

Clarusway



## Backend Teamwork -3-

---

## 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(/\abc|\acd/, (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\d{1}c{2}|\a\d{1}c{3}/, (req, res) => {  
  res.send("path matched");  
})
```

- 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();

// Middleware to parse JSON request body
app.use(express.json());

// Define route to get all students
router.get("/students", (req, res) => {
  res.json(students);
});

// Define route to get information about a specific student
router.get("/students/:id", (req, res) => {
  const id = parseInt(req.params.id);
  const student = students.find(student => student.id === id);
  if (!student) {
    return res.status(404).json({ message: "Student not found" });
  }
  res.json(student);
});

// Define a default route
router.get("/", (req, res) => {
  res.send("Welcome to the student API!");
});

const students = [
  {
    id: 1,
    name: "Alex",
```

```
    },
    {
      id: 2,
      name: "Steve",
    },
  ];

  // Attach the router
  app.use("/", router);

  // Define the port
  const PORT = process.env.PORT || 3000;

  // Start the server
  app.listen(PORT, () => {
    console.log(`Server is running on port ${PORT}`);
  });
```

### 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, start the Express server and listen on port 3000.

#### Answer:

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

// Middleware function
app.use((req, res, next) => {
  console.log("Middleware function is triggered!");
  next(); // Pass control to the next middleware or route handler
});

// Route handler for the root URL ( '/')
app.get("/", (req, res) => {
  res.send("Hello!");
});

// Start the Express server and listen on port 3000
const PORT = 3000;
```

```
app.listen(PORT, () => {  
  console.log(`Server is running on port ${PORT}`);  
});
```

**4. In the code block below, a custom error middleware 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!");  
});  
  
// Özel hata orta yazılımı  
const errorHandler = (err, req, res, next) => {  
  res.status(500).json({  
    error: {  
      message: err.message  
    }  
  });  
};  
  
app.use(errorHandler);  
  
app.listen(8000, () => {  
  console.log("Sunucu 8000 numaralı portta başlatıldı");  
});
```

---

😊 Thanks for Attending 📌

