Assignment 9 (HF, major subject)

Due: Wed 23.12.2015; 12:00h (9 days)

Goals

After doing these exercises,

- You have basic knowledge of using a Mongo DB
- You can hook up an Express App with MongoDB

Task 1: Insert Data into a MongoDB — Difficulty: Easy

Insert the JSON file data.json from the GitHub repository into a mongoDB. We propose you call your database `mmn-<your_last_name>`, to make it easier for us to correct your submission.

a) Provide the full statement that you used.
b) Modify the note with the title “Christmas” and update it with your Christmas wishes. Provide the statement that is necessary for this.
c) Remove the “Example note” from the collection. Provide the statement.

Put all queries / statements in a .txt file and put it into the folder ‘task1’.

Task 2: NotesApp with NodeJS and MongoDB.

For assignment 5 and 6, you’ve already implemented a note-taking app with PHP and MySQL. Now we want to achieve the same with NodeJS, Express and MongoDB.

We provided a large code skeleton for this task, but you are welcome to modify what you have from assignment 06.

If you work with the code skeleton, you only need to adjust the URL of the database (default: `localhost/mmn`) and complete two middleware functions in the **`routes/notes.js`** module:

```
router.get('/', function (req, res) {…})
```

Here you fetch all the notes from the MongoDB. The response should look like this:

```
{
    "status" : "success",
    "message" : "fetched notes",
    "notes" : []
}
```

If there was an error, adjust the response accordingly. The status and message are mandatory.

```
router.post('/insert', function (req, res) {…})
```

Here you insert an arbitrary number of notes into the database. Each note contains a title and a content field.

Respond to the client about what happened, e.g.

```
{
    "status" : "success",
    "message" : "inserted notes",
    "inserted" : []
}
```

The inserted field is an array that should contain all documents that were inserted into the UI, i.e. something a like this:

```
{
    "_id" : "566fdb2c9b8629381cfff85e",
    "title" : "Note title",
    "content" : "Some content"
}
```

The client-side JavaScript in `public/index.html` will handle the response appropriately.

Put all your source code into the folder 'task2' (except for the `node_modules` folder). If you worked with the skeleton, the **`notes.js`** file is enough unless you modified more than this.

Submission

Put the solution to Task1 – Task2 into a ZIP folder. However, to keep the submission small, please **do not include the `node_modules` directory**. The correctors will do an `npm install` and do not need the `node_modules`. Please turn in your solution via UniWorX. You can form groups of up to three people. After the submission deadline, push your solution to our GitHub repository.