Тестовое задание.

Язык программирования: Java
Платформа: не ниже J2SE 1.8

1. Создать простое приложение с использованием Spring Boot.
Приложение должно предоставлять администраторский REST CRUD интерфейс для управления продуктами.
В качестве инструмента для сборки и управления проектом использовать Maven.
Для хранения данных можно использовать PostgreSQL или in-memory DB.
Продукт должен иметь уникальный идентификатор, название и описание, цену, дату создания и дату модификации.
При запросе продукта по идентификатору, если продукт не найден, возвращать пустой результат со статусом 404.

2. Расширить доменную модель и методы REST API так, чтобы цену продукта можно было указывать в разных валютах, а название продукта и его описание - в разных языках.

3. Добавить валидацию на создании и модификации продукта.
Входные данные нужно валидировать следующим образом:
Название продукта - обязательное поле, не должно быть пустым.
Цена продукта - обязательное поле, должна быть больше 0.

4. Добавить в приложение второй REST ресурс, предназначенный для получения продуктов клиентами, ProductClientResource.
В новом ресурсе создать следующие методы:
- метод для поиска продуктов по имени или описанию.
- метод для получения всего списка продуктов.
В случае, когда продукт не имеет данных в указанном языке и/или валюте, такой продукт не должен попадать в результирующий список.
- метод для получения продукта по идентификатору.
Списки продуктов должны возвращаться постранично.
Если у полученного по идентификатору продукта нет данных в указанном языке и/или валюте, кидать специальное исключение.
Для этого ресурса ошибки при получении продукта (404 + новое исключение) должны быть обработаны и возвращены клиенту в виде простого объекта с полями errorCode и errorMessage.
Для преобразования исключения в объект ошибки использовать обработчик на основе аннотации @ExceptionHandler.
Исключительные ситуации должны быть залогированы.
Все методы этого ресурса должны требовать указания конкретного языка и валюты.

Примечания.
Рекомендуется использовать трёхуровневую (трёхслойную) архитектуру приложения.
Для логирования использовать log4j либо logback в связке с SLF4J.
Если возникнут проблемы с логированием можно использовать System.out.println().

Допускается использование не указанных в техническом задании технологий и сторонних библиотек. Например: Swagger, Liquibase.