***Web Application Basic Concepts***

***Web Browser:***

* *It’s a “Desktop Application” which helps us to interact with web applications*
* *Browser is the One & Only application which understands content/data present in HTML and display accordingly*

***Web Resources:***

* *Resources present inside a web application are called as web resources*
* *There are two types of web resources*

1. *Static web Resources:*

* *These resources “are present at web application” before making the request*
* *Content of these resources “does not change”.*
* *In other words, resources which generates “static response” is called as Static web Resources*
* *Few Examples:*

1. *Any Songs Downloads*
2. *Any Books (PDF, MS-WORD, etc.) Download*
3. *Any Software Download*
4. *Any Video/Movie files.*
5. *Dynamic Web Resources:*

* *These resources “****does not*** *present at web application” before making the request & they get generated at the time of request.*
* *Content of these resources “May Change” for every request (Dynamic Response)*
* *In other words, resources which generates “dynamic response” is called as Dynamic Web Resources*
* *Few Examples:*

1. *Any NetBanking Web Application Transaction Statement Download (PDF file)*
2. *Any Post Paid Connection Statement Downloads (PDF file)*
3. *Google Search Page (HTML Page)*
4. *Gmail Inbox Page (HTML Page)*
5. *Facebook Home Page (HTML Page)*
6. *Gmail “Download All Attachments” (ZIP file)*

*Note:*

*Both Static & Dynamic Web Resources can be “HTML” or “Non-HTML” in nature.*

***Web Path:***

* *It’s a Path in Webserver in which Web Applications are present*
* *Web Path varies from Webserver to Webserver, we have to read the manual to get this information.*
* *In case of Tomcat web path is “<Tomcat Location>/webapps”*
* *Hence in Tomcat, WAR file should be kept webapps.*

***Starting the WebServer***

***Note:***

* *When we start webserver it should not throw any exception in the console*
* *At the time of starting the server, webserver extracts the contents of WAR file to a folder by same name inside “webapps” folder.*

***Web Application*** *:*

* *Web Application is an application which is accessed over the network with the help of web browser*
* *Web application is a collection of web resources*
* *If a web application consists of “ONLY static resources then it is called as “Static Web Application”*
* *If a web application consists of “one/more dynamic resources” then it is called as “Dynamic Web Application”*

*Example: Gmail, Facebook, Twiter, Flipkart, etc.,*

* *J2EE helps us to develop “Dynamic Web Applications”*

***Web Server:***

* *Like any other application (Adobe Reader, Media Player, etc.,) , Webserver is also an application which runs on Operating System*
* *Webserver as the name implies “Serves requests to a Web Applications”*
* *In other words, it helps both web browser & web application to interact with each other*
* *Hence every web application (Static/Dynamic) is directly under the control of webserver*
* *Few Examples:*

1. *Apache Tomcat*
2. *Apache JBOSS*
3. *IBM WebSphere*
4. *Oracle WebLogic*
5. *Oracle GlassFish & many more…*

*Different ways to interact with Web Applications*

1. *By Typing an URL in Browser*
2. *By Clicking on the Hyperlink*
3. *By Submitting the HTML Form*

***URI, URL and URN***

* *Uniform Resource Identifier (URI) is a string of characters used to*

*- identify a resource using name or*

*- Locate a resource in the network*

* *A URI identifies a resource either by location, or a name, or both. A URI has two specializations*

*1. URL (Uniform Resource Locator) and*

*2. URN (Uniform Resource Name)*

* * URN ONLY identifies the resource and does not let us know availability of the resource. A URN has to be of this form "urn:"*
* *URL that specifies where an identified resource is available and the mechanism for retrieving it. URL does not have to be HTTP URL (http://), a URL can also be (ftp://) or (smb://) or (jdbc:)*
* *For example,*

*- A URN is similar to a person's name, while*

*- A URL is like a street address.*

*- The URN defines something's identity, while the URL provides a location.*

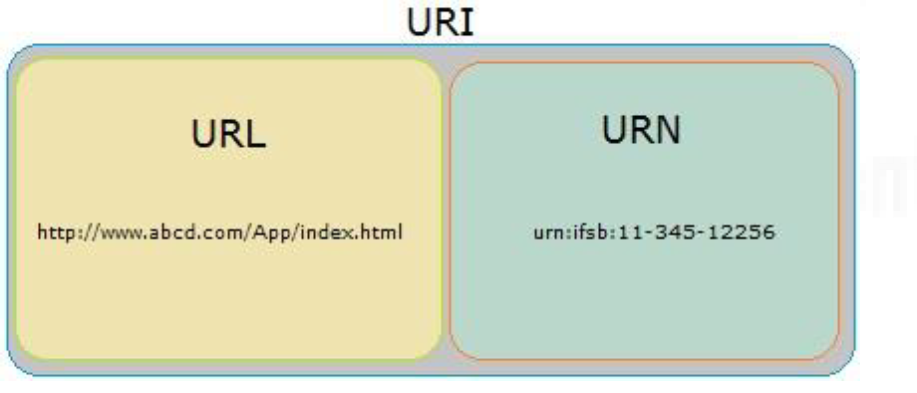
*- Essentially, "what" vs. "where"*

* *To put it differently,*

*- A URL is a URI*

*- A URN is a URI*

*- but URNs and URLs are different, A URI is not necessarily a URL*

**

***WEB URL***

* *Web URL, uniquely indentifies a particular web resource inside a web application*
* *Hence every web resource (Static/Dynamic) must need to have its unique address in the form of “web URL”*

*Note: In case of Static web Resources, URL Consists of Resource file Name*

***Protocol:***

* *When one application wants to communicate with other (or in case browser & server) , there needs to be a common language which both application understands & that language should have set of rules and instructions*
* *This common language is known as “Protocol” where protocol is “Set of Rules”*
* *Web Browser & Web Server application communicate using*

1. *Hyper Text Transfer Protocol (HTTP)*
2. *Hyper Text Transfer Protocol Secure (HTTPS)*

* *As the name implies most of the time HTTP Response contain HTML*
* *In Url it’s an optional informtion & default protoco is HTTP*

***Domain:***

* *IT uniquely identifies a computer in a network in which web application is present*
* *Domain consists of Computer Name / IP address of the computer in which web application is present*
* *In URL it is a Mandatory Information*

***Port:***

* *Port number in Web URL uniquely identifies web server application*
* *Default port number for http IS 80 & https is 443*
* *In URL this is an optional information*
* *When it’s not used default port number is used depending on the protocol present in Web URL*
* *In Tomcat Webserver, default port number for HTTP is changed from 80 to 8080 and default port number for HTTPS is changed from 443 to 8443*

***Path:***

* *We know that web application is a collection of web resources (Static / Dynamic) & also Web Server can consist of one/more applications*
* *Path is the full path of the web resource at web application side*
* *It consists of Web Application Name + (File Name in case of Static Resource OR configred URL in case of Dynamic Resource)*
* *Web Application Name “ uniquely identifies One web Application inside webserver.*
* *“File Name” uniquely identifies Static web resource inside that web application*
* *“Configured URL” uniquely identifies Dynamic web*
* *In URL, it’s an optional Information.*

***Query String:***

* *Query String is a name & value string pair which passes information ONLY to Dynamic Resources such as Servlets & JSPs*
* *In URL, It’s an optional information and if present it starts with question mark followed by one or more name-value pair which are separated by an ampersnd(&)*
* *Matrix Parameters are a set of “name=value”. They can be present anywhere in URL (generally used with* ***path****) & URL can consist of N number of Matrix parameters but they should be separate by a semi colon “;“*
* *The important difference between Query Parameters & Matrix Parameters is that,*

*- Matrix Parameters apply to a particular path element while*

*- Query Parameters apply to the request as a whole*

*- This comes into play when making a complex REST-style query to multiple levels of resources and sub-resources*

*Examples:*

*Few Examples:-*

*URL: ftp://ftp.is.co.za/rfc/rfc1808.txt*

*URL: http://www.ietf.org/rfc/rfc2396.txt*

*URL: ldap://[2001:db8::7]/c=GB?objectClass?one*

*URL: mailto:John.Doe@example.com*

*URL: news:comp.infosystems.www.servers.unix*

*URL: telnet://192.0.2.16:80/*

*URN (not URL): urn:oasis:names:specification:docbook:dtd:xml:4.1.2*

*URN (not URL): urn:isbn:0-486-27557-4*

***Fragment ID***

* *A fragment ID or Fragment Identifier, as the name implies, it refers to a particular section within a web page*
* *In URL, It’s an optional information & if present, it begins with a hash (#) character followed by an identifier.*

*Example:*

[*http://tomcat.apache.org/tomcat-6.0-doc/manager-howto.html#SessionStatistics*](http://tomcat.apache.org/tomcat-6.0-doc/manager-howto.html#SessionStatistics)

***Request and Response Structure***

*Key elements of HTTP Requests are:*

1. *URL*
2. *Form Data (if any)*
3. *HTTP Method*
4. *Cookies (if any)*

*Key elements of HTTP Response are:*

1. *Status code*
2. *Content Type*
3. *Actual Content*
4. *Cookies (if any)*

***Status Code:***

* *Status code represents the status of HTTP Request For example,*

*Status Code: Description*

*200 Server successfully handled the request*

*404 Requested Resource (static/dynamic) is not found at server side*

*500 Server encountered an unexpected condition which prevented it from fulfilling the request.*

* *It’s a Mandatory information & it will be present in Header of HTTP Response*
* *Generally Webserver provides “Status Code” info in Http Response*

***Content Type OR Multipurpose Internet Mail Extensions (MIME)***

* *Content Type OR Multipurpose Internet Mail Extensions (MIME) Type, tells the browser that what type of content it’s going to receive so that it can prepare itself to handle the response data*
* *For example,*
* *Open an Adobe Reader to handle PDF content*
* *Open Media Player to handle media content etc.*
* *It’s a mandatory information & it will be present in Header of HTTP Response*
* *The default content type is “text/html”*
* *Few Examples:*

*text/html*

*appplication/pdf*

*video/quicktime*

*Many more…*

***Actual Content****:*

* *It’s a Mandatory information & it will be present in Body of HTTP Response*
* *In case of static resource content of the resource becomes the “Actual Content”*
* *In case of dynamic resource, content present in servlets/ JSP becomes the “Actual Content”*
* *In case of Error Scenarios webserver generates error information & it becomes the “Actual Content”.*

***Request Components:***

***Web Url***

* *Web application is a “Collection of Web Resources” & every web resource (static/dynamic) should have its unique address in the form of web URL*
* *Hence every request should consist of Web URL (Mandatory information) & it will be present in Header HTTP Request.*

***Form Data:***

* *Data collected using HTML form is called as Form Data*
* *i.e Whenever we make request by Submitting Form, Then ONLY HTTP Request will have Form Data*
* *Hence in HTTP Request it’s an Optional Information*
* ***If Present, it may be present in either Header or Body of HTTP Request which depends on HTTP Method Present in the Request.***

***HTTP Method***

* *It’s a mandatory information present in the header of the HTTP Request*
* *HTTP Method is the first component in the HTTP Request header*
* *HTTP 1.0 had 3 mehods & in HTTP 1.1, Five new Methods got introduced so in total HTTP 1.1 has 8 different methods & every Http Request should consist of “ONE of the 8 HTTP Methods”*

*(Present in HTTP 1.1)*

1. *HEAD*
2. *TRACE*
3. *PUT*
4. *DELETE*
5. *OPTIONS*
6. *GET*
7. *POST*
8. *CONNECT*

*Code Word: htpp dog c*

***POST Method***

* *If this method present in the Request then Webserver invokes doPost(HSR, HSR) Method in that Servlet*
* *This method allows user to Post the data of unlimited size to the server using HTML FORM (i.e Form Data)*
* *Post requests have a body*
* *Hence data sent using the POST is present in the body of the HTTP Request.*

***GET Method***

* ***If this method present in the Request then Webserver invokes doGet(HSR, HSR) Method in that Servlet.***
* ***This method allows us to get the data from server***
* ***It’s a default method***
* ***Get requests “Using the GET Method the form-data is present in the Header part of HTTP Request “in the form of Query String”***

***What determines whether Browser sends GET or POST requests:***

1. ***Typing a URL in Browser makes request to contain GET method***
2. ***Clicking on a Hyper link in Browser makes request to contain GET Method***
3. ***Submitting the form with method= “get” form attribute in Browser makes request to contain Get method.***
4. ***Submitting the form with method = “post” form attribute in Browser makes request to contain “POST” method.***

***Submitting the form with “No method form attribute declaration “ in Browser makes request to contain “GET” method.***

***Differences between GET/doGet() & POST/doPost()***

|  |  |
| --- | --- |
| *GET/ doGet* | *POST/ doPost()* |
| *GET method allows to get the data from server* | *POST method allows to Post data of unlimited size to the server.* |
| *GET is a default method* | *POST is not a default. We have to explicitly declare method =”POST” .* |
| *GET requests “do not have a Body” OR “have Empty Body”* | *POST requests “do have a body”* |
| *Hence incase of GET , Form Data will be present in HEADER IN THE FORM OF Query String.* | *In case of POST, Form Data will be present in Body.* |
| *Insecure; because form data get exposed to the outside world* | *Secure; because form data will be present in the Body & hence it will not be exposed to the outside world* |
| *The amount of data sent using GET is restricted because URL can contain Only limited characters.* | *There is no restriction on the amount of data sent using the POST method.* |
| *We cannot send the files using GET* | *We can entire files using POST for Ex: Resume Upload, video files Etc.* |
| *GET requests, by default, they are “idempotent”. i.e we can perform the same operation again & again without any side effects.* | *POST requests are “Non-idempotent” in nature.* |
| *We “can bookmark” the GET requests* | *We “cannot bookmark” POST requests.* |