

Interview Sample Application

The purpose of the sample application is to show your engineering abilities through code. This is your opportunity to go past the standard interview questions and show us why we should hire you.

The application itself is meant to be a somewhat trivial full stack application which performs CRUD operations on a simplistic data model. What we are looking for with this exercise is not necessarily what is created, but how it's created.

Overview

The application you'll be building is a simple task management tool. The tool will allow an end user to create, manage and delete tasks using a web interface.

Hints

Below are a few hints as to what really matters to us as an engineering team.

- We are strong proponents of 12 Factor Apps delivered via containerization
- Automation is very important to us. Your deliverable should include:
 - A build script showing how you compile and package your code
 - Automated unit tests with mocking where appropriate
 - Automated functional tests
- A well organized and documented code base is preferred. Your deliverable should include:
 - Code comments
 - API documentation for the microservice
- We develop our applications using a microservices architecture where the user interface (HTML, CSS and JavaScript) is separated from the business logic (microservices) each having their own lifecycle
 - If using Docker and Docker Compose, the preference would be two separate containers; one for the UI and one for the microservice

Logistics

The code can be written using the IDE / editor of our choice. A public GitHub repository should be created to store the code and that repository should be shared with the interviewer for review. The programming language(s) and framework(s) used to create the sample application will be determined by the interviewer and sent along with this outline.

User Stories

The user stories below outline the functionality the sample application should provide. Questions about the user stories are welcomed and the interviewer will inform you how best convey and receive answers to your questions.

1. As a user, I need to be able to create a task for myself providing a task name, description, and due date.
2. As a user, I need to be able to see a list of existing tasks.
3. As a user, I need to be able to view the details of a specific task.
4. As a user, I need to be able to mark a task as completed.
5. As a user, I need to be able to remove a task.
6. As a user, I need to be able to filter my list of tasks to just those tasks due tomorrow and / or today.
7. As a user, I need to be able to filter my list of tasks to just those tasks which are overdue.
8. As a user, I need to be able to filter my list of tasks to just those tasks which have been marked as completed.
9. As a user, I need a visual indication of which tasks are due tomorrow and / or today as well as those tasks which are past due (2 different indications).

Technical Considerations

The application will need to be able to store tasks. For this exercise in-memory storage is acceptable. Using a datastore to store the tasks will score you bonus points.

The user interface can be created without style / theming. Style / Theming will score you bonus points, but is not the goal of this exercise.

The application has no need for authentication, authorization or any type of user profile or tracking. It is assumed that the application is used by a single user.

The use of Docker / Docker Compose to run the application is encouraged and will also score you bonus points.