

Project goal

- Create a REST API service that provides all available products of an e-shop and allows partial updates of a single product.
- Create Swagger documentation for all endpoints.
- Create unit tests covering the functionality of the endpoints.
- We expect you to deliver the same quality of solution as you would for any production code.
- Follow best practices for REST API design.
- Follow best practices for application design and implementation, use proper architecture, and apply SOLID design principles.

Technical specification

- The solution should be based on the .NET Core framework (use the latest version or latest LTS version) and C#.

Data Specification

- The design product entity with at least following properties
 - Name of product
 - Url to main product image
 - Product price
 - Description of product
 - Quantity of product stocked
- We want to be able to create new product only with name and image url.

API Specification

- API endpoint to list all available products.
- API endpoint to create new product
- API endpoint to get a single product by ID.
- API endpoint to update only the product stock.

Data layer

- Prepare a simple SQL database to store product data use MSSQL, SQLite or InMemory.
- Use an ORM framework of your choice.
- Prepare initial database seed with sample products
- Prepare mock data that can be used for unit tests.

- Link to a public git repository (showing complete commit history documenting your progress) with the complete solution consisting of at least:
 - A Web API project that can be directly compiled and run in an IDE.
 - A unit test project.
- Complete API documentation accessible in the browser after running the project on localhost (e.g., Swagger).
- README file in the repository describing at least basic information about the application, prerequisites for running the project, steps to run the application, information on how to run unit tests.

Version 2

- Add version 2 of the endpoint with pagination support (default page size 10). Use the versioning method of your choice (ensure that versioning is correctly reflected in the API documentation).
- Add asynchronous queue for handling of product stock update using InMemory queue, Kafka or RabbitMQ.