Take Home Technical Test

September 2023

Entegra

Project Description:

You will be provided with a partially completed Web API project. The project is a simple task management system where users can create, update, delete, and retrieve tasks. The solution is built using ASP.NET Core and C#. Your task is to complete the missing parts and ensure that the API functions correctly. Please read all of this document to understand the project, submission and evaluation.

Instructions:

1. Copy and unzip the project from Dropbox to your local development environment.
2. Review the existing codebase and project structure to understand the current implementation.
3. Ensure you have the required development environment set up, including .NET 7 and any necessary IDE or code editor.
4. Implement the missing functionality to complete the Web API according to the specifications below.

Specifications:

There are 3 kinds of tasks that you need to implment: house, leisure, and work tasks. You will need to consider how you save and retrieve these to and from the database. Consider and demonstrate how you can use design patterns to achieve this.

The project will need to use code first migrations.

The Web API should have the following endpoints:

1. GET /api/tasks:
   - This endpoint should return a list of all tasks.
2. GET /api/tasks/{id}:
   - This endpoint should return a single task with the specified ID.
3. POST /api/tasks:
   - This endpoint should allow the creation of a new task.
   - The task should have the following properties: Id (Guid), Title (string), Description (string), DueDate (DateTime), IsCompleted (bool).
   - The Id should be generated automatically upon task creation.
   - Validate the input data to ensure all required properties are provided.

4. PUT /api/tasks/{id}:

   - This endpoint should allow updating an existing task with the specified ID.
   - Only the Title, Description, DueDate, and IsCompleted properties can be updated.
   - Validate the input data to ensure all required properties are provided and the task with the given ID exists.

5. DELETE /api/tasks/{id}:

   - This endpoint should allow deleting the task with the specified ID.

6. Implement appropriate error handling and status codes for various scenarios, like invalid requests, missing tasks, etc.

7. POST /api/tasks/search1

   This endpoint should allow searching using LINQ for the following:

   - Search for all work and leisure tasks that have a due date in a week within the future.

8. POST /api/tasks/search2

   This endpoint should allow searching using LINQ for the following:

   - Get a count of tasks that is distinct by TaskType (leisure/home/work) for which the DueDate has passed and are not yet completed.

9. Frontend

   - In this section, we will ask you to setup a frontend system of your choice (angularjs or react + html) and create a checkbox for each Task Type (work/leisure/home), an input box for description and a search button below them. Once the search button is clicked, a request will need to be fired to the backend controller that will allow searching for the selected inputs. The results should be returned to the UI and displayed on a table with clear columns marked.

10. Unit Tests:

    - Write unit tests for the controllers and any other critical components to ensure the correctness of your implementation.

Submission Instructions:

1. After completing the task, push your changes to your own Github repository.
2. Provide a link to your repository and any necessary instructions on how to run the project and tests.

Please complete this test and email your submission to Martinos.Evripidou@entegraps.uk by **4pm Friday 8th September.**

Evaluation:

Your project will be evaluated based on the following criteria:

1. Completeness of the implementation and adherence to the specifications.
2. Code quality and organization, including proper separation of concerns.
3. Error handling and validation.
4. Test coverage and effectiveness of unit tests.
5. Use of best practices and design patterns.