

Express-js

Express Js

Express is fast, Unopinionated , minimalist web framework for Node.js

- Express is a lightweight and flexible routing framework with minimal core features meant to be augmented through the use of Express middleware modules.
- Express is Unopinionated means it doesn't enforce a specific structure, giving you flexibility in how you design and organize your application.

What is middleware?

- the work between request submission from client and response sent by the server is handled by middleware. like routing, parsing request bodies, handling authentication, logging etc.,

Install express module from npm

> npm i(install) express

```
const express = require('express') // Creates an Express application
```

```
const app = express() // The express() function is a top-level function exported by the express modu
```

Server creation with or without express

Server creation without express

```
const http=require('http');

const server=http.createServer((req,res)=>{
    // handle request and send responses//res.end("hello client i am listening");
});

server.listen(3000,'localhost')=>{console.log('server is listening on port 3000');});
```

Server creation with express

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

// handle request and send responses by creating endpoints

app.listen(3000,'localhost')=>{console.log('server is listening on port 3000');}
```

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:

- app is an instance of express.
- METHOD is an HTTP request method, in lowercase.
- PATH is a path on the server.
- HANDLER is the function executed when the route is matched.

Example

```
app.get('/', (req, res) => { res.send('Hello World!') })    Respond to get request from route '/'  
  
app.post('/', (req, res) => { res.send('Got a POST request') })    Respond to post request from route '/'  
  
app.put('/user', (req, res) => { res.send('Got a PUT request at /user') })    Respond to put request from route '/user'  
  
app.delete('/user', (req, res) => { res.send('Got a DELETE request at /user') })    Respond to delete request from route '/user'
```

Route
handling
without
express

```
const http=require('http');
const url=require('url');
const server=http.createServer((req,res)=>{
let parsedURL=url.parse(req.url,true);
let path=parsedURL.pathname;
let payload="";
  switch(path){
    case '/': console.log("home page");
              payload+="requested home page";
              break;
    case '/login': console.log("request login page");
                  payload+="requested login page";
                  break;
    case '/register': console.log("requested registration page");
                     payload+="requested register page";
                     break;
    default: console.log("page not found");
            payload+="requested page not found";
  }
  res.writeHead(200,"ok",{ "Content-Type":"text/plain"});
  res.write(payload+"\n\n");
  res.end();
});
server.listen(8082,'localhost',()=>console.log('server is listening
on port no 8082'));
```

Route handling using express

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

```
app.get("/",(req,res)=>{  
  console.log("requested home page");  
  res.writeHead(200,"ok",{ "Content-Type":"text/plain"});  
  res.end("requested home page using express");  
});  
app.get("/login",(req,res)=>{  
  console.log("requested login page");  
  res.writeHead(200,"ok",{ "Content-Type":"text/plain"});  
  res.end("requested login page using express");  
});  
app.get("/register",(req,res)=>{  
  console.log("requested register page");  
  res.writeHead(200,"ok",{ "Content-Type":"text/plain"});  
  res.end("requested register page using express");  
});
```

```
app.listen(1234,()=>{console.log('server is listening on port no 1234 at localhost');});
```

Handling requests without express

```
const http= require('http');
const url=require('url');
// for decoding buffer objects into string , that preserves encoded UTF-8/16
const { StringDecoder } = require('string_decoder');
const server = http.createServer((req,res)=>{
  let parsedURL=url.parse(req.url,"true");
  let buffer="";
  let decoder=new StringDecoder('utf-8');
  res.writeHead(200,"OK",{ "Content-Type":"text/plain"});
  res.write("Response is \n\n");
  if(req.method==='GET'){
    let payLoad= parsedURL.query; // retrieve the query string from parsedurl
    console.log(payLoad); // display it on console
    res.write(JSON.stringify(payLoad)); // send it to requested console
    res.end("\n\n The end of message to browser for get request\n\n");
  }
});
```

```
else (req.method==='POST') {
  req.on('data',(chunk)=>{
    buffer+=decoder.write(chunk);
  });
  req.on('end',()=>{
    buffer+=decoder.end();
    console.log(buffer);
    res.write(buffer);
    res.end("\n\n The end of message to browser for post request\n\n");
  });
}
else{
  console.log("invalid request");
  res.end("\n\nInvalid request \n\n");
}
});
server.listen(8081,'localhost',
  ()=>{console.log('server is listening on 8081 from localhost');}
);
```