

WP-UNIT-3

1.What is Ajax ? Write about its features and applications.

AJAX stands for **A**synchronous **J**avaScript and **X**ML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and JavaScript.

- Ajax uses XHTML for content, CSS for presentation, along with Document Object Model and JavaScript for dynamic content display.
- Conventional web applications transmit information to and from the sever using synchronous requests. It means you fill out a form, hit submit, and get directed to a new page with new information from the server.
- With AJAX, when you hit submit, JavaScript will make a request to the server, interpret the results, and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server.
- XML is commonly used as the format for receiving server data, although any format, including plain text, can be used.
- AJAX is a web browser technology independent of web server software.
- A user can continue to use the application while the client program requests information from the server in the background.
- Intuitive and natural user interaction. Clicking is not required, mouse movement is a sufficient event trigger.
- Data-driven as opposed to page-driven.

Rich Internet Application Technology

AJAX is the most viable Rich Internet Application (RIA) technology so far. It is getting tremendous industry momentum and several tool kit and frameworks are emerging. But at the same time, AJAX has browser incompatibility and it is supported by JavaScript, which is hard to maintain and debug.

AJAX is Based on Open Standards

AJAX is based on the following open standards –

- a) Browser-based presentation using HTML and Cascading Style Sheets (CSS).
 - b) Data is stored in XML format and fetched from the server.
 - c) Behind-the-scenes data fetches using XMLHttpRequest objects in the browser.
 - d) JavaScript to make everything happen.
- Google Maps A user can drag an entire map by using the mouse, rather than clicking on a button.
 - Google Suggest As you type, Google offers suggestions. Use the arrow keys to navigate the results.
 - Gmail Gmail is a webmail built on the idea that emails can be more intuitive, efficient, and useful.
 - Facebook, Twitter, Youtube etc.. **All the browsers support AJAX technologies.**

2. Write about Ajax Technologies

AJAX cannot work independently. It is used in combination with other technologies to create interactive webpages.

JavaScript

- Loosely typed scripting language.
- JavaScript function is called when an event occurs in a page.
- Glue for the whole AJAX operation.

DOM

- API for accessing and manipulating structured documents.
- Represents the structure of XML and HTML documents.

CSS

- Allows for a clear separation of the presentation style from the content and may be changed programmatically by JavaScript

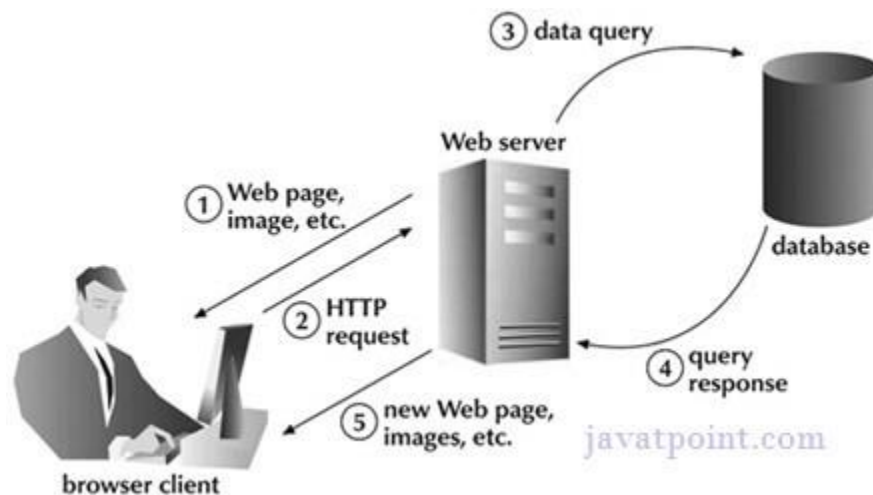
XMLHttpRequest

- JavaScript object that performs asynchronous interaction with the server.

3. How Ajax Works

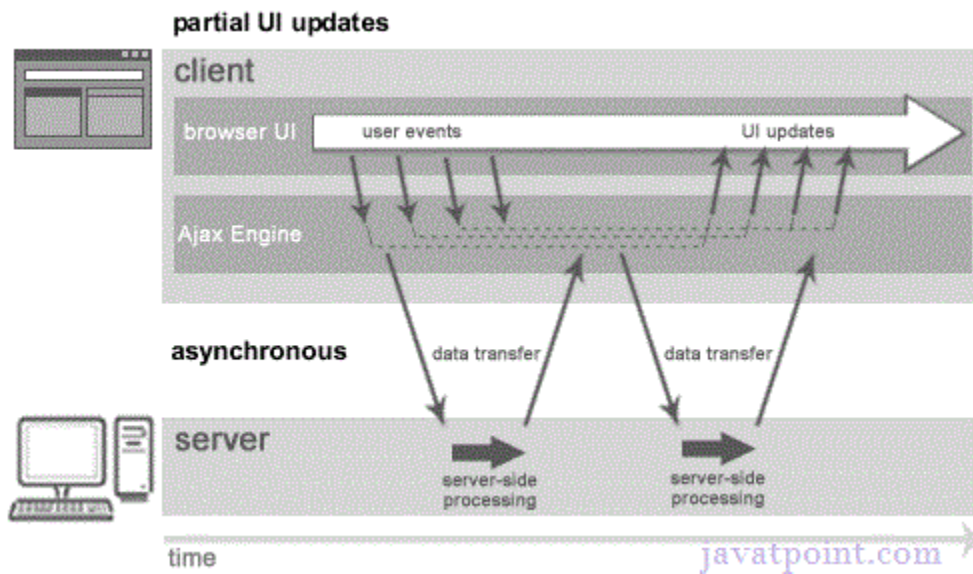
Synchronous (Classic Web-Application Model)

A synchronous request blocks the client until operation completes i.e. browser is unresponsive. In such case, javascript engine of the browser is blocked.



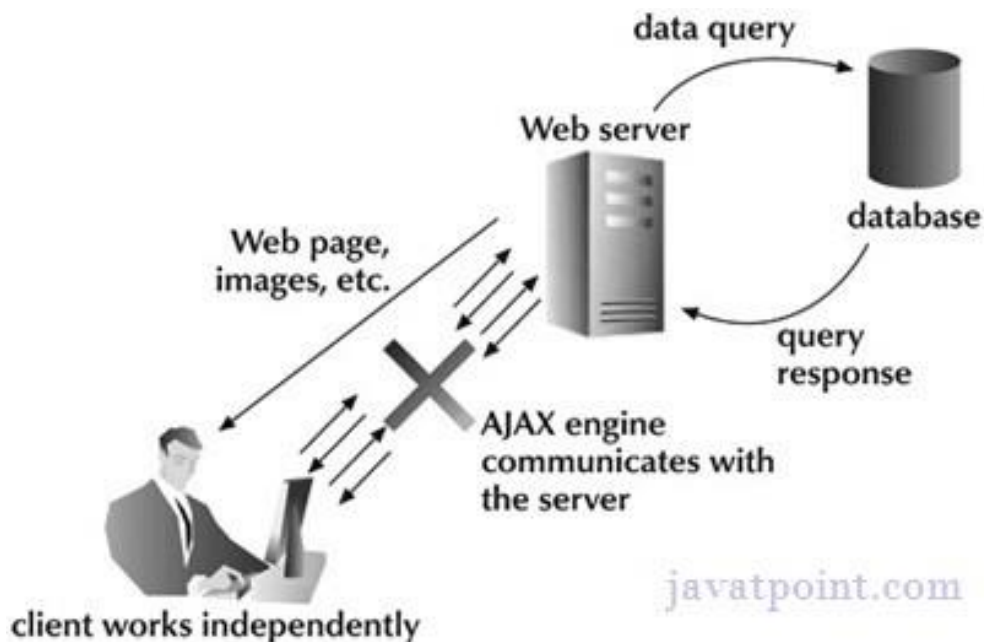
Asynchronous (AJAX Web-Application Model)

An asynchronous request doesn't block the client i.e. browser is responsive. At that time, user can perform another operations also. In such case, javascript engine of the browser is not blocked.



As you can see in the above image, full page is not refreshed at request time and user gets response from the ajax engine.

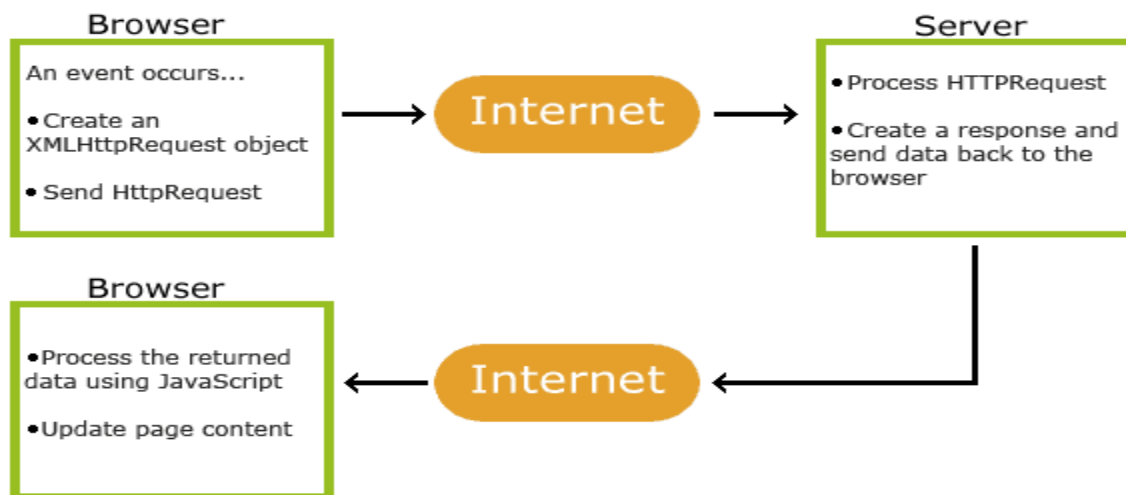
Let's try to understand asynchronous communication by the image given below.



An object of **XMLHttpRequest** is used for asynchronous communication between client and server.

It performs following operations:

- Sends data from the client in the background
- Receives the data from the server
- Updates the webpage without reloading it.



1. An event occurs in a web page (the page is loaded, a button is clicked)
 2. An XMLHttpRequest object is created by JavaScript
 3. The XMLHttpRequest object sends a request to a web server
 4. The server processes the request
 5. The server sends a response back to the web page
 6. The response is read by JavaScript
-
7. Proper action (like page update) is performed by JavaScript

Properties of XMLHttpRequest object

The common properties of XMLHttpRequest object are as follows:

Property	Description
onReadyStateChange	It is called whenever readystate attribute changes. It must not be used with synchronous requests.
readyState	represents the state of the request. It ranges from 0 to 4. 0 UNOPENED open() is not called. 1 OPENED open is called but send() is not called.

	<p>2 HEADERS_RECEIVED send() is called, and headers and status are available.</p> <p>3 LOADING Downloading data;.responseText holds the data.</p> <p>4 DONE The operation is completed fully.</p>
responseText	returns response as text.
responseXML	returns response as XML

The important methods of XMLHttpRequest object are as follows:

Method	Description
void open(method, URL)	opens the request specifying get or post method and url.
void open(method, URL, async)	same as above but specifies asynchronous or not.
void open(method, URL, async, username, password)	same as above but specifies username and password.
void send()	sends get request.
void send(string)	send post request.
setRequestHeader(header,value)	it adds request headers.

4.Example using AJAX

The XMLHttpRequest Object

All modern browsers support the XMLHttpRequest object.

The XMLHttpRequest object can be used to exchange data with a server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

Create an XMLHttpRequest Object

All modern browsers (Chrome, Firefox, IE7+, Edge, Safari Opera) have a built-in XMLHttpRequest object.

Syntax for creating an XMLHttpRequest object: **variable = new XMLHttpRequest();**

The onreadystatechange Property

With the XMLHttpRequest object you can define a function to be executed when the request receives an answer.

The function is defined in the **onreadystatechange property of the XMLHttpRequestResponse** object:

The onreadystatechange Property

The **readyState** property holds the status of the XMLHttpRequest.

The **onreadystatechange** property defines a function to be executed when the readyState changes.

The **status** property and the **statusText** property holds the status of the XMLHttpRequest object.

Property	Description
onreadystatechange	Defines a function to be called when the readyState property changes
readyState	Holds the status of the XMLHttpRequest. 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
status	200: "OK" 403: "Forbidden" 404: "Page not found" For a complete list go to the Http Messages Reference
statusText	Returns the status-text (e.g. "OK" or "Not Found")

```

<!DOCTYPE html>
<html>
<body>

<h1>The XMLHttpRequest Object</h1>

<p id="demo">Let AJAX change this text.</p>

<button type="button" onclick="loadDoc()">Change Content</button>
<script>
function loadDoc() {
  var xhttp;
  if (window.XMLHttpRequest) {
    // code for modern browsers
    xhttp = new XMLHttpRequest();
  } else {
    // code for IE6, IE5
    xhttp = new ActiveXObject("Microsoft.XMLHTTP");
  }
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML = this.responseText;
    }
  };
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();
}
</script>

</body>
</html

```

5. What is WebServer? Give Examples.

Web server" can refer to hardware or software, or both of them working together.

On the hardware side, a web server is a computer that stores web server software and a website's component files (e.g. HTML documents, images, CSS stylesheets, and JavaScript files). It is connected to the Internet and supports physical data interchange with other devices connected to the web.

On the software side, a web server includes several parts that control how web users access hosted files, at minimum an HTTP server. An HTTP server is a piece of software that understands URLs (web addresses) and HTTP (the protocol your browser uses to view webpages). It can be accessed through the domain names (like mozilla.org) of websites it stores, and delivers their content to the end-user's device.

A web server processes incoming network requests over HTTP and several other related protocols. The primary function of a web server is to store, process and deliver web pages to clients. The communication between client and server takes place using the Hypertext Transfer Protocol (HTTP).

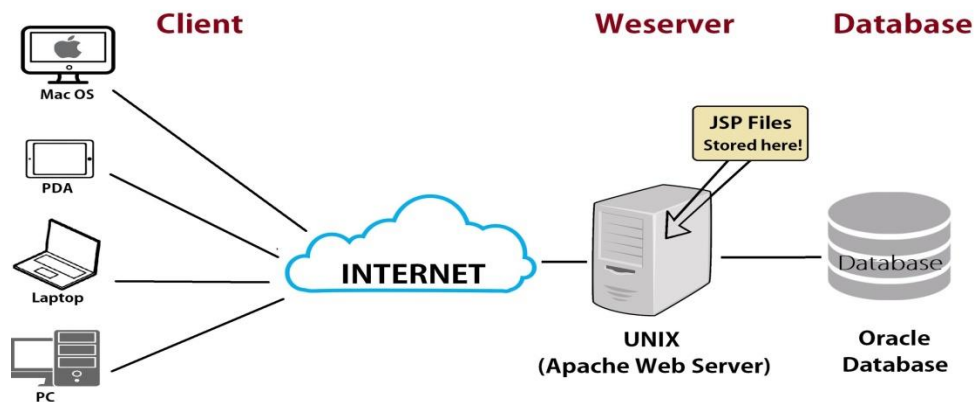
At the most basic level, whenever a browser needs a file which is hosted on a web server, the browser requests the file via HTTP. When the request reaches the correct web server (hardware), the HTTP server (software) accepts request, finds the requested document (if it doesn't then a 404 response is returned), and sends it back to the browser, also through HTTP.

Basic representation of a client/server connection through HTTP

To publish a website, you need either a static or a dynamic web server.

A static web server, or stack, consists of a computer (hardware) with an HTTP server (software). We call it "static" because the server sends its hosted files "as-is" to your browser.

A dynamic web server consists of a static web server plus extra software, most commonly an application server and a database. We call it "dynamic" because the application server updates the hosted files before sending them to your browser via the HTTP server.



Various services provided by the web server are:

1. **Cost Efficient:** Web server is the most cost efficient method to use, maintain and upgrade. Traditional desktop software costs companies a lot in terms of finance. On the other hand, it is available at much cheaper rates. Besides, there are many one-time-payment, pay-as-you-go and other salable options available, which makes it very reasonable for the company.
2. **Resource Sharing:** Web Server has the capability to store unlimited information such as Google Drives, Cloud computing etc. The space where the data can be stored is shared by the other users at the same time like hard disk can be shared on physical network as LAN.
3. **Data Sharing:** With the help of web servers one can easily access the information from anywhere, where there is an Internet connection using Google docs such as Documents, Excel sheets, Drawings, power point presentations etc.
4. **Backup and Recovery:** As all the data now a days is stored on web servers, backing it up and restoring the same is relatively much easier than storing the same on a physical device. Hence, the entire process of backup and recovery is much simpler than other traditional methods of data storage.

6.What are the types of Servers.

1. **Mail Server:** Mail Server provides a centrally-located pool of disk space for network users to store and share various documents in the form of emails. Since, all the data is stored in one location, administrators need only backup files from one computer.
2. **Application Server:** An application server acts as a set of components accessible to the software developer through an API defined by the platform itself. For Web applications. These components are usually performed in the same running environment as its web server(s), and their main job is to support the construction of dynamic pages.
3. **File Transfer Protocol (FTP) Server:** FTP uses separate control and data connections between the client and the server. FTP users may authenticate themselves in the form of a username and password. But can connect anonymously if the server is configured to allow it. For secure transmission username and password must be encrypted using FTP and SSL.
4. **Database Server:** A database server is a computer program that provides database services to other computer programs or computers using client-server functionality, and some DBMSs (e.g., Mysql) depend on the client-server model for database access. Such a server is accessed either through a “front end” running on the user’s computer where the request is made or the “back end” where the request is served such as data analysis and storage.
5. **Domain Name System (DNS) Server:** A name server is a computer server that hosts a network service for providing responses to queries. It maps a numeric identification or addressing component. This service is performed by the server in response to a network service protocol request.

7.Discuss about IIS(INTERNET INFORMATION SERVER) webserver.

A web server processes incoming network requests over HTTP and several other related protocols. The primary function of a web server is to store, process and deliver web pages to clients. The communication between client and server takes place using the Hypertext Transfer Protocol (HTTP).

IIS is a web server that runs on the Microsoft .NET platform on the Windows OS.

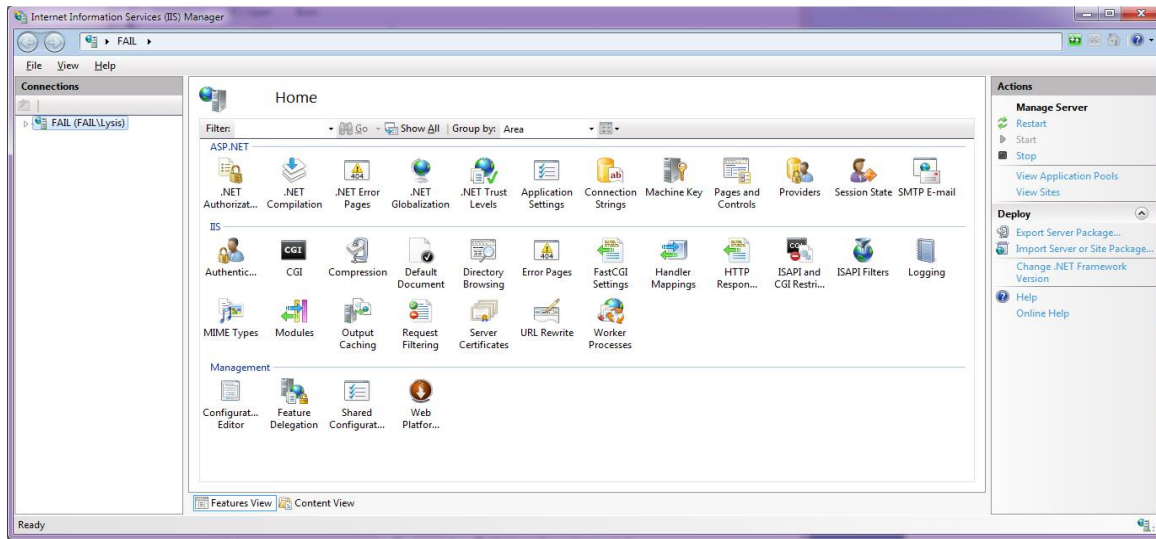
The two main process models for web servers are to either handle all requests on a single thread, or to spawn a new thread for each request. Although the single-thread model (Node.js, for example) has some worker threads available, it typically only uses them for certain kinds of work, such as file system access. **The thread-per-request model that uses will grab a thread from a thread pool for each request.**

IIS is rich with features.

- Most commonly, IIS is used to host ASP.NET web applications and static websites. It can also be used as an FTP server, host WCF services, and be extended to host web applications built on other platforms such as PHP.
- There are built-in authentication options such as Basic, ASP.NET, and Windows auth. The latter is useful if you have a Windows Active Directory environment—users can be automatically signed into web applications using their domain account.
- Other built-in security features include TLS certificate management and binding for enabling HTTPS and SFTP on your sites, request filtering for whitelisting or blacklisting traffic, authorization rules, request logging, and a rich set of FTP-specific security options.

- One key feature of IIS is the application pool. We'll have to take a closer look at the application pool, as it's a critical component of the IIS process model.

The main IIS configuration utility is found in the Windows Control Panel. It's in the "System and Security" section in the "Administrative Tools" option. The below image gives you a quick overview of the IIS 7.5 main manager window:



Virtual Directories

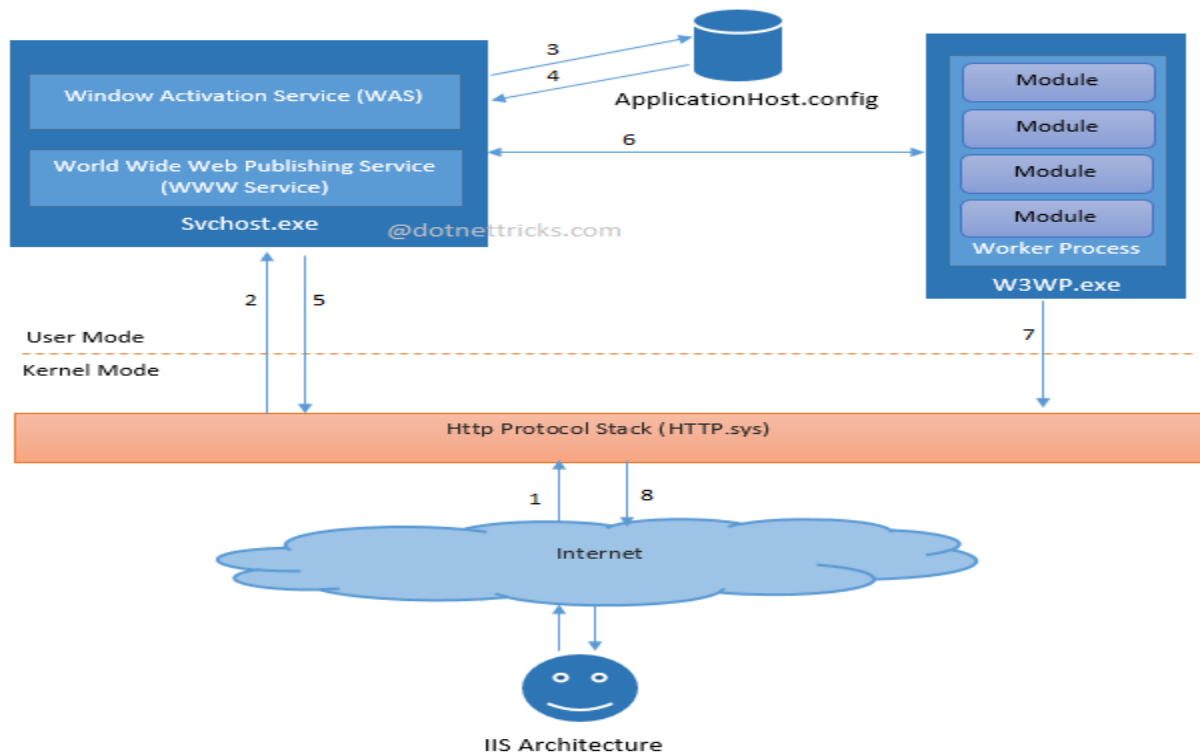
- IIS allows you to create sites, applications, and virtual directories to share information with users over the Internet or internally on an intranet such as a home network.
- A virtual directory is a name that you specify in IIS and that maps to a physical directory on a server similar to how DNS maps a URL to an IP address.

Ports

- Typically your server will use port 80 for HTTP traffic however this can be adjusted to meet your needs or the needs of another application on your computer.
- You can find a full list of ports and the purpose they each serve here. Changing a port within IIS 7 to 10 is simple. First Open Internet Information Services Manage

IIS Architecture:

IIS has two main layers - Kernel Mode and User Mode. The Kernel Mode contains the HTTP.SYS and User Mode contains WAS and W3 service. The subsection of both are shown in fig.



The above diagrams shows the flow of an HTTP request in process. The request-processing flow is described as:

1. An HTTP request first goes to HTTP.sys and now, HTTP.SYS is responsible for passing the request to a particular application pool.
2. HTTP.sys contacts to WAS and WAS requests configuration information from the xml file.
3. The configuration information is sent to WWW service receives.
4. The WWW service uses the configuration information to configure HTTP.sys.
5. Configured HTTP.sys contacts to WAS and now, WAS starts a worker process for the application pool to which the request was made.
6. The worker process processes the request and returns a response to HTTP.sys. The request is passed through an ordered series of module in the processing pipeline.

Role of HTTP.sys in IIS

HTTP.SYS is the part of kernel mode of IIS. Every client request is passes through the kernel mode, Http.sys then makes a queue for each and individual application pool based on the request. Whenever we create any application pool IIS automatically

registers the pool with HTTP.sys to identify the particular during request processing. It provides the following services in IIS:

1. Routing HTTP requests to the correct request queue.
2. Caching of responses in kernel mode.
3. Performing all text-based logging for the WWW service.
4. Implementing quality of service functionality, which includes connection limits, connection timeouts, queue-length limits, and bandwidth throttling.

ISAPI Filter

ISAPI filters are DLL files that can be used to modify and enhance the functionality provided by IIS. ISAPI filters always run on an IIS server, filtering every request until they find one they need to process.

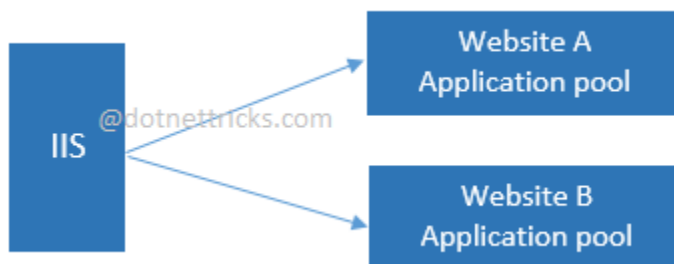
ISAPI filters can be registered with IIS to modify the behavior of a server. It can perform the following tasks:

1. Change request data (URLs or headers) sent by the client
2. Control which physical file gets mapped to the URL
3. Control the user name and password used with anonymous or basic authentication
4. Modify or analyze a request after authentication is complete
5. Modify a response going back to the client
6. Run processing when a request is complete
7. Run processing when a connection with the client is closed
8. Perform special logging or traffic analysis.
9. Handle encryption and compression.

Different Security Settings Available in IIS

IIS provides a variety of authentication schemes:

1. Anonymous (enabled by default)
2. Basic
3. Digest
4. Integrated Windows authentication (enabled by default)
5. Client Certificate Mapping



Key Points About Application Pool

1. Provides isolation between different web applications.
2. Every web application has individual worker process.
3. Improve manageability of web application.
4. Provides better performance.

In IIS 6.0, **WWW Service** manages the following main areas in IIS:

- HTTP administration and configuration
- Process management
- Performance monitoring

Windows Process Activation Service (WAS)

In IIS 7 and later, Windows Process Activation Service (WAS) manages application pool configuration and worker processes instead of the WWW Service. This enables you to use the same configuration and process model for HTTP and non-HTTP sites.

WAS manages application pools and worker processes for both HTTP and non-HTTP requests. When a protocol listener picks up a client request, WAS determines if a worker process is running or not. If an application pool already has a worker process that is servicing requests, the listener adapter passes the request onto the worker process for processing.

An **IIS Worker Process (w3wp.exe)** handles the web requests sent to the IIS web server for the configured IIS application pool. IIS application pools also provide a bunch of advanced settings. These impact the behavior of w3wp and your IIS worker process.

IIS application pools also provide a bunch of advanced settings. These impact the behavior of w3wp and your IIS worker process. Including things like what Windows user account it runs as, auto restarting of the process, auto shutdown, and more. It is also possible for one IIS application pool to create multiple IIS worker processes in what is called a web garden.

8. Discuss about Apache Tomcat WebServer.

Apache Tomcat is a webcontainer which allows to run servlet and JavaServer Pages (JSP) based web applications. Most of the modern Java web frameworks are based on servlets, e.g. JavaServer Faces, Struts, Spring.

Apache Tomcat also provides by default a HTTP connector on port 8080, i.e., Tomcat can also be used as HTTP server. But the performance of Tomcat is not as good as the performance of a designated web server, like the Apache HTTP server.

Tomcat is used in conjunction with Apache HTTP Server where Apache HTTP Server attends static content like html, images etc., and forwards the requests for dynamic content to Tomcat. This is because Apache HTTP Server supports more advanced options than that of Tomcat.

Jasper 2

Jasper is the JSP Engine for Tomcat. Jasper is responsible for parsing JSP files and compilation of JSP's Java code as servlets.

Jasper is capable of background compilation, which means if any changes are made to JSP files, then the older versions of those JSP files are still retained by the server, until the updated JSP files are recompiled.

Installing Apache Tomcat on Ubuntu

To install Tomcat on Ubuntu, you could use command line interface and run the following command :

```
~$ sudo apt-get install tomcat8
```

Start Apache Tomcat

Once you install Tomcat, it is started automatically.

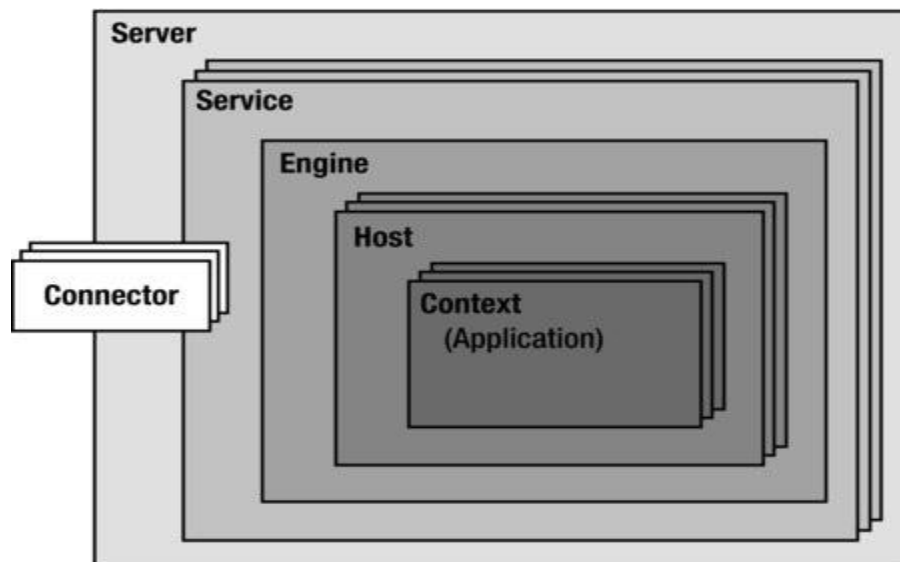
In case if you have stopped it manually, and would like to start Apache Tomcat again, open a terminal and run the following command :

```
~$ sudo /etc/init.d/tomcat8 start
```

Deploying Dynamic Web-Applications with Apache Tomcat

.war is the format of the web application that Apache Tomcat Server could deploy. If you are building a web application using an IDE like Eclipse, you could export the application as a WAR file

Tomcat Architecture



Context

A Context is the innermost element of a group of Tomcat components called containers, and it **represents a single web application**. Tomcat automatically instantiates and configures a standard context upon loading your application. As part of the configuration, Tomcat also processes the properties defined in the `\WEB-INF\web.xml` file of your application folder and makes them available to the application.

Connector

A Connector handles communications with the client. There are multiple connectors available with Tomcat e.g. HTTP connector for most of the HTTP traffic and AJP connector which implements the AJP protocol used when connecting Tomcat to another web server such as Apache HTTPD server.

The default configuration of Tomcat includes a connector to handle HTTP communication. By default, this connector waits for requests coming through port **8080**. This is why the URLs of our examples always start with `http://localhost:8080/`. Note that the requests for all applications go through a single instance of this connector. Each new request causes the instantiation of a new thread that remains alive within the connector for the duration of the request. Tomcat often refer to this connector as "**Coyote**". AJP connector lets Tomcat only handle dynamic web pages and lets a pure HTML server (e.g., the Apache Web Server) handle the requests for static pages. This maximizes the efficiency with which the requests are handled.

Host

A Host is an association of a network name, e.g. `www.yourdomain.com`, to the Tomcat server. A host can contain any number of contexts (i.e. applications). You can define several hosts on the same server. For example, if you have registered the domain `yourdomain.com`, you can define host names such as `w1.yourdomain.com` and `w2.yourdomain.com`. Keep in mind that it will only be accessible from the Internet if a domain name server maps its name to the IP address of your computer. The default configuration of Tomcat includes the host named **localhost**

Engine

An Engine represents request processing pipeline for a specific Service. As a Service may have multiple Connectors, the Engine receives and processes all requests from

these connectors, handing the response back to the appropriate connector for transmission to the client.

An engine must contain one or more hosts, one of which is designated as the default host. The default Tomcat configuration includes the engine Catalina, which contains the host localhost (obviously designated to be the default host because it is the only one). The Catalina engine handles all incoming requests received via the HTTP connector and sends back the corresponding responses. It forwards each request to the correct host and context on the basis of the information contained in the request header.

Service

A Service is an intermediate component which lives inside a Server and ties one or more Connectors to exactly one Engine. Tomcat's default configuration includes the service Catalina which associates the HTTP and AJP connectors to the Catalina engine. Accordingly, Connector and Engine are subelements of the Service element.

Server

The Server is the top component and represents an instance of Tomcat. It can contain one or more services, each with its own engine and connectors.

Working of Tomcat

The lifecycle of a typical servlet running on Tomcat might look something like this:

1. Tomcat receives a request from a client through one of its connectors.
2. Tomcat maps this request to the appropriate Engine for processing. These Engines are contained within other elements, such as Hosts and Servers, which limit the scope of Tomcat's search for the correct Engine.
3. Once the request has been mapped to the appropriate servlet, Tomcat checks to see if that servlet class has been loaded. If it has not, Tomcat compiles the servlet into Java bytecode, which is executable by the JVM, and creates an instance of the servlet.
4. Tomcat initializes the servlet by calling its init method. The servlet includes code that is able to read Tomcat configuration files and act accordingly, as well as declare any resources it might need, so that Tomcat can create them in an orderly, managed fashion.
5. Once the servlet has been initialized, Tomcat can call the servlet's service method to process the request, which will be returned as a response.
6. During the servlet's lifecycle, Tomcat and the servlet can communicate through the use of listener classes, which monitor the servlet for a variety of state changes. Tomcat can retrieve and store these state changes in a variety of ways, and allow other servlets access to them, allowing state to be maintained and accessed by various components of a given context across the span of a single or

multiple user sessions. An example of this functionality in action is an e-commerce application that remembers what the user has added to their cart and is able to pass this data to a checkout process.

7. Tomcat calls the servlet's destroy method to smoothly remove the servlet. This action is triggered either by a state change that is being listened for, or by an external command delivered to Tomcat to undeploy the servlet's Context or shut down the server.

9.Features of web hosting provider..

The first thing to consider when starting your website is to choose a web hosting provider (if you already have a web domain¹). The web hosting provider provides the web **space** (i.e. special computers called web servers) where your website files are stored, as well as the **technologies** and **services** needed for your website to be viewed on the Internet.

Add-on services provided by a web hosting provider typically include data backup, firewall protection, technical support, email services, domain name registration, website building tools, and applications.

- Round the clock support
- Unlimited² storage
- Unlimited (professional) email addresses
(with Unlimited Autoresponders, Mail Forwards, Email Aliases, Mailing Lists, etc.)
- 1-click installation
(WordPress, Joomla, Magento, Drupal, phpBB, Gallery and many other CMSs)
- Latest cPanel

10. Discuss about to Hosting a Website.

Step 1: Decide What Type of Website You Want

You will typically find 2 types of websites:

Static or Basic Websites: Static websites are simple websites with one or more web pages (called HTML pages). You can build them on your computer with software like Dreamweaver and then upload the pages to your host's server using any FTP software (such as FileZilla). Since they cannot be modified dynamically, such websites are called static websites. Static websites are cheaper than dynamic websites (below) but come with limited functionality and no option for e-commerce or interactivity.

Dynamic Websites: Dynamic websites contain information that changes, depending on the time of day, the viewer and other factors. They make use of both client-side and server-side scripts to create and update content. Client-side scripts, which run on a user's computer, are mainly used for appearance and interaction purposes. Server-side scripts, which reside on a server and are extensively used by E-commerce and social networking sites, allow users to have individual accounts and provide a customized response for each user. Examples of dynamic websites include blogs, forums, photo galleries and e-commerce sites.

Step 2: Choose Your Hosting Server

Unlike static HTML sites which can be hosted on most web servers, when it comes to web applications, there are basically two types of hosting platforms. Depending on your hosting needs and what you're most comfortable with, you can choose from:

- **Linux Hosting**, which allows running scripts written in PHP, Perl, Python and other Unix-originated languages, and usually supports PostgreSQL and MySQL databases. This is the most commonly used system today.
- **Windows Hosting**, which allows running ASP scripts utilizing .NET and other Microsoft technologies, and supports Microsoft SQL Server and Access database.

Step 3: Select Your Web Hosting Plan

You will typically find a wide range of services in web hosting, such as:

- **Shared Hosting:** In shared hosting, you get to share the physical server with other website owners. However, you will have your own separate account (secured with login credentials). Shared hosting is very affordable because the cost of operating the server is shared between you and the other website owners.
- **VPS Hosting (Virtual Private Server Hosting):** In VPS hosting, every website is stored on a very powerful server that is divided into several virtual compartments. The server software is configured separately so that each unit can function independently. It should be your preferred option if you have high-security concerns but don't want to invest in a faster (but costlier) dedicated server.
- **Dedicated Hosting:** Dedicated hosting offers you an entire server for yourself, thereby making it faster, more secure...and costlier. It is the ideal solution for larger businesses and high-traffic websites because it allows for maximum customization, configuration, installation and flexibility.
- **Cloud Hosting:** Cloud hosting allows multiple virtual servers (clouds) to work together to host a website or a group of websites. It offers unlimited ability to handle sudden traffic spikes. A cloud-hosted website is not limited to a single server, and the resources allocated to it can shrink or expand dynamically, depending on how much traffic you get. It's a great option for large websites, including e-commerce websites, newsletters and blogs.

Step 4: Change Your DNS Address

After you have purchased your web hosting, you will get Name Servers (also known as Domain Name Servers or DNS) – which is the Internet’s equivalent of a phone book that contains IP Addresses. To get your website up and working, you will need to change the Name Servers of your domain. It’s a simple but mandatory step for you to get started.

1. Go to your Domain Control Panel via <http://manage.hostgator.in/customer>.
2. Enter your registered **email address** and **password**.
3. Click on the **Domain Name** for which you need to change the Name Servers.
4. In the Domain Registration section, click on the **Name Servers** option.
5. Replace the existing Name Servers with the ones **provided by your current web host**, and click on the **Update Name Servers** button.

Step 5: Upload Your Website

You can now upload your website to your account by connecting to the server using either cPanel’s **File Manager** or **FTP Client** (such as FileZilla) – after which your website will go live.

Step 6: share the DNS address of your website to client.