

Full Stack Javascript Test

Introduction

This is a technical test that will allow us to assess your thought process, ability to interpret requirements and technical implementation skills. You will be judged on code design, structure, efficiency, specs, and cleanliness.

Requirements

Todo (Task) management application.

- Should allow to create a task
- Should allow adding multiple subtasks to a parent task
- Should allow marking both task and subtask as done or undone
- Tasks and its subtasks should be displayed as accordion

Tech stack

Backend

- Node JS/TS (LTS version)
- Server libraries such as Nest.JS and Express are encouraged but can use any familiar library.
- Any DB layer (Postgres database preferred)
- Optional: ORM

Frontend

Feel free to pick one of these libraries (React and Vue are encouraged)

- React
- Vue
- jQuery
- Pure Javascript/Typescript

Models

Create **Todo** model that includes the following attributes:

- Title
- Status [pending/completed]
- Created Date

Create **Subtask** model that includes the following attributes:

- Title
- Status [pending/completed]
- Created Date

Each **Subtask** belongs to one **Todo**

API

Create an API that returns a list of **Todo** exposing only the following attributes:

- id
- title
- status
- created_at
- subtasks

Create an API that creates a **Todo** accepting the following attributes:

- title

Create an API that updates a **Todo** accepting the following attributes:

- status

Create an API that creates a **Subtask** and assign it to a specific **Todo** accepting the following attributes:

- title
- todo_id

Create an API that updates a **Subtask** accepting the following attributes:

- status

API Implementation notes and requirements

- The API should follow JSON-API specifications (<http://jsonapi.org/>).
- Provide a mechanism to seed test data.

User Interface

Todo App

☒
Do laundry

3 of 5 completed ▾

☒ Pick up the clothes

☒ Throw the clothes in the machine

☒ Turn on the machine

☐ Bring back the clothes

☐ Dry the clothes

What are the steps?

☐
Do Something else

0 of 5 completed ▲

Note: This is just a wireframe, you have the flexibility over UI

Display the input box with a submit button which will create a new **Todo** that will be displayed right away after submitting.

Display a list of the following information retrieved from the API:

Todo

- Title
- Checkbox reflecting the status
- Summary of completed/total subtasks
- The collapsible list of all related subtasks

Each **Todo** can have its status switched between [complete/pending]. Completing a **Todo** leads to completion of all nested **Subtasks**. In addition, switching a **Subtask** under a completed **Todo** to [pending] will re-open the parent **Todo**.

Testing

Add tests where you deem appropriate. The Jest framework is recommended but not required.

General Implementation Notes

- Provide helpful, descriptive commit messages.
- Avoid adding all files in one commit.

Submission method

- Send us a ZIP file that includes the `.git` folder and all commits and commit messages, or upload to cloud storage.
- Use your full name when naming the folder
- The submission should include both frontend and backend implementation. Partial submission is not acceptable.
- Do not include the `node_modules` directory in .zip file. Instead include related `.lock` file
- Include a README describing how to setup and run locally.

Important Note: Please do not upload this project to the version control system such as GitHub, GitLab, or others.