Web

02 May 2022 09:27

Web Application: It is a software application which executes on top of a *web-server* enabling this server to publish *dynamic content* over its HTTP endpoint so that it can be presented to the user through a *web-browser*.

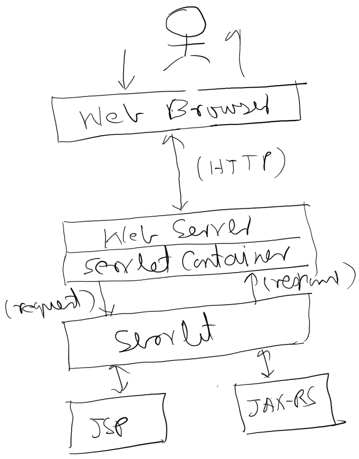
Classic Web Application	**Modern** (AJAX) **Web Application**
The data is acquired and output (HTML) is rendered by the server-side code and this output is transported to the browser for presentation	The data is acquired by server-side code and transported to the browser as objects which are rendered by client-side code for presentation
Input is received by the browser and transported to the server where it is handled by the server-side code.	Input is received by the browser and it is handled by the client-side code by calling server-side code when required.
Application is more secure and is independent of the version and type of browser	Application is less secure and may depend upon the version and type of the browser
Development requires familiarity with server-side frameworks like Spring (Java) and ASP.NET (C#)	Development requires familiarity with server-side frameworks as well as client-side framework like Angular and React
UI is less responsive because every interaction requires its re-rendering on the server	UI is more responsive because interactions do not require its re-rendering.
Commonly used for enterprise applications with multiple web-pages	Commonly used by single page applications (SPA)

REpresentational State Transfer (REST)**:** It is an *architectural style* for exchanging (transferring) the content (state) of some resource in a mutually agreed upon format (representation) between HTTP endpoints. A RESTful service publishes methods (API) of a *stateless* (no instance fields) object using REST so that each of these methods

1. Performs a create, read, update or delete operation identified by the requested verb (POST, GET, PUT or DELETE) on a data-object identified by the requested path.
2. Produces (in response) or Consumes (from request) a data-object using standard media-types (such as application/json) and returns success or failure of its operation using standard HTTP status codes.

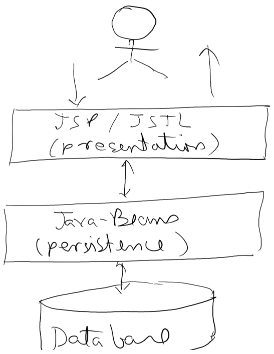
Java Servlet API: It specifies standard support (through jakarta.servlet package) for implementing a server-side Java object known as a servlet required for generating a response to a particular type of request received by the web-server. A web server which supports deployment of Java web-applications (example Glassfish, TomCat, WildFly, WebSphere, WebLogic) includes an environment called a *servlet container* which provides implementation for the servlet API and handles each request received by the server in following steps

1. Get the servlet object mapped to the URL pattern (in WEB-INF/web.xml or through annotation) matching with the requested path if available otherwise initialize a new servlet object from its class (which implements jakarta.servlet.Servlet interface)
2. Invoke the *service* method of the above servlet object passing the current *request* and *response* objects as arguments and when this method returns send the content of the response object to the server so that it can be transported back to its requesting client.



JSP (Java Server Pages)**:** It provides a servlet based support enabling a web-application to publish dynamic content by combining client-side markup (HTML) with server-side elements (expressions and tags) within a single web-page. The URL pattern of a JSP page (*.jsp) is mapped to the built-in JSP servlet which services each request for such a page by executing the server-side elements in that page. The programming model of JSP includes support for

1. **Java Bean** - It is a serializable object whose class supports parameter-less constructor and exposes instance properties using methods which follow standard get-set naming convention.
2. **Tag Extension** - It is a class (which implements jakarta.servlet.jsp.tagext.SimpleTag interface) whose object is used by a JSP page for handling a user-defined server-side markup element (tag) mapped to this class in an XML document called *tag-library descriptor*. JSP includes JSTL (Java Standard Tag Library) which provides implementations for commonly required tag extensions.



JAX-RS (Java API for RESTful Services)**:** It provides a servlet based support enabling a web-application to publish a RESTful service which can be consumed by the client-side code. It includes support for implementing

1. Application configuration class defined with standard annotation which specifies the root path mapped to the servlet responsible for invoking operations identified by the incoming request.
2. Service class containing stateless methods each defined with standard annotations for specifying the verb, the path and the media type of data-object exchanged by this method.