

Backend Coding Exercise (FSE)

Description

As a software engineer working at Techspert, you will be building the frontend tools that enable our users to take advantage of our search technology when looking for experts in areas of interest. The two exercises will assess your understanding of the process of building commercial applications, building features across the stack, backend to frontend.

The focus of this exercise is the API you will build to serve data to the frontend. We will be looking at your practices, architectural choices, and generally the quality of your solution. Particularly, we are interested in how you handle setup and startup, as well as the choices you make for the internal structure of the service.

This is a take-home exercise that you can do **in your own time**. We expect that the task will take **2h to complete**, so try to stick with that guideline. The language and framework of choice will be **Node.js and Express**.

Problem Statement

Big Books Publishing Co. is developing a tool to track the state of different projects that their editors are managing. From the moment they receive a manuscript to the point where the book is in print, the project goes through multiple states and involves multiple roles:

- Their pipeline consists of the following stages: Received, Reviewed, Accepted, Contract Signed, Edited, Copyedited, Typeset, In Print.
- For each project, they need to store:
 - Working title
 - Authors
 - Stage
 - Main editor
 - Person currently processing it
 - Creation date
 - Update date

Using Node and Express, build an REST API that can create a new project, update an existing project, and list all projects. All decisions about parameters, database and internal structure will be up to you.

As an *optional* challenge, you can create an endpoint that for a specific project returns a timeline of the different stages the project has been on, with timestamps.

Deliverables

Provide a zip/tar file with your full project ready to be used. Include a README file with:

- Instructions on how to run the API
- A list with the assumptions you made
- A short description of interesting parts of your code
- A short description of potential improvements

After evaluation, if we decide to move forward with your application you will have a chance to defend your work during a conversation with us. Be ready to provide highlights and alternatives, as well as any points you would like to discuss about your work.

Tips

- Ask us any questions you have about the task.
- Make any reasonable assumptions but justify your choices.
- This is a senior role exercise, so it's not enough to have a working API. Your practices, architectural choices, and the overall quality of your solution are what matters.
- Don't overengineer just to show off, but build an API that will be easy to extend and maintain, following best practices for production services.