Learn The MERN Stack

youtube playlist

Table of Contents

1 Express & MongoDB Rest API

1.1 Preconditions

Create a new repository in gitHub. Clone your new empty repo to your local computer. Start a new npm project:

CLI=> *npm init -y*

Create backend folder to store all the files that related to backend logic.

Create a main file server.js

1.2 Installing 3rd party libraries

CLI=> *npm i express dotenv mongoose colors*

CLI=> *npm i -D nodemon*

1.3 Change default run scripts

Open package.json file (it should be in root folder of your project)

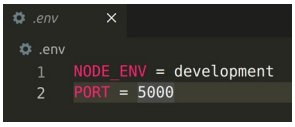
```
"scripts": {
  "start": "node backend/server.js",
  "server": "nodemon backend/server.js"
 },
```

Now you can run your server using next command:

CLI=> *npm run server*

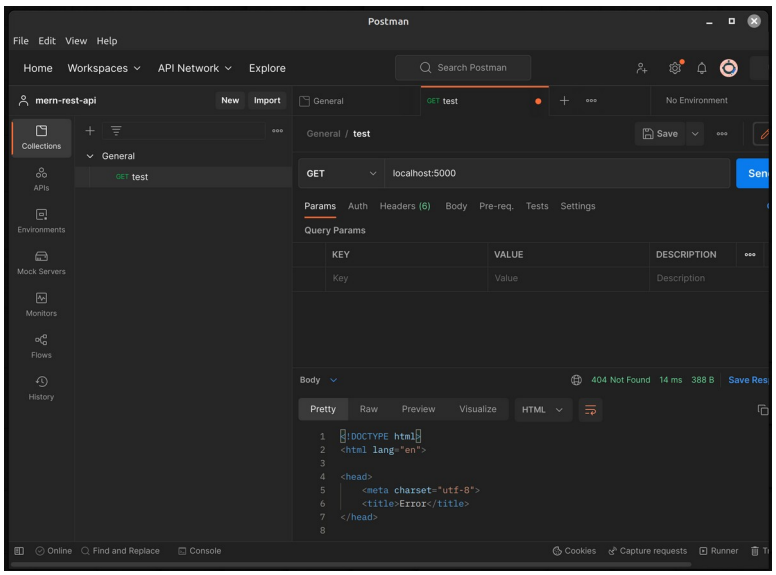
1.4 Create a simple express app

```
const express = require('express');
const dotenv = require('dotenv').config();
const port = process.env.PORT || 5000;
const app = express();
app.listen(port, () => console.log(`Server was starterd on port ${port}`));
```



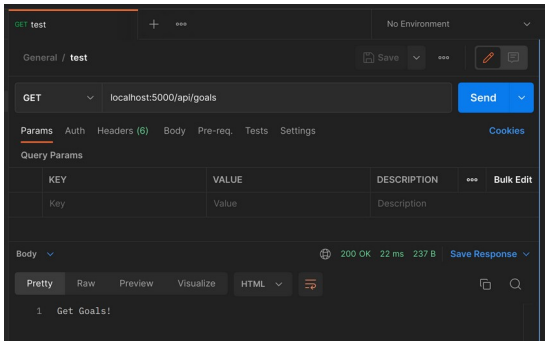
1.5 Postman

Download Postman. Create a new collection and a first test request.



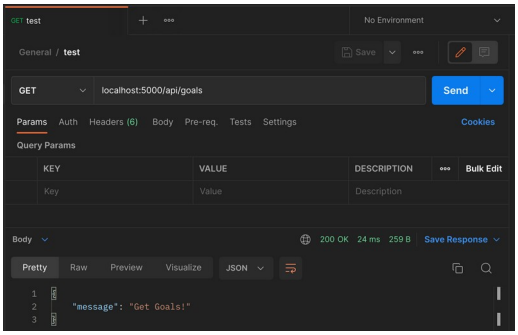
1.6 Create a simple test route with text response

```
JS server.js M ×
backend > JS server.js > ...
1  const express = require('express');
2  const dotenv = require('dotenv').config();
3  const port = process.env.PORT || 5000;
4
5  const app = express();
6
7  app.get('/api/goals', (req, res) => {
8    res.send('Get Goals!');
9  });
10
11 app.listen(port, () => console.log(`Server was starterd on port ${port}`));
```



1.7 Setup a response (as a JSON object) and a status code

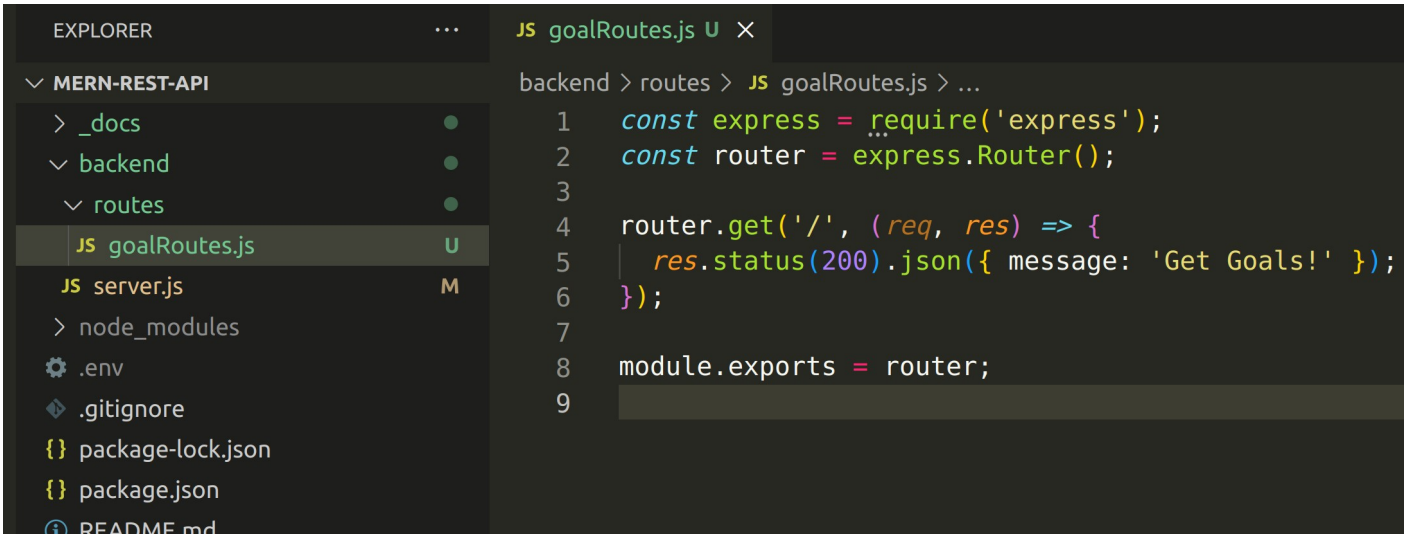
```
JS server.js M ×
backend > JS server.js > ...
1  const express = require('express');
2  const dotenv = require('dotenv').config();
3  const port = process.env.PORT || 5000;
4
5  const app = express();
6
7  app.get('/api/goals', (req, res) => {
8    res.status(200).json({ message: 'Get Goals!' });
9  });
10
11 app.listen(port, () => console.log(`Server was starterd on port ${port}`));
```

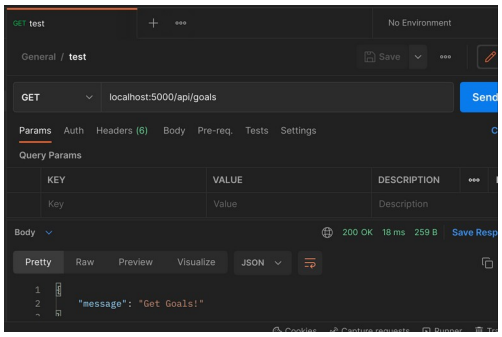


1.8 Split out all the routes into different files (entities)

Create a new folder backend/routes to store a few main routes (for each of main entities).

Create a first route file for Goal entity.





1.9 Add different routes for Goal entity

goalRoutes.js U

backend > routes > goalRoutes.js > ...

```javascript
const express = require('express');
const router = express.Router();

router.get('/', (req, res) => {
  res.status(200).json({ message: 'Get all the Goals!' });
});

router.post('/', (req, res) => {
  res.status(201).json({ message: 'Set a new Goal!' });
});

router.put('/:id', (req, res) => {
  res
    .status(200)
    .json({ message: `Update a Goal with ID = '${req.params.id}'` });
});

router.delete('/:id', (req, res) => {
  res
    .status(200)
    .json({ message: `Delete a Goal with ID = '${req.params.id}'` });
});
module.exports = router;
```

1.10 Add Goal controller

Create a new folder backend/controllers to store all the controllers. Create a new file backend/controllers/goalController.js

JS server.js M ×

backend > JS server.js > ...

```js
const express = require('express');
const dotenv = require('dotenv').config();
const port = process.env.PORT || 5000;

const app = express();

app.use('/api/goals', require('./routes/goalRoutes'));

app.listen(port, () => console.log(`Server was starterd on port ${port}`));
```

JS goalRoutes.js U ×

backend > routes > JS goalRoutes.js > ...

```js
const express = require('express');
const router = express.Router();
const {
  getGoals,
  setGoal,
  updateGoal,
  deleteGoal,
} = require('../controllers/goalController');

router.route('/').get(getGoals).post(setGoal);
router.route('/:id').put(updateGoal).delete(deleteGoal);

module.exports = router;
```

JS goalController.js U ×

backend > controllers > JS goalController.js > ...

```js
// @desc    Get all the Goals
// @route   GET to /api/goals
// @access  Private
const getGoals = (req, res) => {
  res.status(200).json({ message: 'Get all the Goals!' });
};

// @desc    Create a new Goal
// @route   POST to /api/goals
// @access  Private
const setGoal = (req, res) => {
  res.status(201).json({ message: 'Set a new Goal!' });
};

// @desc    Update a Goal by ID
// @route   PUT to /api/goals/:id
// @access  Private
const updateGoal = (req, res) => {
  res
    .status(200)
    .json({ message: `Update a Goal with ID = '${req.params.id}'` });
};

// @desc    Delete a Goal by ID
// @route   DELETE to /api/goals/:id
// @access  Private
const deleteGoal = (req, res) => {
  res
    .status(200)
    .json({ message: `Delete a Goal with ID = '${req.params.id}'` });
};

module.exports = { getGoals, setGoal, updateGoal, deleteGoal };
```