

Senior Backend Engineer - Tech Test

Instructions

Your goal in this task is to implement a simple back-end for an online marketplace. Here is a sample of some of the products available on the site.

Product code	Name	Price
-----	-----	-----
001	Lavender heart	£9.25
002	Personalised cufflinks	£45.00
003	Kids T-shirt	£19.95

Your task is to create a RESTful API to implement CRUD operations on this data. You should provide five endpoints:

- * GET /products - A list of products, names, and prices in JSON.
- * POST /product - Create a new product using posted form data.
- * GET /product/{product_id} - Return a single product in JSON format.
- * PUT /product/{product_id} - Update a product's name or price by id.
- * DELETE /product/{product_id} - Delete a product by id.

The service should be implemented using .Net Core. Use Swagger to define and allow interaction with your API. Make sure there are sensible return values for both successful and unsuccessful requests (e.g. the server should report a code such as 404 for an unknown product ID and not error out).

Implement a database, though the technology you use is your own choice; you may also use any publicly-available (installable through normal package managers or build systems) ORMs, etc.

Provide the finished product as a publicly accessible git repo with all of the code and other files, making sure to include a README that will allow a technically competent user to install and run your app (including any build scripts).

A collection of Postman tests will be provided to you with this document, and will be run against your service to verify the results. Be sure to seed your database with the sample data in the table above before running the tests.

A successful submission will pass all of the functional Postman tests. You will be evaluated on providing a working API as well as basic code style, simplicity, and correctness. You will not be evaluated on the API's robustness or performance, you do not need to secure the API, and you do not need to worry about production server setup (i.e. a framework's default server in debug mode is fine).

Bonus points (not required for a passing solution):

- * Containerise the solution with Docker.