JENSEN yrkeshögskola
TRÄNING FÖR VERKLIGHETEN

Course name: Frontend development - Server
Class: Frontend Developer Web Security
Term: Semester 1 HT23

REST API (ONLINE MARKET APPLICATION)

INTRODUCTION

Background description, question, delimitation and goals	Background: : Our client has just launched a webshop for his new brand "Drezztore" and has signed up a supplier agreement with a Chinese department store. The customer wants us to help them set up an API so that product management will be easier. In this task you will create a REST server to apply what we have learned about APIs, server programming and JSON Objective: A REST server that can receive GET, POST, PUT, and DELETE requests to manage products in an e-commerce application. The application must also have a route to render an html page with all products. Delimitation: You may implement this with any method (package). preferably.
Why should you do this work?	The purpose: • The student should gain an expanded understanding of how he creates a REST application with Node and NPM • The student must work with problem solving and design of JavaScript code according to good practice.
What will you deliver?	The following must be delivered: 1. The following should be delivered: 1. A folder with an NPM project (package.json etc…) a. Do not include /node_modules in projects 2. Video recording of review of application with Postman or Insomnia (max. 10 min) on all routes 3. Written report on use and various choices in design of The API

YOUR PROJECT TASK

What will you do? Create	a REST API for Node.js with Express that interacts with a MongoDB database for an e-commerce application (Note: application does not need a frontend, but the root route (/) should render html. No css is needed for this). Write a written report describing the use of the API

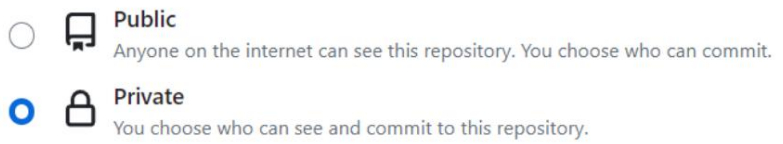
How are you going to solve it?

the task?

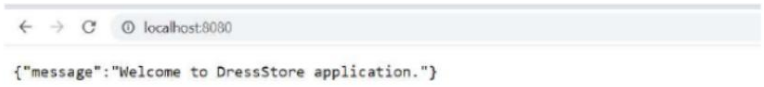
1. Create a folder with your project (ex: submission task) and add your desk
2. Open the terminal and navigate to the folder (cd ~/desktop/submission)
3. Run the command “npm init” in the folder to initialize an npm project
4. Now install the following npm packages (cors, express, mongoose) and/or others package that you want to use in the project
5. Add a file (.gitignore) to the folder with the following content

   node_modules

6. Now we can initialize a git repo and connect it to github. Stay in folder in the terminal and follow the instructions in the tutorial (https://www.digitalocean.com/community/tutorials/how-to-push-an-existing-project-to-github). NOTE: make sure the repo on github is private so that no one can see any MongoDB credentials etc.



7. Now you have a template to start working with your REST API. When you enter new code in the project don't forget to commit and push to github (you have to be in the folder in the terminal)
   a. git add .
   b. git commit -m “my commit message”
   c. git push
8. To get started, start by entering a route that gives the following message when you log in to http://localhost:3000



The following CRUD methods shall be supported by the API

Methods	Urls	Actions
GOAT	api/products	Download all products
GOAT	api/products/:id	Retrieving product with id
MAIL	api/products	Creates a new product
PUT	api/products/:id	Changes product with id
DELETE	api/products/:id	Removes product with id
DELETE	api/products	Removes all products
GOAT	api/products?name=[kw]	Gets product whose name contains "kw"
GOAT	/	Sends back html with

	<table><tr><td></td><td></td><td>all products in a table with all information about the product</td></tr></table> Each product has - name (String) - description (String) - price (Number) - quantity (Number) - category (String) Use Postman or Insomnia to test your API during development.

SUBMISSION AND ACCOUNTING

Submission	Submission takes place via Learn Point **date *10/12 no later than 23:59*** • Video with report (max. 10 min) uploaded with mp4 or link to zoom (alternatively you report live for the teacher in charge of the course) • Zip file with all project files is posted on the learnpoint platform • Written report on use of the API

ASSESSMENT AND FEEDBACK

Assessment takes place against the following grading criteria:	The grading criteria for Pass and Pass with Distinction are: **Approved** • The student has demonstrated knowledge and skills in server programming with Node.js • The student has demonstrated knowledge and skill in structuring a REST API • The student uses an array in javascript as a container for all products **Well passed** • The student has demonstrated knowledge and skills in Backend development (connection with Database) • The student uses MongoDB (in the "cloud-atlas" environment) as a container for all products. • The student has written a report in which he explains the different REST-API routers and how they should be used, the student explains with the help of different test cases (examples of use) and gives nuanced reasoning for different outcomes when calling different routes, i.e. about what should happen, for example, if you send the wrong data when you send the right data, etc.

Feedback	Oral feedback is given to the groups during the presentation. The groups receive written feedback via the learning platform LearnPoint **by the date 20/12 at the latest.**