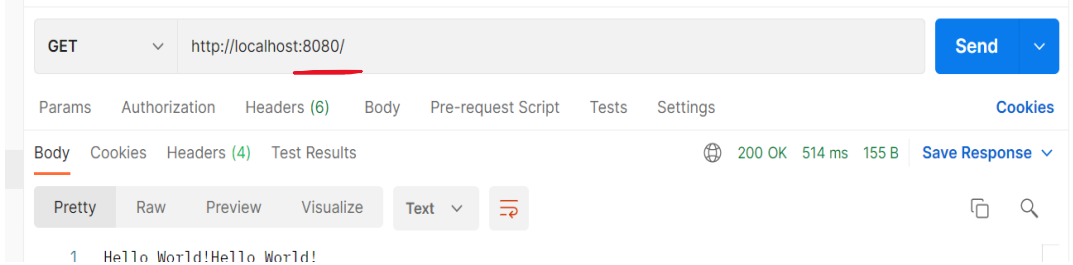
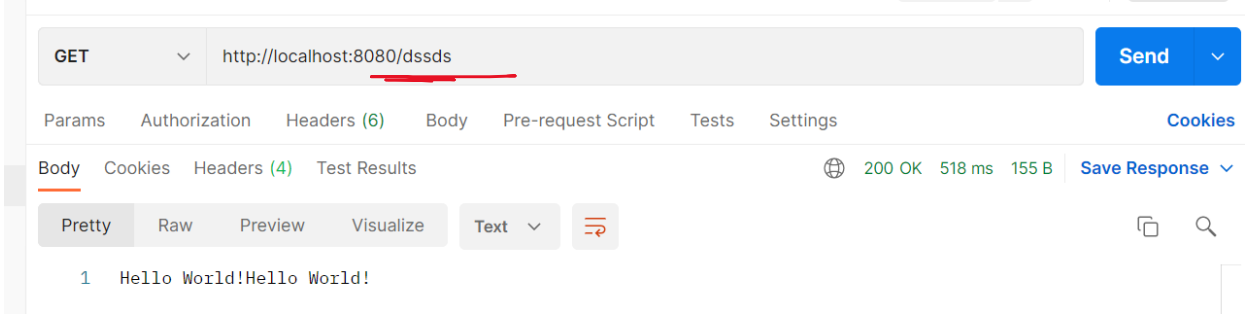
Node.js Routing

Introduction

- Check that the code we wrote in our first webserver-> does not react to the URL that we're requesting.





Introduction

What is Routing?

- Routing defines the way in which client requests are *handled* by the application endpoints (endpoint ie /api/users)
- **Routing actually means implementing different actions for different urls**

Routing with node.js

- No we will see how we can implement routing without framework.
- A developer may need to build up their own server without other dependencies!
- We will know how to do so!

URL Module

- **URL** node.js module -> splits up a **web address** into readable parts
- For very simple urls we do not actually need the url module
- We do need it for more complex ones

Lets see an example

- We use some conditions to handle different urls and display different content
- **req object** represents the **HTTP request** and has **properties**

```
const server= http.createServer((req, res) => {
    //we store the requested url in a variable
    const path_name= req.url;
    console.log(path_name);

    if(path_name==='/' || path_name==='/all'){
        res.write('Here are all my Projects');
    }else if(path_name==='/project'){
        res.write('This is one project');
    }else{
        //inspect web page and check the console
        //we can also send headers, a piece of info regar
        res.writeHead('404',{'Content-Type': 'text/html'});
        res.write('<h2> 404 <br> Page not Found </h2>');
    }
    res.end(); //end the response
});
```

200 - OK

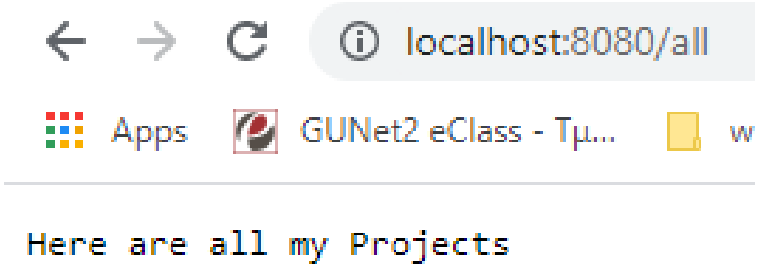
301 - Resource moved

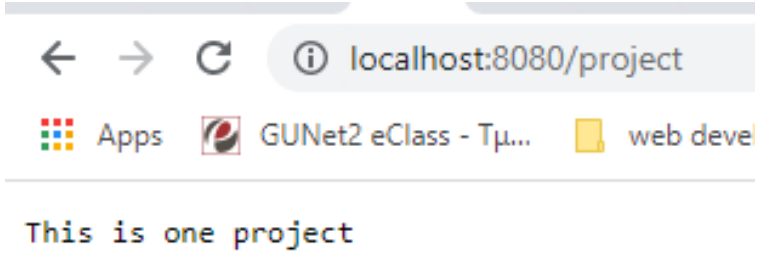
404 - Not found

500 - Internal server error

Lets see an example

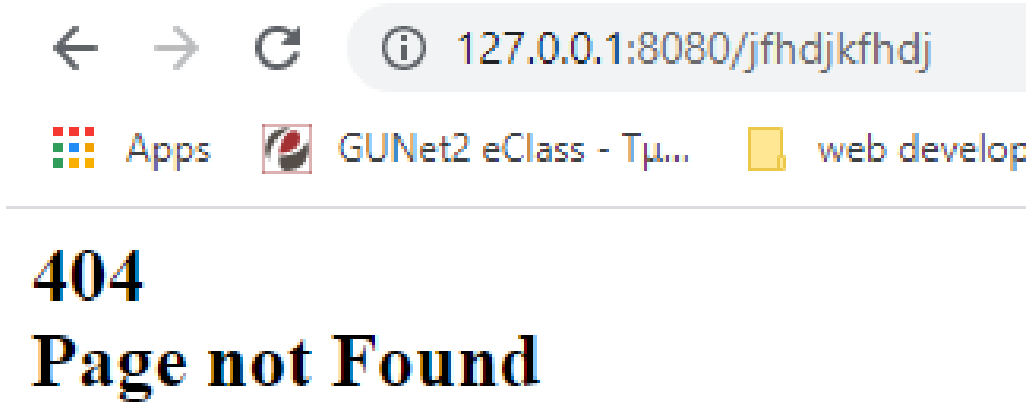
- Lets see the results in the browser





Lets see an example

- Lets see the results in the browser



- Lets see how we can perform routing and serve an html file

UNIVERSITY OF PIRAEUS

To be continued…