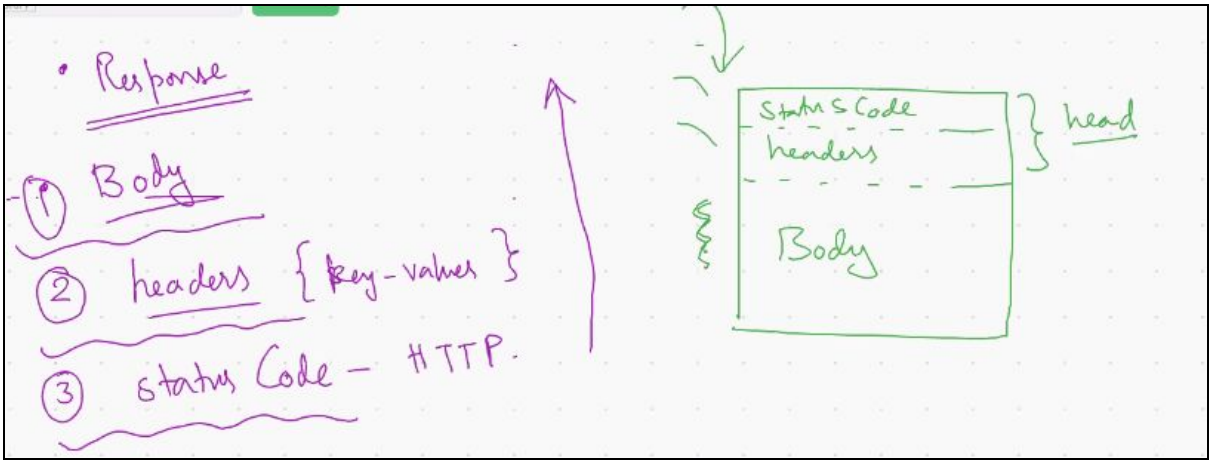
Date: 27th Nov 2020

Title: Rest apis
Topics covered:

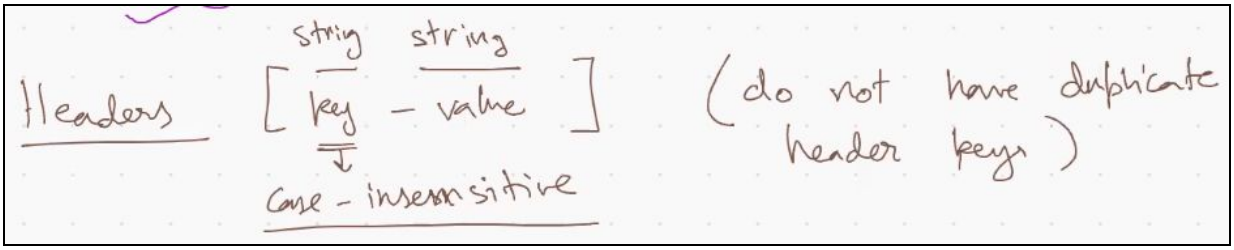
- **Response apis**
  - **Body**
  - **Headers**
  - **Status Code-http**
- **1st Assessment question on backend node server**
- **Rest api**

Response is sent in this order: **Status Code -> Headers -> Body**



First status code is sent then the headers then the body

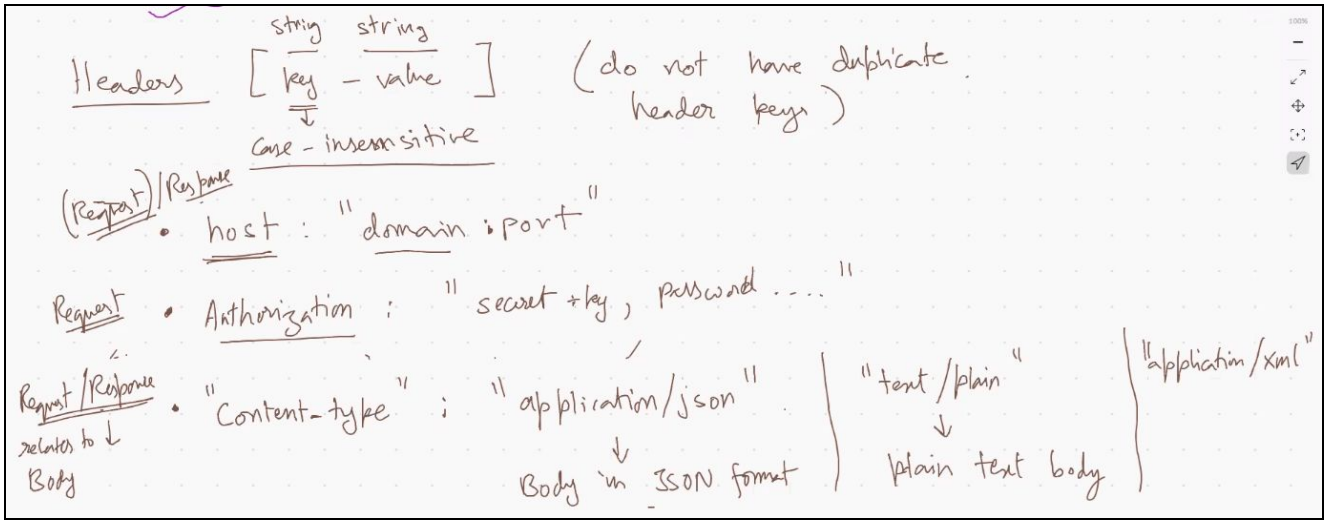
Headers



Headers should not have duplicate keys

Certain headers

1. Host header: conventionally it is present in the request
2. Authorization header, it has a secret key or password used by server to detect whether you authenticated it or not.
3. Content-type header: It is used to tell what type of content we are sending with the response, whether it is json or text or xml or something else.

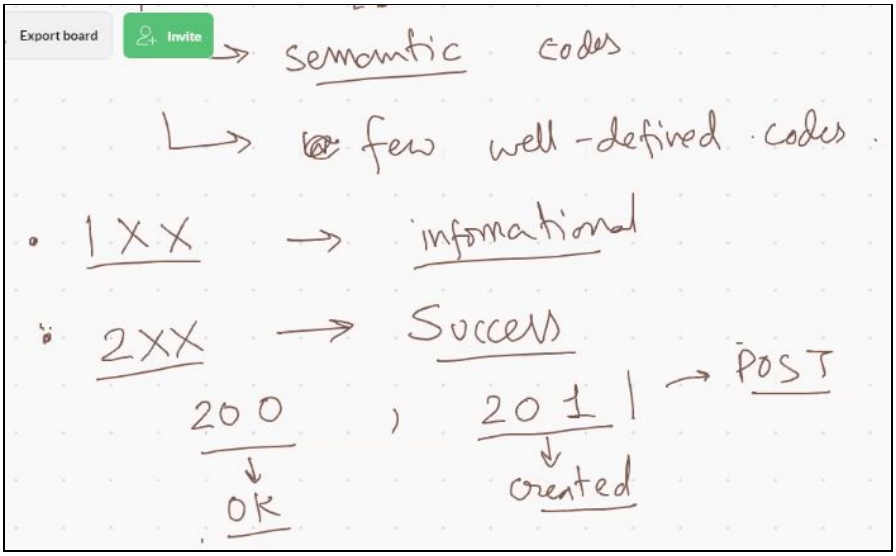


Status Codes:

https://www.restapitutorial.com/httpstatuscodes.html

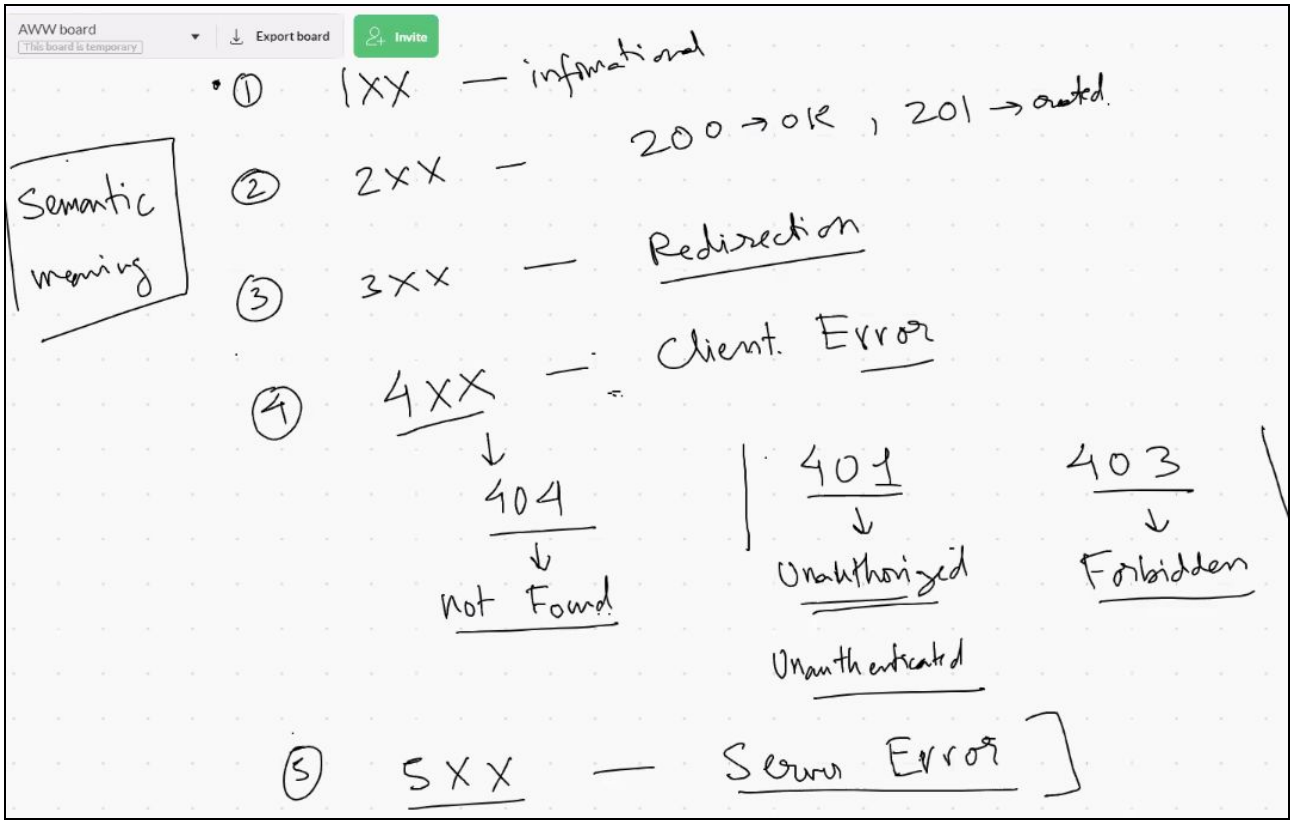
Most of the time 200 is used for success response “OK”.

201: when something is created on Database / Server (POST method uses it).



401 is when you are not logged in, but still trying to access the site.
403 is server can figure out who you are but you don't have permission
404 - page not found

5xx: Server not able to connect to Database.



Three response apis

```
17      response.writeHead(statusCode, responseHeaders);
18      response.write(responseBody);
19      response.end();
```

server.js | package.json | utils.js

src > server.js > http.createServer() callback

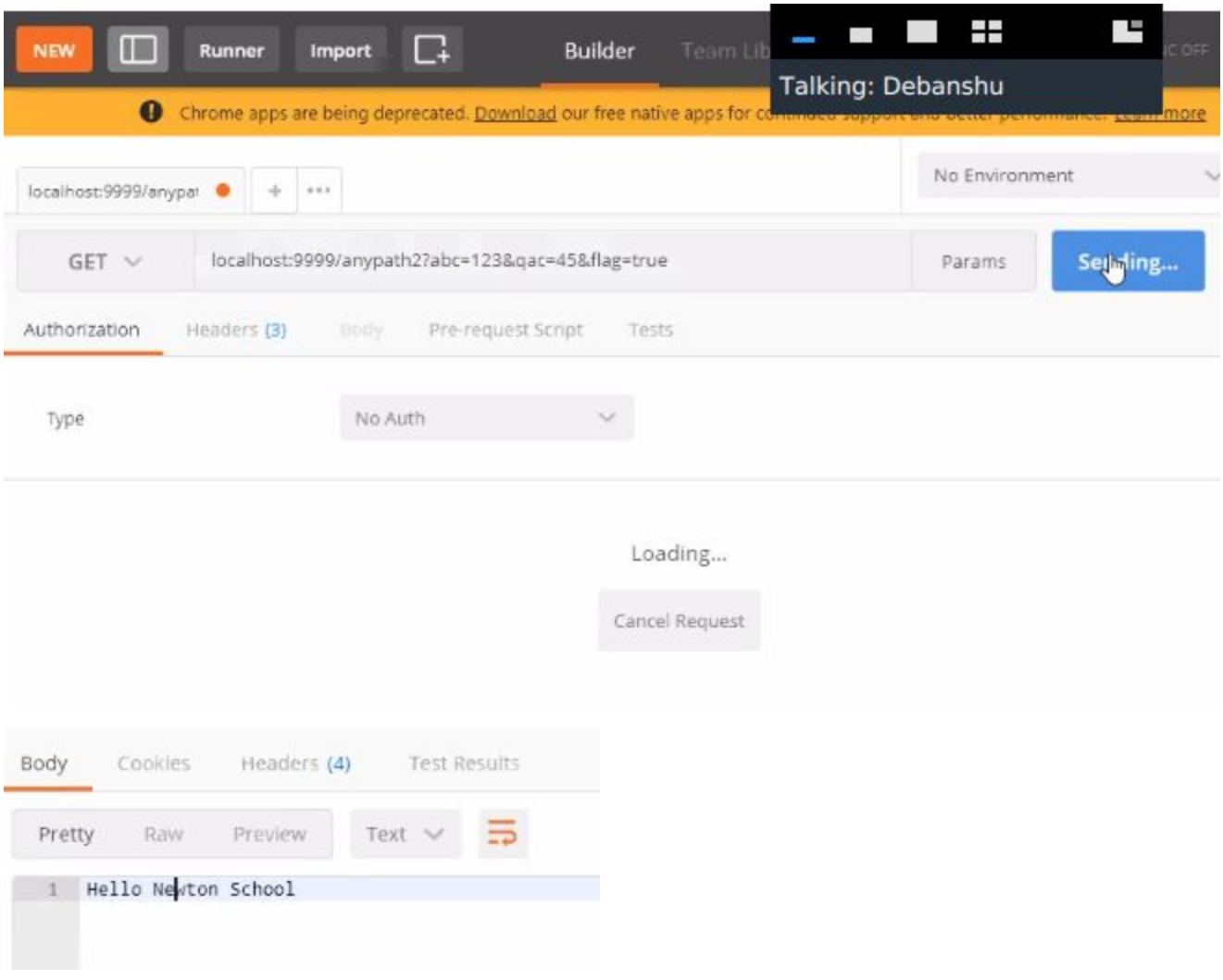
```
11  http.createServer((request, response) => {
12      console.log('Request to Host: ', request.headers.host);
13      console.log('Request Verb', request.method);
14      console.log('Request headers', request.headers);
15      console.log('Request Path/query params', request.url);
16
17      response.writeHead(statusCode, responseHeaders);
18      response.write(responseBody);
19      response.end();
20
21  }).listen(9999);
```

Response headers are key value pair

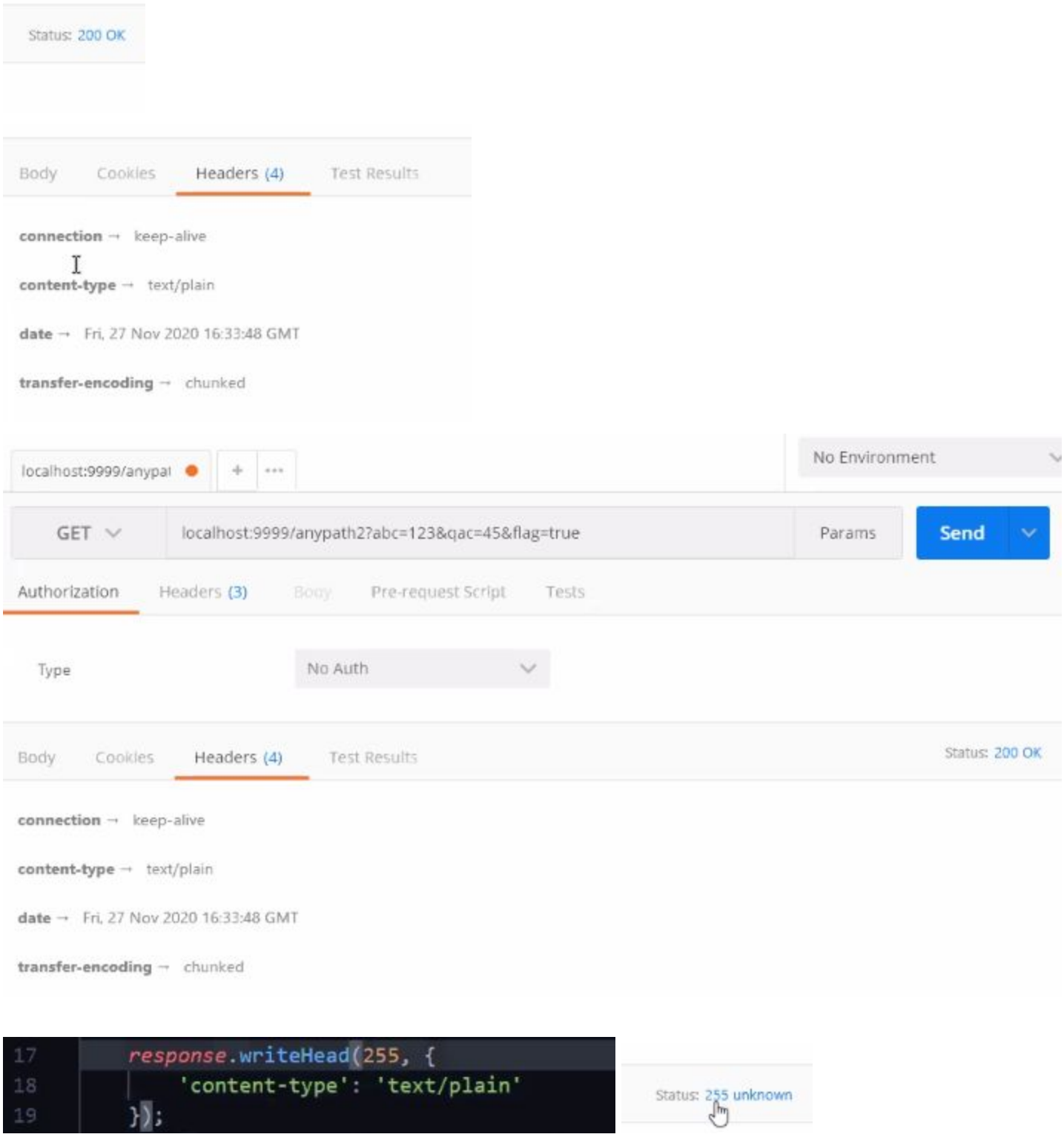
```
17      response.writeHead(200, {
18          'content-type': 'text/plain'
19      });
20      response.write('Hello Newton School');
```

src > JS server.js > http.createServer() callback

```
http.createServer((request, response) => {
    console.log('Request to Host: ', request.headers.host);
    console.log('Request Verb', request.method);
    console.log('Request headers', request.headers);
    console.log('Request Path/query params', request.url);

    response.writeHead(200, {
        'content-type': 'text/plain'
    });
    response.write('Hello Newton School');
    response.end();

}).listen(9999);
```

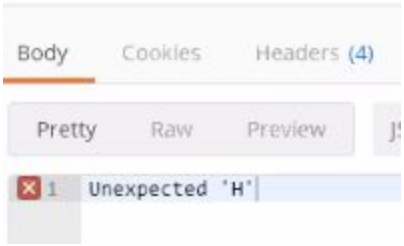


Response body should always be string



Unknown because did not recognize the code, because its not a standard code

```
17      response.writeHead(255, {
18          'content-type': 'application/json'
19      });
```

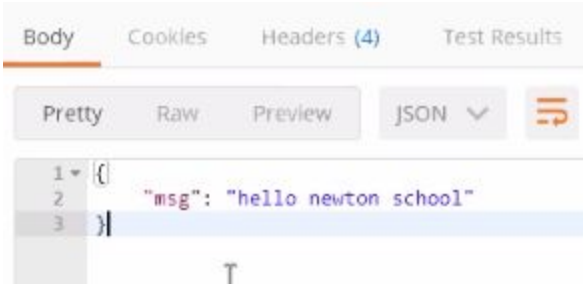


Because written json was not correct

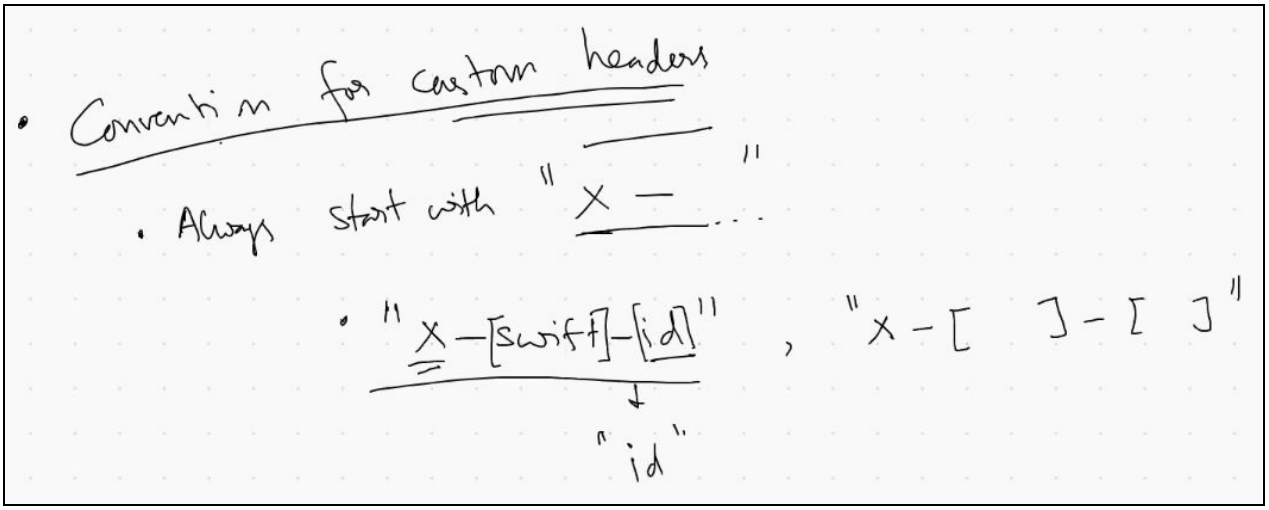
```
20      response.write('{ "msg" : "hello newton school" }');
```

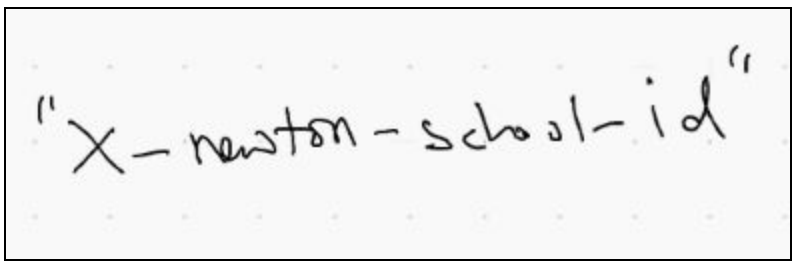
Corrected

Output

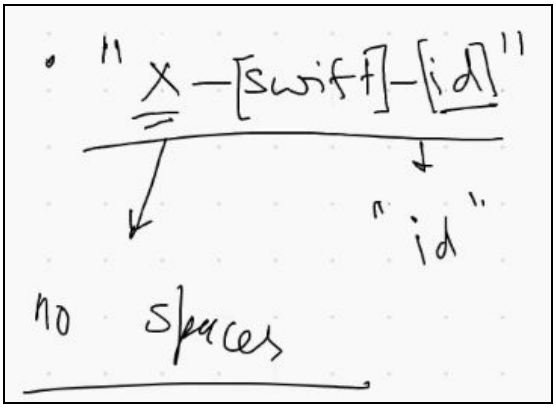


Convention for custom headers

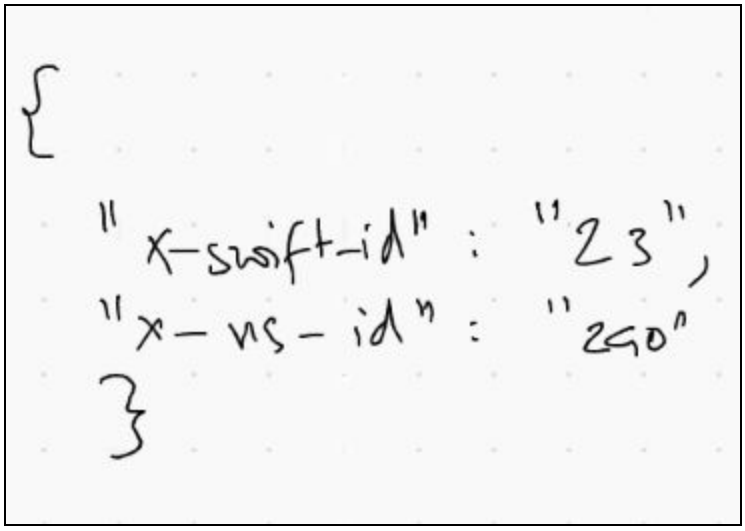




Another example



Key value pair this way

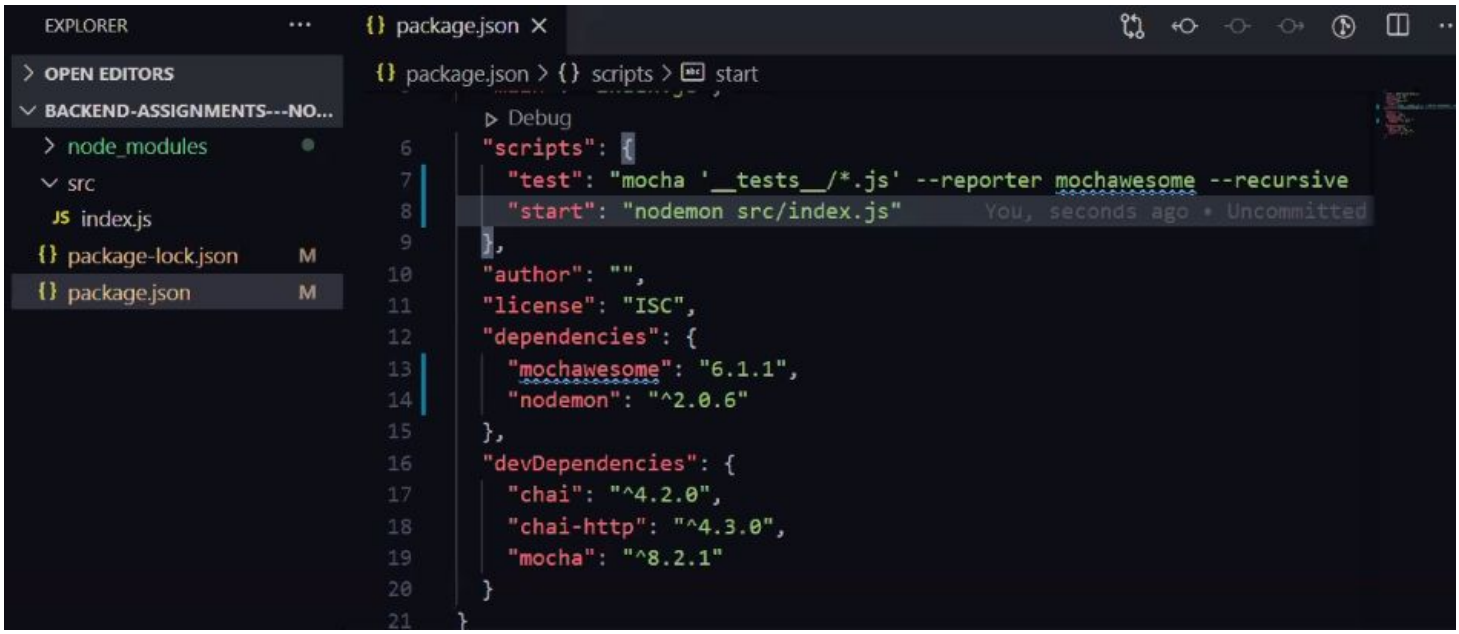


Solving in call assignment

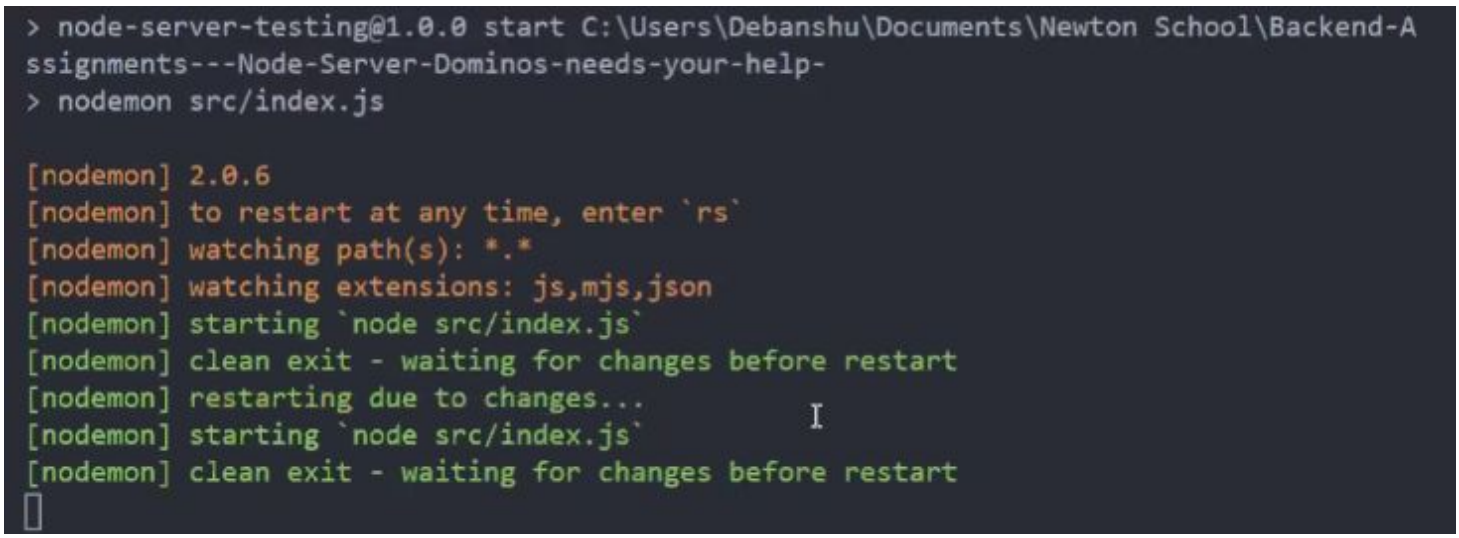
```
C:\Users\Debanshu\Documents\Newton School\Backend-Assignments---Node-Server-Dominos-n
eeds-your-help->npm install nodemon
[                    ] \ finalize:diff: sill finalize C:\Users\Debanshu\Documents\Newt
```

```
8     "start": "nodemon src/index.js"
```

This will restart the server automatically whenever there is a change in the code in index.js


EXPLORER
package.json
OPEN EDITORS
BACKEND-ASSIGNMENTS---NO...
node_modules
src
index.js
package-lock.json M
package.json M
package.json > scripts > start
Debug
"scripts": {
"test": "mocha '__tests__/*.js' --reporter mochawesome --recursive
"start": "nodemon src/index.js"
You, seconds ago • Uncommitted
},
"author": "",
"license": "ISC",
"dependencies": {
"mochawesome": "6.1.1",
"nodemon": "^2.0.6"
},
"devDependencies": {
"chai": "^4.2.0",
"chai-http": "^4.3.0",
"mocha": "^8.2.1"
}
}

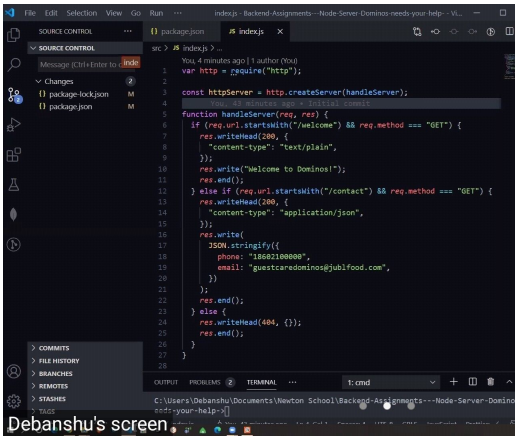
Npm start

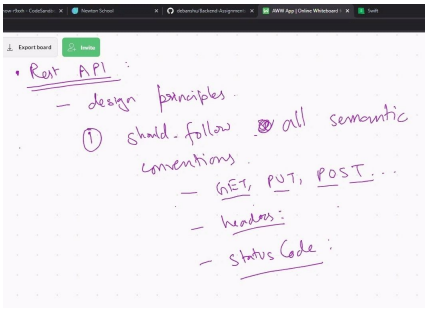

> node-server-testing@1.0.0 start C:\Users\Debanshu\Documents\Newton School\Backend-Assignments---Node-Server-Dominos-needs-your-help-
> nodemon src/index.js
[nodemon] 2.0.6
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node src/index.js`
[nodemon] clean exit - waiting for changes before restart
[nodemon] restarting due to changes...
[nodemon] starting `node src/index.js`
[nodemon] clean exit - waiting for changes before restart

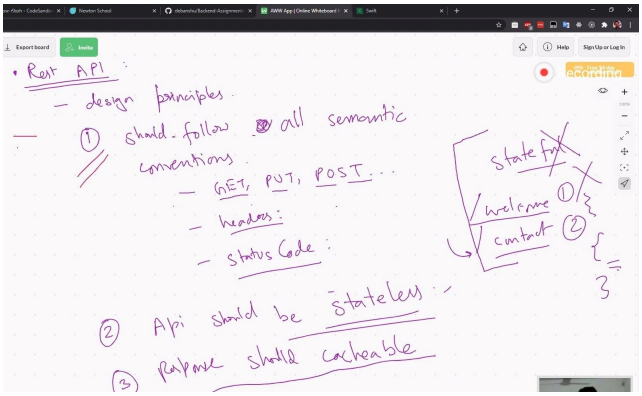
{} package.json JS index.js ●

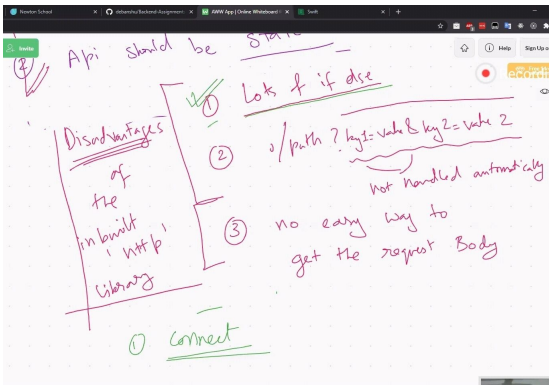
src > JS index.js > handleServer

```
...
var http = require("http");

const httpServer = http.createServer(handleServer);


function handleServer(req, res) {
    if(req.url.startsWith("/welcome") && req.method === "GET") {
        res.writeHead(200, {
            'content-type': 'text/plain'
        });
        res.write('Welcome to Dominos!');
        res.end();
    }


}

httpServer.listen(8081);

module.exports = httpServer;
```



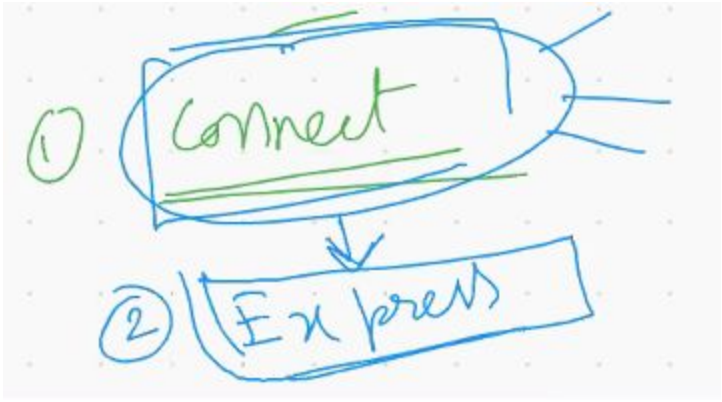






Disadvantages of the in-built http library:

Two famous external libraries:



Note: We can use Switch instead of If-Else in http, but we will not be able to do the “startsWith” matching in that case.