

Intro to Express; Web Servers; Routing; Serving Frontend Code (HTML, CSS); Serving static files and images; Middlewares; Setting up a simple REST API and testing using Postman Assignment Solutions



Assignment Solutions

Problem

Let us create a simple Express server and use Morgan Middleware.

Solution Code

```
var expressJs = require("express");  
var logger_morgan = require("morgan");  
  
var expressApp = expressJs();  
  
expressApp.use(logger_morgan("short"));  
  
expressApp.listen(3000);
```

Testing

1. Start the server and let us hit any endpoint on the port. We should get custom morgan logs.

```
:::1 - GET / HTTP/1.1 404 139 - 2.932 ms
```

Add error logic in API's.

Problem

If anyone creates a user with a repeating userid return 400 code with an Error message.

Eg if we already have a user with id=2, we can't create a new user with different details but the same id.

Solution Code

```
const expressJs = require('express')
const bodyParser = require('body-parser');

const expressApp = expressJs();

// using as a proxy for database
let users = [{"id": "1", "name": "tushar Raina",
"email": "rainatushar22@gmail.com"}
, {"id": "2", "name": "Sachin tendulkar", "email": "sachin112@gmail.com"}];

// Configuring body parser middleware
expressApp.use(bodyParser.urlencoded({ extended: false }));
expressApp.use(bodyParser.json());

//Returns list of all the users
expressApp.get('/users', (request, response) => {
  response.json(users).status(200);
});

//Deletes user with given id
expressApp.delete('/user/:id', (req, res) => {
  const id = req.params.id;
  users = users.filter(user => {
    if (user.id !== id) {
      return true;
    }
    return false;
  });
});
```

```
    res.send('User is deleted').status(400);
  });

//adds a new user
expressApp.post('/user', (req, res) => {
  const user = req.body;
  for(count in users){
    if(users[count]["id"] === user.id){
      res.status(400).send('User cant be added to the database');
      return;
    }
  }
  users.push(user);
  res.send('User is added to the database').status(200);
});

//Listens to given Port
expressApp.listen(3000,
  () => console.log(`Listening on port 3000`));
```