Публикация на тему

Бэкенд API на Laravel

Инструкция по установке и использованию готового бэкенд API на Laravel. BackendAPI доступно по ссылке https://github.com/mikhalkevich/backendapi

Анотация

Данный бэкенд API предназначен для обслуживания запросов клиентов, как публичных, так и админских. Для админской части, после авторизации становятся доступны CRUD-запросы в модели: каталог, продукты, компании, медиа-библиотека. Отдельными запросами админ может формировать связми между этими таблицами, а загружаемые файлы (например, изображения) могут привязываться к любой модели.

This backend API is designed to serve client requests, both public and admin. For the admin part, after authorization, CRUD requests in the model become available: catalog, products, companies, media library. With separate requests, the admin can form links between these tables, and uploaded files (for example, images) can be linked to any model.

Автор

Михалькевич Александр Викторович

Публикация

Наименование Бэкенд API на Laravel

Автор А.В.Михалькевич

Специальность Инструкция по установке и использованию готового бэкенд API на Laravel. BackendAPI доступно по ссылке https://github.com/mikhalkevich/backendapi,

Анотация

Данный бэкенд API предназначен для обслуживания запросов клиентов, как публичных, так и админских. Для админской части, после авторизации становятся доступны CRUD-запросы в модели: каталог, продукты, компании, медиа-библиотека. Отдельными запросами админ может формировать связми между этими таблицами, а загружаемые файлы (например, изображения) могут привязываться к любой модели.

Anotation in English

This backend API is designed to serve client requests, both public and admin. For the admin part, after authorization, CRUD requests in the model become available: catalog, products, companies, media library. With separate requests, the admin can form links between these tables, and uploaded files (for example, images) can be linked to any model.

Ключевые слова backend, laravel, api, бэкенд, RESTFull

Количество символов 9459

Содержание

Введение
1 Установка бэкенд API
2 Настройки
    2.1 Подключение к базе данных
    2.2 Установка и запуск приложение приложения
3 Тестирование бэкенд
    3.1 My Workspace
    3.2 Запросы на бэкенд
    3.3 Публичные запросы
    3.4 Админские запросы
Заключение
Список использованных источников
Приложения

Введение

1 Установка бэкенд API

Для установки бэкенд API сперва необходимо перейти в нужную папку, и там выполнить команду:

```
git clone https://github.com/mikhalkevich/backendAPI
```

2 Настройки

Сперва необходимо создать и подключиться к базе данных. Затем установить необходимые модули, и выполнить миграции.

2 .1 Подключение к базе данных

Подключение к базе данных осуществляется в файле .env

Файл .env отсутствует в репозитории и его можно воссоздать по примеру .env.example

2 .2 Установка и запуск приложение приложения

Для установки необходимых модулей необходимо выполнить команду:

```
composer install
```

После чего необходимо выполнить миграции:

```
php artisan migrate
```

Запустить проект с помощью команды

```
php artisan serve
```

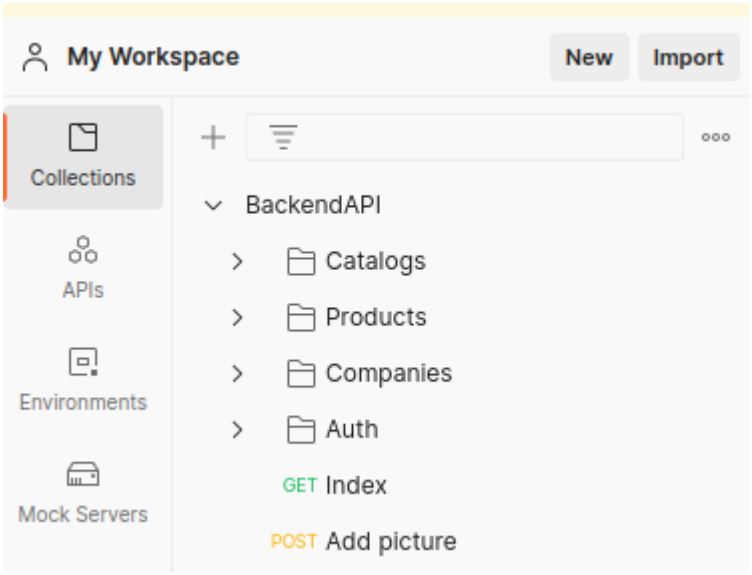
Теперь бэкенд станет доступен для клиентских запросов, по умолчанию по хосту: `http://localhost:8000`

3 Тестирование бэкенд

Для тестирования и совместной с фронтендером разработкой можно воспользоваться программой Postmen.

3 .1 My Workspace

Все необходимые запросы в бэкенд API уже протестированы Postman в рабочем пространстве My Workspace

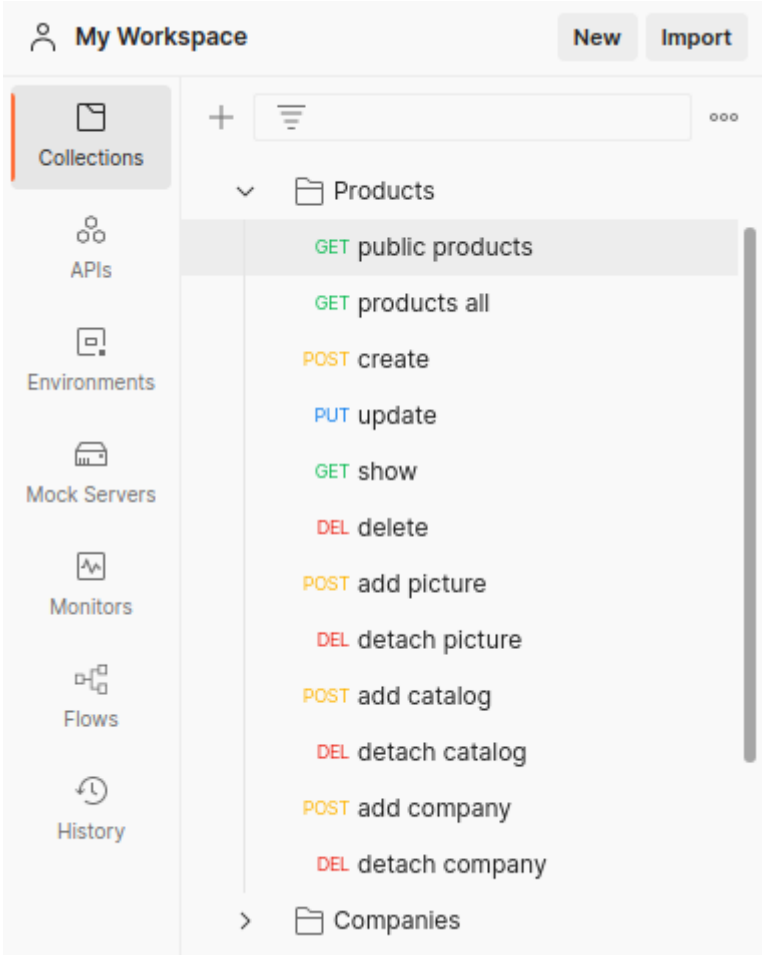


3 .2 Запросы на бэкенд

Со всеми маршрутами приложения можно ознакомиться по адресу:

```
http://localhost:8000/routes
```

В приложении есть админские и публичные запросы.



К публичным запросам относятся запрос на получение списка всех продуктов, и запрос на получение одного продукта.

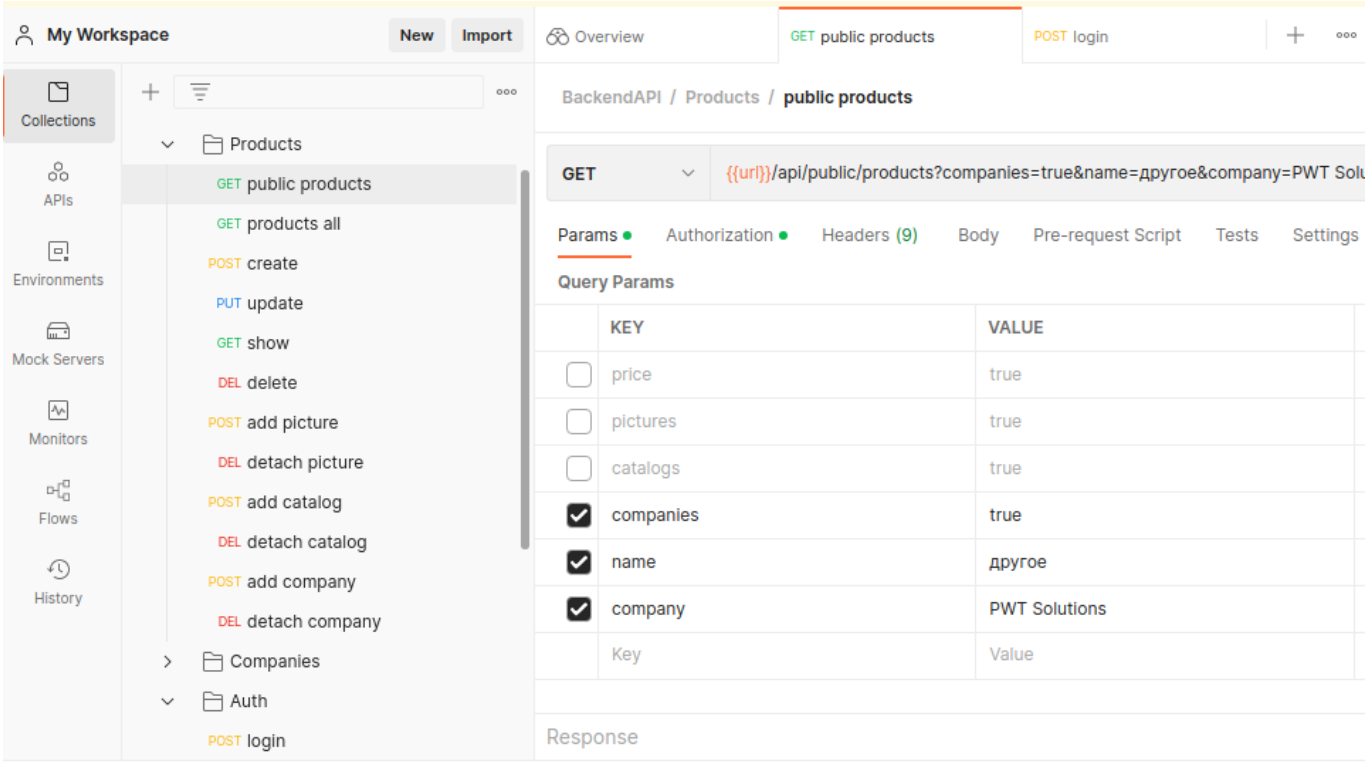
Остальные запросы - админские. К ним относятся CRUD-запросы, а также запросы на привязку и отвязку к продукту компании, каталога и изображения.

Для отправки публичных запросов авторизации не требуется. Для админских - нужна barear-авторизация, т.е. со стороны клиента должен быть отправлен barear token. Рассмотрим публичные и админские запросы подробнее.

3 .3 Публичные запросы

К публичным запросам можно добавлять параметры, с помощью которых осуществляется фильтр и вывод расширенных данных.

Рассмотрим подробнее запрос public products



Запрос в публичную часть начинается с переменной url, значение которой должно быть реализовано в разделе Globals в Postmen. Значение initial value и curent value переменной url для localhost должно быть равно

```
http://localhost:8000
```

Запрос на отображение всех продуктов выглядит так:

```
{{url}}/api/public/products
```

На данный GET запрос бэкенд ответит списком продуктов, которые находятся в массиве data. По умолчанию, бэкенд возвращает только name, description и small_description каждого товара:

```
{
    "data": [
        {
            "id": 1,
            "name": "другое слово",
            "description": null,
            "small_description": "test small description"
        },
        {
            "id": 2,
            "name": "test",
            "description": null,
            "small_description": "test small description"
        }
    ],
    "collection": "public products"
}
```

К ответу бэкенда с помощью соответствующих запросов можно добавить дополнительные данные.

Например, запрос

```
{{url}}/api/public/products?price=true&pictures=true&catalogs=true&companies=true
```

уже вернет более полный массив данных:


```
{
    "data": [
        {
            "id": 1,
            "name": "другое слово",
            "description": null,
            "small_description": "test small description",
            "price": "test",
            "pictures": [
                "http://localhost:8000/storage/1/1665560445587.jpeg"
            ],
            "catalogs": [
                {
                    "id": 1,
                    "name": "Яблоки",
                    "created_at": "2022-09-14T16:21:01.000000Z"
                },
                {
                    "id": 3,
                    "name": "Груши",
                    "created_at": "2022-09-14T17:00:02.000000Z"
                }
            ],
            "companies": [
                {
                    "id": 2,
                    "name": "PWT Solutions",
                    "description": "Разработка сайтов",
                    "small_description": null,
                    "logo": null,
                    "status": null,
                    "link": null
                },
                {
                    "id": 1,
                    "name": "Обновленное название компании",
                    "description": "Описание компании полное",
                    "small_description": null,
                    "logo": null,
                    "status": null,
                    "link": null
```

```
                }
            ]
        },
        {
            "id": 2,
            "name": "test",
            "description": null,
            "small_description": "test small description",
            "price": "test",
            "pictures": [],
            "catalogs": [],
            "companies": []
        }
    ],
    "collection": "public products"
}
```


Для фильтрации данных products тоже имеются соответстующие запросы.

Например, чтобы получить продукты компании PWT, необходимо выполнить следующий запрос:

```
{{url}}/api/public/products?company=PWT
```

Чтобы получить продукты в диапазоне цен от 100 до 200 у.е., можно воспользоваться таким запросом:

```
{{url}}/api/public/products?price_min=100&price_max=200
```

Чтобы получить список продуктов в названии которых присутствует слово phone, можно воспользоваться таким запросом:

```
{{url}}/api/public/products?name=phone
```

Чтобы получить полную информацию об одном продукте, необходимо выполнить запрос:

```
{{url}}/api/public/product/1
```

Тогда последним значением необходимо передать id продукта.

3 .4 Админские запросы

Админ - это пользователь из таблицы users, со значением в is_admin=1

Для выполнения непубличных запросов необходима авторизация в приложении.

После авторизации, клиенту становятся доступны CRUD-запросы, а также запросы на привязку и отвязку к продукту компании, каталога и изображения.

Заключение

Список использованных источников

Приложения