Code Challenge

Purpose

The main purpose of this Code Challenge, is to provide a common starting point for an informal conversation on how you develop and structure your code, and your general knowledge about .NET Core. During the interview, some of the topics and questions we will address are the following:

- How did you approach the challenge?
- How much time you've taken on each part of the challenge (exploring the Product Catalog API, defining the architecture, testing, etc)?
- Why you have arranged the code as you have?
- What have surprised you the most during the coding challenge?
- What would you change if you had more time?

Challenge

We expect you to invest 2 hours on this Code Challenge, we don't expect you to meet all expectations. We advise you to find a quiet time and space to be focused on the Code Challenge, so that you can work without interruptions during this period.

Use our Code Challenge API to **build your own Basket API.** Your Basket API should support the regular operations of a Web Cart, e.g., add product to basket, remove product from basket, decrease and increase product quantity. Note that the mentioned operations do not correspond directly to endpoints. It is up to you to create the Basket API to support the operations according to your own design decisions.

Our Code Challenge API has a Swagger UI available here https://azfun-impact-code-challenge-api.azurewebsites.net/api/swagger/ui.

The Code Challenge API will provide a Product Catalog containing 10.000 ranked products. **Warning**: we will return the whole 10.000 products without pagination.

From these 10.000 products, your API can only allow the top 100 products to be handled by the end user.

The Code Challenge API requires authentication. You can obtain a Token with the Login endpoint.

Once the end user has selected the desired products via your API, you should allow him/her to submit the basket, which can be done in our API via the CreateOrder endpoint.

Additionally, your API should provide the following endpoints:

1. Return the top ranked 100 products.
2. Return a paginated result of the product catalog ordered by price in ascending order and properly paginated.
    a. The page size should be defined by the caller of your API, but your API should not allow page sizes above 1000.
3. Get a basket via its Id, which we prefer it to be a GUID.
4. Get the 10 cheapest products from all the 10.000 products.

Code delivery

We expect you to commit your code to a public git repository, such as GitHub or Bitbucket.

Expectations

The following are our main expectations for the solution you provide:

- A solution designed to adhere to SOLID principles.
- Unit and end-to-end testing.
- Production ready code.
- Code submitted with multiple commits, ideally with a first commit once you start the Code Challenge.
- Comments on what improvement points you'd have.
- We don't expect you to implement Authentication in your Basket API.
- We expect to be able to run your solution locally and run the tests you've implemented without any changes to your code.
- We don't expect you to store the **baskets** in any persistence layer, e.g. database or file. **Storing it in memory is enough**.