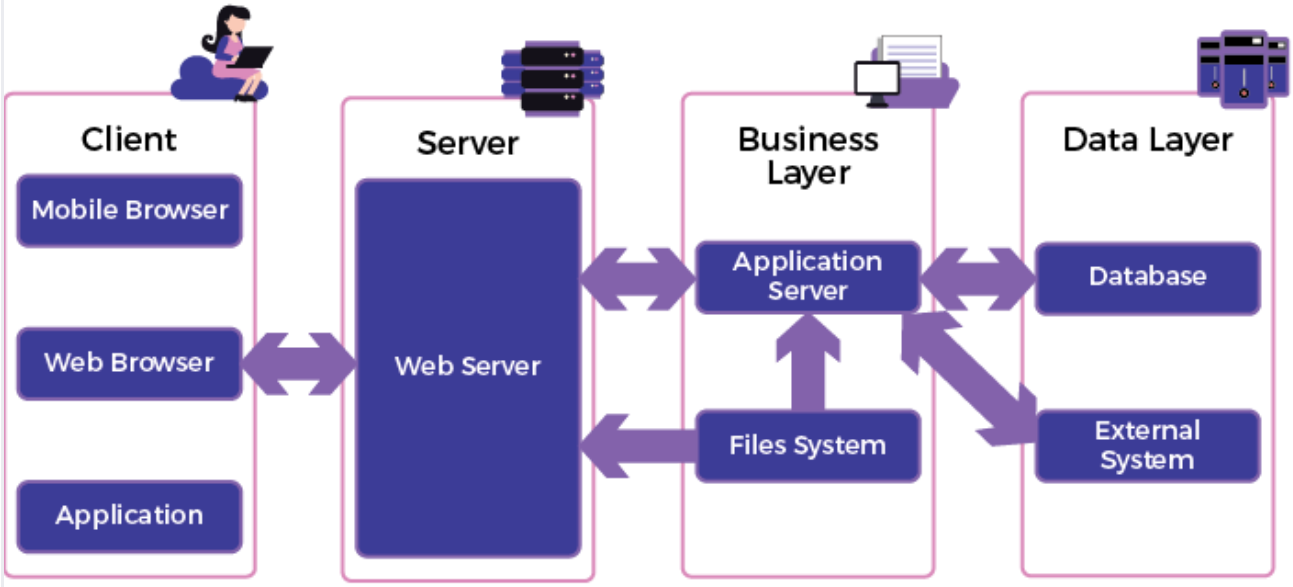
What is a Web Server?

A Web Server is a software application which handles HTTP requests sent by the HTTP client, like web browsers, and returns web pages in response to the clients. Web servers usually deliver HTML documents along with images, style sheets, and scripts.

Most of the web servers support server-side scripts. They use scripting languages and redirect the task to an application server which retrieves data from a database and performs complex logic. Finally a result is sent to the HTTP client through the web server.

Web Application Architecture

A Web application is usually divided into four layers:



- **Client** : This layer consists of web browsers, mobile browsers or applications which can make HTTP requests to the web server.
- **Server** : This layer has the web server which can intercept the requests made by the clients and pass them the response.
- **Business** : This layer contains the application server which is utilized by the web server to do the required processing. This layer interacts with the data layer via the database or some external programs.
- **Data** : This layer contains the databases or any other source of data.

Creating a Web Server using Node

Node.js provides an **HTTP** module which can be used to create an HTTP client of a server. The following is the bare minimum structure of the HTTP server which listens to the 8081 port.

Create a js file named server.js :

File: server.js

```
var http = require('http');
var fs = require('fs');
var url = require('url');

// Create a server
http.createServer( function (request, response) {
   // Parse the request containing file name
   var pathname = url.parse(request.url).pathname;

   // Print the name of the file for which request is made.
   console.log("Request for " + pathname + " received.");

   // Read the requested file content from file system
   fs.readFile(pathname.substr(1), function (err, data) {
      if (err) {
         console.log(err);

         // HTTP Status: 404 : NOT FOUND
         // Content Type: text/plain
         response.writeHead(404, {'Content-Type': 'text/html'});
      } else {
         //Page found
         // HTTP Status: 200 : OK
         // Content Type: text/plain
         response.writeHead(200, {'Content-Type': 'text/html'});

         // Write the content of the file to response body
```

```
            response.write(data.toString());
        }

        // Send the response body
        response.end();
    });
}).listen(8081);

// Console will print the message
console.log('Server running at http://127.0.0.1:8081/');
```

Creating a Web Server using Node

Next, let's create the following HTML file named index.html in the same directory where you have created server.js.

File: index.html

```
<html>
   <head>
      <title>Sample Page</title>
   </head>

   <body>
      Hello World!
   </body>
</html>
```

Now let us run the server.js to see the result:

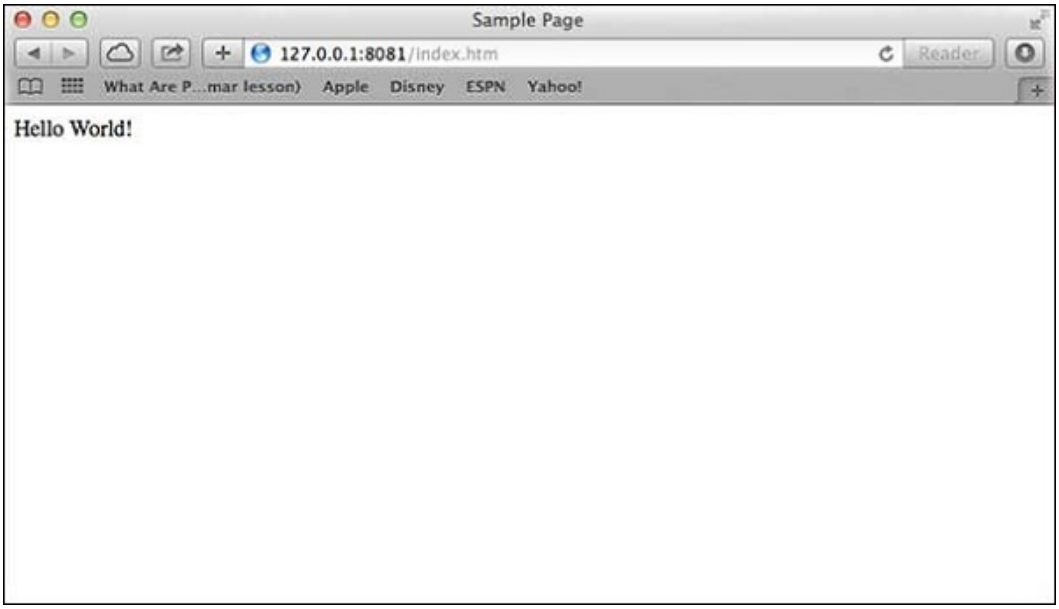
```
$ node server.js
```

Verify the Output

```
Server running at http://127.0.0.1:8081/
```

Make a request to Node.js's server.

Open http://127.0.0.1:8081/index.htm in any browser and you should see our "Hello World" displayed on the page.



Verify the Output at server end:

```
Server running at http://127.0.0.1:8081/
Request for /index.htm received.
```

Creating Web client using Node

A web client can be created using **HTTP** module. Let's check the following example.

Create a .js file named client.js:

File: client.js

```
var http = require('http');

// Options to be used by request
var options = {
   host: 'localhost',
   port: '8081',
   path: '/index.html'
};

// Callback function is used to deal with a response
var callback = function(response) {
```

```
   // Continuously update stream with data
   var body = '';
   response.on('data', function(data) {
      body += data;
   });

   response.on('end', function() {
      // Data received completely.
      console.log(body);
   });
}
// Make a request to the server
var req = http.request(options, callback);
req.end();
```

Now run the client.js from a different command terminal other than server.js to see the result:

```
$ node client.js
```

Verify the output:

```
<html>
   <head>
      <title>Sample Page</title>
   </head>

   <body>
      Hello World!
   </body>
</html>
```

Verify the output at the server's end:

```
Server running at http://127.0.0.1:8081/
Request for /index.htm received.
```