Clarusway

Backend
Teamwork
-4-

Teamwork

Subject: Expressjs

Learning Goals

To reinforce our Expressjs knowledge with applications.

Introduction

As with other frameworks / programming languages, the best way to learn Expressj is by practicing it and trying to fix the errors we encounter. With this teamwork, we aim to reinforce our theoretical and practical knowledge about Expressjs by performing the required tasks.

Pre-requirements

We will use the VSCode.

Lets start

1. Write the regex expression that matches the required conditions.

- Write a router that matches /abc or /acd path.

Answer:

```
router.get(/\a[b|c]d/, (req, res) => {
    res.send("<h1>path matched</h1");
})
```

- Write a router that matches /a(any single digit)/ followed by 2 times c or 3 times c or

Answer:

```
router.get(/\a[0-9]\/c{2,3}/, (req, res) => {
    res.send("path matched");
})s
```

- Routes that must end with string "Hello" and can have any no. of any character before that .

Answer:

```
router.get(/\/*Hello$/, (req, res) => {
    res.send('<h1>Route Fourth</h1>')
})
```

- Matching routes must end with the string "Hello" and must not have any characters before that.

Answer:

```
router.get(/\/*Hello$/, (req, res) => {
    res.send('<h1>Route Fourth</h1>')
})
```

2. I have an object with student information. Code the desired routers.

- Returns all students in json format with the get method
- returns information about the requested student in json format

- If there is any missing/error in the code, please correct it.

```
'use strict'
const express = require("express");
const app=express()

const router = express.Router();
app.use(router)

const students = [{
    id: 1,
    name: "Alex",
},
{
    id: 2,
    name: "Steve",
}];
```

Answer:

```
'use strict'
const express = require("express");
const app=express()

const router = express.Router();
router.use(express.json());
app.use(router)

const students = [{
    id: 1,
    name: "Alex",
},
{
    id: 2,
    name: "Steve",
}];


router.get("/", (req, res) => {
    res.json(students);
});
// http://localhost:8000/?id=2
router.get("/:id", (req, res) => {
    const results = students.filter(students => students.id == req.params.id);
    res.json(results);
});
app.listen(8000, () => console.log(`Running on http://localhost:/8000` ))
```

3. Code the instructions below.

- import the Express framework and create an Express application.
- define a middleware function using app.use(). This function logs a message and then calls next() to pass control to the next middleware or route handler.
- define a route handler for the root URL ('/') using app.get(). This handler will respond with "Hello!" when a GET request is made to the root URL.
- Finally, srart the Express server and listen on port 3000.

Answer:

```
const express = require('express');
const app = express();

app.use((req, res, next) => {
  console.log('This is first middleware.');
  next(); // Call the next middleware function in the chain
});
app.get('/', (req, res) => {
  res.send('Hello Backend');
});
const port = 8000;
app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});
```

4.In the code block below, a custom error middlaware is missing which returns a 500 status code and the detail of the error generated. Complete the code.

```
const express = require('express');
const app = express();

app.use((req, res, next) => {
    throw new Error('Something went wrong!');
});

app.use(errorHandler);
app.listen(8000, () => {console.log('Server started on port 8000');});
```

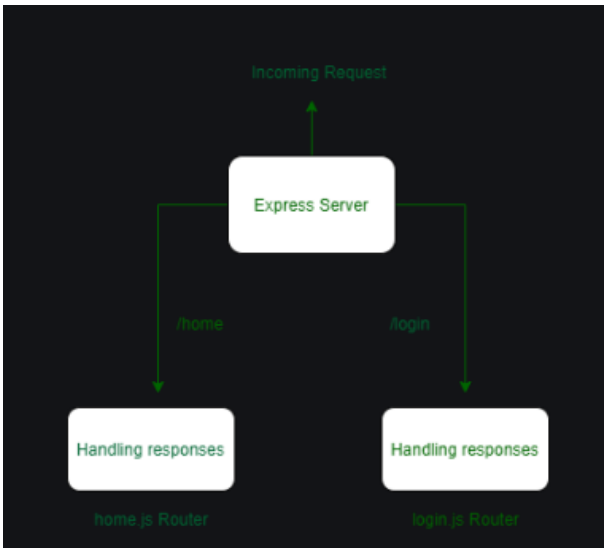
Answer:

```
const express = require('express');
const app = express();
// Custom error handling middleware
const errorHandler = (err, req, res, next) => {
    res.status(500).send('Error-free code is itself an error!'+err.stack);
};

app.use((req, res, next) => {
    throw new Error('Something went wrong!');
});

app.use(errorHandler);
app.listen(8000, () => {console.log('Server started on port 8000');});
```

5. How to create multiple routes on a single express server according to structure Model.



Answer:

/routes/home.js

```
const express=require("express")
const router=express.Router()
router.get("/",(req,res,next)=>{
    res.send("This is the homepage request")
})
module.exports=router
```

/routes/login.js

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

```
router.get("/",(req,res,next)=>{
    res.send("This is the login request")
})
module.exports=router
```

index.js

```
const express=require("express")
const homeroute=require("./routes/Home.js")
const loginroute=require("./routes/login")
const app=express()
app.use("/home",homeroute)
app.use("/login",loginroute)
app.listen((3000),()=>{
    console.log("Server is Running")
})
```

☺ **Thanks for Attending** ✍

Clarusway