BIG-7 methods
============
  What is the difference b/w GET and POST request type?

GET
      if we want to get some information from the server then we need to for GET request.
            eg:fetching the train ticket details by supllying source and destination information.
      usually get request are read only request and at the server side update operation won't be performed.
      In case of GET request,end user provided information will be attached to the url in the form of "QueryString"
                  eg: http://localhost:9999/App-01/test ?source=bengaluru&destination=shivamoghaa
      In case of GET request,end user data is available inside url as query string, it is less secured.
      In case of GET request,since the data is attached in the url as querystring, only small volume of data can be sent.
      Bookmarking and caching is supported in case of GET request.

POST
      we use POST request to send huge amount of data to the server.
            eg: uploading resume
      usally post request are write only request and at the server side update/insert operation would be performed.
       In case of POST request,end user provided information will be not be attached to the url in the form of "QueryString"
                  eg: http://localhost:9999/App-01/test
      In case of POST request,end user data is not available inside url as query string, so it is more secured.
      In case of POST request,since the data is not attached in the url as querystring, large volume of data can be sent.4
      Bookmarking and caching is not supported in case of POST request.

Note: What is idempotent request?
            By repeating the request multiple times, if there is no change in the response then such type of request we call
            as "idempotent" request.
                        eg: GET request is idempotent request,but POST is not Idempotent.

       What is safe request?
            By repeating the request multiple times,if there is no side effect at the server side then such type of request we call
            as "safe" request.
                        eg:GET request is safe request,but POST is not safe request.

How to send GET request?
      1. Type the address in the url bar and hit enter key(request is sent)
      2. clicking the hyperlink.(CLICK HERE)
      3. submit the form with method attribute as "GET".
            <form method ="GET">
                        <!--- -->
            </form>
      4. submit the form without method attribute(default is GET only)

```
            <form>
                  <!--- -->
            </form>
```

How to send POST request?

1. submit the form with method attribute as "POST".

```
                  eg: <form method ="POST">
                                    <!--- -->
                        </form>
```

Note:

When we send the request,automatically the HTTPProtocol create the HTTPRequest Object.

The relevant information will be assigned in the respective sections of HTTPRequest object.

This HTTPRequest object will be taken by HTTP Protocol and it will send it to the respective Server.

Upon sending the response, automatically the HTTPProtocol will create HTTPResponse Object.

The relevant information will be assigned in the respective sections of HTTPResponse Object.

This HTTPResponse object will be taken by HTTPProtocol to the browser and browser uses this information for

displaying the output.

To build webaplication we need to follow standard directory structure given by Server vendor

==========

```
      ProjectName
            |=> WEB-INF
                        |=> web.xml(deployment descriptor)
                        |=> classes
                                    |=> .class
                        |=> lib
                                 |=> *.jar

            |=> src/main/java
                        |=> .java
            |=> pages
                        |=> .jsp
```

How to create Servlet in Java?

To create Servlet in java there are 3 approaches

a. Servlet(I)

b. GenericServlet(AC)

c. HttpServlet(AC)

Servlet(I)

```
public interface Servlet {
      public abstract void init(ServletConfig config) throws ServletException;
      public abstract ServletConfig getServletConfig();
      public abstract void service(ServletRequest request,ServletResponse response)
throws ServletException, IOException;
      public abstract String getServletInfo();
      public abstract void destroy();
}
```

Whenever we write Servlet,Automatically for the written servlet container will perform the following actions

1. Depending on the url pattern supplied by the user for a dynamic resource
    a. Servlet Loading will happen ===========>static block will be executed.
    b. Servlet Instantiation will happen=======>constructor will be called and object will be created.
    c. Servlet Initialization will happen =======>void init(ServletConfig config) throws ServletException
    d. Serlvet Request Processing will happen==>void service(ServletRequest request,ServletResponse response)

       throws ServletException,IOException
    e. Servlet De-Instantiation will happen ====>void destroy();

eg#1.

```
import java.io.*;
import javax.servlet.*;
public class FirstServlet implements Servlet
{
	static
	{
		System.out.println("FirstServlet.class file is loading...");
	}
	public FirstServlet()
	{
		System.out.println("FirstServlet Object is instantiated...");
	}
	//For Servlet Initialization container calls this method
	public  void init(ServletConfig config) throws ServletException
	{
		System.out.println("Servlet initialziation...");
	}

	public ServletConfig getServletConfig()
	{
		return null;
	}

	//Request Processing logic
	public  void service(ServletRequest request,ServletResponse response)
		throws ServletException, IOException
	{
		System.out.println("Servlet Request Processing ...");
	}

	public  String getServletInfo()
	{
		return null;
	}

	//Servlet DeInstantion logic
	public  void destroy()
	{
		System.out.println("Servlet De-Instantiation...");
	}
}
```

2.

After creating a Servlet, for every servlet url-pattern matching should be provided and it should be informed to the
container via XMl,Annoation.

XML
====

```
<web-app>
	<servlet>
		<servlet-name>DemoServlet</servlet-name>
		<servlet-class>FirstServlet</servlet-class>
	</servlet>
	<servlet-mapping>
		<servlet-name>DemoServlet</servlet-name>
		<url-pattern>/test</url-pattern>
	</servlet-mapping>
</web-app>
```

3. Before compilation set path and classpath environmental variable.

```
		set path= C:\programfiles\jdk1.8\bin
		set classpath=;.;C:\Tomcat 9.0\lib\servlet-api.jar
```

After compilation of FirstServlet.java copy the .class file to classes folder present under WEB-INF/classes.

4. Now start the server by going to tomcat9.0/bin/tomcat9.exe file
5. Now send the request by typing the url as shown below

```
		http://localhost:9999/FirstApp/test
```