GOAL SYSTEMS

optimizing your world

Code Challenge
Backend Engineer

PROBLEM STATEMENT

Design and implement an inventory manager.

Provide a README file with:

- Instructions to run the application.
- Brief documentation on the design, code structure and any annotation that you want to add about extensibility, maintenance, security, performance, etc., that you have not had time to implement.
- Brief description of assumptions or not implemented requirements.

Design and implement a solution with .Net Core 3.1 or higher using C# as programming language that models and supports the requirements defined in the following section:

- Use DDD paradigm to model the solution, you must necessarily use the application, infrastructure and domain layers in addition to REST API, i.e., your solution must have at least 4 projects: API, Application, Infrastructure and Domain. You can optionally separate some of the sublayers of these logical layers into additional Visual Studio projects.
- Correctly manage the communications between the different logical layers, considering that the Domain layer is not a technological layer and does not depend on the other layers.
- Do not include database (t-sql) code, it's enough with in-memory repositories.
- Document all design patterns and principles of Object-Oriented design you use.
- Use the Net Core dependency injection and interfaces to abstract layers from each other.
- Document the reasons why you use a third-party nuget package.
- Write unit tests of all layers and cover the proposed use case.
- Use API REST standard.

Write production-quality code that do not crash into static code analysis tools like SonarQube.

Make the technical documentation of all the classes and methods that you include with the /// (xml) notation.

Please ask us any questions you may have related to this technical interview.

OPTIONAL:

1. Include Swagger package for API documentation.
2. Include CQRS pattern for Command and Query segregation.
3. Validate the input parameters of public methods (Fluent validation).
4. Secure the API (for example, basic security).
5. Add traces with ILogger interface in some method.
6. Upload your solution to GitHub repository for easy review.
7. Implement a FrontEnd with VUE 3 or REACT that consumes the API.

REQUIREMENTS

1. ADD AN ITEM TO THE INVENTORY

When you add an item to the inventory, it contains information about the item just added, such as Name, Expiration Date, Type, etc.

2. REMOVE AN ITEM FROM THE INVENTORY BY NAME

When you take an item out of the inventory, the item is no longer available in the inventory.

3. NOTIFY THAT AN ITEM HAS BEEN REMOVED FROM THE INVENTORY

When you remove an item from the inventory, fire and “ItemRemoved” event.

4. NOTIFY WHEN AN ITEM EXPIRES

When an inventory item expires, fire and “ItemExpired” event.

GOAL SYSTEMS

optimizing your world

goal@goalsystems.com
www.goalsystems.com