Express.js-
Creating a Server

Table of content

1-What is Express.js?

2- init+install packages

3-Key Concepts

4-Live Demo

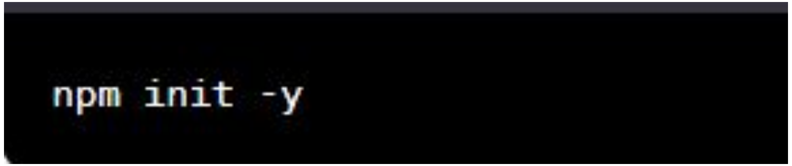
5-Conclusion

What is Express.js?

- Express.js is a minimal and flexible Node.js web application framework.

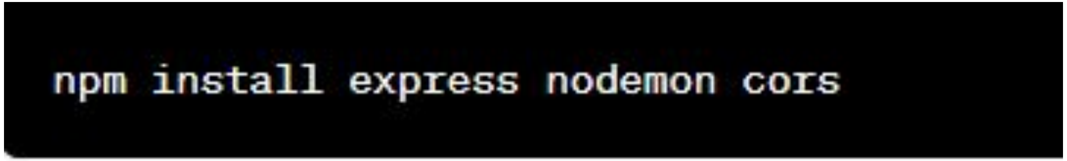
- It simplifies the process of building robust and scalable web applications.

- It provides a set of features for web and mobile applications.



```
npm init -y
```

Initialize a new Node.js project using the npm init -y command. This command will create a package.json file with default values without asking you to interactively provide information



```
npm install express nodemon cors
```

This command will download and install the specified packages in your project directory. After this step, you should see a node_modules folder in your project directory containing the installed packages.

Now you have initialized a new Node.js project with a package.json file and installed the necessary dependencies (Express, Nodemon, and Cors) for your project. You can proceed to develop your application using these packages.

Key Concepts

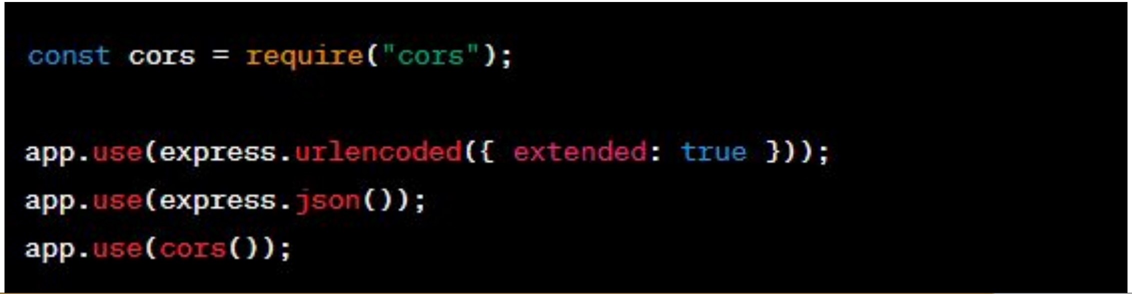
- Middleware
- Routes
- Static Files

Middleware

- Middleware functions are functions that have access to the request and response objects.

- This middleware is used to handle Cross-Origin Resource Sharing (CORS) for your server. It allows or restricts web pages from making requests to your server from different domains.

- Middleware is used for tasks such as authentication, logging, parsing request data, and more. You can add middleware functions globally or on specific routes.



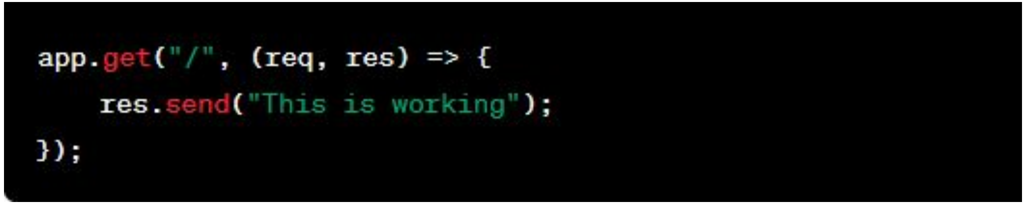
```
const cors = require("cors");

app.use(express.urlencoded({ extended: true }));
app.use(express.json());
app.use(cors());
```

Routes

Routes define how your application responds to client requests.

Express provides methods for different HTTP verbs (GET, POST, PUT, DELETE) to define routes.

Here, you define a route handler for the root URL ("/"). When a client sends an HTTP GET request to the root URL

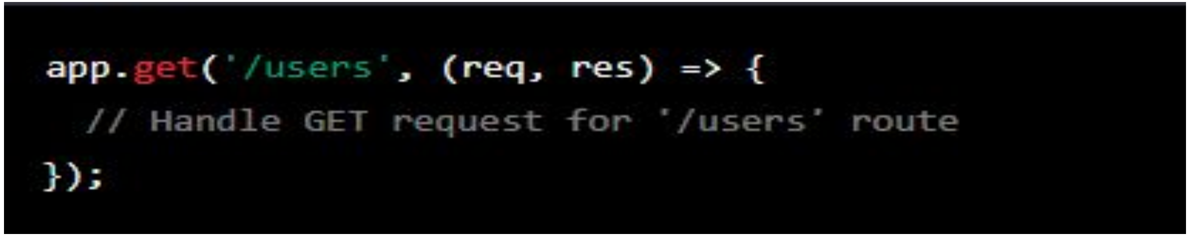


```
app.get("/", (req, res) => {
    res.send("This is working");
});
```

Static Files

You can serve static files (e.g., HTML, CSS, images) using

express.static middleware.



```
app.get('/users', (req, res) => {
  // Handle GET request for '/users' route
});
```

LIVE DEMO

CONCLUSION

Express.js simplifies building web servers and APIs.

It offers robust routing, middleware, and request/response handling.

A great choice for developing modern web applications.

Resources

Express.js Official Documentation

Youtube/DarwinTech:https://www.youtube.com/watch?v=7W61Ep1lKFY