

Техническое задание SBI LARAVEL

Техническое задание SBI LARAVEL.....	1
Задача.....	2
1. Сущности и структура данных.....	2
1.1. Сущность "Категория".....	2
1.2. Сущность "Товар".....	2
2. Инициализация данных (seeder/factory).....	2
3. Реализация API.....	2
Для категорий:.....	2
Для товаров:.....	2
4. Валидация данных.....	3
Правила валидации товара:.....	3
Правила валидации категории:.....	3
5. Представление данных.....	3
6. Unit-тестирование.....	3
7. Интеграция с Excel.....	3
8. GIT.....	4
9. Требования.....	4
10. Сроки выполнения.....	4
11. Видео-презентация работы (обязательно).....	4

Задача

Разработать на фреймворке **Laravel** простой модуль учета товаров с возможностью выгрузки данных в Excel. При разработке необходимо использовать **Repository** и **Service** паттерны. Потребуется использование unit-тестов и размещение в открытом репозитории (например, на GitHub, GitLab).

1. Сущности и структура данных

1.1. Сущность "Категория"

- Поля:
 - `id` (автоинкремент)
 - `name` (строка, обязательное)

1.2. Сущность "Товар"

- Поля:
 - `id` (автоинкремент)
 - `name` (строка, обязательное)
 - `price` (decimal/float, обязательное)
 - `barcode` (строка, уникальное)
 - `category_id` (внешний ключ на категорию)

2. Инициализация данных (seeder/factory)

- Создать в наполнителе 3 категории:
 - Смартфоны
 - Зарядк
 - Чехлы
- Создать **200 фейковых** товаров с случайными категориями (смартфоны, зарядки, чехлы) через **фабрику**

3. Реализация API

Разработать следующие **API-эндпоинты**:

Для категорий:

- `GET /api/categories` — список категорий
- `POST /api/categories` — создание
- `GET /api/categories/{id}` — просмотр
- `PUT /api/categories/{id}` — обновление
- `DELETE /api/categories/{id}` — удаление

Для товаров:

- `GET /api/products` — список товаров
- `POST /api/products` — создание
- `GET /api/products/{id}` — просмотр
- `PUT /api/products/{id}` — обновление
- `DELETE /api/products/{id}` — удаление

4. Валидация данных

Каждый API-метод должен использовать валидацию через Form Request с логическим обоснованием правил:

Правила валидации товара:

- `name` — строка, **обязательное поле**, минимум 2 символа
- `price` — числовое значение, **обязательное поле**, минимум 0 (логика: цена не может быть отрицательной)
- `barcode` — строка, **обязательное поле**, должна соответствовать формату **EAN-13** (логика: стандартный международный формат штрихкода)
- `category_id` — **обязательное поле**, должен существовать в таблице категорий (логика: каждая товарная позиция должна принадлежать категории)

Правила валидации категории:

- `name` — строка, **обязательное поле**, минимум 2 символа

5. Представление данных

- Использовать **Laravel API Resources** при возврате данных из API
- Обязательно использовать метод `whenLoaded()` для работы с отношениями, чтобы избежать лишней загрузки данных

6. Unit-тестирование

Создать модульный тест, который проверяет:

- Создание нового товара
- Обновление цены у уже существующего товара


7. Интеграция с Excel

- Подключить библиотеку для работы с Excel-файлами (например, [maatwebsite/excel](https://maatwebsite.com/excel) для Laravel).
- Реализовать метод в сервисе для экспорта товаров в Excel-файл с колонками:
 - Название товара
 - Штрихкод
 - Цена
 - Название категории
- Важно! Необходимо использовать **очереди** при выгрузке данных

8. GIT

- Разместить результат в публичном репозитории **GitHub** или **GitLab**
- Название проекта должно быть осмысленным и отражать суть задачи
- В **README.md**:
 - Краткое описание проекта
 - Как установить
 - Как запустить

9. Требования

- Структура кода проекта должна быть аккуратной и логически организованной
- Использовать методичку в коде:  LARAVEL - Методичка разработчика в SBI
- Обязано использовать, даже если и маленькая логика: **Service Pattern** и **Repository Pattern**
- Использовать Laravel 10+ версию

10. Сроки выполнения

- Время на выполнение задания: **2 календарных дня** с момента получения задания.

11. Видео-презентация работы (обязательно)

После завершения разработки необходимо записать короткое видео с демонстрацией и **объяснением голосом** по всей выполненной работы. **Ограничение видео - 7 минут.** В видео показать по шагам (сценарий):

1. Код проекта (максимум 1,5 минуты)
 - Кратко показать структуру проекта: **routes, controllers, requests, services, repositories, resources.**
2. Таблицы базы данных (максимум 30 секунд)
 - Открыть базу данных и показать созданные таблицы (структуру таблиц).
3. Работа сидеров (максимум 1 минута)
 - Запустить сидеры и показать, какие данные были добавлены в таблицы.

4. Unit-тесты (максимум 1 минута)

- Запустить unit-тесты.
- Показать код это работает
- Показать, какие изменения произошли в базе данных после выполнения тестов.

5. Выгрузка в Excel через очередь (максимум 3 минуты)

- Запустить очередь в терминале (`php artisan queue:work`).
- Вызвать метод экспорта товаров и показать код метода
- На видео **должно быть видно, что функция была передана в очередь и выполнена** (например, появление строки выполнения задачи в терминале).
- Открыть сгенерированный Excel-файл и показать, что данные успешно загрузились в документ.

Видео должно быть сохранено и передано вместе с проектом (например, загрузить в облако и приложить ссылку).