social computing

UNIVERSITÄT DUISBURG ESSEN

Open-Minded

Advanced Web Technologies
SS 2019

Assignment 3 – MongoDB

Submit your results into the “MongoDB” section in Moodle as a group.

Task 3.0: Setup environment

- Download “MongoDB Assignment Template” available in the Moodle “Assignment 3 (MongoDB)”.
- Extract the zip file which should contains “app.js” file and “models” folder
- Open a terminal in the folder containing “app.js” file and execute the following commands
  - npm init (execute only once)
    - package name: (mongodb-assignment template) <Type> mongodb-assignment <Press Enter>
    - version: (1.0.0) <Press Enter>
    - description: <Press Enter>
    - entry point: (app.js) <Type> app.js <Press Enter>
    - test command: <Press Enter>
    - git repository: <Press Enter>
    - keywords: <Press Enter>
    - author: <Press Enter>
    - license: (ISE) <Press Enter>
    - Is this OK? (yes) <Press Enter>
  - npm install --save express mongodb mongoose nodemon (execute only once)
  - nodemon app.js (this will start the server on port 3000)
  - Ctrl + C (to terminate the server)

Task 3.1: Create and Delete Operations

Create an endpoint (e.g. /GenerateDatabase) in Node / Express that will create two collections (*Students* and *Courses* as shown below) and fill them up with sample data. Every time the endpoint is accessed, the initial sample data should be restored in both the collections.

- Students
  - Matriculation number (_id)
  - First Name
  - Last Name
  - Age
  - Study Program (e.g.ISE, AI, Komedia)
  - Attending Courses (Reference)
- Courses
  - Long Title
  - Short Title
  - Professor
  - Type (e.g. Lecture, Seminar, Lab)
  - Credits
  - Language
  - Registered Students (Reference)

Task 3.2: Update and Read Operations

Create two endpoints (/Register and /Deregister) that requires two parameters (/<Student: Matriculation>/<Course: Short Title>). On registration, the specified course should be added to the Attending Courses list of the specified student and the Registered Students list of the course should be updated with the specified student. Similarly, generate the deregistration endpoint. After registration/deregistration, show the updated student information together with the attending courses information (not just id).

Task 3.3: Read Operation

Create an endpoint that requires two parameters <student: study program> and <course: short title>. Generate a list of all the students belonging to the specified study program and has taken the specified course (e.g. how many students from ISE has taken AdvWebTech course). The generated list should contain only Last name, First name, and Matriculation number. Also, the list should be sorted in ascending order based on the Matriculation number.

Task 3.4: Mongoose Package

This task has to be performed using the “mongoose” package of Node / Express. A sample code is available in the “MongoDB Assignment Template” to get you started with mongoose. Create the schema for the *Students* and *Courses* collections. Perform task 3.2 and task 3.3 again using the mongoose package.