

Belle Croissant Lyonnais API Documentation

Версия: 1.0

Последнее обновление: 11 июля 2024 г.

1. Введение

API Belle Croissant Lyonnais обеспечивает программный доступ к данным о продукте, заказчиках и заказах для системы управления пекарней. Данный RESTful API обеспечивает безопасное и эффективное взаимодействие с базой данных пекарни, поддерживая такие операции, как поиск информации о продукте, управление данными заказчиков и обработка заказов.

2. Начало работы

2.1 Проверка подлинности

API использует веб-токены JSON (JWT) для аутентификации. Для получения токена необходимо отправить POST-запрос в конечную точку `/api/v1/auth/login`, указав свои учетные данные.

3. Конечные точки API

3.1 Проверка подлинности

Конечные точки для аутентификации пользователей и управления сессиями.

Метод	POST
Конечная точка	<code>/api/v1/auth/login</code>
Описание	Аутентификация и получение токена JWT.
Тело запроса (JSON): <pre>{ "username": "staff", "password": "BCLyon2024"}</pre>	
Ответ (JSON): <pre>{ "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...", "session_id": "550e8400-e29b-41d4-a716-446655440000"}</pre>	

3.2 Продукты

Конечные точки для управления информацией о продукте.

Метод	GET
Конечная точка	/api/v1/products
Описание	Извлечение списка всех продуктов.

Метод	GET
Конечная точка	/api/v1/products/{id}
Описание	Поиск конкретного продукта по идентификатору.

Метод	POST
Конечная точка	/api/v1/products
Описание	Создание нового продукта.
<p><i>Тело запроса (JSON):</i></p> <pre>{ "product_name": "Baguette", "category": "Bread", "price": 3.00, "cost": 1.50, "ingredients": "Flour, Water, Yeast, Salt", "seasonal": "FALSE", "active": "TRUE" }</pre>	
<p><i>Ответ (JSON):</i></p> <pre>{ "product_id": "2", "product_name": "Baguette", "category": "Bread", "price": "3.00", "cost": "1.50", "ingredients": "Flour, Water, Yeast, Salt", "seasonal": "FALSE", "active": "TRUE" }</pre>	

Метод	PUT
Конечная точка	/api/v1/products/{id}
Описание	Обновление существующего продукта.

Тело запроса (JSON):

```
{
  "price": 3.50,
  "active": "FALSE"
}
```

Ответ (JSON):

```
{
  "product_id": "2",
  "product_name": "Baguette",
  "category": "Bread",
  "price": "3.50",
  "cost": "1.50",
  "ingredients": "Flour, Water, Yeast, Salt",
  "seasonal": "FALSE",
  "active": "FALSE"
}
```

Метод	DELETE
Конечная точка	/api/v1/products/{id}
Описание	Удаление продукта.

3.2 заказчики

Конечные точки для управления данными заказчика.

Метод	GET
Конечная точка	/api/v1/customers
Описание	Получение списка всех заказчиков.

Метод	GET
Конечная точка	/api/v1/customers/<int:id>
Описание	Поиск конкретного заказчика по идентификатору.

Метод	POST
Конечная точка	/api/v1/customers
Описание	Создание нового заказчика.

Тело запроса (JSON):

```
{
  "first_name": "Jean",
  "last_name": "Martin",
  "email": "jean.martin@email.com",
  "age": "35",
  "gender": "M",
  "postal_code": "69006",
  "phone_number": "33612345678",
  "membership_status": "Basic"
}
```

Ответ (JSON):

```
{
  "customer_id": "103",
  "first_name": "Jean",
  "last_name": "Martin",
  "age": "35",
  "gender": "M",
  "postal_code": "69006",
  "email": "jean.martin@email.com",
  "phone_number": "33612345678",
  "membership_status": "Basic",
  "join_date": "7/11/2024",
  "last_purchase_date": "",
  "total_spending": "0",
  "average_order_value": "0",
  "frequency": "0",
  "preferred_category": "",
  "churned": "FALSE"
}}
```

Метод	PUT
Конечная точка	/api/v1/customers/<int:id>
Описание	Обновление информации о существующем заказчике.
Тело запроса (JSON):	
<pre>{ "membership_status": "Gold", "phone_number": "33687654321" }</pre>	

Ответ (JSON):

```
{
  "customer_id": "103",
  "first_name": "Jean",
  "last_name": "Martin",
  "age": "35",
  "gender": "M",
  "postal_code": "69006",
  "email": "jean.martin@email.com",
  "phone_number": "33687654321",
  "membership_status": "Gold",
  "join_date": "7/11/2024",
  "last_purchase_date": "",
  "total_spending": "0",
  "average_order_value": "0",
  "frequency": "0",
  "preferred_category": "",
  "churned": "FALSE"
}
```

Метод	DELETE
Конечная точка	/api/v1/customers/<int:id>
Описание	Удаление заказчика.

3.3 Заказы

Конечные точки для создания, получения, обновления и удаления заказов.

Метод	GET
Конечная точка	/api/v1/orders
Описание	Извлечение списка всех заказов.

Метод	GET
Конечная точка	/api/v1/orders/<int:id>
Описание	Извлечение конкретного заказа по идентификатору.

Метод	POST
Конечная точка	/api/v1/orders
Описание	Создание нового заказа.

Тело запроса (JSON):

```
{
  "customer_id": "103",
  "date": "2024-07-11",
  "time": "14:30:00",
  "product_id": "2",
  "quantity": 3,
  "price": 3.50,
  "payment_method": "Credit Card",
  "channel": "In-store",
  "store_id": "1",
  "promotion_id": "",
  "discount_amount": 0.0
}
```

Ответ (JSON):

```
{
  "transaction_id": "3",
  "customer_id": "103",
  "date": "2024-07-11",
  "time": "14:30:00",
  "product_id": "2",
  "quantity": "3",
  "price": "3.50",
  "payment_method": "Credit Card",
  "channel": "In-store",
  "store_id": "1",
  "promotion_id": "",
  "discount_amount": "0.0"
}
```

Метод	PUT
Конечная точка	/api/v1/orders/<int:id>
Описание	Обновление существующего заказа.
<i>Тело запроса (JSON):</i>	
<pre>{ "quantity": 4, "discount_amount": 1.0 }</pre>	

Ответ (JSON):

```
{
  "transaction_id": "3",
  "customer_id": "103",
  "date": "2024-07-11",
  "time": "14:30:00",
  "product_id": "2",
  "quantity": "4",
  "price": "3.50",
  "payment_method": "Credit Card",
  "channel": "In-store",
  "store_id": "1",
  "promotion_id": "",
  "discount_amount": "1.0"
}
```

Метод	DELETE
Конечная точка	/api/v1/orders/<int:id>
Описание	Удаление заказа.

Метод	GET
Конечная точка	/api/v1/customers/<int:customer_id>/orders
Описание	Извлечение всех заказов для конкретного заказчика.

Метод	GET
Конечная точка	/api/v1/products/<int:product_id>/orders
Описание	Извлечение всех заказов на определенный продукт.

3.4 Управление сеансами

Метод	POST
Конечная точка	/api/v1/reset
Описание	Сброс данных сеанса.
Ответ (JSON): <pre>{"message": "Session data reset successfully"}</pre>	

4. Обработка ошибок

API использует обычные коды ответа HTTP для указания на успех или неудачу запроса API. В общем:

диапазон 2xx указывает на успех

диапазон 4xx указывает на ошибку, при которой произошел сбой с учетом предоставленной информации (например, был пропущен обязательный параметр, ресурс не был найден или не удалось выполнить аутентификацию).

диапазон 5xx указывает на ошибку в работе наших серверов

Все ответы об ошибках будут содержать объект JSON с полями `error` и `message`.

Метод	
Конечная точка	
Описание	Пример ошибки 404:
Ответ (JSON): <pre>{ "error": "Not Found", "message": "The requested resource was not found" }</pre>	

Метод	
Конечная точка	
Описание	Пример ошибки 401:
Ответ (JSON): <pre>{ "error": "Unauthorized", "message": "Authentication credentials were missing or incorrect" }</pre>	

Метод	
Конечная точка	
Описание	Пример ошибки 500:
Ответ (JSON): <pre>{ "error": "Internal Server Error", "message": "An unexpected error occurred on the server" }</pre>	

5. Ограничение скорости

В настоящее время API не реализует ограничение скорости. Однако мы рекомендуем клиентам использовать разумную частоту запросов, чтобы обеспечить оптимальную производительность для всех пользователей.

6. Управление версиями

API управляется версиями с использованием управления версиями URL-путей. Текущая версия - v1. После внесения обратно несовместимых изменений в API мы выпустим новый номер версии.

7. Безопасность

- В производственных средах все запросы к API должны выполняться по протоколу HTTPS
- Токены аутентификации передаются надежно
- Пароли никогда не передаются в ответах API
- Ключи API следует хранить в безопасном месте и не передавать другим пользователям

8. Передовой опыт

- - Используйте HTTPS для всех запросов в производственных средах
- - Внедрите надлежащую обработку ошибок в ваших приложениях
- - Кэшируйте ответы API по мере необходимости
- - Используйте сжатие для больших запросов или ответов
- - Реализуйте экспоненциальный откат при получении ошибок сервера

9. Список изменений

Версия 1.0 (11 июля 2024)

- - Первоначальный выпуск API
- - Внедрены CRUD-операции для продуктов, заказчиков и заказов
- - Добавлено управление сессиями и проверка подлинности