

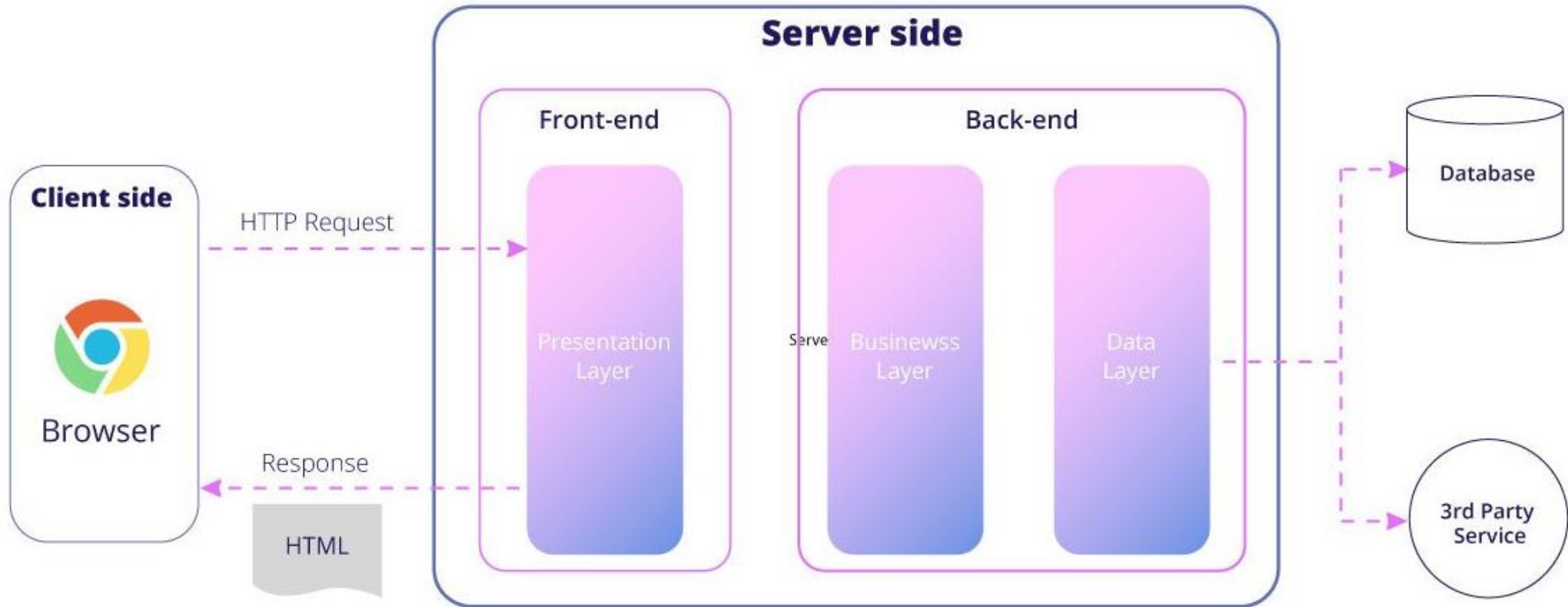
CHANDAN MUKHERJEE

**MTech (IT), BE (Computer Science)
SCJP (Java), Oracle (SQL) GLOBAL CERTIFIED
Microsoft Certified Innovative Educator
Corporate Trainer
chan.muk@gmail.com**

Web Application

- A Web application (Web app) is an application program that is stored on a remote server and delivered over the Internet through a browser interface.
- A web application can be developed for several uses, which can be used by anyone like it can be used as an individual or as a whole organization for several reasons.

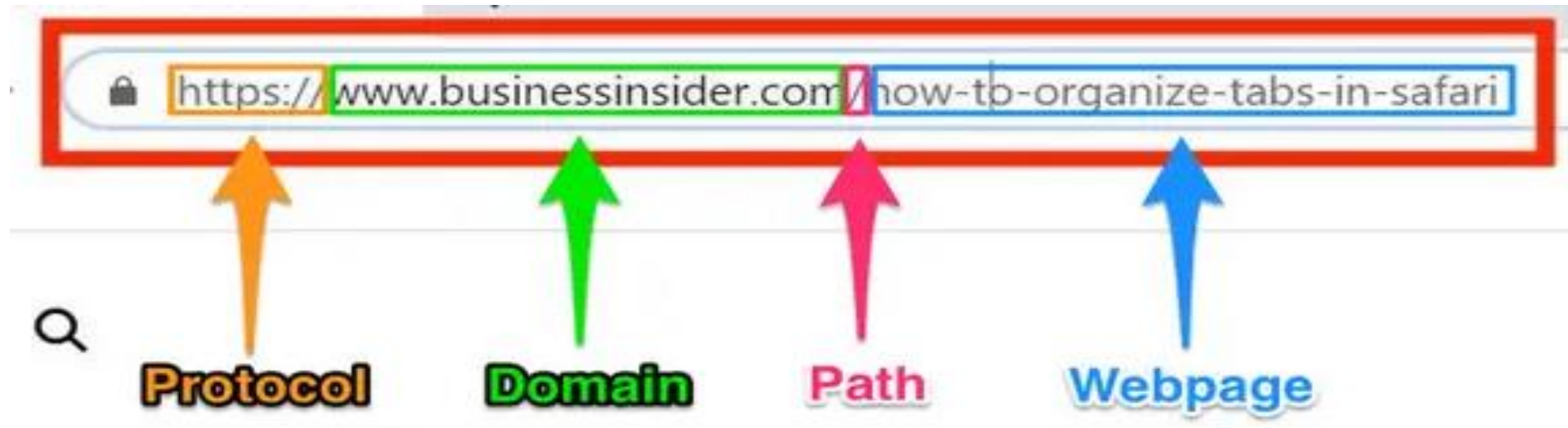
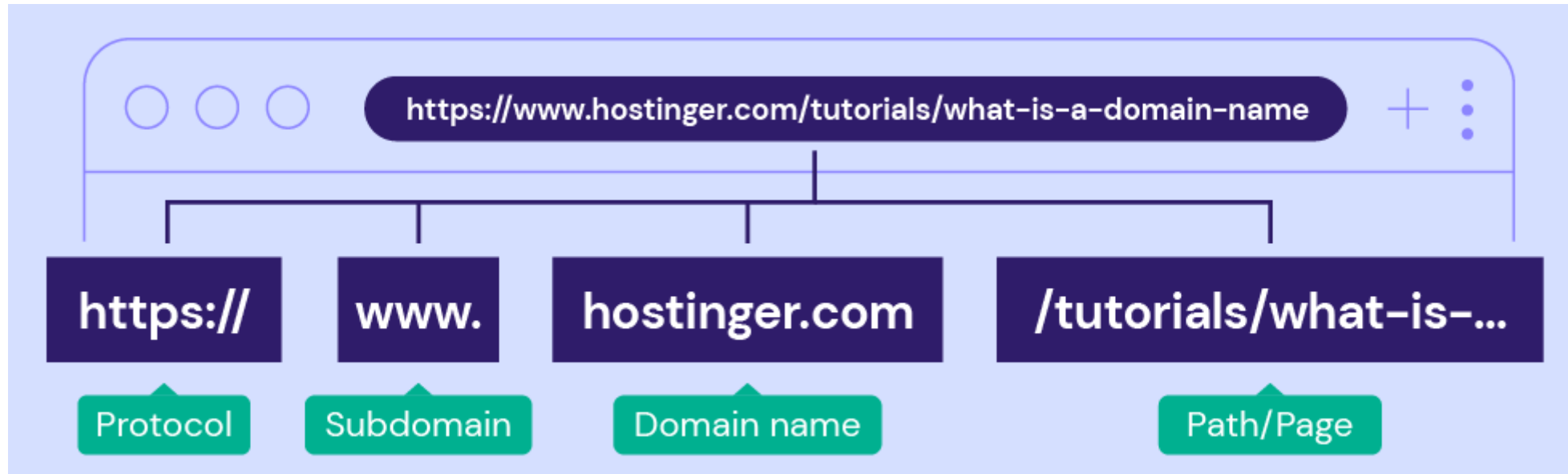
3 Tire Architecture



Web App vs. website - summary

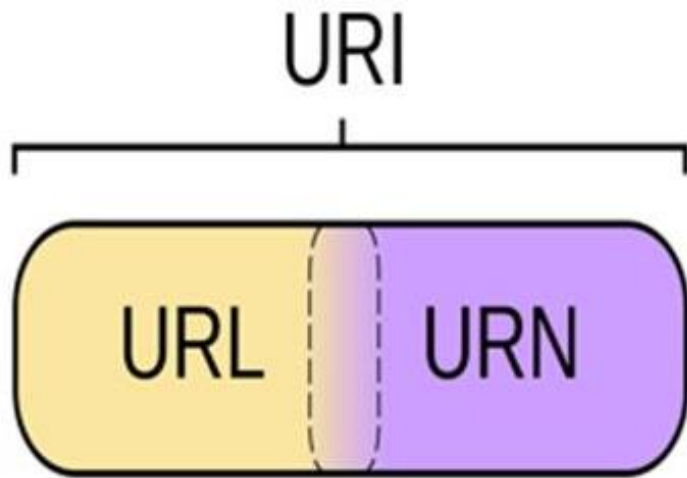
Parameter	Website	Web App
Main purpose	provide information to visitors	interaction with the users
Functionality	limited possibilities	many complex features
Authorization	not obligatory	obligatory
Average production time	can be finished in a few days	few months
Costs	usually less expensive than a web app	depends on the complexity of the project
Technologies used	HTML, CSS	Ruby on Rails Python, React and many more
Compilation and modifications	needs pre-compiling	needs pre-compiling before deployment
Risk of errors	low	relatively high
Specialists involved	UX designer, frontend developer and content creators	Frontend Developer, Backend Developer, UX Designer, Test Engineer ,Project Manager, Scrum Master, Content Creator

URL



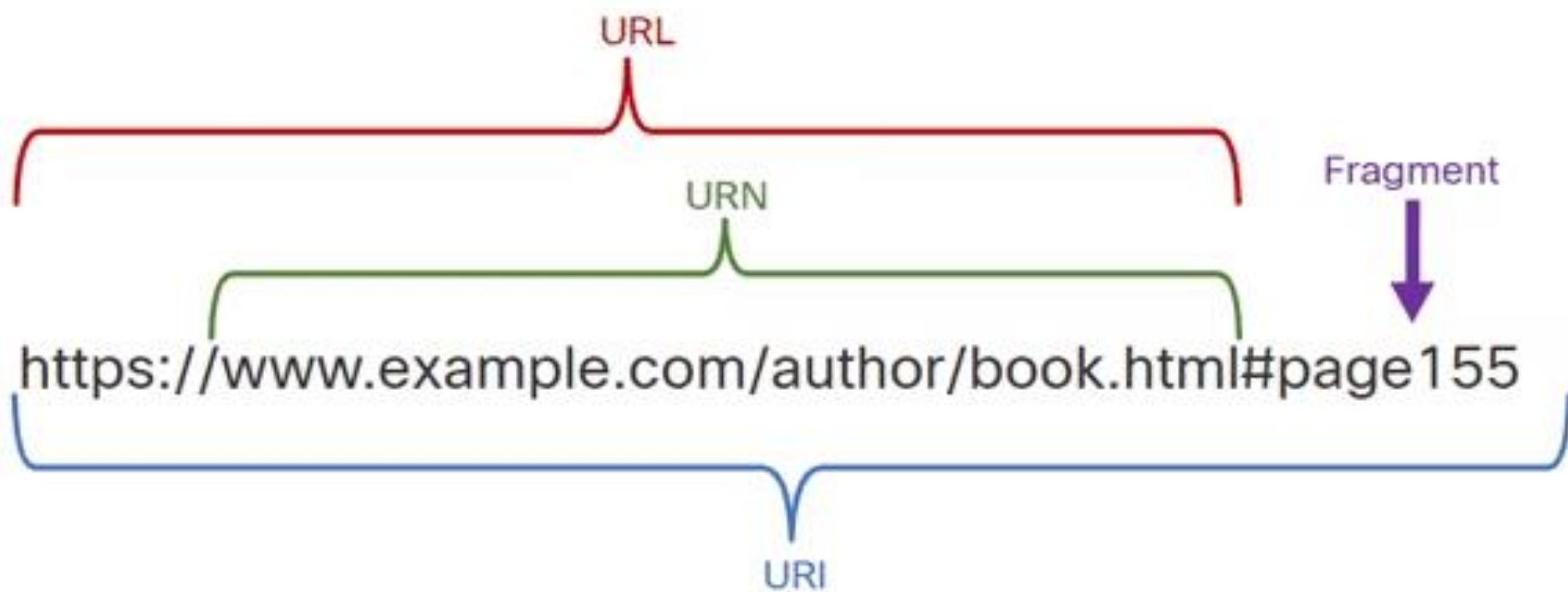
URI vs. URL vs. URN

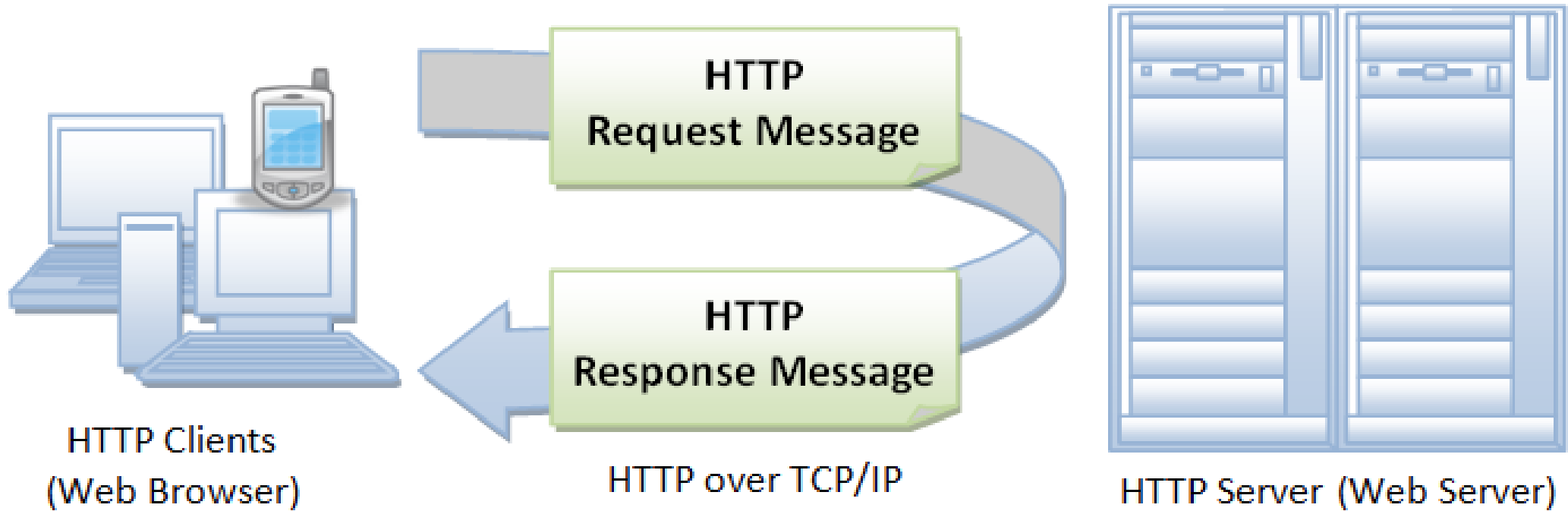
- ▶ Uniform Resource Identifier (URI)
- ▶ Uniform Resource Locator (URL)
- ▶ Uniform Resource Name (URN)



All URLs are URIs
but not vice versa

URL includes the Name of the
resource AND how to access it





HTTP Request Header

POST / HTTP/1.1

Host: localhost:8000

User-Agent: Mozilla/5.0 (Macintosh;...)... Firefox/51.0

Accept: text/html,application/xhtml+xml,...,*/*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Connection: keep-alive

Upgrade-Insecure-Requests: 1

Content-Type: multipart/form-data; boundary=-12656974

Content-Length: 345

-12656974

(more data)

Request headers

General headers

Representation
headers

HTTP Responses

The Response consists of a

- STATUS code And Description
- 1 or more optional headers
- Optional Body message can be many lines including binary data

Response Codes

- Response Status codes are split into 5 groups each group has a meaning and a three digit code.
- 1xx – Informational
- 2xx – Successful
- 3xx - Multiple Choice
- 4xx – Client Error
- 5xx - Server Error

For example

- a successful page request will return a **200 response code**
- an unsuccessful a **404 response code**.

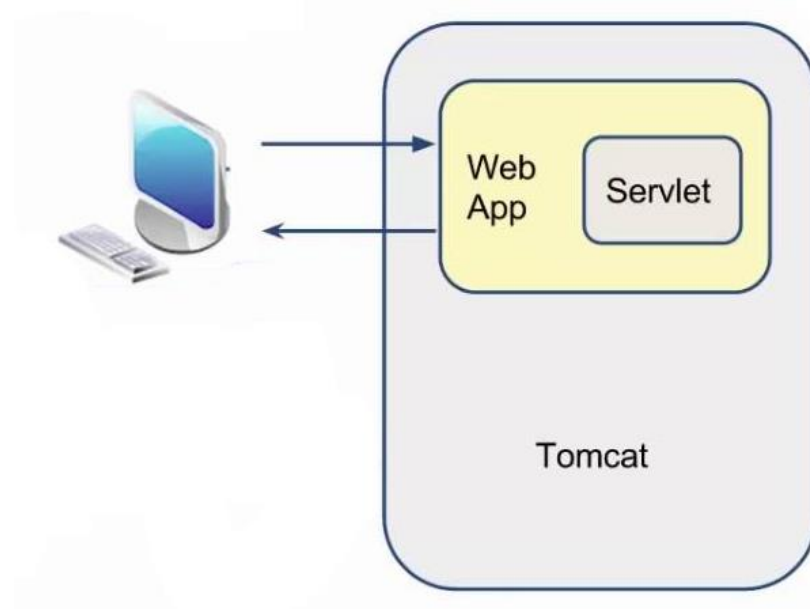


Servlet



SERVLET

- Java Servlet is the first web technology for Java.
- A Java Servlet is a Java object that responds to HTTP requests.
- A Servlet is part of a Jaa web application
- Java Servlet runs inside a Servlet container such as Tomcat, Jetty.

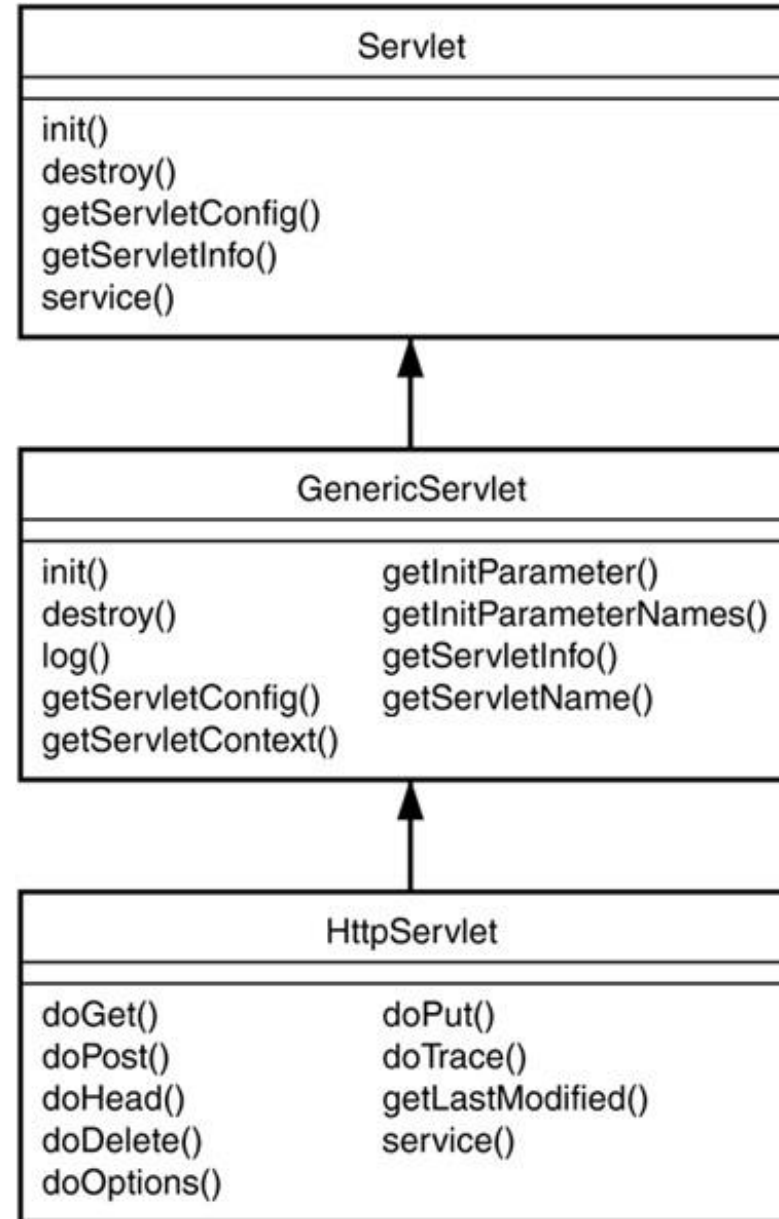




Interface

Abstract Class
service method is
not override

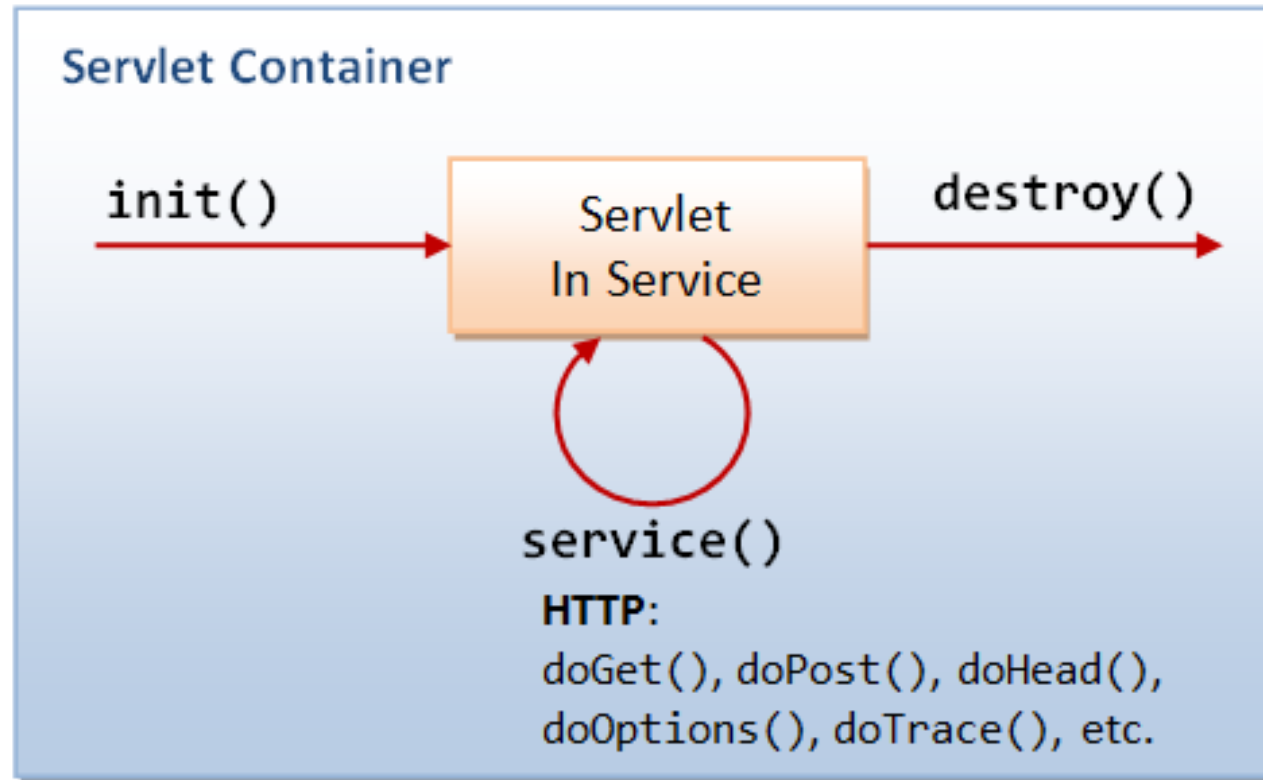
Abstract Class



Implements Servlet
Interface

Child of GenericServlet
Class

Servlet Life Cycle



Servlet Hierarchy

Servlet

service(ServletRequest,
ServletResponse)

Generic Servlet

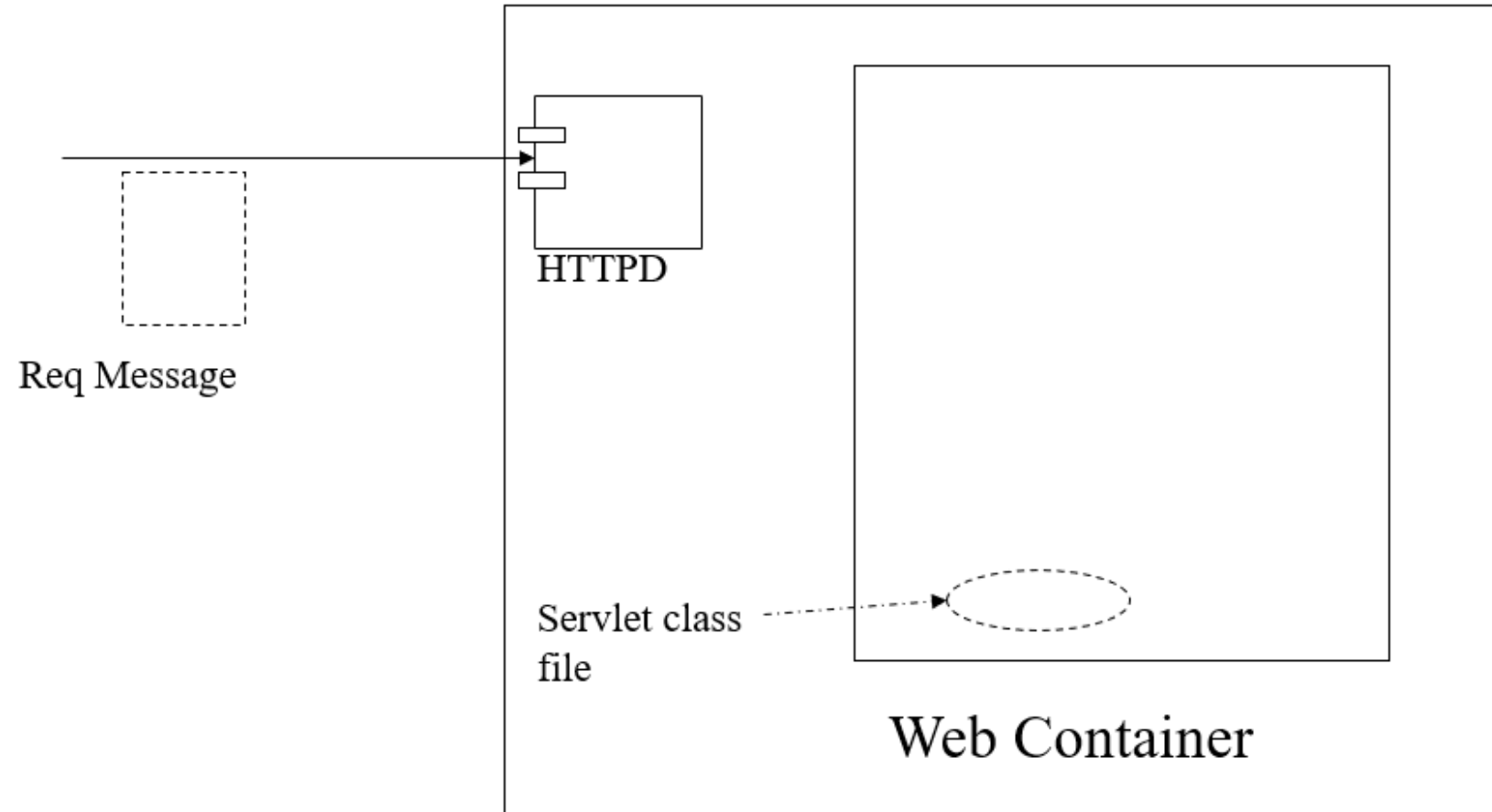
HttpServlet

doGet(HttpServletRequest ,
HttpServletResponse)
doPost(HttpServletRequest
HttpServletResponse)
doPut
doTrace
...

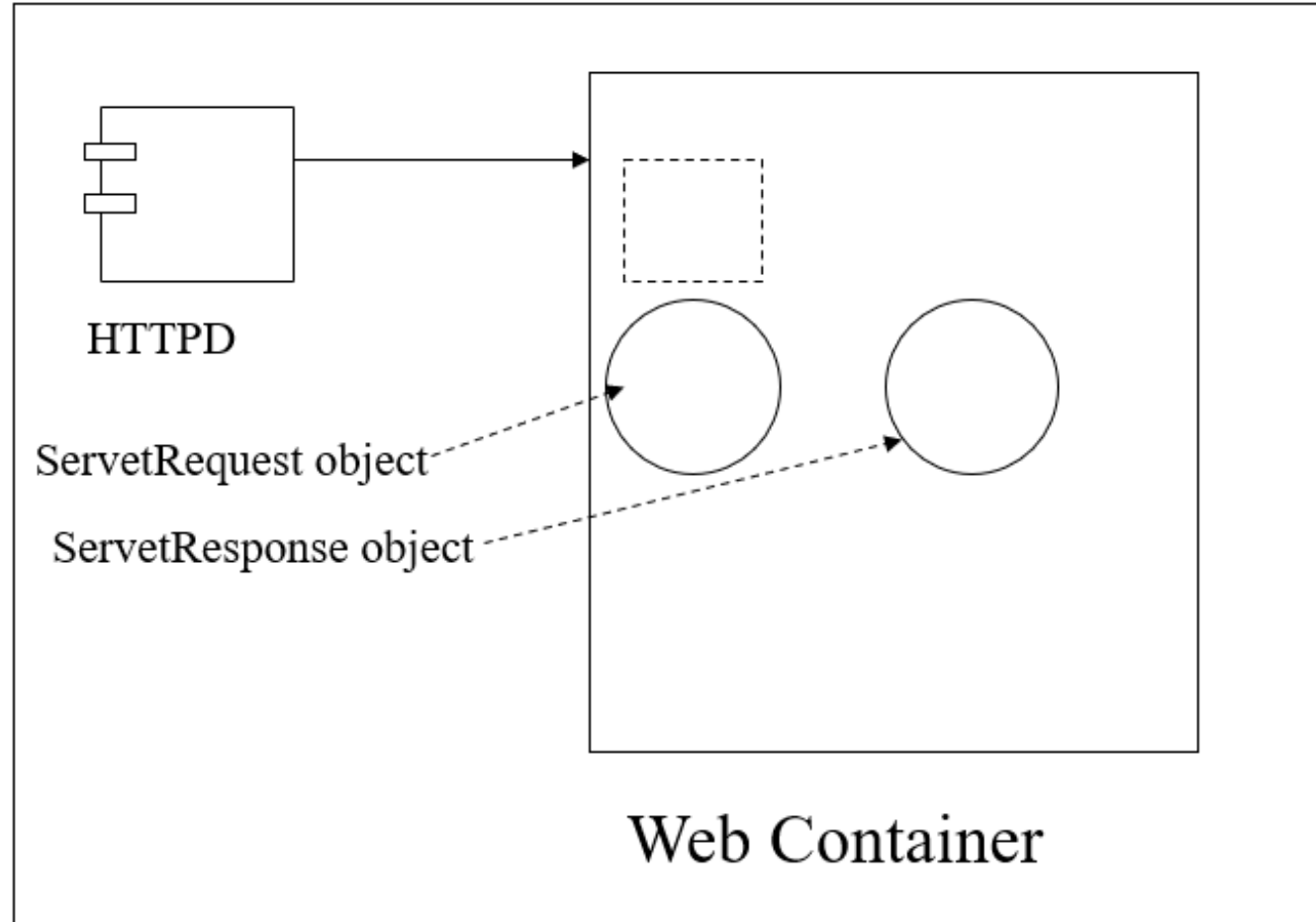
YourOwnServlet



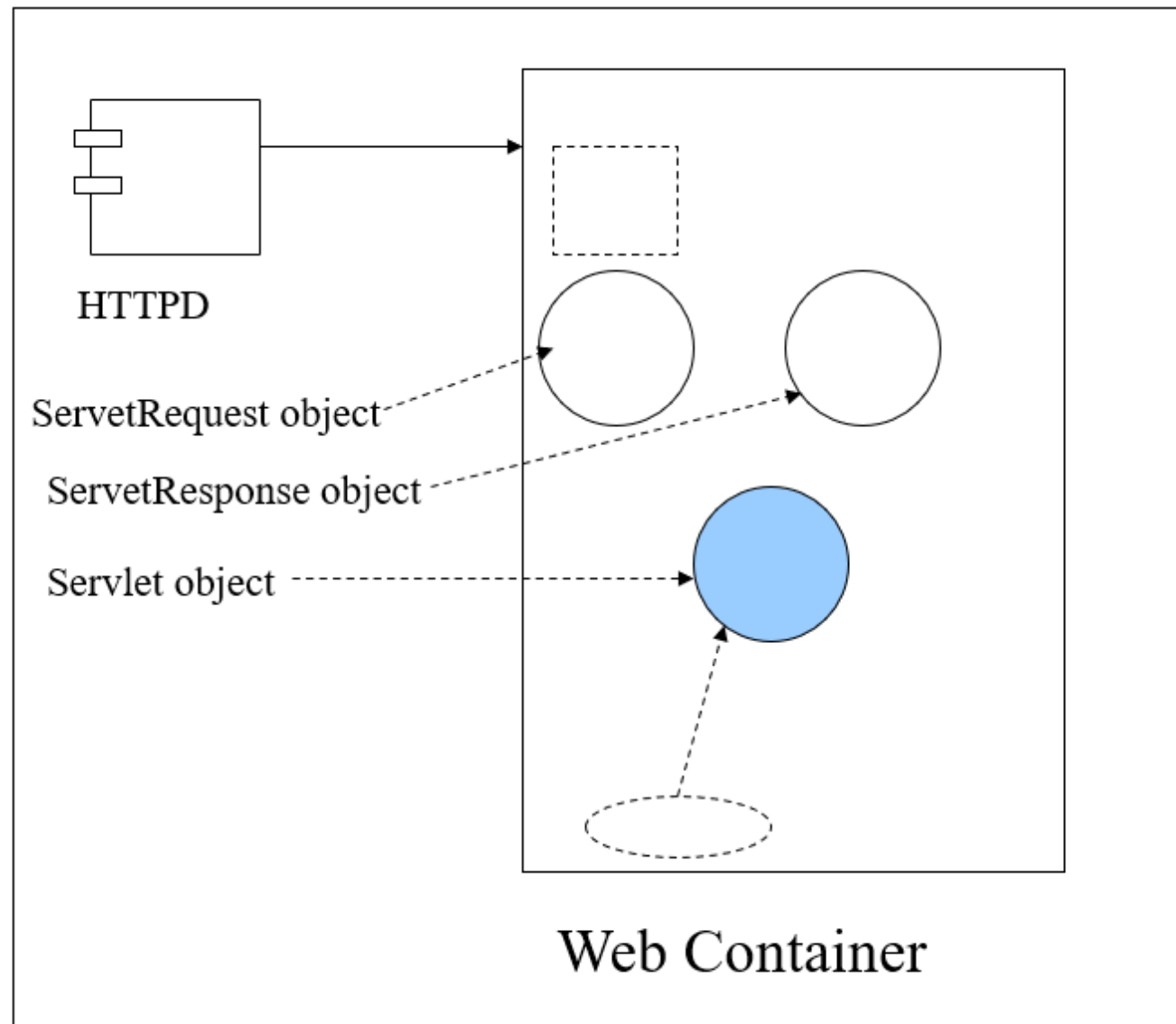
Handling Request



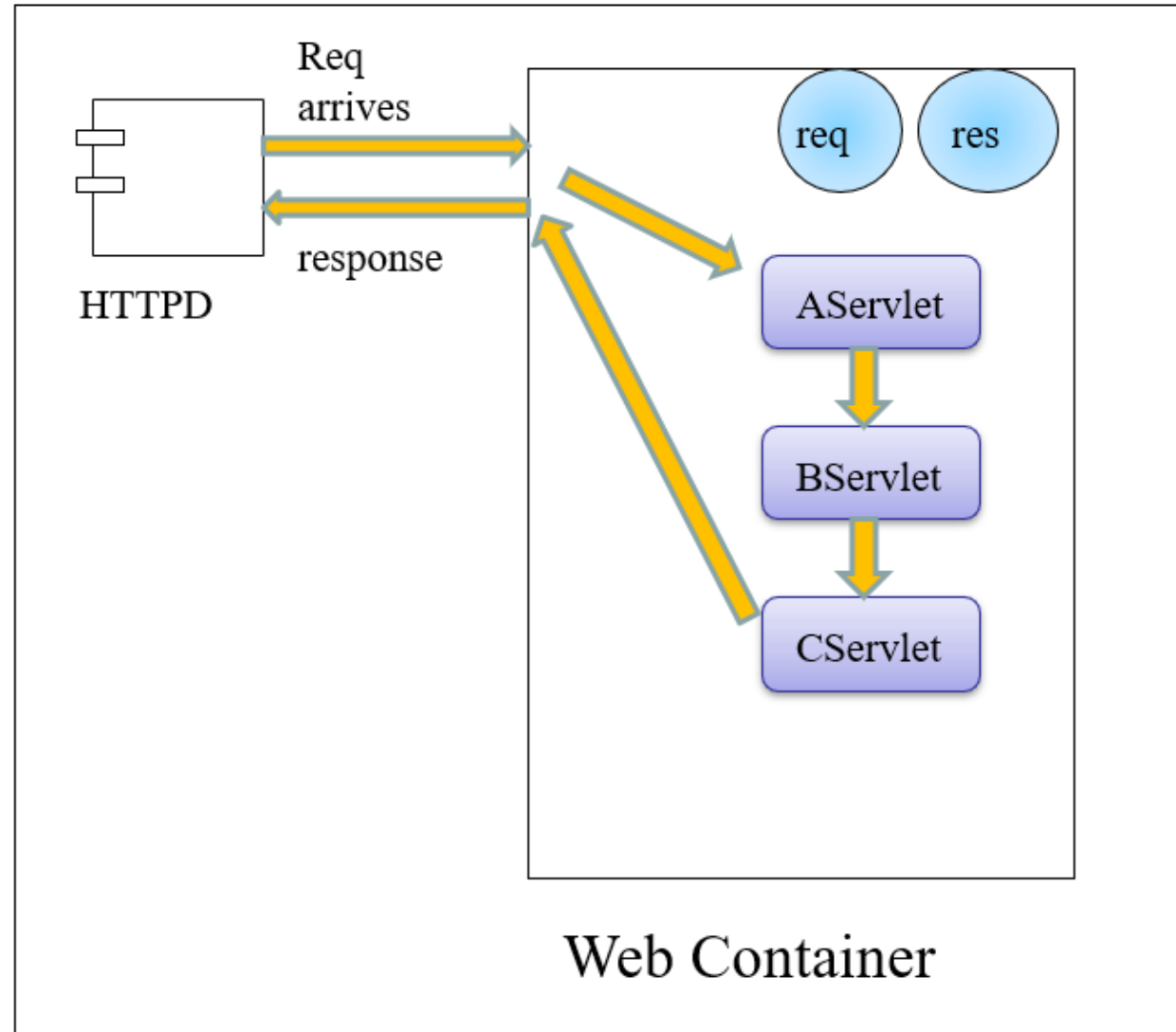
Handling Request



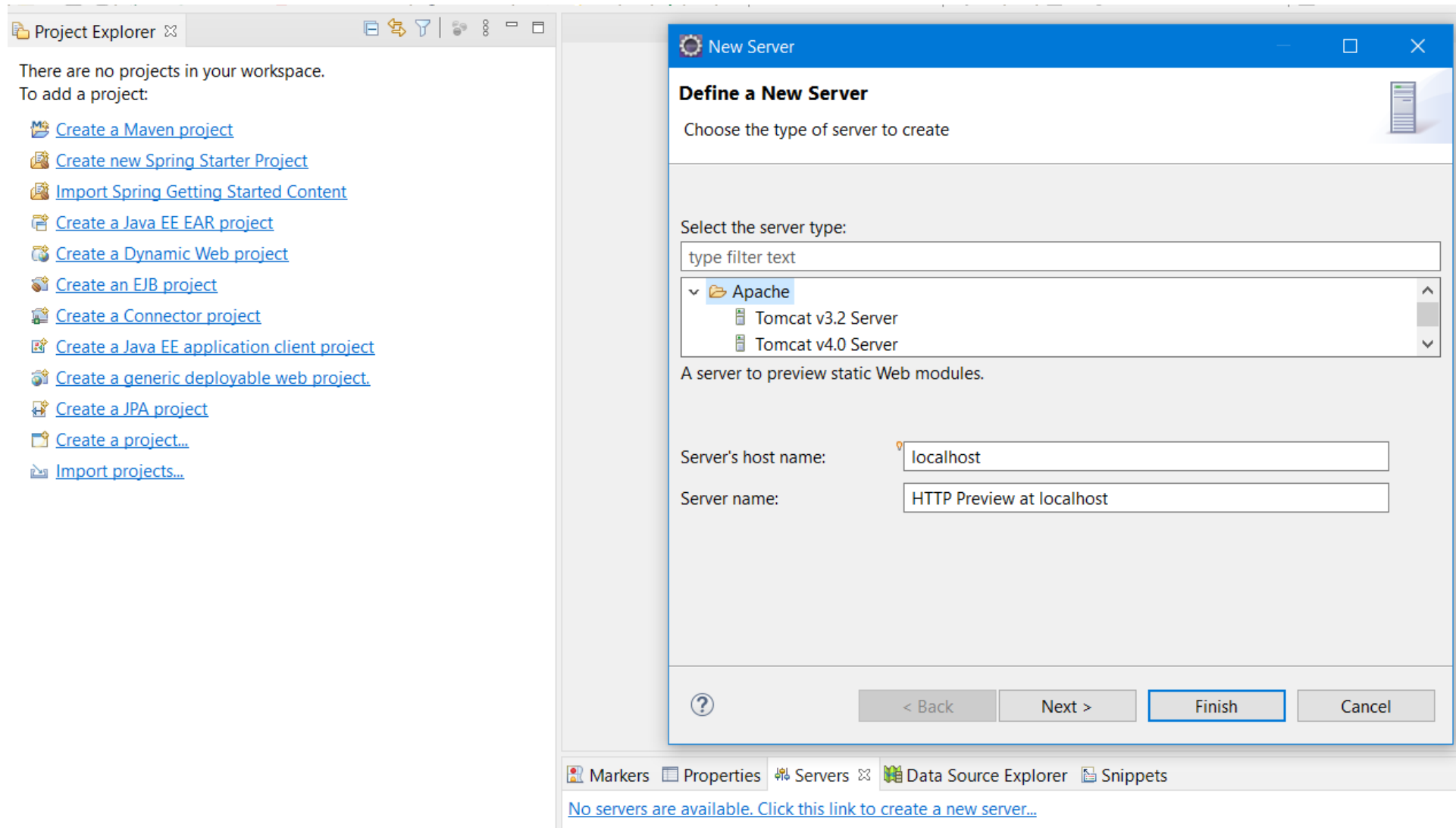
Handling Request



Servlet Chaining



CREATE JEE ENVIRONMENT AND WRITE
SERVLET CODE



SELECT TOMCAT VERSION

VodaFone_JEE - Eclipse IDE

File Edit Navigate Search Project Run Window Help



Project Explorer

There are no projects in your workspace.

To add a project:

- [Create a Maven project](#)
- [Create new Spring Starter Project](#)
- [Import Spring Getting Started Content](#)
- [Create a Java EE EAR project](#)
- [Create a Dynamic Web project](#)
- [Create an EJB project](#)
- [Create a Connector project](#)
- [Create a Java EE application client project](#)
- [Create a generic deployable web project.](#)
- [Create a JPA project](#)
- [Create a project...](#)
- [Import projects...](#)

New Server

Tomcat Server

Specify the installation directory

Name:

Apache Tomcat v9.0

Tomcat installation directory:

Browse...

apache-tomcat-9.0.43

Download and Install...

JRE:

jre

Installed JREs...

Select Folder

← → ▾ ↑ 📁 > This PC > SSD_D (D:) >

🔍 Search SSD_D (D:)

Organize ▾

New folder

🖥 This PC

📁 3D Objects

Name

📁 apache-tomcat-9.0.37

Date modified

30-06-2020 21:14

Type

File fold



New 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:

Add >

< Remove

Add All >>

<< Remove All

Configured:



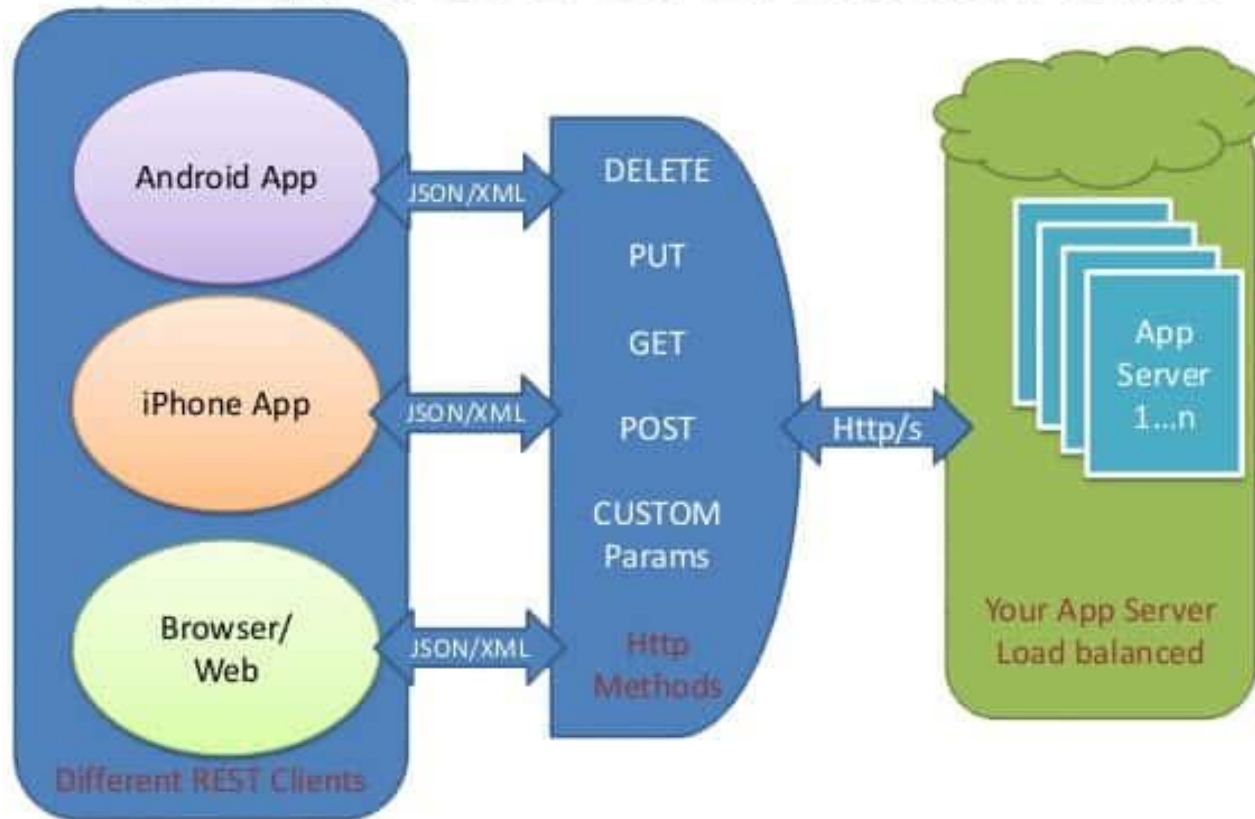
< Back

Next >

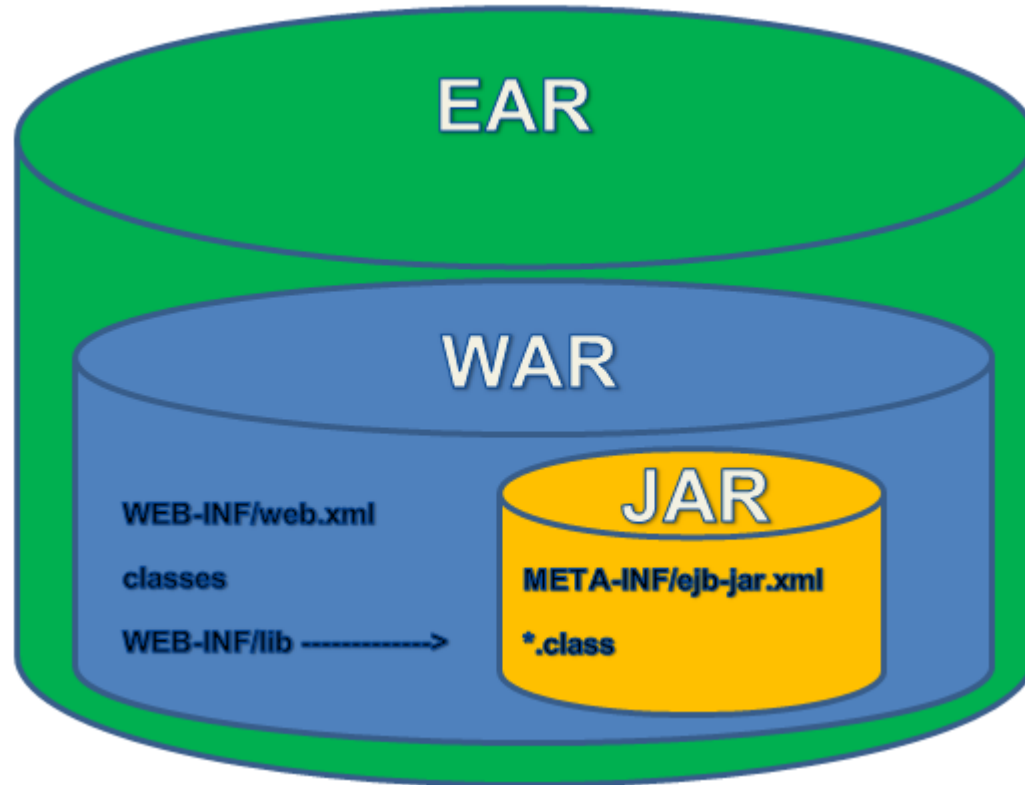
Finish

Cancel

REST API Architecture



	GET	POST
History	Parameters remain in browser history because they are part of the URL	Parameters remain in browser history because they are part of the URL
Bookmarked	Can be bookmarked.	Cannot be bookmarked.
BACK button re-submit behavior	GET requests are re-executed but may not be re-submitted to server if the HTML is stored in the browser cache.	The browser usually alerts the user that data will need to be re-submitted.
Parameters	Can send but the parameter data is limited to what we can stuff into the request line (URL)	Can send parameters, including uploading files, to the server.
Hacked	Easier to hack for script kiddies	More difficult to hack
Restrictions on form data type	Yes, only ASCII characters allowed.	No restrictions. Binary data is also allowed.
Security	GET is less secure compared to POST because data sent is part of the URL. So it's saved in browser history and server logs in plaintext.	POST is a little safer than GET because the parameters are not stored in browser history or in webserver logs.
Restrictions on form data length	Yes, since form data is in the URL and URL length is restricted. A safe URL length limit is often 2048 characters but varies by browser and web server.	No restrictions
Usability	GET method should not be used when sending passwords or other sensitive information.	POST method used when sending passwords or other sensitive info
Visibility	GET method is visible to everyone in address bar & has limits on amount of information to send.	POST method variables are not displayed in the URL.
Cached	Can be cached	Not cached



Enterprise Archive (EAR)

Web application Archive (WAR)

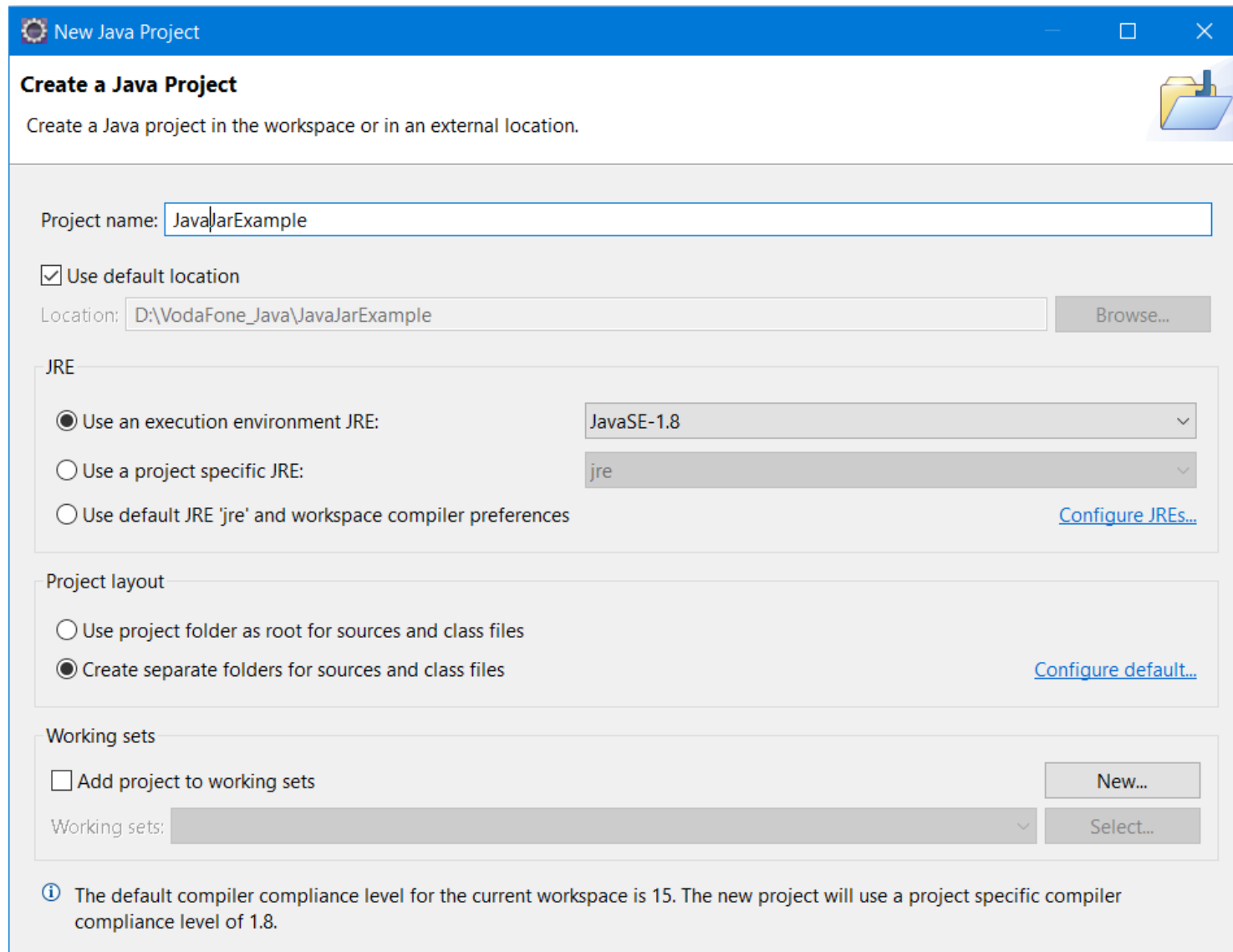
Java Archive (JAR)



CHECK JAVA /JRE VERSION

```
C:\Users\User>java -version
java version "1.8.0_341"
Java(TM) SE Runtime Environment (build 1.8.0_341-b10)
Java HotSpot(TM) 64-Bit Server VM (build 25.341-b10, mixed mode)
```

CREATE PROJECT AND SET JRE VERSION



The screenshot shows the 'New Java Project' dialog box in the Eclipse IDE. The title bar is blue with the text 'New Java Project' and standard window controls. The main area has a light gray background. At the top, it says 'Create a Java Project' followed by the instruction 'Create a Java project in the workspace or in an external location.' and a folder icon. Below this, there are several sections: 'Project name:' with a text field containing 'JavaJarExample'; 'Use default location' checked, with a 'Location:' text field containing 'D:\VodaFone_Java\JavaJarExample' and a 'Browse...' button; 'JRE' section with three radio buttons: 'Use an execution environment JRE:' (selected) with a dropdown showing 'JavaSE-1.8', 'Use a project specific JRE:' with a dropdown showing 'jre', and 'Use default JRE 'jre' and workspace compiler preferences' with a 'Configure JREs...' link; 'Project layout' section with two radio buttons: 'Use project folder as root for sources and class files' and 'Create separate folders for sources and class files' (selected) with a 'Configure default...' link; 'Working sets' section with an 'Add project to working sets' checkbox, a 'Working sets:' dropdown, a 'New...' button, and a 'Select...' button. At the bottom, there is an information icon and a message: 'The default compiler compliance level for the current workspace is 15. The new project will use a project specific compiler compliance level of 1.8.'

New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: JavaJarExample

☒ Use default location

Location: D:\VodaFone_Java\JavaJarExample [Browse...](#)

JRE

☒ Use an execution environment JRE: JavaSE-1.8

☐ Use a project specific JRE: jre

☐ Use default JRE 'jre' and workspace compiler preferences [Configure JREs...](#)

Project layout

☐ Use project folder as root for sources and class files

☒ Create separate folders for sources and class files [Configure default...](#)

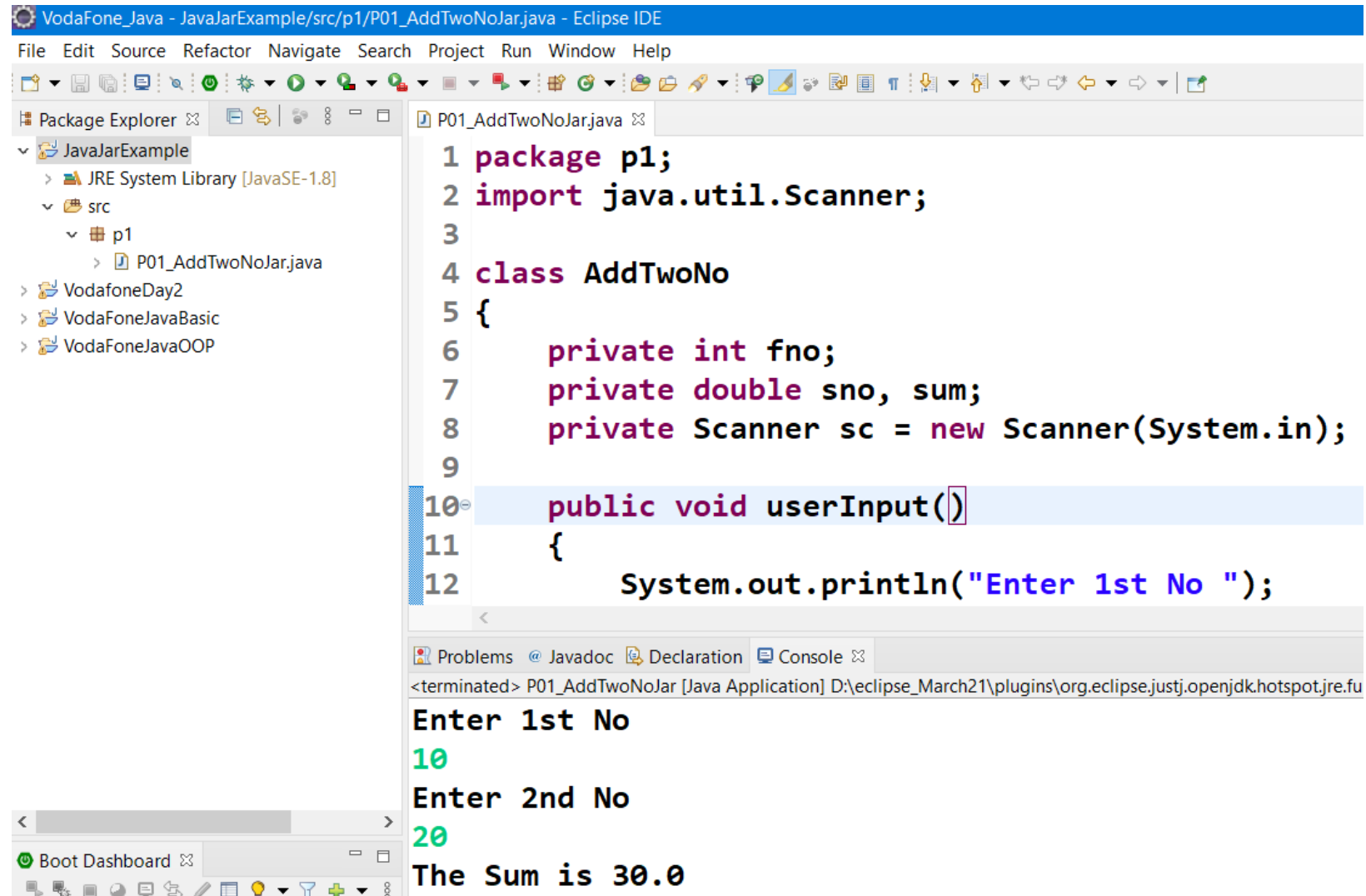
Working sets

☐ Add project to working sets [New...](#)

Working sets: [Select...](#)

i The default compiler compliance level for the current workspace is 15. The new project will use a project specific compiler compliance level of 1.8.

WRITE CODE AND RUN (TO GENERATE CLASS FILES)



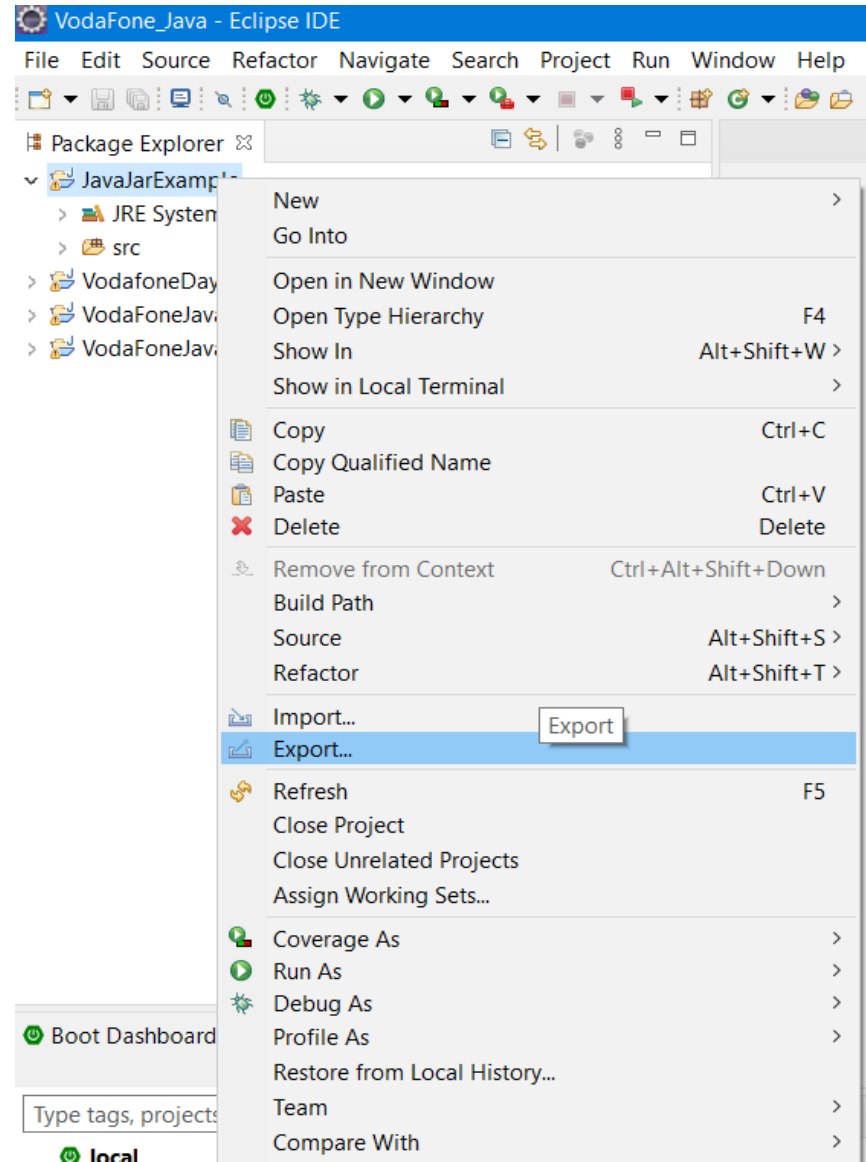
The screenshot displays the Eclipse IDE interface. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The Package Explorer on the left shows the project structure: JavaJarExample, JRE System Library [JavaSE-1.8], src, p1, and P01_AddTwoNoJar.java. The main editor window shows the code for P01_AddTwoNoJar.java. The code defines a package p1, imports java.util.Scanner, and creates a class AddTwoNo with private variables fno, sno, and sum, and a Scanner object sc. The userInput method is highlighted in blue. The bottom console window shows the execution output: Enter 1st No, 10, Enter 2nd No, 20, and The Sum is 30.0.

```
1 package p1;
2 import java.util.Scanner;
3
4 class AddTwoNo
5 {
6     private int fno;
7     private double sno, sum;
8     private Scanner sc = new Scanner(System.in);
9
10    public void userInput()
11    {
12        System.out.println("Enter 1st No ");
```

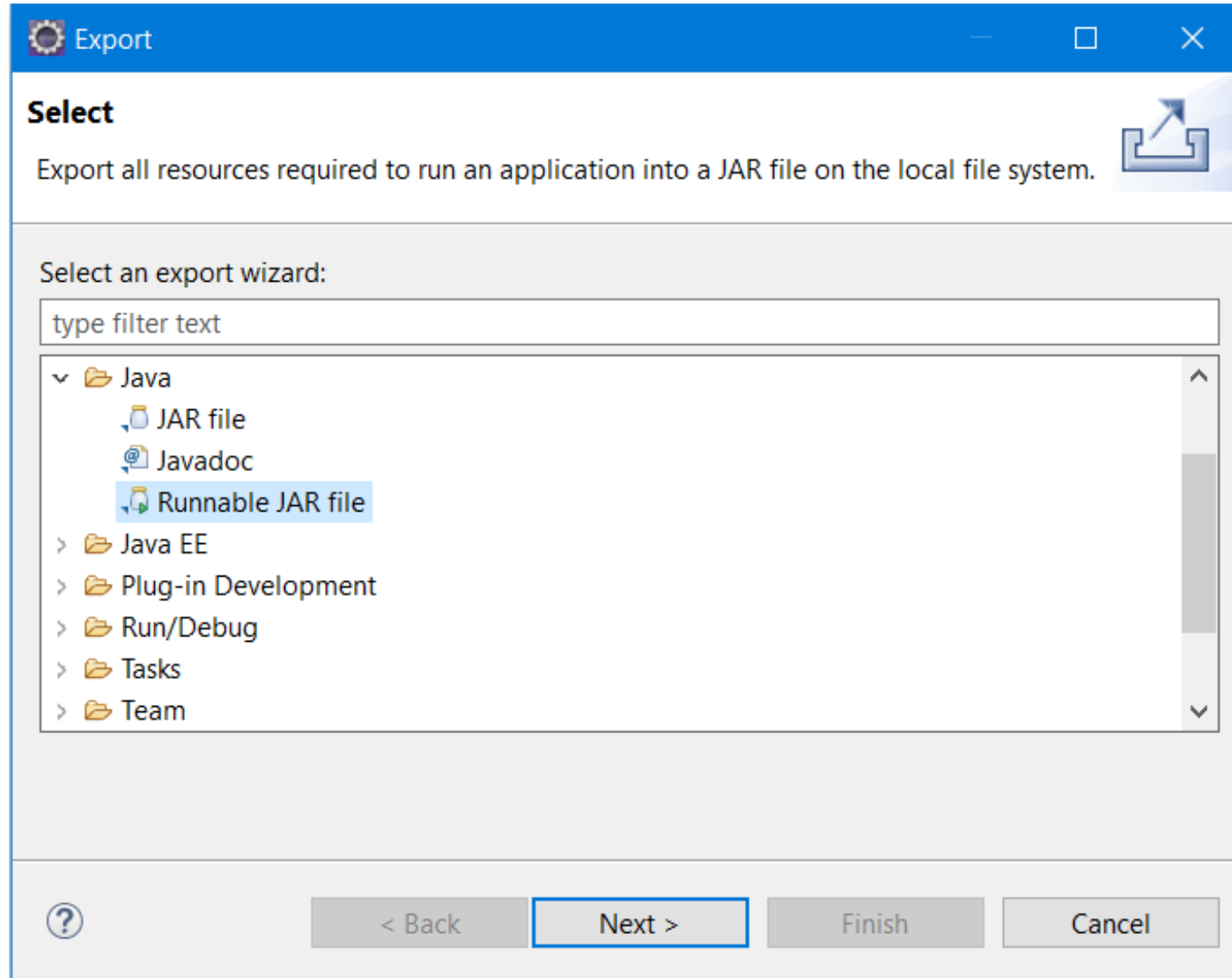
<terminated> P01_AddTwoNoJar [Java Application] D:\eclipse_March21\plugins\org.eclipse.justj.openjdk.hotspot.jre.fu

Enter 1st No
10
Enter 2nd No
20
The Sum is 30.0


CREATE JAR FILE




CREATE JAR FILE



CREATE JAR FILE

 Runnable JAR File Export

Runnable JAR File Specification

Select a 'Java Application' launch configuration to use to create a runnable JAR.

Launch configuration:

P01_AddTwoNoJar - JavaJarExample

Export destination:

C:\Users\User\Desktop\a.jar

Browse...

Library handling:

☒ Extract required libraries into generated JAR


☐ Package required libraries into generated JAR

☐ Copy required libraries into a sub-folder next to the generated JAR

☐ Save as ANT script

ANT script location: D:\VodaFone_Java

Browse...



< Back

Next >

Finish

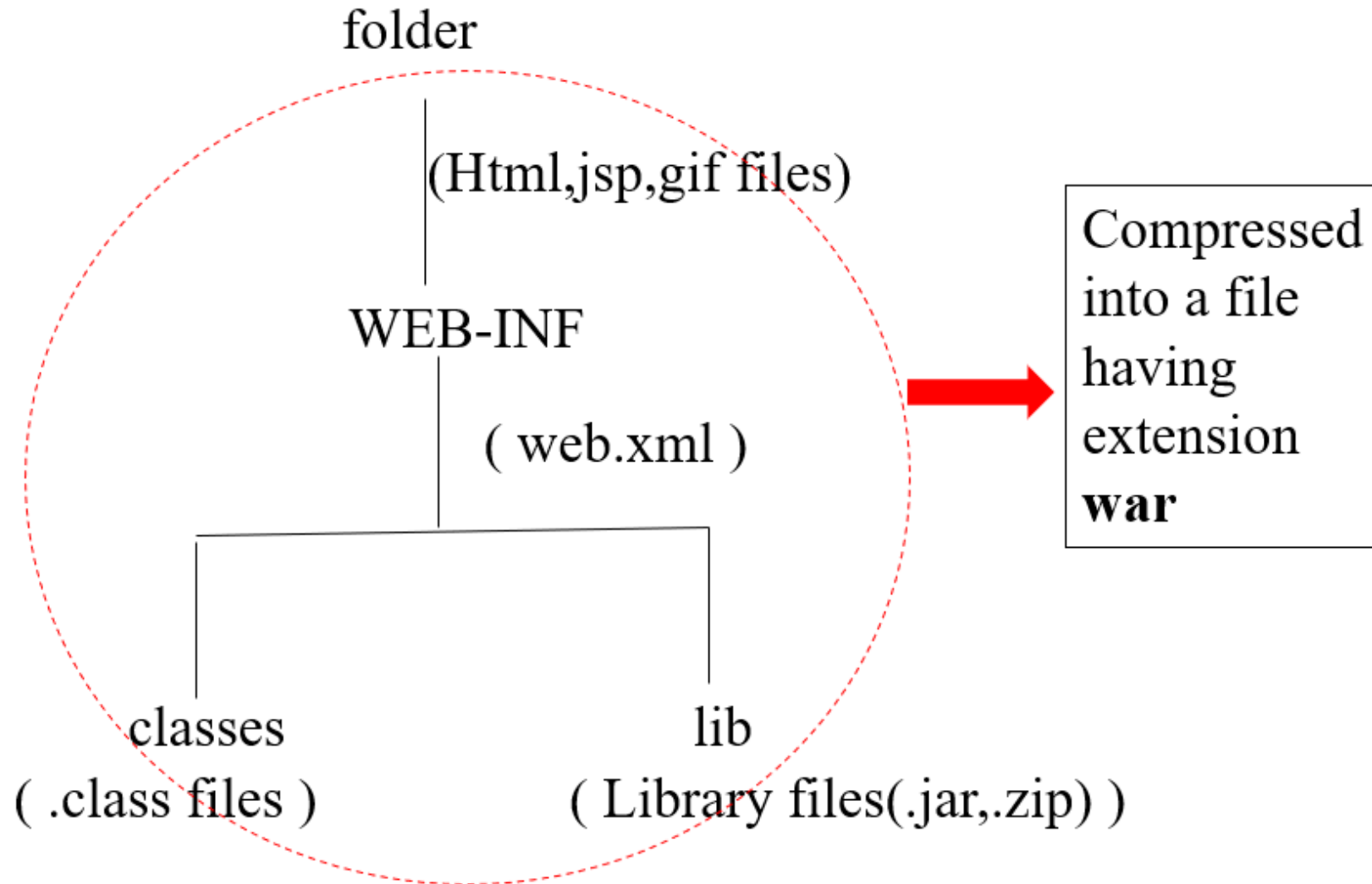
Cancel

RUN JAR FILE

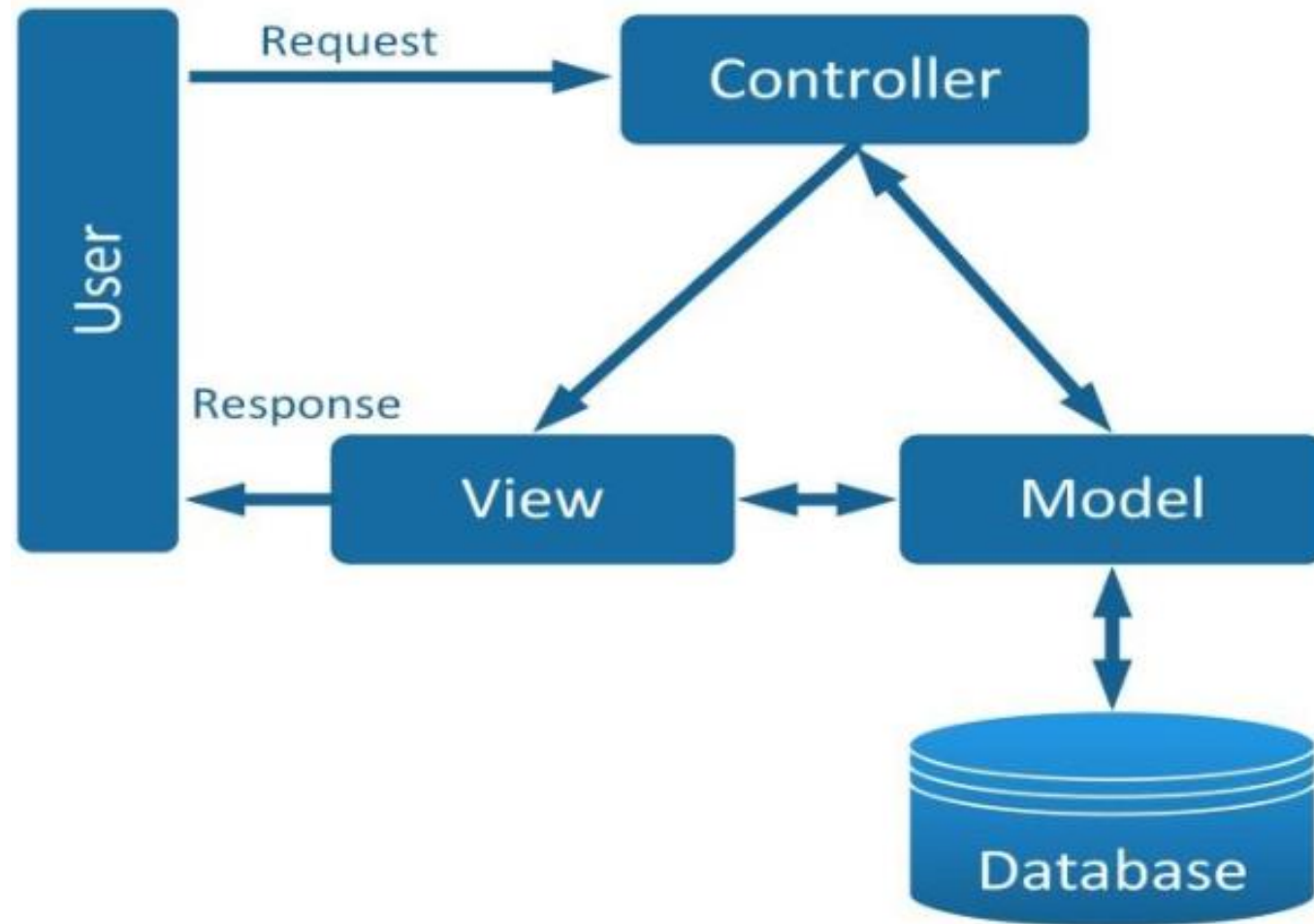
```
C:\Users\User\Desktop>java -jar a.jar  
Enter 1st No  
10  
Enter 2nd No  
20  
The Sum is 30.0
```



WAR Structure



MVC – Model View Controller





Create Maven Project