Leonardo.Ai API Team

Technical Challenge V2

In this technical interview test, you are required to implement API endpoints for managing schedules and tasks using TypeScript. The project involves designing and building RESTful or Lambda API endpoints to handle scheduling and task management. Below are the details of the resources you'll be working with:

Schedule

- `id`: Universally unique identifier (UUID) for the schedule.
- `account_id`: Integer representing the account associated with the schedule.
- `agent_id`: Integer representing the agent assigned to the schedule.
- `start_time`: DateTime indicating the start time of the schedule.
- `end_time`: DateTime indicating the end time of the schedule.
-

Tasks

- `id`: UUID for the task.
- `account_id`: Integer representing the account associated with the task.
- `schedule_id`: UUID referencing the schedule to which the task belongs.
- `start_time`: DateTime indicating the start time of the task.
- `duration`: Integer representing the duration of the task.
- `type`: String enumeration with values 'break' or 'work', indicating the type of task.

There's a one-to-many relationship between Schedule and Tasks, where a Schedule can have multiple Tasks associated.

Guidelines

- Please follow the guidelines below while working on this test:
- Implement your solution using TypeScript, and feel free to use any packages or frameworks of your choice (We recommend NestJS).
- Utilise a SQL-based database, preferably PostgreSQL, for data storage. You can consider using Prisma as an ORM (Object-Relational Mapping) tool.
- You should build RESTful API endpoints.
- Consider organising your code with relevant design patterns suitable for CRUD (Create, Read, Update, Delete) applications.
- Aim to future-proof your codebase by writing clean, maintainable, and extensible code.
- Keep the SOLID principles (Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, Dependency Inversion) in mind while designing your solution.

Unit Tests
Writing unit tests is an essential part of this test. Please ensure that you write thorough unit tests to validate the functionality of your API endpoints.

Integration Tests
Writing integration tests are not part of this test, but it will be nice to have.

Submission

- Upload your solution to a Git repository and share the repository link with us (Please ensure we can access your project and relevant code for grading).

  Sharing a ZIP file is not okay, as there are serious security concerns associated with it. Please instead leverage one of the many free git hosting services, such as GitHub, to publicly share your solution.