1. What is HTTP?

HTTP (**H**yper**T**ext **T**ransfer **P**rotocol) is the communication protocol between server and client. It is a request response protocol and It is stateless means each request is considered as the new request.

1. How does a request and response model work?

When the user enters the URL in the browser, the client generates the get request and sends it to the server. As the server receives the request it sends it to the web container. In the deployment descriptor pattern it searches for the pattern. If the pattern is found the request goes to the particular servlet class. Else the container searches for the page and send the response back to the server and server sends the response back to the client.

1. GET vs POST?

**GET** -Requests data from a specified resource

GET requests can be cached

GET requests remain in the browser history

GET requests can be bookmarked

**POST** - Submits data to be processed to a specified resource

POST requests are never cached

POST requests do not remain in the browser history

POST requests cannot be bookmarked

POST requests have no restrictions on data length

1. What are different HTTP status codes?

200 for success, 404 for Not Found and 403 for Access Forbidden.

200 for success, 300 for redirect, 400 for client error, 500 internal server error

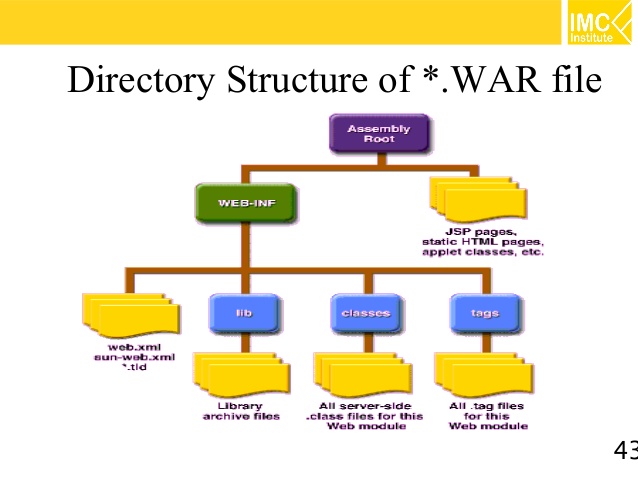
201 created success

400 bad request

401 unauthorized

405 unsupported media type

1. Web application folder structure?



1. What is web.xml?

**web.xml** file is the deployment descriptor of the web application which contains mapping for servlets, welcome pages, security configurations, session timeout settings

1. What is the artifact that web application is packaged into?

War file. The war file is a web archival file that **contains the web application** that can be deployed on any servlet/web container. It **contains jsp, html, JavaScript, Class files** and other files necessary for the development of web applications. It reduces the time duration for transferring file.

1. How do you secure web application?

**Metadata annotations** (or simply, **annotations**) are used to specify information about security within a class file. When the application is deployed, this information can either be used by or overridden by the application deployment descriptor.

**Declarative security** expresses an application’s security structure, including security roles, access control, and authentication requirements in a deployment descriptor, which is external to the application.

**Programmatic security** is embedded in an application and is used to make security decisions. Programmatic security is useful when declarative security alone is not sufficient to express the security model of an application.

1. What is web container?

Web container also known as a Servlet container is the **component** of a web server that interacts with Java servlets. A web container is responsible for managing the lifecycle of servlets, mapping a URL to a particular servlet and ensuring that the URL requester has the correct access rights.

1. What is the difference between webserver and application server?

Application Server supports distributed transaction and EJB. While Web Server only supports Servlets and JSP.

Application Server can contain web server in them. most of App server e.g. JBoss or WAS has Servlet and JSP container.

Though it’s not limited to Application Server but they used to provide services like Connection pooling, Transaction management, messaging, clustering, load balancing and persistence. Now Apache tomcat also provides connection pooling.

In terms of logical difference between web server and application server. web server is supposed to provide http protocol level service while application server provides support to web service and expose business level service e.g. EJB.

Application server are more heavy than web server in terms of resource utilization.

1. How do you implement session management in web application?

**Session** simply means a particular interval of time. **Session Tracking** is a way to maintain state (data) of a user. It is also known as **session management** in servlet. HTTP is stateless that means each request is considered as the new request.

There are four techniques used in Session tracking:

1. **Cookies**
2. **Hidden Form Field**
3. **URL Rewriting**
4. **HttpSession**

In HTTP Session, the container creates a session id for each user. The container uses this id to identify the particular user. An object of HttpSession can be used to perform two tasks:

1. bind objects
2. view and manipulate information about a session, such as the session identifier, creation time, and last accessed time.

In case of Hidden Form Field, **a hidden (invisible) text field** is used for maintaining the state of a user. In such case, we store the information in the hidden field and get it from another servlet. This approach is better if we have to submit form in all the pages and we don't want to depend on the browser. It is widely used in comment form of a website. In such case, we store page id or page name in the hidden field so that each page can be uniquely identified.

A **cookie** is a small piece of information that is persisted between the multiple client requests. A cookie has a name, a single value, and optional attributes such as a comment, path and domain qualifiers, a maximum age, and a version number.

1. What is MVC?

MVC is a design pattern which is used to develop web application and it separates the business layer with presentation layer. It consists of 3 components namely, Model, view and controller. In the model component the developer develops the java classes which connects to the database. In the view component all the presentation files are developed and in the controller component the servlets are developed by the developer.

1. What is welcome-file list in web.xml?

The **welcome-file-list** element of **web-app**, is used to define a list of welcome files. Its sub element is **welcome-file** that is used to define the welcome file. A **welcome file** is the file that is invoked automatically by the server, if you don't specify any file name. If welcome-file-list entry doesn't exist in web.xml file, priority goes to index.html file then index.htm and at last index.jsp file.

1. What is load on startup?

The **load-on-startup** element of **web-app** loads the servlet at the time of deployment or server start if value is positive. It is also known as **pre initialization of servlet**. You can pass positive and negative value for the servlet. servlet is loaded at first request. That means it consumes more time at first request. If you specify the load-on-startup in web.xml, servlet will be loaded at project deployment time or server start. So, it will take **less time** for responding to first request.

1. What is the difference between forward and sendRedirect?

This method is declared in HttpServletResponse**Interface**.

This method is used to redirect client request to some other location for further processing**,** the new location is available on different server or different context. Web container handles this and transfer the request using browser, and this request is **visible in browser as a new request**. Sometime this is also called as client side redirect.

This method is declared in RequestDispatcherInterface.

This method is used to pass the request to another resource for further processing **within the same server**, another resource could be any servlet, jsp page or any kind of file. This process is taken care by web container when we call forward method request is sent to another resource **without the client being informed,** which resource will handle the request. This is also called server side redirect.

1. How can you achieve a response from cross browser domain?

Through sendRedirect() we can get a response from cross browser domain.

17.What is URL?

URL is acronym of Universal Resource Locator and it’s used to locate the server and resource. Every resource on the web has its own unique address.

**http://localhost:8080/FirstServletProject/jsps/hello.jsp**

**http://** – This is the first part of URL and provides the communication protocol to be used in server-client communication.

**localhost** –The hostname of the server that maps to unique IP address.

**8080** – This is the port on which server is running. Port numbers 0 to 1023 are reserved ports for well-known services, for example 80 for HTTP, 443 for HTTPS, 21 for FTP etc.

**FirstServletProject/jsps/hello.jsp** – FirstServletProject is the context-root. Resource requested from server. It can be static html, pdf, JSP, servlets, PHP etc.