Routing/Routes

```
const express = require("express")
const router = express.Router()

router.get("/", function(req,res){
    console.log("This is my first API")
    res.send("This is my first API request implemented")
})


// router.get("/userinfo/:id/:username", function(req,res){…
// })

// router.get("/userinfo", function(req,res){…
// })

// router.post("/userinfo", function(req,res){…
// })

// router.put("/userinfo", function(req,res){…
// })

// router.delete("/userinfo", function(req,res){…
// })


module.exports = router;
```

Connection to MySQL Database

Npm install mysql

```
1  const mysql = require("mysql")
2  const dotenv = require("dotenv").config()
3
4  var connection = mysql.createPool({
5      "connectionLimit":process.env.MYSQL_CONNECTIONLIMIT,
6      "host":process.env.MYSQL_HOST,
7      "user":process.env.MYSQL_USER,
8      "password":process.env.MYSQL_PASSWORD,
9      "database":process.env.MYSQL_DATABASE
10 })
11
12 module.exports = connection
```

Connection to MySQL Database

```
const express = require("express")
const router = express.Router()
const connection = require("../mysql/mysql.js")
const error_controller = require("../controllers/error_handler")

router.get("/", function(req, res){
    console.log("GET request")
    get_users = "select * from users"
    connection.query(get_users, function(err, results, fields) {
    if (err) {
        error_controller.error_handling(err, req, res,"400");
    } else {
        console.log(fields)

        if(results.length){
            var response_body = results;
        }
        else {
            var response_body = null
        }

        res.statusCode = 200
        res.set({
            "Content-Type":"application/json"
        })
        res.end(JSON.stringify(response_body));
    }
    });
})

```

Error Handling

demo-project > controllers > error_handler.js > error_controller > error_handling

```
var express = require("express"),
  router = express.Router();

var error_controller = {
  error_handling: function(err, req, res, statusCode) {
    var response_body = {
      msg: err,
    };
    res.statusCode = statusCode
    res.set({ "Content-Type": "application/json" });
    res.end(JSON.stringify(response_body));
  }
};

module.exports = error_controller;

```

```
const express = require("express")
const router = express.Router()
const connection = require("../mysql/mysql.js")
const error_controller = require("../controllers/error_handler")

router.get("/", function(req, res){
    console.log("GET request")
    get_users = "select * from users"
    connection.query(get_users, function(err, results, fields) {
    if (err) {
        error_controller.error_handling(err, req, res,"400");
    } else {
        console.log(fields)

        if(results.length){
            var response_body = results;
        }
        else {
            var response_body = null
        }

        res.statusCode = 200
        res.set({
            "Content-Type":"application/json"
        })
        res.end(JSON.stringify(response_body));
    }
    });
})
```

MYSQL query with query parameters

```
const express = require("express")
const router = express.Router()
const connection = require("../mysql/mysql.js")
const error_controller = require("../controllers/error_handler")

router.get("/", function(req, res){
    console.log("GET request")
    get_users = "select * from users"
    connection.query(get_users, function(err, results, fields) {
    if (err) {
        error_controller.error_handling(err, req, res,"400");
    } else {
        console.log(fields)

        if(results.length){
            var response_body = results;
        }
        else {
            var response_body = null
        }

        res.statusCode = 200
        res.set({
            "Content-Type":"application/json"
        })
        res.end(JSON.stringify(response_body));
    }
    });
})

```

Register API

```
router.post("/register", function(req,res){
    console.log("Register API")
    const email = req.body.email
    console.log(email)
    const user_details = req.body
    user_check_query = "select * from users where email=?"
    connection.query(user_check_query, [email], function(err,results){
        if(err){
            error_controller.error_handling(err, req, res, 400)
        }else{
            if(results.length>0){
                console.log("user already exists")
                res.send({"message":"user already exists. Go and Login re baba"})
            }else{
                const user_create_query = "INSERT INTO users SET email=?,  password=?, first_name=?, last_name=?, phone_number=?"

                connection.query(user_create_query,[user_details.email, user_details.password, user_details.first_name, user_details.last_name,
                user_details.phone_number], function(error1, results1){
                    if(error1){
                        console.log(error1)
                        error_controller.error_handling(error1, req, res, 400)
                    }else{
                        console.log(results1)
                        res.send({"message":"user Created successfully. Maja madko"})
                    }
                })
            }
        }
    })
```

Register API

- What should happen in the register API?

Register API

Logic

- Check whether all required parameters exists
- Format of each parameter is correct or not
- Check whether already exists in the system
- If user does not exists
  - Hash the password
  - Save all details into database