Express-Web framework for nodejs

Api-application programming interface

Set of rules that allow programs to talk to each other.

Developer creates api on server allows client to talk to it.

REST api

REST determines how api looks like.

Api-set of programs on server side

Representational state transfer

Transfers only state not object

Ex-weather forecasting app

Are you going to place sensors everywhere?no

Use third-party data via an api.

Rest uses HTTP protocol.

HTTP methods-you knew some earlier

* GET – *safe, idempotent, cacheable*
* For reading
* PUT – *idempotent*
* POST
* DELETE – *idempotent*
* HEAD
* OPTIONS

Table

Description automatically generated

Post-creates an object.

Put-replaces object with object supplied in the body(updating data)

Delete-deletes the object.

**body-parser** − This is a node.js middleware for handling JSON, Raw, Text and URL encoded form data.

**cookie-parser** − Parse Cookie header and populate req.cookies with an object keyed by the cookie names.

**multer** − This is a node.js middleware for handling multipart/form-data.

**Pug**

Pug (earlier known as Jade) is a language for writing HTML templates. It is one of the most popular template language used with Express.

**Mongoose**

It is a client API for **node.js** which makes it easy to access our database from our Express application.

**The most common HTTP Methods are**

* **GET**

The GET method requests a representation of the specified resource. Requests using GET should only retrieve data and should have no other effect.

* **POST**

The POST method creates a new object/entity of the resource identified by the URI.

* **PUT**

The PUT method modifies the existing object identified by the URI. If it does not exist then the PUT method should create one.

* **DELETE**

The DELETE method requests that the server delete the specified resource.

* The following function is used to define routes in an Express application

**app.method(path, handler)**

* app is just an instance of Express.js. You can use any variable of your choice.
* METHOD is an HTTP request method such as get, set, put, delete, etc.

PATH is the route to the server for a specific webpage

Seen get and post till now.

In case, the current function doesn’t terminate the request-response cycle

then it must invoke the **next() function** in order to pass on the control to the next available middleware function.

If not done, then the request will be left incomplete.

In case, the current function doesn’t terminate the request-response cycle

then it must invoke the **next() function** in order to pass on the control to the next available middleware function.

If not done, then the request will be left incomplete.

**body-parser**

This is used to parse the body of requests which have payloads attached to them. To mount body parser, we need to install it using

**npm install body-parser**

and to mount it, include the following lines (Parse incoming request bodies in a middleware before your handlers, available under the req.body property)

var bodyParser = require('body-parser');

//To parse URL encoded data

app.use(bodyParser.urlencoded({ extended: false }))

//To parse json data

app.use(bodyParser.json())

**cookie-parser**

It parses Cookie header and populate req.cookies with an object keyed by cookie names. To mount cookie parser, we need to install it using

npm install cookie-parser

and to mount it, include the following lines.

var cookieParser = require('cookie-parser');

app.use(cookieParser())

Next() moves on to next middleware.

Middle ware functions are executed in order and response is sent.

A picture containing timeline

Description automatically generated

var app = express();

starts server