

## Lecture 8

The last class was focussed on a brief introduction to the networking terminologies like protocol, IP address, MAC which might be required when you would be handling servers. This class is aimed at giving a hands on experience of working to server and client **request** and **response**.

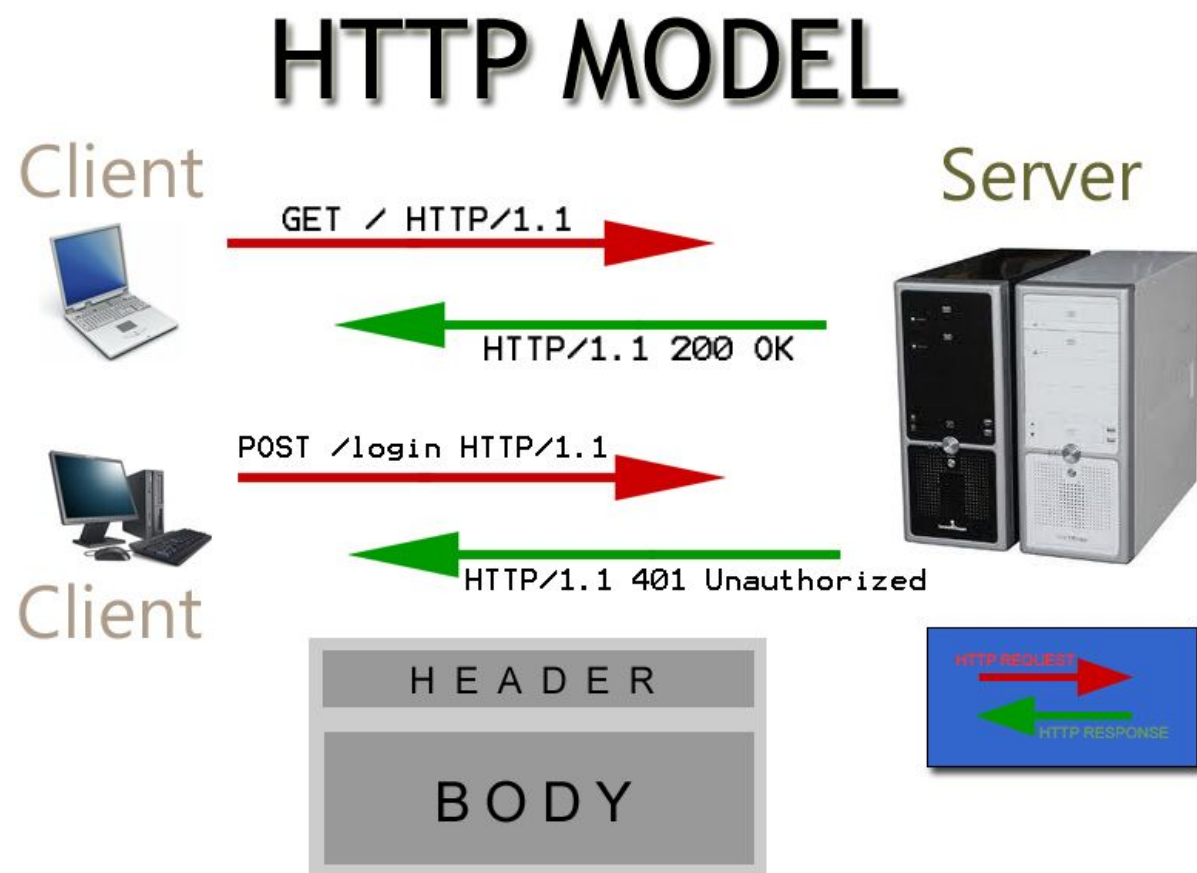
### Request and Response

**Rule 0** : Any device that sends a request is client and one that responds is the server for the particular HTTP communication.

Request and response work like a table tennis table. One side requesting a resource and the other side responding with a resource

In the HTTP Model,

Generally the web browser is the client sending GET / POST / PUT / DELETE requests to the server and the server responds with either the HTML page or JSON.



Now we would try to create a node server, send a request print the request on the server, send a response and print the response on the client.

First create a new directory :

As taught in the previous classes, create a node server running on port 3000.

```
express myapp
npm install
npm start
```

This would start a server at the localhost at port 3000. Having address <http://localhost:3000/>

With the server ready, let us create a mock client.

Make a new directory named client.

Run the command,

```
npm install request
```

This would install request in the current directory. Request is a node module that allows to send HTTP requests to a server.

Open a terminal window and go to the client directory in the terminal.

Open the node interpreter by typing

```
node
```

Once you are in the node interpreter, type the following code

```
var request = require('request');
var req = function() {

    request('http://localhost:3000/', function (error, response, body) {
        console.log('error:', error); // Print the error if one occurred
        console.log('statusCode:', response && response.statusCode); // Print
the response status code if a response was received
        console.log('body:', body); // Print the HTML for the Google homepage.
    });

}
```

The response in the code prints the response header received. Usually the data that the server intended to send would be present in the **response.body** property of the **response object**.

The body brings the HTML page that has been sent by the server to be displayed by the browser.

Now to print the request on the server side,

Go to the routes in the node server application that you have running and change the code in the index file to print the request.

```
var express = require('express');
var router = express.Router();

/* GET home page. */
router.get('/', function(req, res, next) {
  console.log(req);
  res.body = 'Message';
  res.send('Hello World');
});

module.exports = router;
```

This would start printing the request on the server console whenever you send a new request from the client by typing req() in the client. Now with this modified code you would be sending Hello World as the body instead of the HTML code being sent earlier.

You can experiment a bit more with the request response model to get a better understanding of the HTTP request response model.