**Express.js**

خب [express.js](https://expressjs.com/) همانطور که در سایت خودش نوشته یک فریمورک سریع ، بی درد سر و سبک وزن یا مینیمال برای [نود جی اس](https://nodejs.org/en/) است .

کار اکسپرس این است که یک ابزاری برای دستان برنامه نویسان وب بسازد که ساخت سایت با Node.js را راحت بکند .

express.js بهترین گزینه برای استارتاپ هایی است که میخواهند سریعا یک محصول تولید کنند و زیاد درگیر مسائل پیچیده کدنویسی نشوند .

باید این نکته را بدونید که پیش نیاز  express.js فهمیدن Node.js است.

محیط Node.js بخشی اساسی برای ساخت و استقرار express.js است . تقریبا میشه گفت Node.js یک مفسر کراس - پلتفرم برای جاوا اسکریپت است  که قادر به اجرای جاوا اسکریپت بصورت نامحدود برای مرورگرهاست .

در واقع Node.js به جاوا اسکریپت کمک میکند تا بتواند در سمت سرور هم نقش بازی کند و بعنوان یک زبان برنامه نویسی سمت سرور بجای زبان های PHP, Python, Java و ... اانتخاب شود .

توسط چند خط ساده شما میتوانید یک سرور درست کنید و به مرورگر بگید در ادرس  localhost:3000 بهش دسترسی داشته باش !

|  |  |
| --- | --- |
| 3  4  5  6  7  8  9 | var express = require('express')  var app = express()    app.get('/', function (req, res) {    res.send('Hello World!')  })  app.listen(3000, function () {    console.log('Server is listening at 3000')  }) |

* [Node.js](https://7learn.com/programming/what-is-nodejs): محیط اجرایی جاوااسکریپت است و اجازه می‌دهد کدهای جاوااسکریپت را در سرور اجرا کنید.
* Express.js: یک فریم ورک سمت سرور برای نود جی اس است.
* [MongoDB](https://7learn.com/tutorials/what-is-mongo-db): یک پایگاه داده NoSQL و چند سکویی است.

**چرا از فریم‌ورک اکسپرس (Express js) استفاده کنیم؟**

1. روتینگ یا مسیرنویسی را بسیار ساده می‌کند.
2. به دلیل اینکه برپایه‌ی نود جی‌ اس بنا شده، بسیار سریع است.

REST API: Representational State Transfer

REST ساختاری است که مجموعه‌ای از قوانین را برای توسعه و استفاده از وب سرویس‌ها بر بستر HTTP تعریف می‌کند. REST API هم ساده است و به سادگی اجرا می‌گردد، هم اینکه بدون وابستگی به سکویی قابلیت اجرایی با هر زبان برنامه نویسی را دارد.

**Express**-**generator** is an all-in-one tool that allows you to quickly create a node-js application using the **express** library. Thanks to this **generator**, it will be possible to specify the template (view), the CSS preprocessor, etc. We want to install **express**-**generator**.

**Express Routes**

* In the**locationRouter.js,** create an Express router with the name of locationRouter.
* Handle **GET**and **POST**requests to the '/' path for locationRouter. For the **GET**request, simply return all the locations. Use the**res.send()** method to handle returning an array, rather than res.end().
* For the **POST**request, this is where you will deal with submissions of the HTML form in **server.js**. You will look for the **location**property from the request **body**and add it to the locations array, then return the locations array as a response.
* Be sure to **export**the locationRouter from the module.

**locationRouter.js**

const locations = ["Tacoma", "Marysville", "Seattle", "Spokane", "Bellingham"];

const express = require("express");

const bodyParser = require("body-parser");

const **locationRouter** = express.Router();

locationRouter.use(bodyParser.json());

locationRouter.route("/")

.all((req, res, next) => {

res.statusCode = 200;

res.setHeader("Content-Type", "text/plain");

next();

})

.get((req, res) => {

res.send(locations);

})

.post((req, res) => {

locations.push(req.body.location);

res.send(locations);

});

module.exports = locationRouter;

**server.js:**

const express = require('express');

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

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

const app = express();

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

app.use('/locations', locationRouter);

app.get('/', (req, res) => {

console.log(req.headers);

res.setHeader('Content-Type', 'text/html');

res.status(200).send(`

<html>

<body>

<form action='/locations' method='post'>

<label for='location'>Add a location:

<input type='text' name='location' />

</label>

<button type='submit'>Submit</button>

</form>

</body>

</html>

`);

});

const listener = app.listen(process.env.PORT, function() {

console.log('Your app is listening on port ' + listener.address().port);

});

/\* Forked from https://codesandbox.io/s/oqprp9m6oy \*/