**Take-Home Test for Java API Developer**

**Objective:**  
Design a simple RESTful API using Java to manage a basic product inventory for a small store.

**Requirements:**

1. **Data Model**:
   * Product
     + id (Long, auto-generated)
     + name (String)
     + description (String)
     + price (Double)
     + quantity (Integer)
2. **API Endpoints**:
   * GET /products: Retrieve a list of all products.
   * GET /products/{id}: Retrieve details of a specific product by ID.
   * POST /products: Add a new product. The payload should accept name, description, price, and quantity.
   * PUT /products/{id}: Update the details of a specific product by ID.
   * DELETE /products/{id}: Delete a specific product by ID.
3. **Functional Requirements**:
   * Implement CRUD (Create, Read, Update, Delete) operations for the product.
   * Ensure data validation for the inputs.
   * Handle potential exceptions and return appropriate HTTP status codes and messages.
4. **Technical Requirements**:
   * Use Java 11 or later.
   * Use Spring Boot for the REST API.
   * Use an in-memory database (e.g., H2) for storage.
   * (Bonus) Implement basic authentication using Spring Security.
5. **Testing**:
   * Write unit tests for the service layer.
   * Write integration tests for the API endpoints.
6. **Documentation**:
   * Provide a README.md file that:
     + Explains the design decisions you made.
     + Gives instructions on how to run the application and tests.
     + Provides example API calls using curl or any other method.

**Submission**:  
Package your code and tests into a ZIP archive or provide a link to a public Git repository. Email your submission to us.

**Evaluation Criteria**:

* Code quality, readability, and organization.
* Implementation of the requirements.
* Proper error handling and validation.
* Quality and coverage of tests.
* Quality of the documentation.

**Note**: This test is meant to assess your skills and is not used for any actual business purposes. You don't need to spend more than 2-3 hours on it. Focus on the core requirements first and consider the bonus tasks if time permits.