MEAN Application

Agenda

- Create Angular App
- Replacing Server with Express.js
- Installling Mongoose.js
- Creating Student Model
- Creating Routes
- Crating Angular Component
- Creating Angular Routes
- Student List

1) Update Angular CLI and Create Angular 6 Application

- **New App using Angular CLI and serve**
  - ng new students
  - cd students
  - ng serve

2) Replacing Web Server with Express.js

- **Adding Express.js modules and its dependencies.**
  - npm install --save express body-parser morgan body-parser serve-favicon
- **Add bin folder and www file inside bin folder. Www contains node enviornment information and starting point of app**
  - mkdir bin
  - vi bin/www

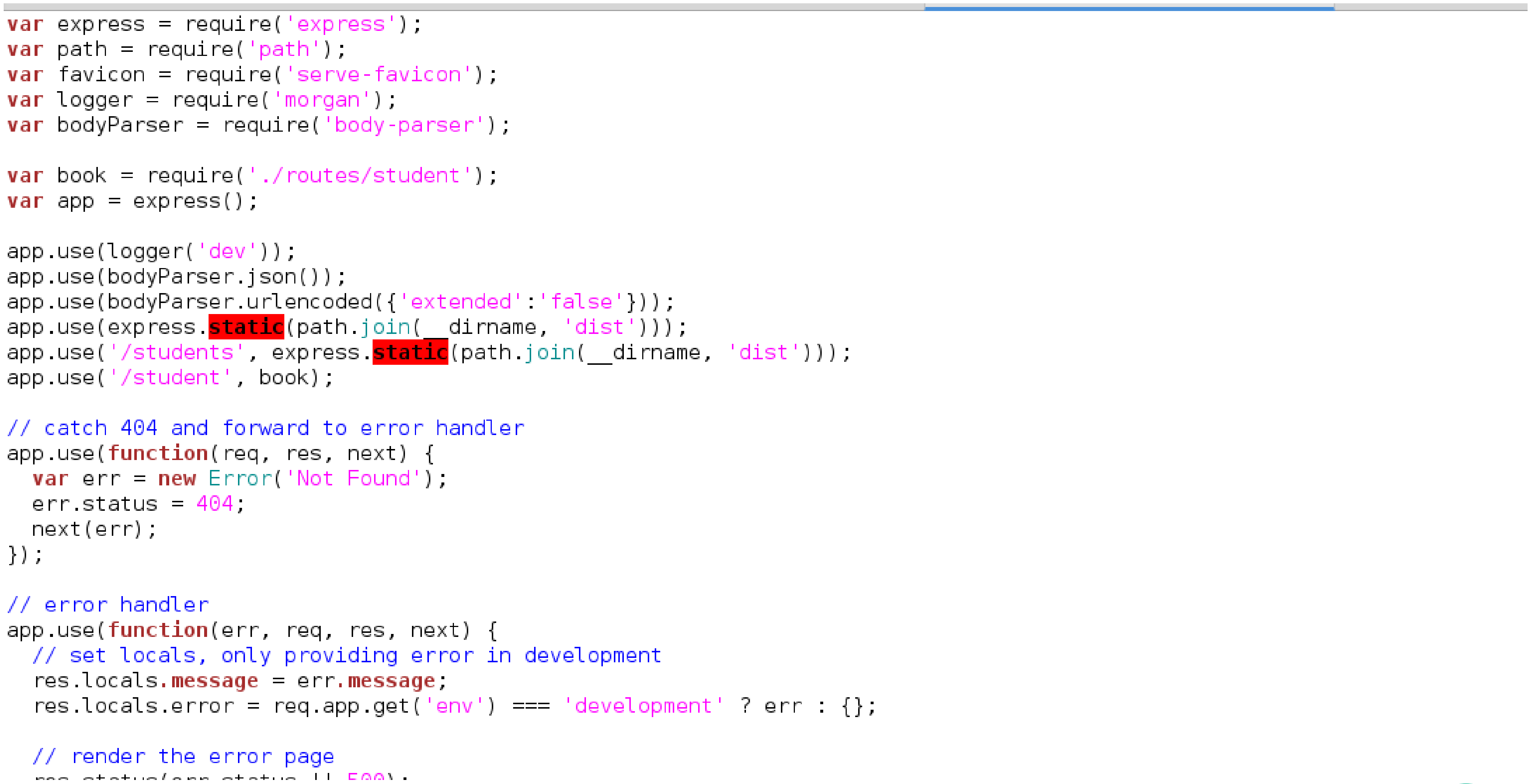
Www file

- **To make the server run from bin/www, open and edit "package.json" then replace "start" value.**
  - "scripts": {
  -     "ng": "ng",
  -     "start": "ng build && node ./bin/www",
  -     "build": "ng build",
  -     "test": "ng test",
  -     "lint": "ng lint",
  -     "e2e": "ng e2e"
  -   }

App.js

- Create app.js in root folder and give properties.

App.js

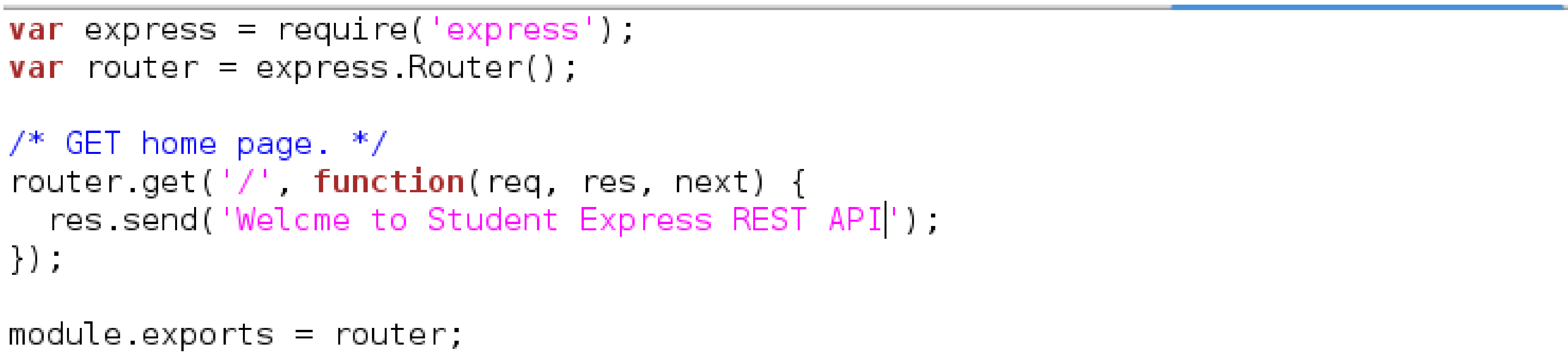


```
var express = require('express');
var path = require('path');
var favicon = require('serve-favicon');
var logger = require('morgan');
var bodyParser = require('body-parser');

var book = require('./routes/student');
var app = express();

app.use(logger('dev'));
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({'extended':'false'}));
app.use(express.static(path.join(__dirname, 'dist')));
app.use('/students', express.static(path.join(__dirname, 'dist')));
app.use('/student', book);

// catch 404 and forward to error handler
app.use(function(req, res, next) {
  var err = new Error('Not Found');
  err.status = 404;
  next(err);
});

// error handler
app.use(function(err, req, res, next) {
  // set locals, only providing error in development
  res.locals.message = err.message;
  res.locals.error = req.app.get('env') === 'development' ? err : {};

  // render the error page
  res.status(err.status || 500);
```

Routes

- **Create routes folder and routes file for students**
  - mkdir routes
  - vi routes/student.js

Student.js



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

/* GET home page. */
router.get('/', function(req, res, next) {
  res.send('Welcme to Student Express REST API');
});

module.exports = router;
```

3) Installing and Configuring Mongoose.js

- **Mongoose.js is ORM for node and mongoDb**
  - npm install --save mongoose bluebird
  - In app.js
- **//Mongoose properties and connection verification message**
  - //Set up mongoose connection
  - var mongoose = require('mongoose');
  - var mongoDB = 'mongodb://127.0.0.1/students';
  - mongoose.connect(mongoDB);
  - mongoose.Promise = global.Promise;
  - var db = mongoose.connection;
  - db.on('error', console.error.bind(console, 'MongoDB connection error:'));

4) Create Student Model

- **Create models in root dir and student.js in models**
  - var mongoose = require('mongoose');
  - var StudentSchema = new mongoose.Schema({
  -   id: String,
  -   name: String,
  -   subject: String,
  -   description: String,
  -   join_year: String,
  -   address: { street:String, state: String },
  -   updated_date: { type: Date, default: Date.now },
  - });
  - module.exports = mongoose.model('Student', StudentSchema);

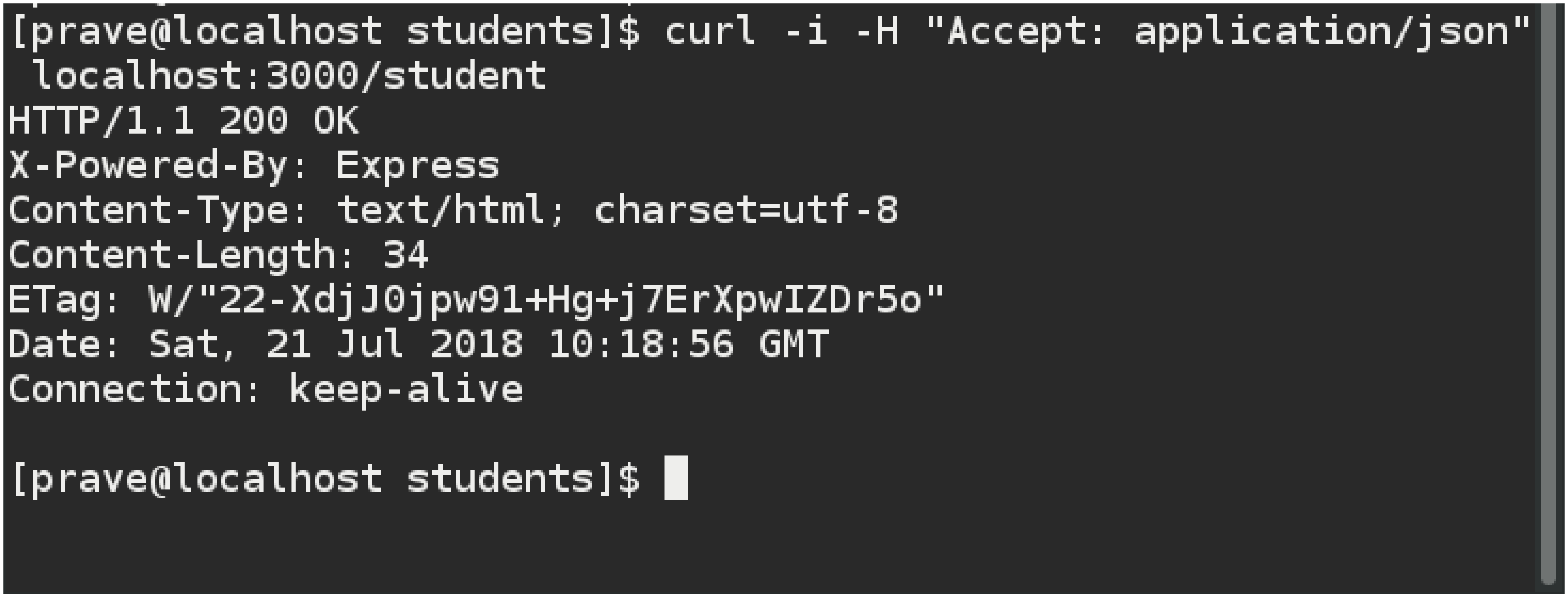
5) Creating Routes for Accessing Restful Student data

```
• var express = require('express');
• var router = express.Router();
• var mongoose = require('mongoose');
• var Student = require('../models/student.js');
•
• /* GET ALL Students */
• router.get('/', function(req, res, next) {
•   Student.find(function (err, products) {
•     if (err) return next(err);
•     res.json(products);
•   });
• });
•
• /* GET SINGLE Student BY ID */
• router.get('/:id', function(req, res, next) {
•   Student.findById(req.params.id, function (err, post) {
•     if (err) return next(err);
•     res.json(post);
•   });
• });
```

```
/* SAVE Student */
router.post('/', function(req, res, next) {
  Student.create(req.body, function (err, post) {
    if (err) return next(err);
    res.json(post);
  });
});

/* UPDATE Student */
router.put('/:id', function(req, res, next) {
  Student.findByIdAndUpdate(req.params.id, req.body, function (err, post) {
    if (err) return next(err);
    res.json(post);
  });
});

/* DELETE Student */
router.delete('/:id', function(req, res, next) {
  Student.findByIdAndRemove(req.params.id, req.body, function (err, post) {
    if (err) return next(err);
    res.json(post);
  });
});

module.exports = router;
```

Check using cURL

- **curl -i -H "Accept: application/json" localhost:3000/student**

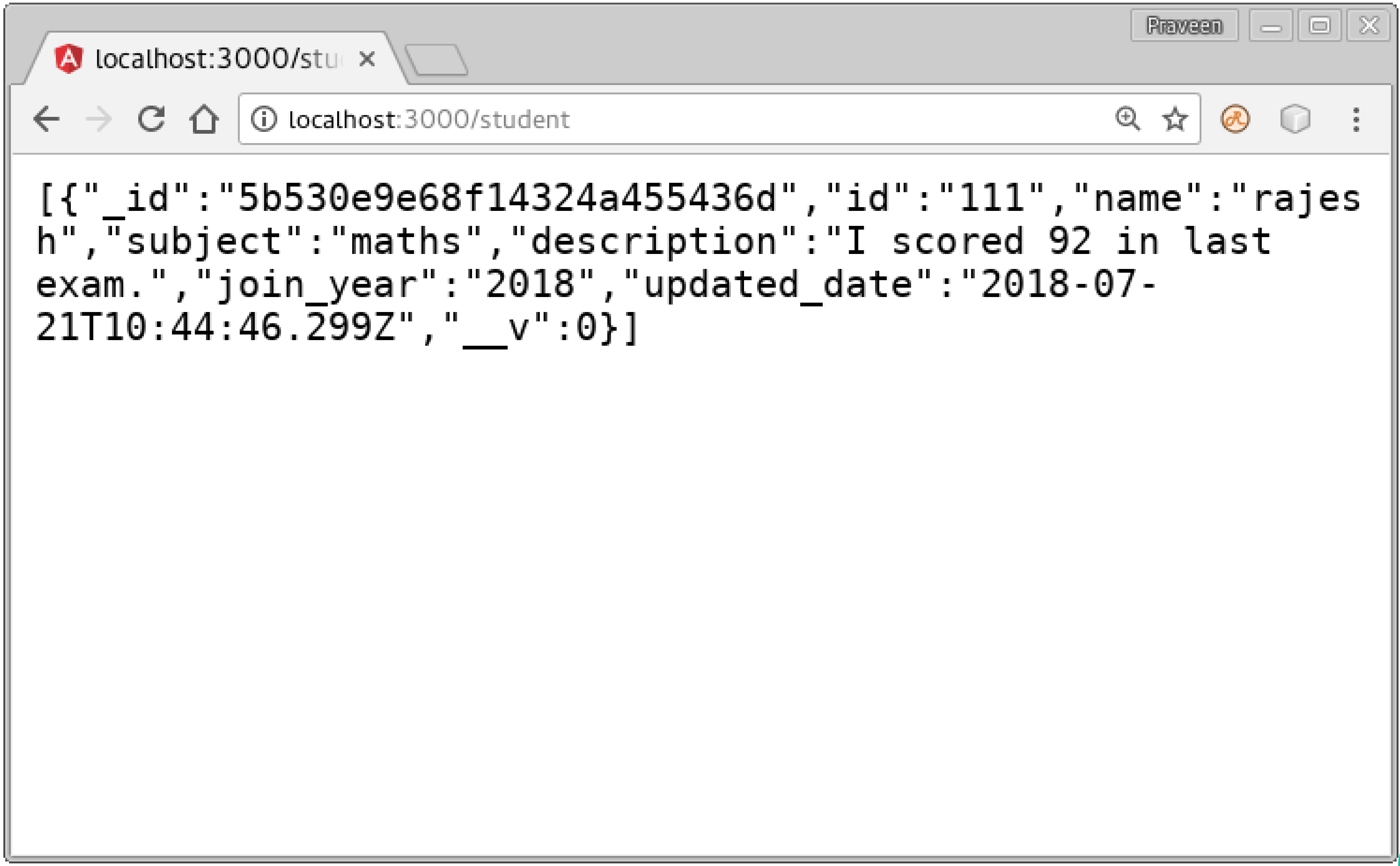


```
[prave@localhost students]$ curl -i -H "Accept: application/json"
 localhost:3000/student
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: text/html; charset=utf-8
Content-Length: 34
ETag: W/"22-XdjJ0jpw91+Hg+j7ErXpwIZDr5o"
Date: Sat, 21 Jul 2018 10:18:56 GMT
Connection: keep-alive

[prave@localhost students]$ 
```

Populate Student collection

- **Populate Student collection with initial data sent from RESTful API.**
  - curl -i -X POST -H "Content-Type: application/json" -d '{ "id":"111","name":"rajesh","subject": "maths","description":"I scored 92 in last exam.","join_year":"2018","address":"Bangalore"}' localhost:3000/student

Refresh http://localhost:3000/student



```
[{"_id":"5b530e9e68f14324a455436d","id":"111","name":"rajes
h","subject":"maths","description":"I scored 92 in last
exam.","join_year":"2018","updated_date":"2018-07-
21T10:44:46.299Z","__v":0}]
```

6) Creating Angular Component

- **Creating Angular Component for Displaying Student List**
  - ng g component student
- **we need to add `HttpClientModule` and 'FormsModule' to `app.module.ts`**
  - import { FormsModule } from '@angular/forms';
  - import { HttpClientModule } from '@angular/common/http';
  - imports: [
  -     BrowserModule,
  -     FormsModule,
  -     HttpClientModule
  -   ]

- **In student.component.ts**
  - import { HttpClient } from '@angular/common/http';
  - constructor(private http: HttpClient) { }
  -   students: any;
  -   ngOnInit() {
  - this.http.get('/student').subscribe(data => {
  -     this.students = data;
  -   });
  -   }

Student.component.html

```
• <div class="container">
•   <h1>Student List</h1>
•   <table class="table">
•     <thead>
•       <tr>
•         <th>Name</th>
•         <th>Subject</th>
•         <th>Description</th>
•       </tr>
•     </thead>
•     <tbody>
•       <tr *ngFor="let student of students">
•         <td>{{ student.name }}</td>
•         <td>{{ student.subject }}</td>
•         <td>{{ student.description }}</td>
•       </tr>
•     </tbody>
•   </table>
• </div>
```

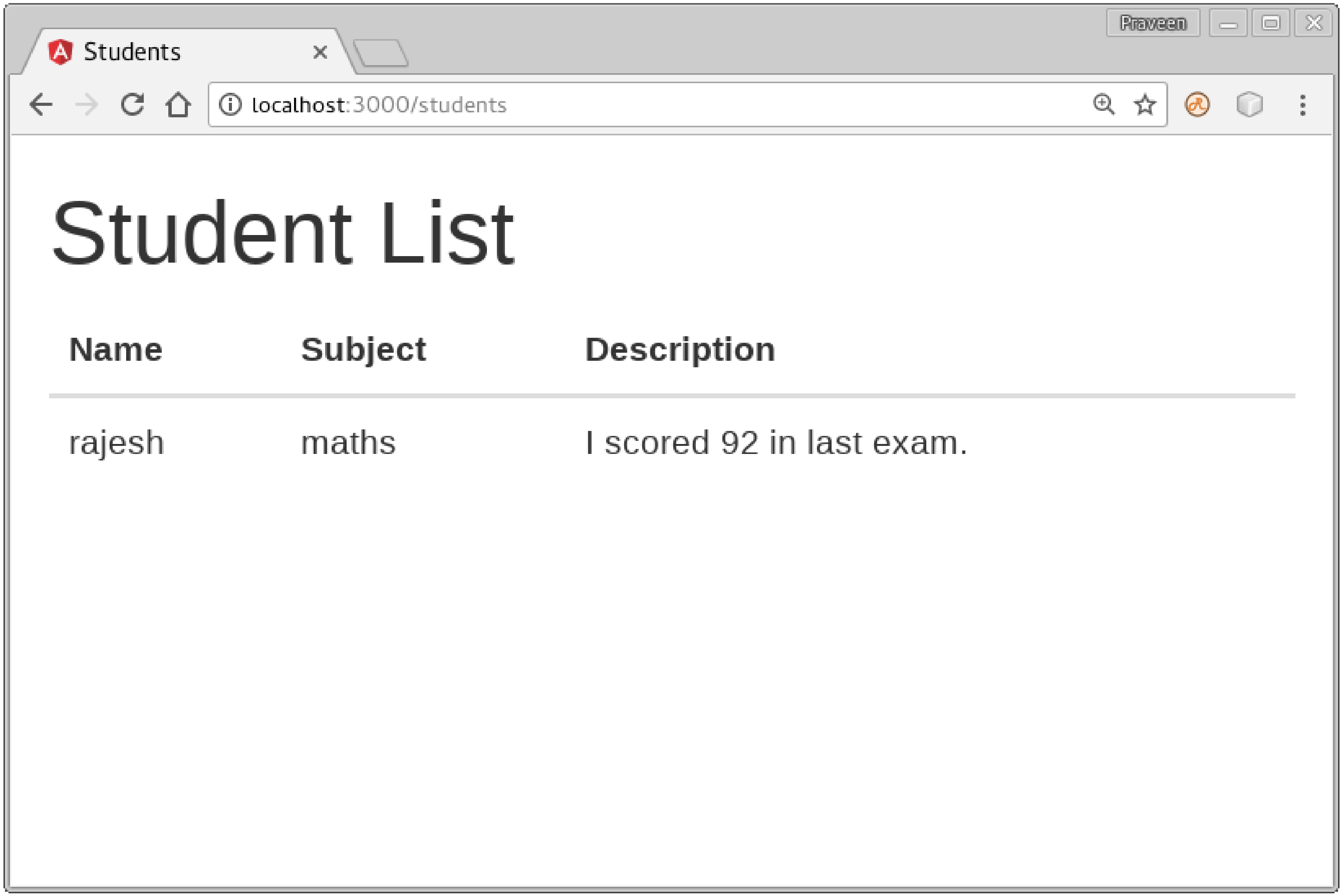
7) Creating Angular Routes to Student Component

- **In app.module.ts**
  - import { RouterModule, Routes } from '@angular/router';
  - const appRoutes: Routes = [
  -   {
  -     path: 'students',
  -     component: StudentComponent,
  -     data: { name: 'Student List' }
  -   },
  -   { path: '',
  -     redirectTo: '/students',
  -     pathMatch: 'full'
  -   }
  -];
  - imports: [
  -     BrowserModule,
  -     FormsModule,
  -     HttpClientModule,
  -     RouterModule.forRoot(
  -     appRoutes,
  -     { enableTracing: true }
  -   )
  -   ]

Angular Routes

- **In app.component.html**
  - <router-outlet></router-outlet>

Student List



Student List

Name	Subject	Description
rajesh	maths	I scored 92 in last exam.

Resources

- https://angular.io/
- https://expressjs.com/
- https://nodejs.org/
- https://www.mongodb.com