### **Why advance Java?**

* It simplifies the complexity of a building n-tier application.
* Standardizes and API between components and application sever container.
* JEE application Server and Containers provides the framework services.

### Benefits of Advance Java

The four major benefits of advance Java that are, network centric, process simplification, and futuristic imaging standard.

* JEE (advance Java) provides libraries to understand the concept of **Client-Server architecture** for web- based applications.
* We can also work with web and application servers such as **Apache Tomcat** and **Glassfish** Using these servers, we can understand the working of HTTP protocol. It cannot be done in core Java.
* It is also important understand the advance Java if you are dealing with trading technologies like **Hadoop, cloud-native** and **data science**.
* It provides a set of services, **API** and **protocols**, that provides the functionality which is necessary for developing **multi-tiered** application, web-based application.
* There is a number of advance Java frameworks like, **Spring, Hibernate, Struts,** that enables us to develop secure **transaction-based** web applications such as banking application, inventory management application.

**What is web application?**

A web application (or web app) is [application software](https://en.wikipedia.org/wiki/Application_software) that runs on a [web server](https://en.wikipedia.org/wiki/Web_server), unlike computer-based software programs that are run locally on the [operating system](https://en.wikipedia.org/wiki/Operating_system) (OS) of the device. Web applications are accessed by the user through a [web browser](https://en.wikipedia.org/wiki/Web_browser) with an active network connection. These applications are programmed using a [client–server](https://en.wikipedia.org/wiki/Client%E2%80%93server_model) modeled structure—the user ("client") is provided services through an off-site server that is hosted by a third-party. Examples of commonly-used web applications include: [web-mail](https://en.wikipedia.org/wiki/Webmail), [online retail sales](https://en.wikipedia.org/wiki/Online_shopping), [online banking](https://en.wikipedia.org/wiki/Online_banking), and [online auctions](https://en.wikipedia.org/wiki/Online_auction).

The general distinction between a [dynamic web page](https://en.wikipedia.org/wiki/Dynamic_web_page) of any kind and a "web app" is unclear. Web sites most likely to be referred to as "web applications" are those which have similar functionality to a desktop software application, or to a [mobile app](https://en.wikipedia.org/wiki/Mobile_app). [HTML5](https://en.wikipedia.org/wiki/HTML5) introduced explicit language support for making applications that are loaded as web pages, but can store data locally and continue to function while offline.

[Single-page applications](https://en.wikipedia.org/wiki/Single-page_application) are more application-like because they reject the more typical web paradigm of moving between distinct pages with different [URLs](https://en.wikipedia.org/wiki/URL). This is due to individual components being able to be replaced or updated without having to refresh the whole web page, (Jadhar, Sawant and Desbmukh, 2015). Single-page frameworks might be used for speed development of such a web app for a mobile platform as it is able to save bandwidth, as well as the extinction of loading external files, (Jadhar, Sawant and Desbmukh, 2015).

**What are Servlets?**

Servlets | Servlet Tutorial



Servlet technology is used to create a web application (resides at server side and generates a dynamic web page).

Servlet technology is robust and scalable because of java language. Before Servlet, CGI (Common Gateway Interface) scripting language was common as a server-side programming language. However, there were many disadvantages to this technology. We have discussed these disadvantages below.

There are many interfaces and classes in the Servlet API such as Servlet, GenericServlet, HttpServlet, ServletRequest, ServletResponse, etc.

What is a Servlet?

Servlet can be described in many ways, depending on the context.

* Servlet is a technology which is used to create a web application.
* Servlet is an API that provides many interfaces and classes including documentation.
* Servlet is an interface that must be implemented for creating any Servlet.
* Servlet is a class that extends the capabilities of the servers and responds to the incoming requests. It can respond to any requests.
* Servlet is a web component that is deployed on the server to create a dynamic web page.



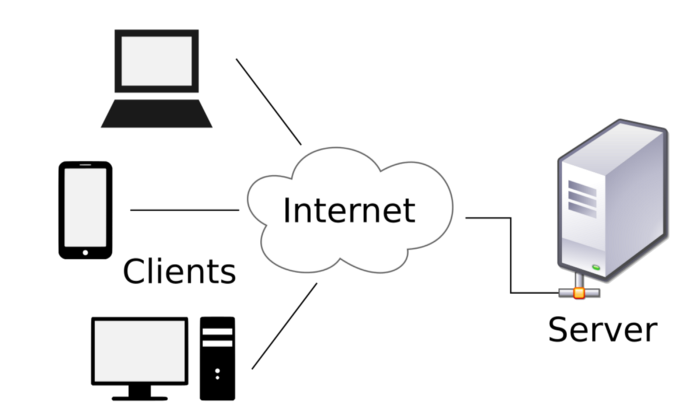
**Difference between Core and Advance Java.**

The **main difference** between Core Java and Advanced Java is that**the Core Java is used to build general applications while the Advanced Java is used to build enterprise level applications.**

[Java](https://pediaa.com/what-is-the-difference-between-java-and-python/#Java)is a general-purpose high-level [programming language](https://pediaa.com/what-is-the-difference-between-markup-language-and-programming-language/#Programming%20Language) that helps to build a variety of applications. Java is popular as it provides platform as it provides various features such as independency, security, multithread support. There are two types of Java as Core Java and Advanced Java. Core Java covers the fundamental concepts in the Java programming language. On the other hand, Advanced Java is the next level after Core Java.

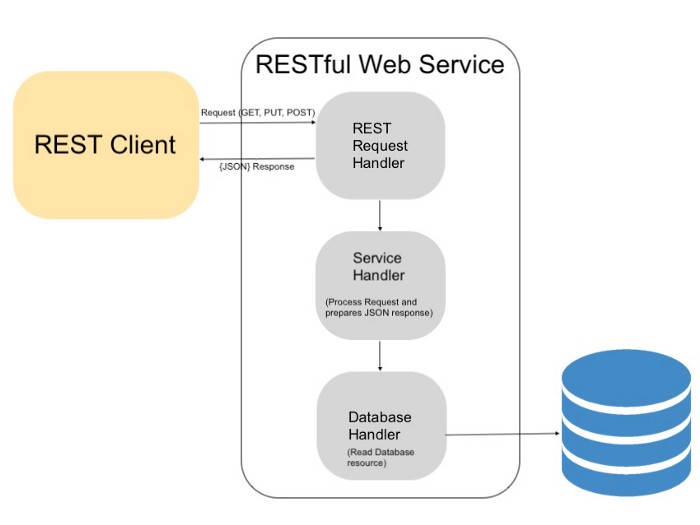
**How do client and server communicate?**

In client-server communication we have first of all two obvious partners: The client and the server.



To understand the communication between these two partners, we need to know some simple topics:

* Requests: Requests are sent from the client in order to ask the server for some data like files, or tell the server about things that happen, like that a user wants to login with his credentials
* Response: A response is sent from the server to the client and is the reaction of the server to a request of the client. This could for example be an authentication result.
* Service: A Service is a specific task that the server provides for the client to use, like downloading image

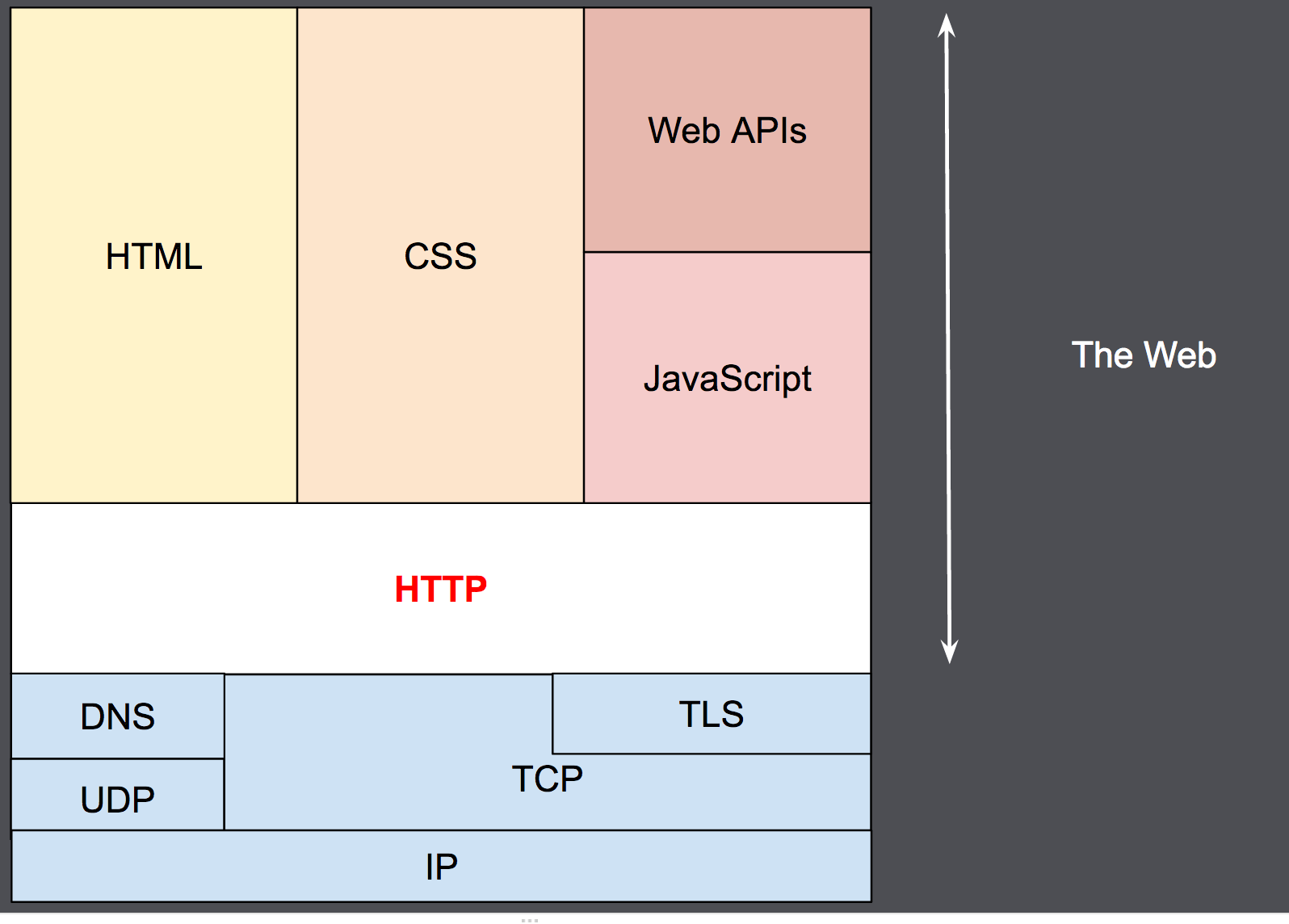


**What is HTTP PROTOCOL?**

HTTP is a [protocol](https://developer.mozilla.org/en-US/docs/Glossary/Protocol) which allows the fetching of resources, such as HTML documents. It is the foundation of any data exchange on the Web and it is a client-server protocol, which means requests are initiated by the recipient, usually the Web browser. A complete document is reconstructed from the different sub-documents fetched, for instance text, layout description, images, videos, scripts, and more.



Clients and servers communicate by exchanging individual messages (as opposed to a stream of data). The messages sent by the client, usually a Web browser, are called requests and the messages sent by the server as an answer are called responses.



Designed in the early 1990s, HTTP is an extensible protocol which has evolved over time. It is an application layer protocol that is sent over [TCP](https://developer.mozilla.org/en-US/docs/Glossary/TCP), or over a [TLS](https://developer.mozilla.org/en-US/docs/Glossary/TLS)-encrypted TCP connection, though any reliable transport protocol could theoretically be used. Due to its extensibility, it is used to not only fetch hypertext documents, but also images and videos or to post content to servers, like with HTML form results. HTTP can also be used to fetch parts of documents to update Web pages on demand.

The exchange of information among servlets of a particular Java web application is known as Servlet Collaboration. This enables passing/sharing information from one servlet to the other through method invocations.

What are the principle ways provided by Java to achieve Servlet Collaboration?

The servlet api provides two interfaces namely:

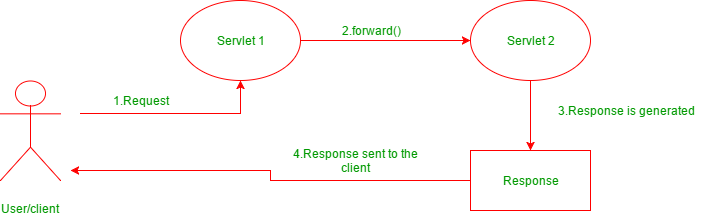
1. javax.servlet.RequestDispatcher
2. javax.servlet.http.HttpServletResponse

These two interfaces include the methods responsible for achieving the objective of sharing information between servlets.

**What is Servlet Collaboration?**

The RequestDispatcher interface provides the option of dispatching the client’s request to another web resource, which could be an HTML page, another servlet, JSP etc. It provides the following two methods:

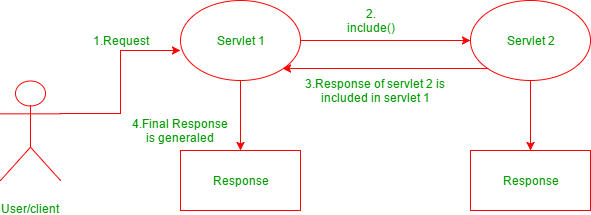
1. public void forward(ServletRequest request, ServletResponse response)throws ServletException, java.io.IOException:

The forward() method is used to transfer the client request to another resource (HTML file, servlet, jsp etc). When this method is called, the control is transferred to the next resource called. On the other hand, the include() method is used to include the content of the calling file into the called file. After calling this method, the control remains with the calling resource, but the processed output is included into the called resource.  
The following diagram explains the way it works:  
  


1. public void include(ServletRequest request, ServletResponse response)throws ServletException, java.io.IOException:

The include() method is used to include the contents of the calling resource into the called one. When this method is called, the control still remains with the calling resource. It simply includes the processed output of the calling resource into the called one.

The following diagram explains how it works:



**Introduction to JSP and need for JSPs**

* It stands for Java Server Pages.
* It is a server side technology.
* It is used for creating web application.
* It is used to create dynamic web content.
* In this JSP tags are used to insert JAVA code into HTML pages.
* It is an advanced version of Servlet Technology.
* It is a Web based technology helps us to create dynamic and platform independent web pages.
* In this, Java code can be inserted in HTML/ XML pages or both.
* JSP is first converted into servlet by JSP container before processing the client’s request.

JSP pages are more advantageous than Servlet:

* They are easy to maintain.
* No recompilation or redeployment is required.
* JSP has access to entire API of JAVA .
* JSP are extended version of Servlet.

Features of JSP

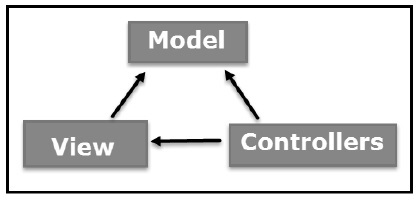
* Coding in JSP is easy :- As it is just adding JAVA code to HTML/XML.
* Reduction in the length of Code :- In JSP we use action tags, custom tags etc.
* Connection to Database is easier :-It is easier to connect website to database and allows to read or write data easily to the database.
* Make Interactive websites :- In this we can create dynamic web pages which helps user to interact in real time environment.
* Portable, Powerful, flexible and easy to maintain :- as these are browser and server independent.
* No Redeployment and No Re-Compilation :- It is dynamic, secure and platform independent so no need to re-compilation.
* Extension to Servlet :- as it has all features of servlets, implicit objects and custom tags

**What is MVC?**

The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development framework to create scalable and extensible projects.

## MVC Components

Following are the components of MVC −



### Model

The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. For example, a Customer object will retrieve the customer information from the database, manipulate it and update it data back to the database or use it to render data.

### View

The View component is used for all the UI logic of the application. For example, the Customer view will include all the UI components such as text boxes, dropdowns, etc. that the final user interacts with.

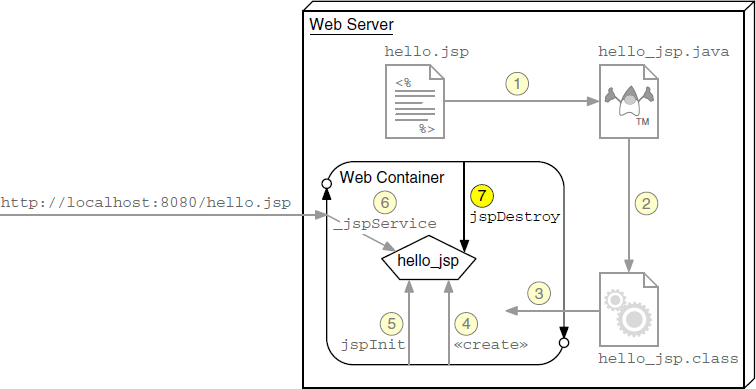
### Controller

Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output. For example, the Customer controller will handle all the interactions and inputs from the Customer View and update the database using the Customer Model. The same controller will be used to view the Customer data.

**Life cycle of JSP**

* Difficulty Level : [Medium](https://www.geeksforgeeks.org/medium/)
* Last Updated : 03 Jul, 2018

A Java Server Page life cycle is defined as the process started with its creation which later translated to a servlet and afterward servlet lifecycle comes into play. This is how the process goes on until its destruction.



Following steps are involved in JSP life cycle:

* Translation of JSP page to Servlet
* Compilation of JSP page(Compilation of JSP into test.java)
* Classloading (test.java to test.class)
* Instantiation(Object of the generated Servlet is created)
* Initialization(jspInit() method is invoked by the container)
* Request processing(\_jspService()is invoked by the container)
* JSP Cleanup (jspDestroy() method is invoked by the container)