

Express.js

Setting up basic project in Express.js

Ashish Alok
Kunal Singh Chauhan
Shubhankar Shankar

Introduction

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. It is an open source framework developed and maintained by the Node.js foundation.



Advantages of Express.js

Makes Node.js web application development fast and easy.

Easy to configure and customize.

Allows you to define routes of your application based on HTTP methods and URLs.

Includes various middleware modules which you can use to perform additional tasks on request and response.

Easy to serve static files and resources of your application.

Allows you to create REST API server.

Easy to connect with databases such as MongoDB, Redis, MySQL



Installing Express.js


```
npm install -g express
```



Listening on Port

JS data.js

JS index.js X

JS index.js >  app.delete('/api/student/:id') callback

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

```
2  const student=require('./data')
```

```
3
```

```
4  const app = express()
```

```
5  app.use(express.json()) // this will make the body as object and set it to its property
```

```
6
```

```
7  app.listen(3000, () => {
```

```
8    |    console.log('Listening On Port 3000');
```

```
9  |  })
```

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

```
11  |    res.json({ message: "API is Working" })
```

```
12  |  })
```

```
13
```

HTTP GET Request

```
// get |  
app.get('/api/student', (req, res) => {  
  res.json(student)  
})
```



HTTP POST Request

```
// post
app.post('/api/student', (req, res) => {
  const user = {
    id : student.length+1,
    first_name : req.body.first_name,
    last_name : req.body.last_name,
    email : req.body.email
  }
  student.push(user)
  res.json(user)
})
```



HTTP PUT Request

```
32 // update data
33 app.put('/api/student/:id', (req, res) => {
34   let id = req.params.id
35   let first_name = req.body.first_name
36   let last_name = req.body.last_name
37   let email = req.body.email
38   let index = student.findIndex((student) => {
39     return (student.id == Number.parseInt(id))
40   })
41
42   if (index >= 0){
43     let std = student[index]
44     std.last_name = last_name
45     std.first_name = first_name
46     std.email = email
47     res.json(std)
48   } else {
49     res.status(404)
50     res.end()
51   }
52 })
53
```


HTTP DELETE Request

```
// delete data
app.delete('/api/student/:id', (req, res) => {
  let id = req.params.id;
  let index = student.findIndex((student) => {
    return (student.id == Number.parseInt(id))
  })
  if (index >= 0) {
    let std = student[index]
    student.splice(index, 1)
    res.json(std)
  } else {
    res.status(404)
  }
})
```