Q. What is express & what it used for ?

Ans. Express is used to create and manage sever in backend

Express is a node js web application framework that provides broad features for building web and mobile applications. It is used to build a single page, multipage, and hybrid web application.

Express.js, or simply Express, is a back end web application framework for building RESTful APIs with Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js.

2. What is the Express.js package?

Express is a flexible Node.js web application framework that provides a wide set of features to develop both web and mobile applications

3. What are the different types of HTTP requests?

HTTP defines a set of request methods used to perform desired actions. The request methods include:

GET: Used to retrieve the data

POST: Generally used to make a change in state or reactions on the server

HEAD: Similar to the GET method, but asks for the response without the response body

DELETE: Used to delete the predetermined resource

|  |  |  |
| --- | --- | --- |
| What | why | Commands |
| Installing express |  | npm install express |
| Hellow World- How do you create a simple Express.js application?   * The request object represents the HTTP request and has properties for the request query string, parameters, body, HTTP headers, and so on * The response object represents the HTTP response that an Express app sends when it receives an HTTP request * Set the server to listen to port 8080 and assign an IP address | This app starts a server and listens on port 3000 for connections. The app responds with “Hello World!” for requests to the root URL (/) or route. For every other path, it will respond with a 404 Not Found. | // import statement  const express = require("express")  // config    //server init  const server = express()  //server routes or services  //Get = to read the data  server.get()  //server start  server.listen(8000, ()=> console.log("hello world...")); |
| Dynamic routes/service |  |  |
| Middlewares | A code which will be in the middle of the service(routes)  To secure private data  The middleware in node. js is a function that will have all the access for requesting an object, responding to an object, and moving to the next middleware function in the application request-response cycle. | const express = require('express')  const app = express()  const myLogger = function (req, res, next) {  console.log('LOGGED')  next()  }  app.use(myLogger)  app.get('/', (req, res) => {  res.send('Hello World!')  })  app.listen(3000) |
| Postman |  |  |

|  |  |  |
| --- | --- | --- |
| What | Why | command |
| nodemon | nodemon is a tool that helps develop Node. js based applications by automatically restarting the node application when file changes in the directory are detected. nodemon does not require any additional changes to your code or method of development. nodemon is a replacement wrapper for node | npm i nodemon |