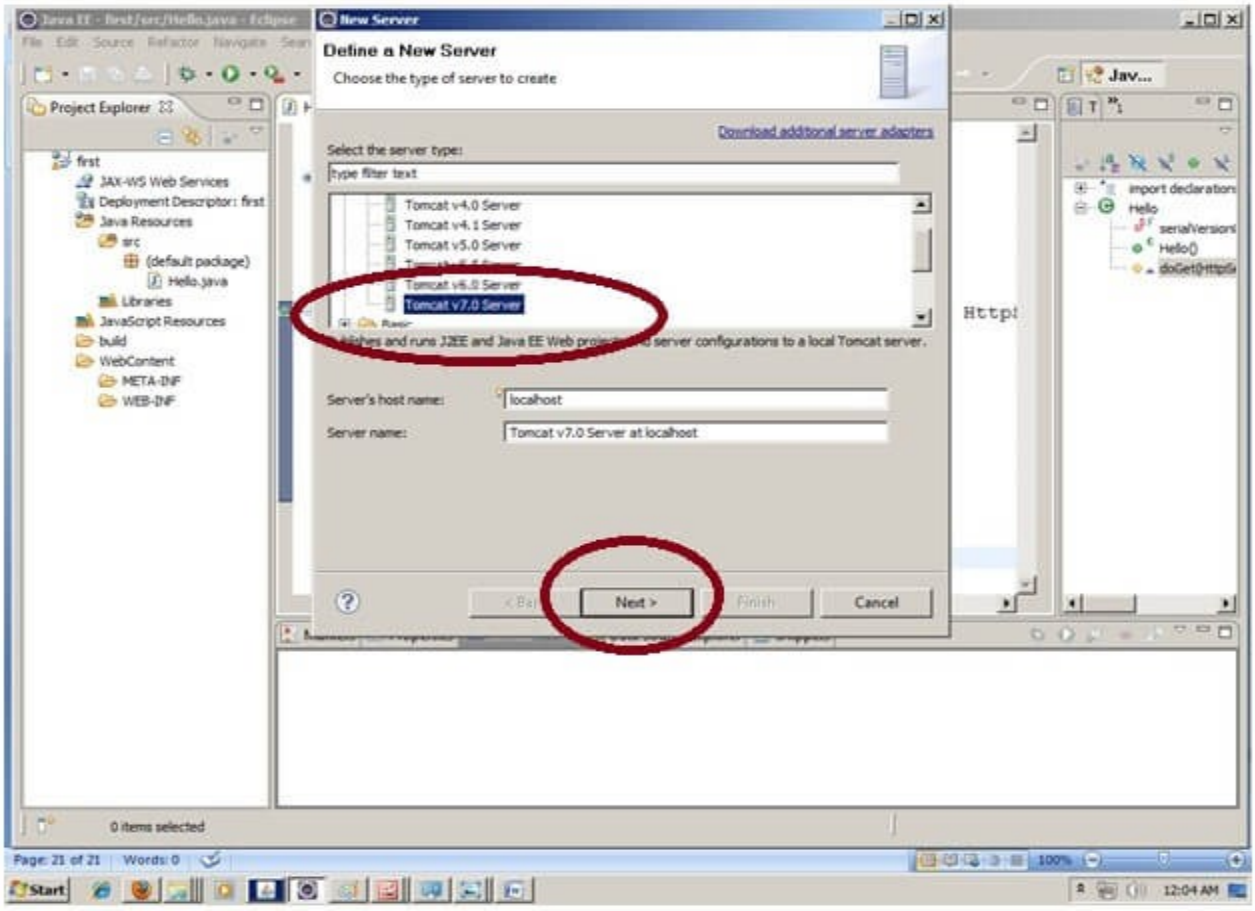
If you are using **Eclipse IDE first time**, you need to configure the tomcat server First.

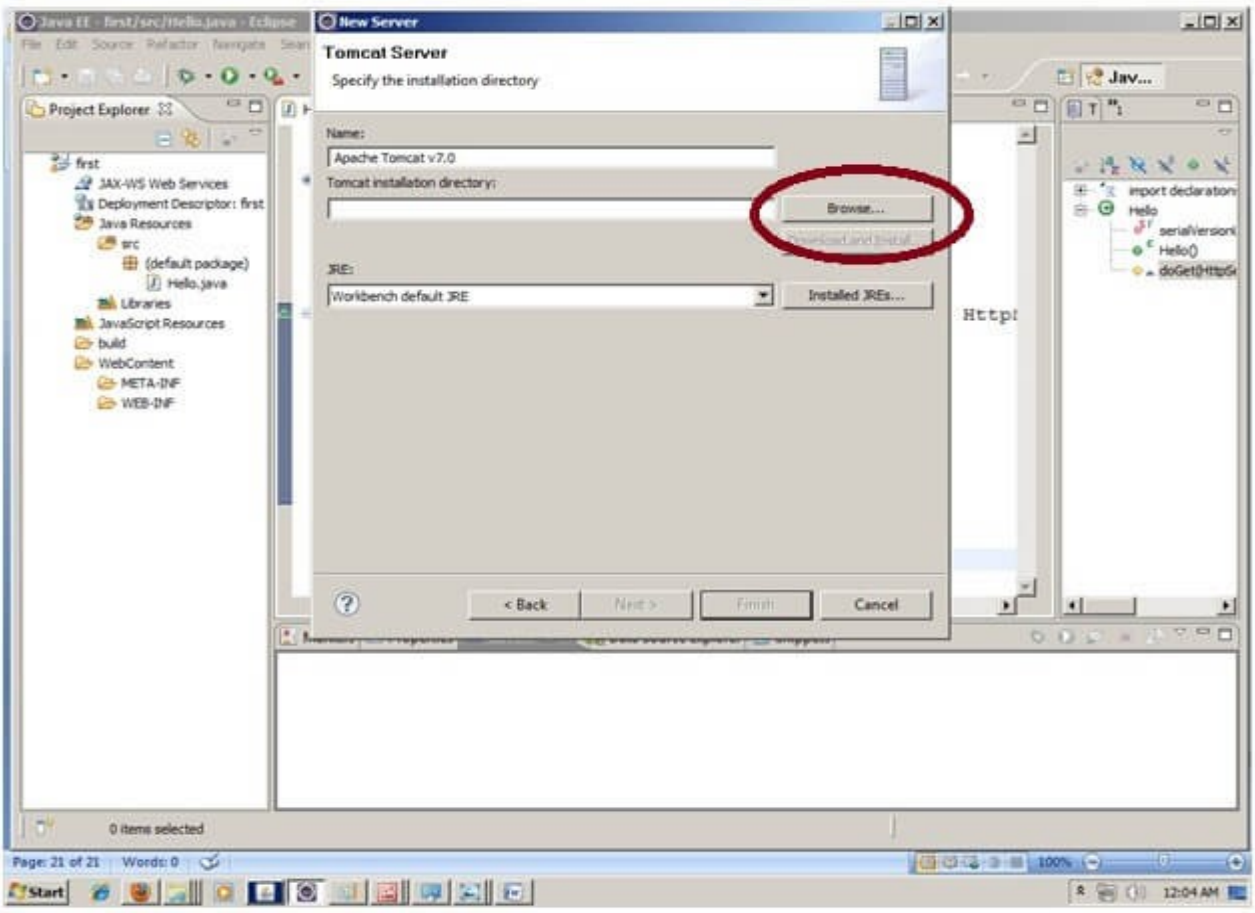
For configuring the tomcat server in eclipse IDE, **click on servers tab at the bottom side of the IDE -> right click on blank area -> New -> Servers -> choose tomcat then its version -> next -> click on Browse button -> select the apache tomcat root folder previous to bin -> next -> addAll -> Finish.**

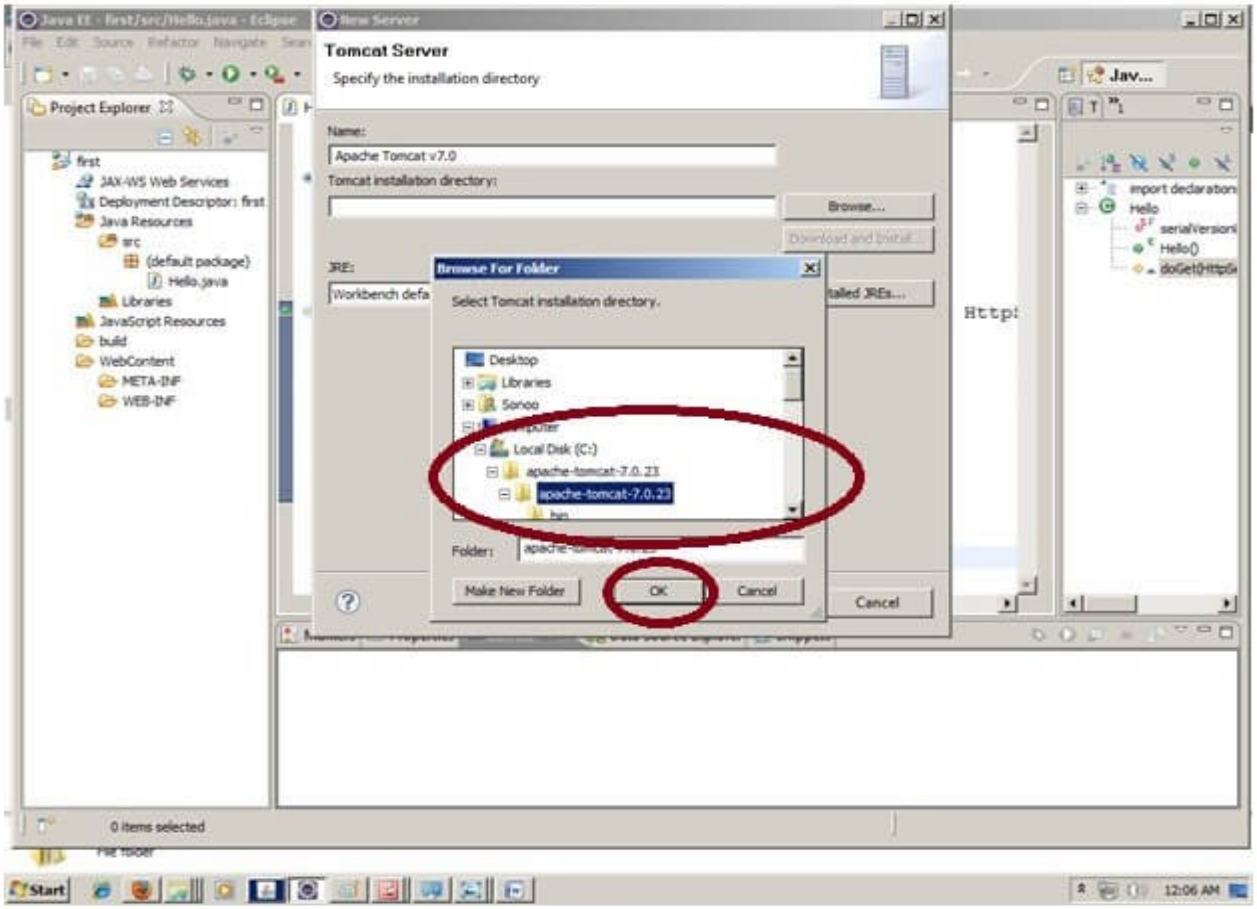

Java EE - first/src/Hello.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
first
JAX-WS Web Services
Deployment Descriptor: first
Java Resources
src
(default package)
Hello.java
Libraries
JavaScript Resources
build
WebContent
META-INF
WEB-INF
Hello.java
import java.io.IOException;
public class Hello extends HttpServlet {
private static final long serialVersionUID = 1L;
public Hello() {}
protected void doGet(HttpServletRequest request, Http
response.setContentType("text/html");
PrintWriter out=response.getWriter();
out.print("<html><body>");
out.print("<h3>Hello Servlet</h3>");
out.print("</body></html>");
}
}
Markers
Servers
Snippets
0 items selected
Page: 16 of 16
Words: 0
100%
Start
12:03 AM

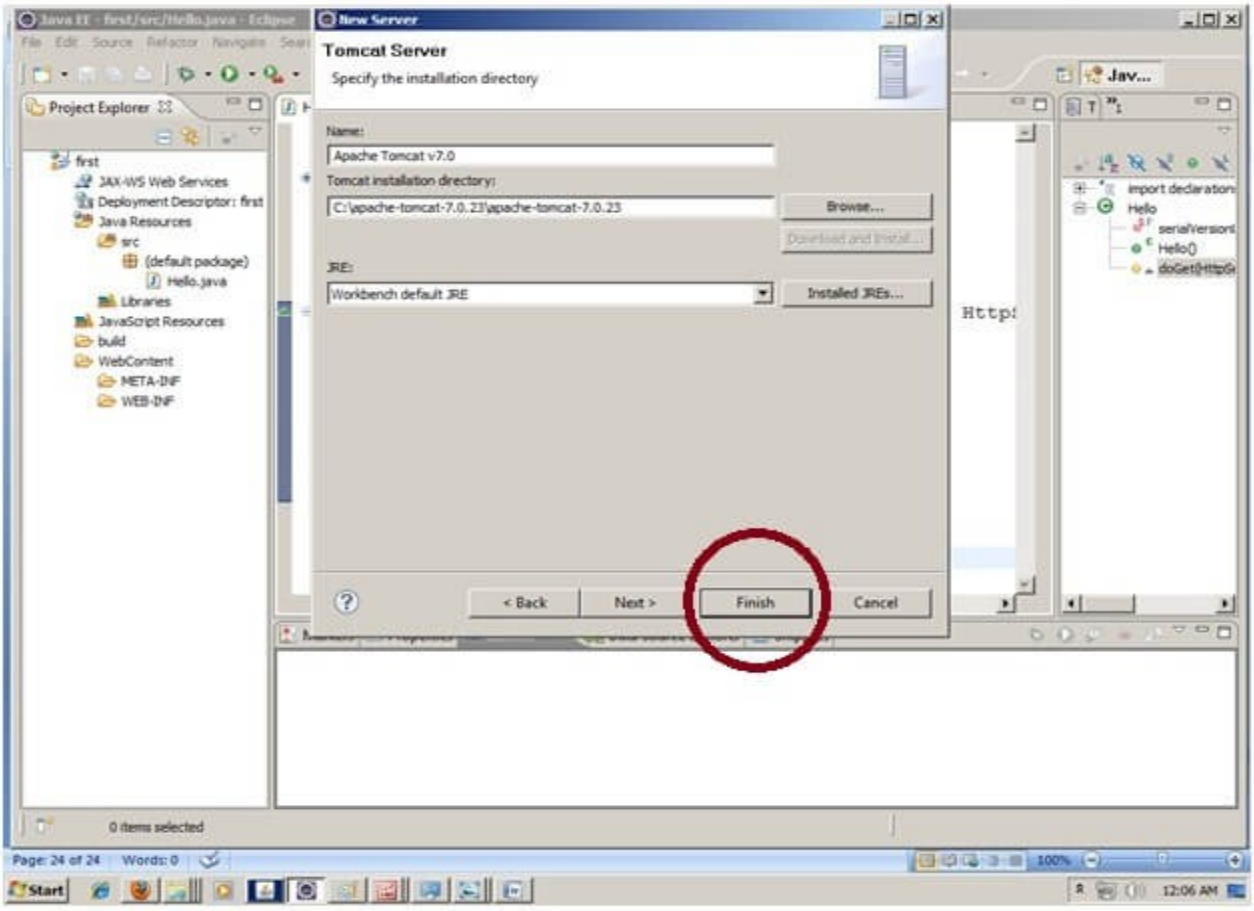

Java EE - first/src/Hello.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
first
JAX-WS Web Services
Deployment Descriptor: first
Java Resources
src
(default package)
Hello.java
Libraries
JavaScript Resources
build
WebContent
META-INF
WEB-INF
Hello.java
import java.io.IOException;
public class Hello extends HttpServlet {
private static final long serialVersionUID = 1L;
public Hello() {}
protected void doGet(HttpServletRequest request, Http
response.setContentType("text/html");
PrintWriter out=response.getWriter();
out.print("<html><body>");
out.print("<h3>Hello Servlet</h3>");
out.print("</body></html>");
}
}
Markers Properties Servers Data Source Explorer Snippets
New
Server
Properties Alt+Enter
0 items selected
Page: 16 of 16
100%
Start
12:03 AM


New Server
Define a New Server
Choose the type of server to create
Download additional server adapters
type filter text
Apache
IBM
JBoss
ObjectWeb
Oracle
Server's host name:
localhost
Server name:
< Back
Next >
Finish
Cancel
Project Explorer
first
JAX-WS Web Services
Deployment Descriptor: first
Java Resources
src
(default package)
Hello.java
Libraries
JavaScript Resources
build
WebContent
META-INF
WEB-INF
0 items selected
Page: 21 of 21
Words: 0
Start
12:04 AM

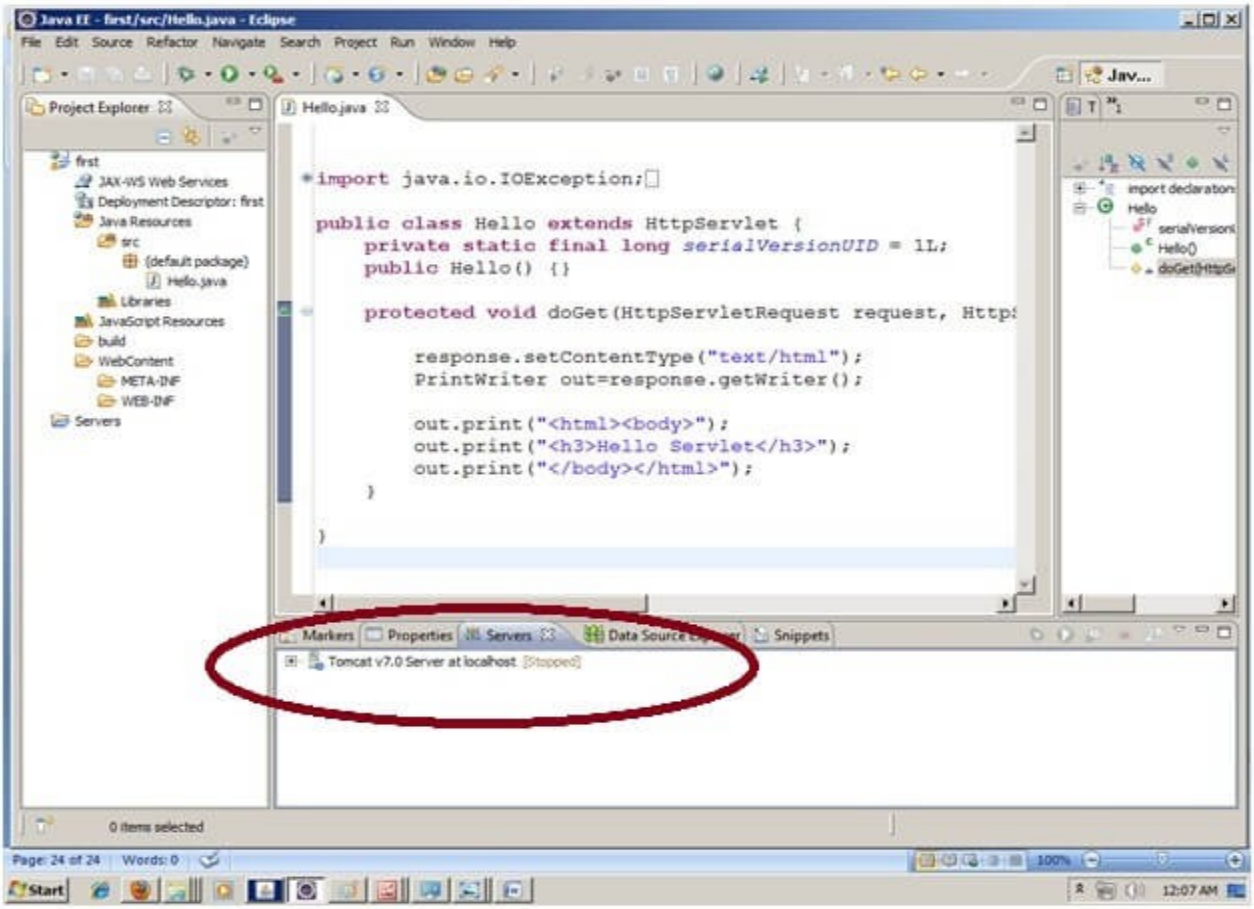

New Server
Define a New Server
Choose the type of server to create
Download additional server adapters
Select the server type:
type filter text
Tomcat v4.0 Server
Tomcat v4.1 Server
Tomcat v5.0 Server
Tomcat v6.0 Server
Tomcat v7.0 Server
Server's host name:
localhost
Server name:
Tomcat v7.0 Server at localhost
< Back
Next >
Finish
Cancel
Project Explorer
first
JAX-WS Web Services
Deployment Descriptor: first
Java Resources
src
(default package)
Hello.java
Libraries
JavaScript Resources
build
WebContent
META-INF
WEB-INF
0 items selected
Page: 21 of 21
Words: 0
Start
12:04 AM


New Server
Tomcat Server
Specify the installation directory
Name:
Apache Tomcat v7.0
Tomcat installation directory:
Browse...
Download and Install...
JRE:
Workbench default JRE
Installed JREs...
< Back
Next >
Finish
Cancel
Project Explorer
first
JAX-WS Web Services
Deployment Descriptor: first
Java Resources
src
(default package)
Hello.java
Libraries
JavaScript Resources
build
WebContent
META-INF
WEB-INF
0 items selected
Page: 21 of 21
Words: 0
100%
Start
12:04 AM


Tomcat Server
Specify the installation directory
Name:
Apache Tomcat v7.0
Tomcat installation directory:
Browse...
Download and Install...
JRE:
Workbench defa
Installed JREs...
Browse For Folder
Select Tomcat installation directory.
Desktop
Libraries
Sonoo
Local Disk (C:)
apache-tomcat-7.0.23
apache-tomcat-7.0.23
bin
Folder:
Make New Folder
OK
Cancel
Cancel
Project Explorer
first
JAX-WS Web Services
Deployment Descriptor: first
Java Resources
src
(default package)
Hello.java
Libraries
JavaScript Resources
build
WebContent
META-INF
WEB-INF
0 items selected
Start
12:06 AM




Now tomcat7 server has been configured in eclipse IDE.


Java EE - first/src/Hello.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
first
JAX-WS Web Services
Deployment Descriptor: first
Java Resources
src
(default package)
Hello.java
Libraries
JavaScript Resources
build
WebContent
META-INF
WEB-INF
Servers
Hello.java
import java.io.IOException;
public class Hello extends HttpServlet {
private static final long serialVersionUID = 1L;
public Hello() {}
protected void doGet(HttpServletRequest request, Http
response.setContentType("text/html");
PrintWriter out=response.getWriter();
out.print("<html><body>");
out.print("<h3>Hello Servlet</h3>");
out.print("</body></html>");
}
}
Markers
Properties
Servers
Snippets
Tomcat v7.0 Server at localhost [Stopped]
0 items selected
Page: 24 of 24
Words: 0
100%
Start
12:07 AM

<<prev next>>