

Clarusway



Backend Workshop -3-

Workshop

Subject: Expressjs

Learning Goals

Learning the basic structures we will use when doing a project with Express.js

Introduction

Information about the framework we use can take shape in our minds. But when necessary, we must be able to express them verbally or in writing. This workshop aims to reinforce the basic information we may encounter during an interview.

Pre-requirements

We will use the VSCode .

Lets start

1. What is Express?

Answer:

2. Where does the popularity of Express come from?

Answer:

3. Have you ever taken a look, which famous companies use Express, what is Express place among other frameworks? Lets googling.

Answer:

4. What are the differences between Express and Nodejs?

Answer:

5. What is Middleware, and what are its functions?

Answer:

6. Explain what the keys we see in the package.json file mean.

Answer:

7. What is the difference between `res.send()` and `res.json()` in Express.js?

Answer:

8. What is the difference between `app.use()` and `app.get()` in Express.js?

Answer:

9. Write a middleware to log the incoming request method and URL to the console.

Answer:

10. Write a function in Express.js that calculates the sum of two numbers passed as query parameters.

Answer:

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:

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

Answer:

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

Answer:

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

Answer:

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:

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:

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!");
});

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

Answer:

😊 Thanks for Attending 🙌

Clarusway

