

## Agenda

- Express
- Mysql
- ~~Crypto-Js~~
- ~~JWT~~

## nodemon

- It is a Node monitor which is used to monitor changes in server and refreshes the server

## Express

- Developed by many developers around the world
- It is a fast,minimilistic web framework for nodejs
- It is a lightweight and flexible routing framework with minimal core features meant to be augmented through the use of Express middleware modules.

```
# to start using express application install express using npm or add using yarn
npm install express
# OR
yarn add express
```

## Routing

- Routing refers to determining how an application responds to a client request to a particular endpoint, which is a URI (or path) and a specific HTTP request method (GET, POST, and so on).
- Each route can have one or more handler functions, which are executed when the route is matched.
- Route definition takes the following structure:

```
app.METHOD(PATH, HANDLER)
```

- Where:
  1. app is an instance of express.
  2. METHOD is an HTTP request method, in lowercase.
  3. PATH is a path on the server.
  4. HANDLER is the function executed when the route is matched.
- we define routing using methods of the Express app object that correspond to HTTP methods
- for example, app.get() to handle GET requests and app.post to handle POST requests.
- These routing methods specify a callback function (sometimes called "handler functions") called when the application receives a request to the specified route (endpoint) and HTTP method.
- In other words, the application "listens" for requests that match the specified route(s) and method(s), and when it detects a match, it calls the specified callback function.

## express.Router

- Use the express.Router to create modular, mountable route handlers.
- A Router instance is a complete middleware and routing system.
- for this reason, it is often referred to as a "mini-app".

## Middleware

- Middleware is a function that runs between the request and the response in a web application.
- It is commonly used in backend frameworks like Express.js to modify, process, or control HTTP requests before sending a response.
- It Access Request (req) and Response (res) Objects
- It Modify Requests/Responses → Example: Adding headers, parsing JSON
- Run Code Before Sending a Response → Example: Authentication, Logging
- Call next() to Continue Execution → Passes control to the next middleware
- It is commonly used in express for
  1. Handling CORS (cors package)
  2. Parsing JSON data (express.json())
  3. Authentication & authorization

## Connecting with mysql database

1. add/install the mysql module

```
npm install mysql2
```

2. Create the pool object

```
const pool = mysql.createPool({  
  host: 'localhost',  
  user: 'root',  
  password: 'root',  
  database: 'kdac_db'  
})
```

3. Call the query method

```
const sql = `SELECT * FROM product`  
pool.query(sql, (error, data) => {  
  if(data)  
    response.send(data)  
  else  
    response.send(error)  
})
```

