**EXPRESSJS**

**INTRODUCTION:**

Express js is a open-source Node.js framework for quickly creating dynamic end-to-end **REST APIs.**

**SETUP:**

1. Install **node.js**
2. Create a folder myapp using **mkdir myapp**
3. Move to myapp folder using **cd myapp**
4. Create package.json using **npm init**
5. Install expressjs module using **npm install express –save**

**Simple GET API Hello World Example:**

1. **C**reate a file app.js inside **myapp** folder.
2. Write below snippet of code in **app.js** file.

const express = require('express');

const app = express();

const port = 3000;

app.get('/', (req, res) => res.send('Hello World!'));

app.listen(port, () => console.log(`Example app listening on port ${port}!`));

1. Run this file from **command prompt** using **node app.js**
2. Test Hello word example using URL **:** [**http://localhost:3000/**](http://localhost:3000/)
3. Output: **Hello World!**

**Route methods:**

Express supports methods that correspond to all HTTP request methods.

A route method is derived from one of the HTTP methods,

The following code is an example of routes that are defined for the GET and the POST methods to the root of the app.

const express = require('express');

const app = express();

const port = 3000;

app.get('/', (req, res) => res.send('GET Request Example'));

app.post('/', (req, res) => res.send('POST Request Example'));

app.listen(port, () => console.log(`Example app listening on port ${port}!`))

**MiddleWares:**

Simple Middlewares for Logging middleware and enabling CORS:

You’ll start by making your app log requests. Put the code in the following listing

inside **app.js.**

var express = require("express");

var app = express();

app.use(function(req, res, next) { //Logging Every income request

console.log("Request IP: " + req.url);

console.log("Request date: " + new Date());

next(); //It needs to call next to continue on to the next function in the middleware

//stack and needs to finish responding to the current incoming request Also.

});

//CORS Middleware

app.use(function (req, res, next) {

//Enabling CORS

res.header("Access-Control-Allow-Origin", "\*");

res.header("Access-Control-Allow-Methods", "GET,HEAD,OPTIONS,POST,PUT");

res.header("Access-Control-Allow-Headers", "Origin, X-Requested-With, contentType,Content-Type, Accept, Authorization");

next();

});

app.get('/api', (req, res) => res.send('Hello World!'));

listen(3000, function() {

console.log("App started on port 3000");

});

DEAL WITH PATH PARAMS :

Sample URI For Below snippet code: [**http://localhost:3000/**course/express](%20http://localhost:3000/course/express)

**app.get('/course/:name', function(req, res) {**

**res.send('course name :' + req.params.name);**

});

Output: course name : express

DEAL WITH QUERY STRING:

Sample URI For Below snippet code: [**http://localhost:3000/**course?id= 1](http://localhost:3000/course?id=%201)

**app.get('/course?id ', function(req, res) {**

**res.send('course id :' + req.query.id); //**

});

Output: course id : 1

**DEAL WITH REQUEST BODY:**

Parsing Request body using body-parser middleware.

Install body-parser middleware using npm **install body-parser –save.**

var express = require('express')

var bodyParser = require('body-parser')

const port = 3000;

var app = express()

app.use(bodyParser.urlencoded({ extended: false })); // parse application/x- www-form-urlencoded

app.use(bodyParser.json());// parse application/json

app.post("/",function (req, res) {

res.setHeader('Content-Type', 'application/json');

res.write('you posted:\n')

res.end(JSON.stringify(req.body, null, 2))

});

app.listen(port, () => console.log(`Example app listening on port ${port}!`))

**Testing url for above code snippet:**

**URL:** [**http://localhost:3000**](http://localhost:3000)

**HTTP Method :** POST

**REQUEST BODY :** {

"name" :"gopi"

}

**Routes:**

By Using Routes we can Segregate api’s into different files instead of writing in one file called app.js.

In Below example we are writing API’s in a separating file called **api.js** by using **Router** and mounting with help of **app.use.**

**app.js**

// Bring in our dependencies

const app = require('express')();

const routes = require('./api');

// Connect all our routes to our application

app.use('/api', routes);

// Turn on that server!

app.listen(3000, () => {

console.log('App listening on port 3000');

});

**api.js**

// Bring in our dependencies

const app = require('express')();

const routes = require('./api');

// Connect all our routes to our application

app.use('/api', routes);

// Turn on that server!

app.listen(3000, () => {

console.log('App listening on port 3000');

});

Start application using **node** **app.js**

**Testing url for above code snippet:**

**URL:** [**http://localhost:3000**/api](http://localhost:3000/api)