Документация

**Controllers**

**Add(<Т> item)**

Този метод добавя нов ентитет от тип **<Т>**в базата данни. Първо, той търси в DbSet за съществуваща категория със същия ID, какъвто е подаденият. Ако такава категория не съществува (т.е. **existingItem** е **null**), новата категория се добавя към DbSet , и промените се запазват в базата данни. Ако категория със същия ID вече съществува, се хвърля **ArgumentException**, съобщаващо, че категорията вече присъства.

**Delete(int id)**

Този метод изтрива ентитет от тип **<T>** от базата данни по неговото ID. Първо, методът извлича категорията с посоченото ID, използвайки метода **Get(int id)**. След това, той премахва намерената категория от DbSet и запазва промените в базата данни.

**Get(int id)**

Този метод извлича единичен ентитет от тип **<T>** по неговото ID. Използва LINQ, за да търси в DbSet за първата категория, която съответства на даденото ID. Ако такава категория не е намерена, се връща **null**.

**GetAll()**

Този метод извлича всички ентитети от тип **<T>** от базата данни. Просто връща списък с всички категории в DbSet , като ги превръща в списък.

**Update(<T> item)**

Този метод обновява съществуващ ентитет от тип **<T>** в базата данни. Първо, извлича категорията със същия ID като този, който е подаден, използвайки метода **Get(int id)**. Ако такава категория съществува, използва метода **Entry**, за да намери информацията за проследяване на този ентитет и обновява неговите текущи стойности с тези от подадения обект **<T>**. След това запазва промените в базата данни. Ако не съществува категория с подаденото ID, се хвърля **ArgumentException**, указващо, че категорията не е намерена.

Всеки метод гарантира, че операциите спазват ограниченията и целостта на базата данни, като проверява съществуването на ентитети преди тяхното добавяне или обновяване и като директно манипулира ентитети за изтриване.

**ProductController()**

* **Описание:** Създава нова инстанция на **ProductController** със стандартен **ShopContext**.

**ProductController(ShopContext context)**

* **Описание:** Създава нова инстанция на **ProductController** с възможност за използване на съществуващ **ShopContext**. В момента този конструктор повторно инициализира **ShopContext**, което може да не е очакваното поведение.

**Методи**

**AddProduct()**

* **Описание:** Добавя нов продукт в контекста. Проверява за съществуването на производител и категория преди добавянето. Ако някой от тях не съществува, хвърля **ArgumentException**.

**DeleteProduct(string barcode)**

* **Описание:** Изтрива продукт по зададен баркод. Ако продуктът не е намерен, хвърля **ArgumentException**.

**EditProduct()**

* **Описание:** Обновява детайли на съществуващ продукт, идентифициран по баркод. Ако продуктът не е намерен, хвърля **ArgumentException**. Този метод също така валидира наличието на новия производител и категория, ако са предоставени.

**ReadProduct(string barcode)**

* **Описание:** Връща продукт по зададен баркод. Ако продуктът не е намерен, хвърля **ArgumentException**. Включва информация за производителя и категорията на продукта.

**GetProductsByProducer(string producerName)**

* **Описание:** Връща списък на продуктите, произведени от конкретен производител.

**GetProductsByCategory(string categoryName)**

* **Описание:** Връща списък на продуктите, принадлежащи към конкретна категория.

**AddProduct()**

* **Описание:** Добавя нов продукт в базата данни. Проверява дали съществува производител и категория с предоставените имена и ако да, създава нов продукт с тези данни.
* **Параметри:** Включва баркод, име, количество на склад, цена, име на производител, име на категория и допълнителен текст.

**DeleteProduct(barcode)**

* **Описание:** Изтрива продукт от базата данни по даден баркод. Ако продукт с такъв баркод не съществува, хвърля **ArgumentException**.
* **Параметри:** Баркод на продукта за изтриване.

**EditProduct()**

* **Описание:** Обновява информацията за съществуващ продукт. Проверява за наличието на продукт, производител и категория, и обновява данните му с новопредоставените стойности.
* **Параметри:** Включва баркод, ново име, ново количество на склад, нова цена, ново име на производител, ново име на категория и нов допълнителен текст.

**ReadProduct(barcode)**

* **Описание:** Връща продукт по даден баркод. Ако продукт с такъв баркод не съществува, хвърля **ArgumentException**.
* **Параметри:** Баркод на продукта, който да бъде прочетен.

**GetProductsByProducer(producerName)** и **GetProductsByCategory(categoryName)**

* **Описание:** Връщат списък на продукти, филтрирани по име на производител или категория съответно.
* **Параметри:** Име на производителя или категорията за филтриране на продуктите.