# Express.js Routing

app.get(route, callback)

- This function tells what to do when a **get** request at the given route is called.
- The callback function has 2 parameters, ***request(req)*** and ***response(res)***.
- The request **object(req)** represents the HTTP request and has properties for the request query string, parameters, body, HTTP headers, etc.
- Similarly, the response object represents the HTTP response that the Express app sends when it receives an HTTP request.

res.send()

- This function takes an object as input and it sends this to the requesting client.
- Here we are sending the string *"Hello World!"*.

app.listen(port, [host], [backlog], [callback]])

- This function binds and listens for connections on the specified host and port.
- Port is the only required parameter here.

# Express.js Routing

app.get(route, callback)

- This function tells what to do when a **get** request at the given route is called.

res.send()

- This function takes an object as input and it sends this to the requesting client.

app.listen(port, [host], [backlog], [callback]])

- This function binds and listens for connections on the specified host and port.

# Express.js Routing

HTTP Methods

| Method | Description |
|---|---|
| 1. GET | The HTTP GET method helps in requesting for the representation of a specific resource by the client. The requests having GET just retrieves data and without causing any effect. |
| 2. POST | The HTTP POST method helps in requesting the server to accept the data that is enclosed within the request as a new object of the resource as identified by the URI. |
| 3. PUT | The HTTP PUT method helps in requesting the server to accept the data that is enclosed within the request as an alteration to the existing object which is identified by the provided URI. |
| 4. DELETE | The HTTP DELETE method helps in requesting the server to delete a specific resource from the destination. |

# Express.js Middleware

- Express.js Middleware are different types of functions that are invoked by the Express.js routing layer before the final request handler.

- As the name specified, Middleware appears in the middle between an initial request and final intended route.

- In stack, middleware functions are always invoked in the order in which they are added.

- Middleware is commonly used to perform tasks like body parsing for URL-encoded or JSON requests, cookie parsing for basic cookie handling, or even building JavaScript modules on the fly.