Software Developer Case - API

Please read the context and instructions below. All the necessary information is provided. You are welcome to make assumptions, use real-life experiences and common sense to fill the gaps.

Scenario

Using an HTTP REST API approach and C# .NET Core, develop a solution to fulfill the following stories:

Story API-001	As a Developer, I want to have an API to save JSON object in a secure way based on my credentials so I can retrieve the same object later
Acceptance Criteria	• The data must be persisted after being encrypted; using the username and password as the encryption key • The endpoint should only accept valid JSON objects • The API must generate a random ID for when the object is created, and return it as part of the response ○ The API must support updates of existing objects • The API must accept only one JSON object per request and handle the exception properly • The API must handle properly the not found and not authorized scenarios

Story API-002	As a Developer, I want to have an API endpoint to retrieve a JSON object I've saved before with my credentials so I can use the data
Acceptance Criteria	• The API must return a previously saved object, decrypted properly • The API must handle properly not found and not authorized scenarios

Story API-003	As a Developer, I want to have an API endpoint to delete a JSON object I've saved before with my credentials so the data is properly deleted
Acceptance Criteria	• The API must delete the data properly when the data belongs to the authenticated user • The API must handle properly not found and not authorized

	scenarios

Story API-004	As a Developer, I want to have an API endpoint to list the JSON object ID of objects I have saved before using my credentials so I can interact with them
Acceptance Criteria	• The API must handle properly not authorized scenarios

General Acceptance Criteria

- The API follows HTTP REST standard and best practices and using *application/json* as the Content-Type for data exchange
- The data must be persisted somehow not in-memory
- Proper documentation for developers to use the API must be provided; Swagger, Postman or any other standard/best practice method is fine
- Simple documentation/steps for DevOps on how to execute the application must be provided
- The authenticated API endpoints must use Basic Authentication, following the RFC7617 and RFC7235 standards
    - Consider the following credentials will be used to test the API
        - Username: JohnSmith Password: Test1234!:|
        - Username: User2 Password: password2

Delivery and General Requirements

- Keep it simple; develop the best solution you can, using a minimum of your time like 2h or so
- The code will be evaluated from the quality and functionality perspectives; therefore Unit Tests are expected also
- Please use Github, create a private repository and add @wgcorrea as an admin
    - Please commit often and regularly, the idea is to see the “sausage-making process”
    - Use the README to provide the required documentation
- Please document assumptions and decisions as well
- When you have finished, please send a quick note to willian.correa@rangle.io

Presentation

You will have the chance to present your solution in a session after you submit your code, with the following agenda:

- 15 minutes for you to present
- 10 min for you to ask questions
- 20 minutes for discussion, where you can suggest improvements to your solution based on your learning from the previous answers