

REST-API mit Node, Express und MongoDB

1 Installation von Node.js, MongoDB

Download

nodejs.org, mongodb.com

Test

```
$ node -v          v10.15.0
$ npm -v          6.4.1
$ mongo --version MongoDB shell version v4.0.5 (evtl. PATH setzen)
```

2 Projekt erstellen

```
$ npm init  erstellt im aktuellen Verzeichnis die Datei package.json
```

3 Hallo Welt in Konsole

index.js

```
console.log('Hallo Welt!');
```

Script starten

```
$ node index
```

4 Webserver nur mit Node

index.js

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

http.createServer(function(request, response) {
  response.writeHead(200, {'Content-Type': 'text/plain'});
  response.end('Hallo Welt!');
}).listen(3000);

console.log('Server läuft auf Port 3000');
```

Script starten

```
$ node index.js
```

Mit Browser auf Webserver zugreifen

```
http://localhost:3000
```

5 Webanwendung mit express

express installieren

\$ npm install express --save Installation im aktuellen Verzeichnis, Dependency in package.json

index.js

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

app.get('/', function(req, res) {
  res.send('Hallo Welt!');
});

app.listen(3000, function() {
  console.log('Server läuft auf Port 3000');
});
```

6 API-Routes

api-routes.js

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

router.get('/', function (req, res) {
  res.json({
    status: 'API funktioniert',
    message: 'Willkommen zur REST-API!',
  });
});

module.exports = router;
```

zu index.js hinzufügen:

```
const apiRoutes = require('api-routes');
app.use('/api', apiRoutes);
```

Zugriff

```
http://localhost:3000      Hallo Welt!
http://localhost:3000/api  {'status': 'API funktioniert', 'message': 'Willkommen zur REST-API!'}
```

7 MongoDB

MongoDB starten

```
$ mongod
```

mongoose und body-parser installieren

```
$ npm install mongoose --save      MongoDB object modeling  
$ npm install body-parser --save   Parse incoming request bodies
```

index.js

```
const express = require('express');  
const bodyParser = require('body-parser');  
const mongoose = require('mongoose');  
const apiRoutes = require('./api-routes');  
  
const app = express();  
  
app.use(bodyParser.urlencoded({  
  extended: true  
}));  
app.use(bodyParser.json());  
  
mongoose.connect('mongodb://localhost/mongorest', { useNewUrlParser: true });  
let db = mongoose.connection;  
  
app.get('/', function(req, res) {  
  res.send('Hallo Welt!');  
});  
  
app.use('/api', apiRoutes);  
  
app.listen(3000, function() {  
  console.log('Server läuft auf Port 3000');  
});
```

8 Controller und Model

model.js

```
var mongoose = require('mongoose');

// Setup schema
var contactSchema = mongoose.Schema({
  name: {
    type: String,
    required: true
  },
  email: {
    type: String,
    required: true
  },
  gender: String,
  phone: String,
  create_date: {
    type: Date,
    default: Date.now
  }
});

// Export Contact model
var Contact = module.exports = mongoose.model('contact', contactSchema);

module.exports.get = function (callback, limit) {
  Contact.find(callback).limit(limit);
}
```

controller.js

```
Contact = require('./model');
exports.index = function (req, res) {
  Contact.get(function (err, contacts) {
    if (err) {
      res.json({ status: "error", message: err });
    }
    res.json({ status: "success", message: "Contacts retrieved successfully", data: contacts });
  });
};

exports.new = function (req, res) {
  var contact = new Contact();
  contact.name = req.body.name ? req.body.name : contact.name;
  contact.gender = req.body.gender;
  contact.email = req.body.email;
  contact.phone = req.body.phone;

  contact.save(function (err) {
    res.json({ message: 'New contact created!', data: contact });
  });
};

exports.view = function (req, res) {
  Contact.findById(req.params.contact_id, function (err, contact) {
    if (err)
      res.send(err);
    res.json({ message: 'Contact details loading..', data: contact });
  });
};

exports.update = function (req, res) {
  Contact.findById(req.params.contact_id, function (err, contact) {
    if (err)
      res.send(err);

    contact.name = req.body.name ? req.body.name : contact.name;
    contact.gender = req.body.gender;
    contact.email = req.body.email;
    contact.phone = req.body.phone;

    contact.save(function (err) {
      if (err)
        res.json(err);
      res.json({ message: 'Contact Info updated', data: contact });
    });
  });
};

exports.delete = function (req, res) {
  Contact.remove({
    _id: req.params.contact_id
  }, function (err, contact) {
    if (err)
      res.send(err);

    res.json({ status: "success", message: 'Contact deleted' });
  });
};
```

Quellen

- How To Build Simple RESTful API With NodeJs, ExpressJs And MongoDB