**What is a servlet?**

|  |
| --- |
| A servlet is a server side progam, which takes the request from the client, process the request and sends a response back to the client. |

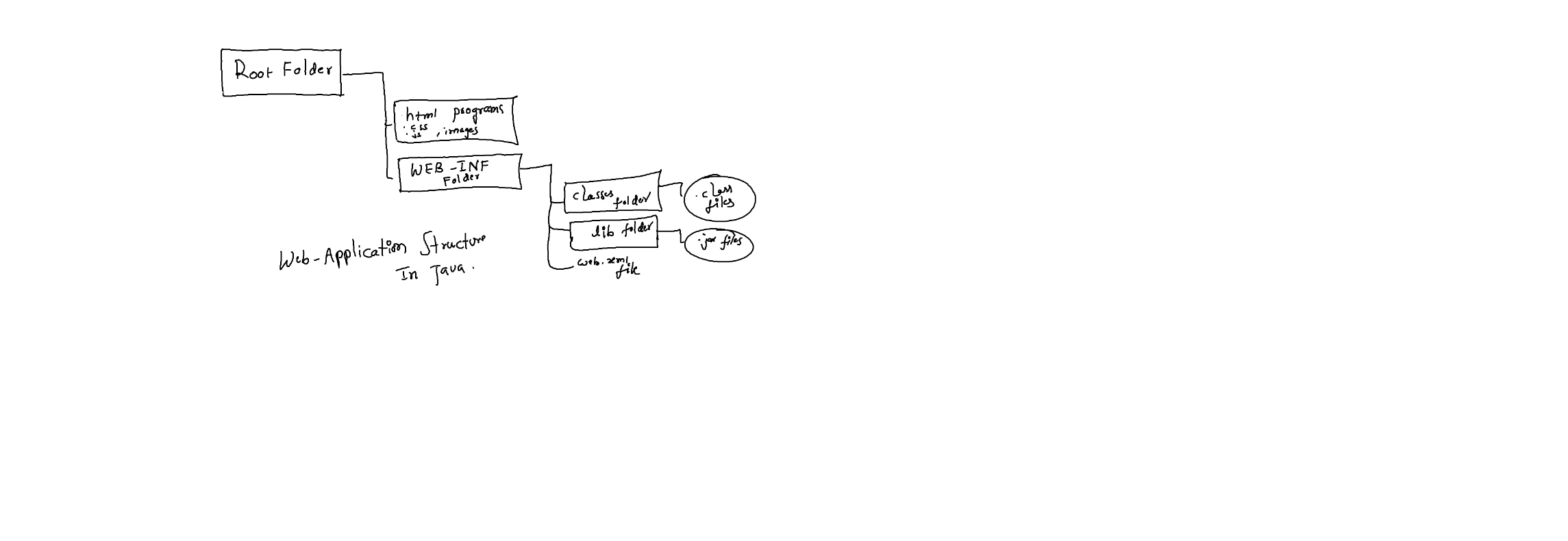
**In how many ways we can create a servlet?**

|  |
| --- |
| **There are 3 ways to create a servlet**   1. By implementing a Servlet interface 2. By extending a GenericServlet class 3. By extending HttpServlet class. |

**Creating a first web application manually in tomcat server**

|  |
| --- |
| 1. Install jdk 2. Install tomcat server |

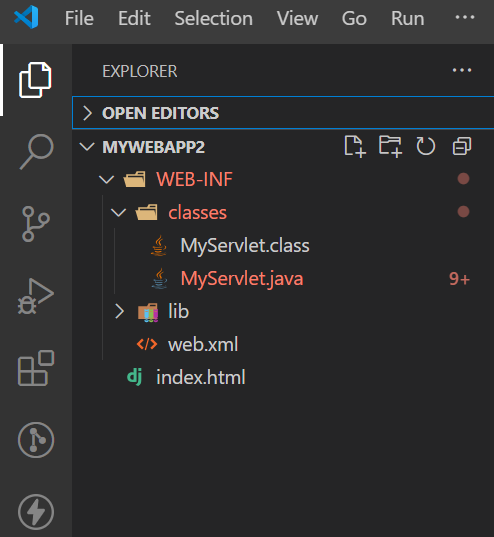
**Web application structure in Java**



**We can create projects directly in webapps folder of tomcat server**

|  |  |
| --- | --- |
| 1. Create a root folder (It behaves like a project name) it is also called as contextual root 2. I have created a folder called MyWebApp in webapps 3. Create a WEB-INF folder in contextual root. 4. To put the .class files, we have to create classes (small letters) folder and to put the libraries we have to create lib folder in contextual root. 5. If you want to place html, css, js and images etc. we have to place them in the contextual root directly. 6. I have written index.html file in root folder   **Index.html**   |  | | --- | | <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>Document</title>  </head>  <body>      <h1>Index Page</h1>  </body>  </html> |  1. We haven’t written any servlet but we can run this application, but we need to start the server      1. Run the web site like below |

**Second WebApplication (With a Servlet)**

****

|  |
| --- |
| 1. Create contextual root (root folder cum web application name) in webapps folder of tomcat server. |

**Can I run a web site developed in Java in tomcat server without providing WEB-INF folder in a web site?**

|  |
| --- |
| Yes, if you are creating a web site without write java programs then it is possible  Note: we create WEB-INF folder only if you want to put .class files. |

**Where we have to place .class files in a web application in Java?**

|  |
| --- |
| In classes folder |

**Where we have to place .jar files?**

|  |
| --- |
| In a lib folder |

**Where we place .html,.css,.js, images etc… in a web application?**

|  |
| --- |
| Directly in the contextual root (root folder) |

**I have created index.html file in root folder**

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>Document</title>  </head>  <body>      <h1>Index Page</h1>      <a href="iphone">InvokeServlet</a>  </body>  </html> |

**Where we have to place web.xml file?**

|  |
| --- |
| We have to create/place in WEB-INF folder  Web.xml file is also called as deployment descriptor |

**I am going to create a servlet?**

|  |
| --- |
| We can create a servlet by using servlet api? |

**What is servlet-api?**

|  |
| --- |
| It is an api which which contains two packages which are used to build a servlet.  Old version:   1. javax.servlet 2. javax.servlet.http   Latest:   1. jakarta.servlet 2. Jakarta.servlet.http |

**Where I can find these two packages?**

|  |
| --- |
| We can find these packages in servlet-api.jar file which is located at the following path: |

**Is it necessary to set the classpath for servlet-api.jar file?**

|  |
| --- |
| 1. Yes it is, if you are compiling the servlet manually, it is mandatory to set the classpath. 2. If it is IDE, no need to set the classpath for servlet-api.jar file |

**Create a servlet in classes folder**

|  |
| --- |
| I am going to create a servlet by implementing a Servlet interface |

**Where the getWriter() method is existed?**

|  |
| --- |
| In ServletResponse object |

**What the getWriter () method returns?**

|  |
| --- |
| PrintWriter object. |

**What is the use of PrintWriter object?**

|  |
| --- |
| By using the methods of PrintWriter object we can send the response to the client(browser). |

**Why we set the contentType?**

|  |
| --- |
| By using setContentType() method of HttpResponse object we can set what type of content a servlet is going to send to the client(browser). |

**MyServlet.java**

|  |
| --- |
| import java.io.PrintWriter;  import jakarta.servlet.\*;  public class MyServlet implements Servlet  {      private ServletConfig config;      public void init(ServletConfig config) throws jakarta.servlet.ServletException{          System.out.println("init method is invoked..");          this.config=config;      }      public ServletConfig getServletConfig(){          return config;      }      public void service(ServletRequest req, ServletResponse res) throws jakarta.servlet.ServletException, java.io.IOException{          res.setContentType("text/html");          PrintWriter out=res.getWriter();          out.println("hello<br>");          out.close();      }      public String getServletInfo(){          return "Myservlet Created by Madhu";      }      public void destroy(){          System.out.println("servlet destroy method is invoked...");      }  } |

**Web.xml**

|  |
| --- |
| <web-app>      <servlet>          <servlet-name>apple</servlet-name>          <servlet-class>MyServlet</servlet-class>      </servlet>      <servlet-mapping>          <servlet-name>apple</servlet-name>          <url-pattern>/iphone</url-pattern>      </servlet-mapping>  </web-app> |

**Who manages a servlet?**

|  |
| --- |
| **Servlet engine (or) servlet container** |

**Where the servlet engine/container contains?**

|  |
| --- |
| It is existed in a web-container. |

**Where we can find a web-container?**

|  |
| --- |
| In a server |

**Which constructor will be invoked of a servlet?**

|  |
| --- |
| During servlet creation, Servlet engine calls the default constructor. It wont’ call the parameterized constructor. That is the reason why we won’t write the parameterized constructor in a servlet. |

**What methods are called as life cycle methods?**

|  |
| --- |
| 1. init 2. service 3. destroy   These methods are called as life cycle methods because, these methods are invoked (after servlet creation and before servlet get destroyed) during the life of a servlet. |

**When the init() method will be invoked?**

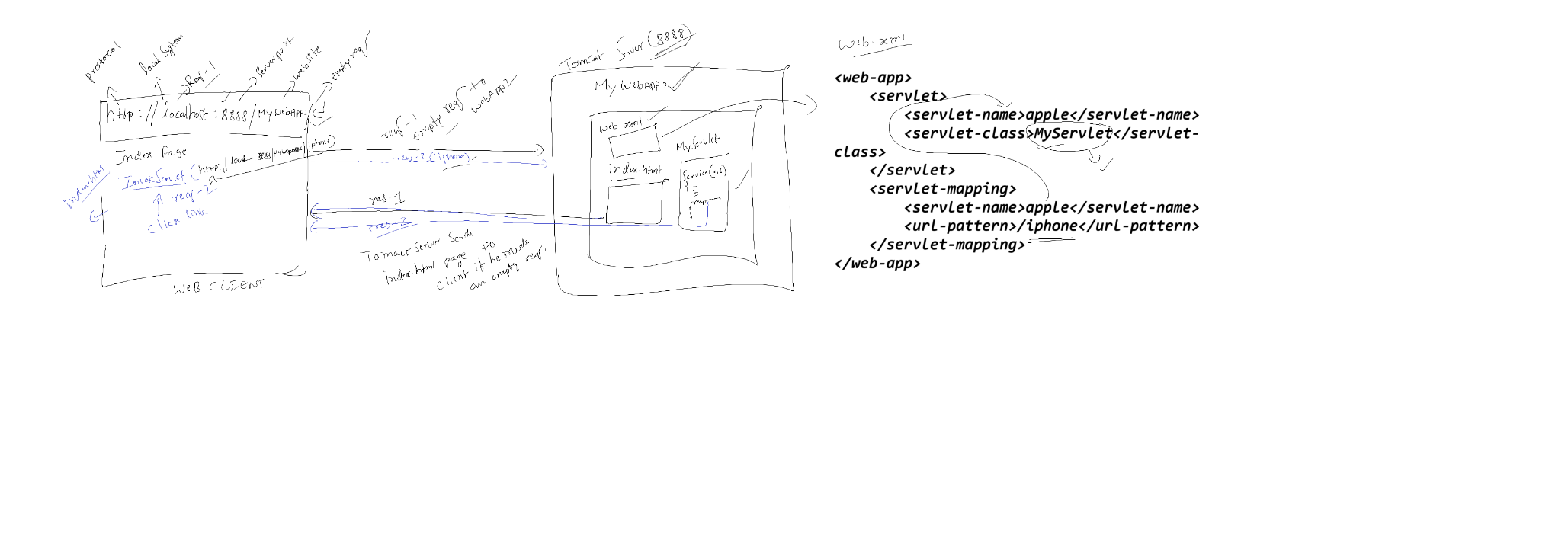
|  |
| --- |
| 1. This is the first method which is invoked after servlet creation by the servlet engine. 2. It is invoked only once during the life of a servlet |

**When the service method will be invoked?**

|  |
| --- |
| It is invoked each time whenever a servlet takes a request from the client. |

**When the destroy() method will be invoked?**

|  |
| --- |
| * It will be invoked just before object get destroyed. * It will be invoked only once during the life of a servlet |



**Who will pass ServletConfig object to the init() method?**

|  |
| --- |
| At the time of init() method invocation, servlet engine creates ServletConfig object and passes it to the init() method. |

**Who passes ServletRequest, ServletResponse objects to service method?**

|  |
| --- |
| Servlet engine passes these two objects to the service method during invocation. |

**Third WebApp Creation of A Servlet By Extending Generic Servlet class**

**Index.html**

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>Document</title>  </head>  <body>      <h1>Index Page</h1>      <a href="gs">Click Here To Open Generic Servlet</a>  </body>  </html> |

**MyServlet.java**

|  |
| --- |
| import jakarta.servlet.\*;  import java.io.\*;  public class MyServlet extends GenericServlet  {      @Override      public void service(ServletRequest request,ServletResponse response)throws ServletException,IOException{          response.setContentType("text/html");          PrintWriter out= response.getWriter();          out.println("<h1>Generic Servlet Response...</h1>");          out.close();      }  } |

**Web.xml**

|  |
| --- |
| <web-app>      <servlet>          <servlet-name>bunny</servlet-name>          <servlet-class>MyServlet</servlet-class>      </servlet>      <servlet-mapping>          <servlet-name>bunny</servlet-name>          <url-pattern>/gs</url-pattern>      </servlet-mapping>  </web-app> |

**Explaining about important methods of Generic Servlet class in brief**

|  |
| --- |
| javax.servlet  Class GenericServlet   * [java.lang.Object](http://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true)   + javax.servlet.GenericServlet * All Implemented Interfaces:   [Serializable](http://docs.oracle.com/javase/7/docs/api/java/io/Serializable.html?is-external=true), [Servlet](https://docs.oracle.com/javaee/7/api/javax/servlet/Servlet.html), [ServletConfig](https://docs.oracle.com/javaee/7/api/javax/servlet/ServletConfig.html" \o "interface in javax.servlet) |

**Servlet interface contains 5 methods**

|  |
| --- |
|  |

**ServletConfig methods**

|  |
| --- |
|  |

**Methods of GenericServlet class**

|  |
| --- |
|  |

**Why extra init() method is provided in GenericServlet class?**

|  |
| --- |
| * It was introduced for the programmer’s convenience purpose. i.e. if you want to override the you can directly override the this init() method without overriding life cycle init() method. |

**Who calls this extra init() method?**

|  |
| --- |
| Life cycle init() method calls it. |

**What is query string?**

|  |
| --- |
| <http://localhost:8888/WebApp4/WelcomeServlet?uname=Shekar>  in the URL after question mark whatever added it is called as query string |

**What is request parameter name?**

|  |
| --- |
| <http://localhost:8888/WebApp4/WelcomeServlet?uname=Shekar>  In the above url uname is the request parameter and Shekar is the request parameter value. |

**What a ServletRequest object contains?**

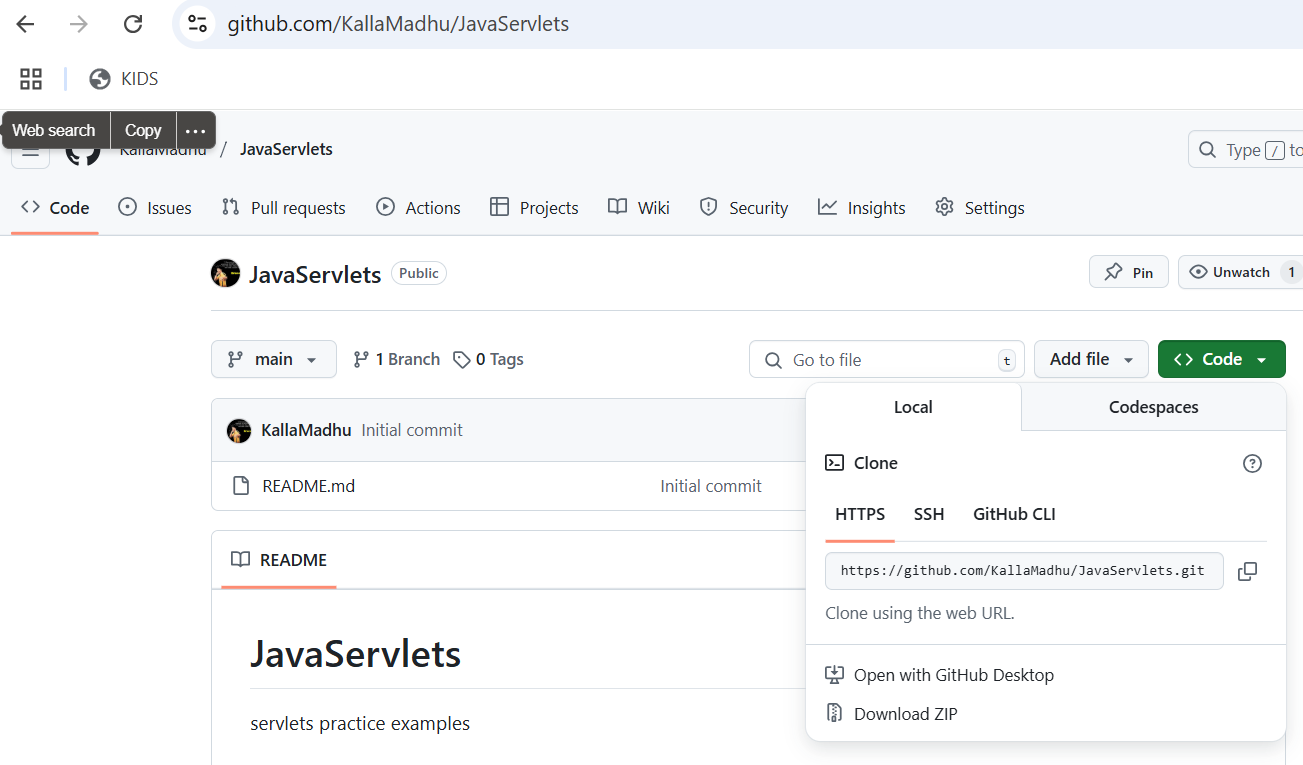
|  |
| --- |
| It contains client submitted form details(data) |

**How to get the data existed in ServletRequest object?**

|  |
| --- |
| * + 1. By using getParameter() method     2. getParameter() method takes the request parameter name as an argument and returns the request parameter value as a string to us. |

**How to upload/push our websites into remote repository**

* **Create a remote repository**

****

* Initialize the git in you local system (in a particular folder), see the below command which initialize the local repository.

|  |
| --- |
|  |

* Connect the local repository with remote repository

|  |
| --- |
| Git remote add origin <https://github.com/KallaMadhu/JavaServlets.git> |

* Pull the project from remote repository

|  |
| --- |
| git pull <https://github.com/KallaMadhu/JavaServlets.git> main |

* Command to push the files from present folder to staging area

|  |
| --- |
| C:\Program Files\Apache Software Foundation\Tomcat 10.1\webapps\WebApp3> git add . |

* Commit the files

|  |
| --- |
|  |

* Now we can push the files existed in local repository to remote repository

|  |
| --- |
|  |

**Why we write HttpServlet?**

|  |
| --- |
| * It is the child class of GenericServlet * It is a servlet which takes only http requests * It doesn’t take the request coming through protocols other than http |

**What is the use of http protocol?**

|  |
| --- |
| * It is a stateless protocol * It is a protocol which doesn’t maintain any information about the previous request.   What is the advantage of http protocol?  It reduces the network traffic  What is the dis-advantage of http protocol?  Session tracking becomes difficult |

**What is statfull protocol?**

|  |
| --- |
| It is a protocol which maintains the previous request information, so it increases the network traffic  **Advantage:**   * + 1. Session tracking will be easy |

**What is a Generic Servlet?**

|  |
| --- |
| * It is a protocol independent servlet so it can take the requests coming through any protocol * We won’t write generic servlets in real time because, this might increate network traffic |

**How many service methods are there In HttpServlet class?**

|  |
| --- |
| **Two service methods**   * publicc void service(ServletRequest,ServletResponse) * protected void service(HttpServletRequest, HttpServletResponse) |

**What is the use of public service method? And what kind of code is written in public service() method of HttpServlet?**

|  |
| --- |
| * It calls the protected service method and dispatches HttpServletRequest and HttpServletResponse objects to it, if the request is coming through http protocol. * Simply we can say it filters the requests. If the request is http request then only it will dispatch to protected service method. |

**What is the use of protected method of HttpServlet?**

|  |
| --- |
| It dispatches the request to the doXXX() method, based on the reques made by client.  If client makes get request then doGet() method will be invoked  If client makes post request the doPost() method will be invoked….etc.. |

**What is the difference between get and post?**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | **Feature** | **GET Request** | **POST Request** | | --- | --- | --- | | **Purpose** | Retrieves data from the server | Sends data to the server to be processed | | **Data Location** | Appended to the URL (e.g., ?id=123) | Sent in the body of the request | | **Visibility** | Visible in the URL (less secure) | Not visible in the URL (more secure) | | **Length Limit** | Limited by URL length (varies by browser) | No significant length limit | | **Caching** | Can be cached by browsers | Not cached by default | | **Bookmarkable** | Yes, since data is in the URL | No, because data is in the body | | **Idempotency** | Yes (should not change server state) | No (typically changes server state) | | **Use Cases** | Search queries, retrieving resources | Submitting forms, uploading files | |