**Документация**

API, разработанного на основе Laravel 10

**Оглавление**

[**1.** **Установка приложения** 3](#_Toc151805003)

[1.1. Установка путем скачивания ZIP файла проекта 3](#_Toc151805004)

[1.2. Установка путем использования консоли, поддерживающей Git 3](#_Toc151805005)

[1.3. Установка путем использования IDE 3](#_Toc151805006)

[**2.** **Подключение базы данных** 3](#_Toc151805007)

[**3.** **Запуск приложения** 3](#_Toc151805008)

[3.1. База данных и миграции 3](#_Toc151805009)

[3.2. Запуск сервера приложения 4](#_Toc151805010)

[**4.** **Функции API** 4](#_Toc151805011)

[4.1. Функции CRUD 5](#_Toc151805012)

[4.1.1. Сущность «Пользователи» 5](#_Toc151805013)

[4.1.2. Сущность «Компании» 7](#_Toc151805014)

[4.1.3. Сущность «Комментарии» 10](#_Toc151805015)

[4.2. Функции статистики 12](#_Toc151805016)

1. **Установка приложения**
   1. Установка путем скачивания ZIP файла проекта

Перейдите на страницу проекта GitHub (https://github.com/d4n1ss1m0/API-Laravel10) и проследуйте по пути «Code/Download ZIP».

* 1. Установка путем использования консоли, поддерживающей Git

Откройте консоль, поддерживающую Git (например, GitBash), перейдите в директорию, где будет храниться проект и выполните команду «git clone https://github.com/d4n1ss1m0/API-Laravel10.git»

* 1. Установка путем использования IDE

Откройте среду разработки приложений, поддерживающую Git и создайте проект используя систему контроля версий и при создании укажите ссылку на проект GitHub «https://github.com/d4n1ss1m0/API-Laravel10.git».

1. **Подключение базы данных**

Создайте базу данных в СУБД, которую поддерживает Laravel.

Откройте файл «.env». Найдите строку «DB\_CONNECTION» и задайте для нее параметр используемого Вами СУБД. Например, для PostgreSQL этот параметр будет «pgsql».

Аналогично задайте параметры для остальных параметров базы данных (рис. 1).

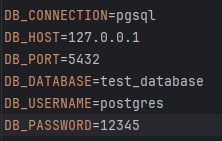


Рисунок 1 – Параметры базы данных в файле «.env»

1. **Запуск приложения**
   1. База данных и миграции

В проекте присутствуют две миграции:

- add\_tables – миграция, которая создает таблицы users, companies, comments;

- add\_data\_to\_tables – миграция, которая заполняет упомянутые выше таблицы тестовыми данными для проверки функций API.

В среде разработки откройте терминал.

Выполните миграцию базы данных для создания таблиц и заполнения их тестовыми данными, используя команду «php artisan migrate».

* 1. Запуск сервера приложения

В среде разработки откройте терминал и введите команду «php artisan serve». Программа будет доступна по ссылке «http://localhost:8000».

1. **Функции API**

API принимает запросы GET, POST, PATCH, DELETE, а возвращает данные объектов в формате JSON для простого обмена данными с внешним приложением. Ссылки на маршруты функций представлены в таблице 1.

Таблица 1.

|  |  |  |  |
| --- | --- | --- | --- |
| Функция | Метод запроса | Маршрут | Параметры |
| Список пользователей | GET | /users |  |
| Отобразить  одного пользователя | GET | /user/{id} | Id – ID пользователя, число |
| Создание пользователя | POST | /users | Form-data, содержащая все атрибуты сущености |
| Изменение пользователя | PATCH | /user/{id}/edit | Form-data, содержащая все атрибуты сущености |
| Удаление пользователя | DELETE | /user/{id} | Id – ID пользователя, число |
| Список компаний | GET | /companies |  |
| Отобразить одну компанию | GET | /company/{id} | Id – ID компании, число |
| Создание компании | POST | /companies | Form-data, содержащая все атрибуты сущености |
| Изменение компании | PATCH | /company/{id}/edit | Form-data, содержащая все атрибуты сущености |
| Удаление компании | DELETE | /company/{id} | Id – ID компании, число |
| Список комментариев | GET | /comments |  |
| Отобразить один комментарий | GET | / comment/{id} | Id – ID комментария, число |
| Создание комментария | POST | / comments | Form-data, содержащая все атрибуты сущености |
| Изменение комментария | PATCH | /comment/{id}/edit | Form-data, содержащая все атрибуты сущености |
| Удаление комментария | DELETE | /comment/{id} | Id – ID комментария, число |
| Комментарии компании | GET | /company\_comments/{id} | Id – ID компании, число |
| Общая оценка компании | GET | /company/rate  /{id} | Id – ID компании, число |
| Рейтинг компаний | GET | /companies/rating |  |

* 1. Функции CRUD

Для удобства тестирования и использования API были предусмотрены формы для создания, изменения и удаления данных. Список маршрутов на эти формы указаны в таблице 2.

Таблица 2.

|  |  |  |
| --- | --- | --- |
| Название формы | Маршрут | Параметры |
| Создание пользователя | /users/create |  |
| Изменение пользователя | /users/{id}/edit | Id – ID пользователя, число |
| Удаление пользователя | /users/{id}/delete | Id – ID пользователя, число |
| Создание компании | /companies/create |  |
| Изменение компании | /companies/{id}/edit | Id – ID компании, число |
| Удаление компании | /companies/{id}/delete | Id – ID компании, число |
| Создание комментария | /comments/create |  |
| Изменение комментария | /comments/{id}/edit | Id – ID комментария, число |
| Удаление комментария | /comments/{id}/delete | Id – ID комментария, число |

* + 1. Сущность «Пользователи»

Отображение всех пользователей (рисунок 2).

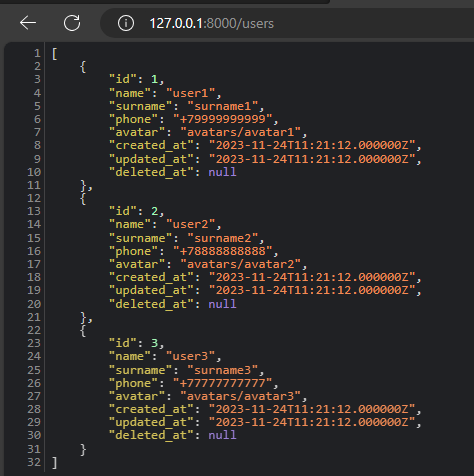


Рисунок 2 – Демонстрация функции отображения пользователей

Отображение одного пользователя

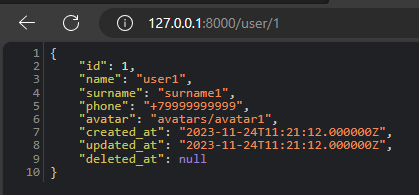


Рисунок 3 – Демонстрация функции отображения одного пользователя

Создание пользователя с параметрами (new\_user, new\_surname,+71111111111) и тестовым изображением.

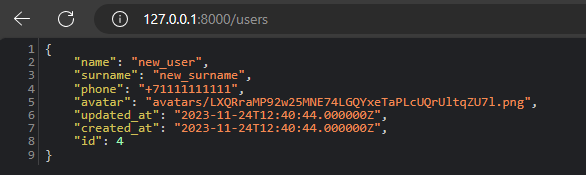


Рисунок 4 – Демонстрация функции создания пользователя

Изменение имени пользователя

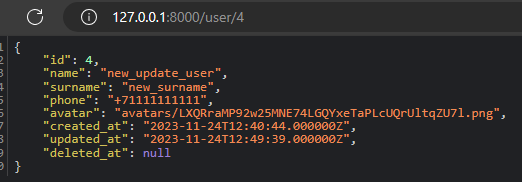


Рисунок 5 – Демонстрация функции изменения пользователя

Удаление пользователя

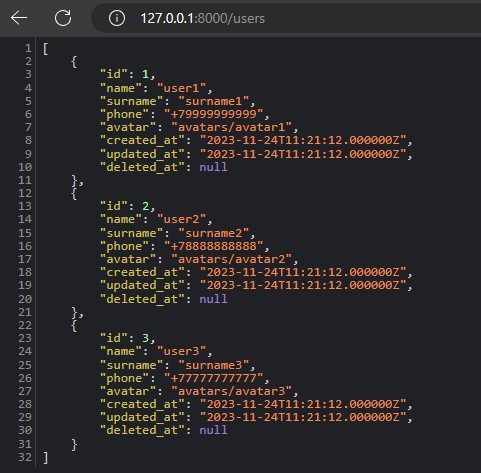


Рисунок 6 – Демонстрация функции удаление пользователя

* + 1. Сущность «Компании»

Отображение всех компаний

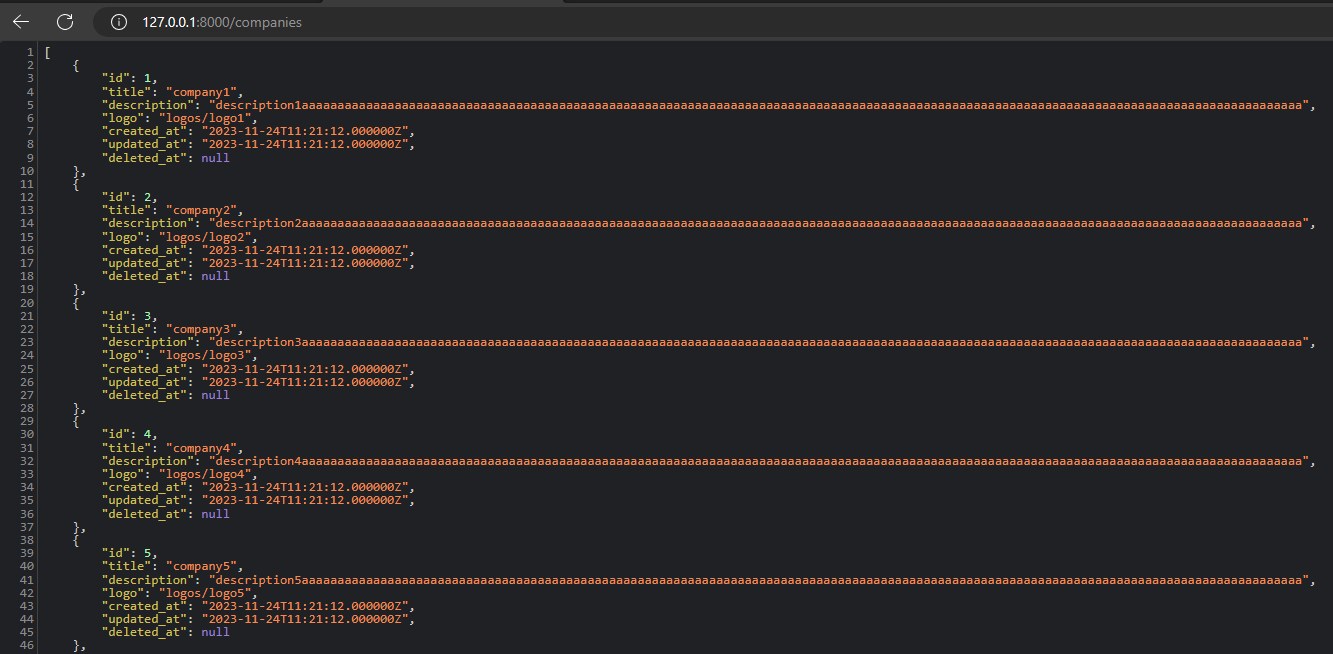


Рисунок 7 – Демонстрация функции отображения всех компаний

Отображение одной компании

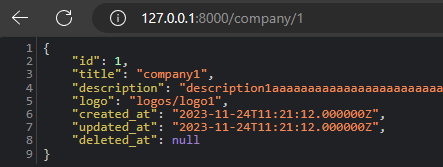


Рисунок 8 – Демонстрация функции отображения одного пользователя

Создание компании с параметрами (new\_company, new\_companyfff….fff) и тестовым изображением

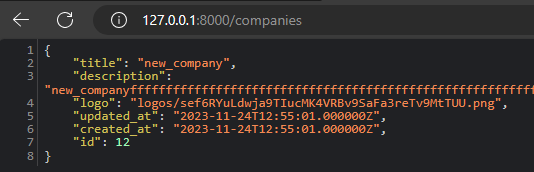


Рисунок 9 – Демонстрация функции создания компании

Изменение названия компании на new\_update\_company

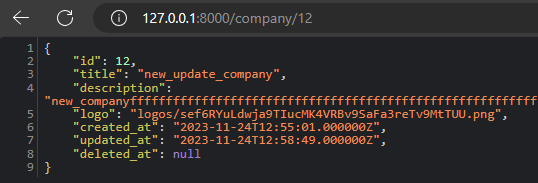


Рисунок 10 – Демонстрация функции изменения компании

Удаление компании.

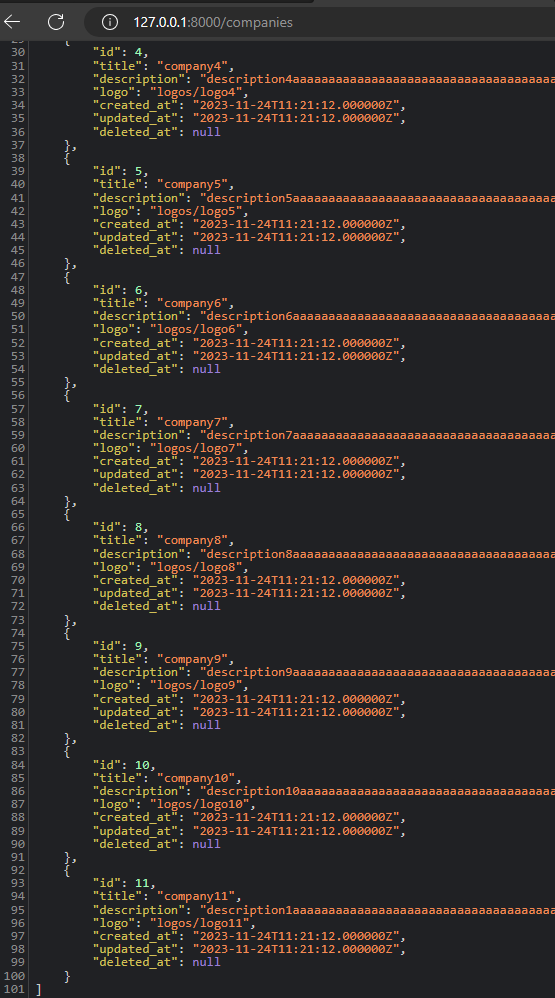


Рисунок 11 – Демонстрация функции удаления компании

* + 1. Сущность «Комментарии»

Отображение всех комментариев

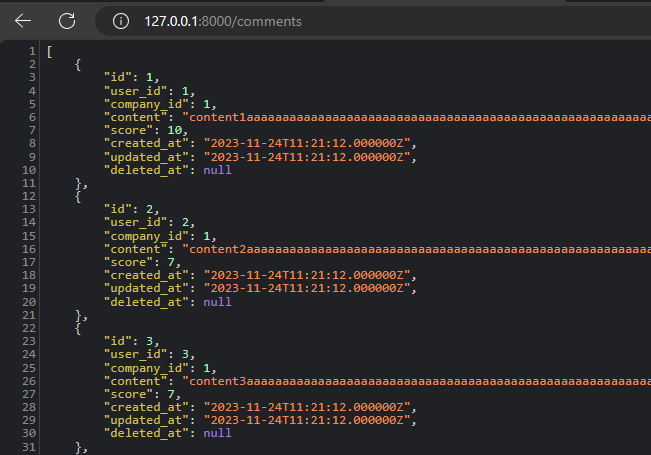


Рисунок 12 – Демонстрация функции отображения всех комментариев

Отображение одного комментария

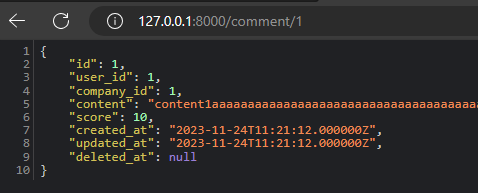


Рисунок 13 – Демонстрация функции отображения одного комментария

Создание комментария с параметрами (1,1, new\_commentddd…ddd, 9)

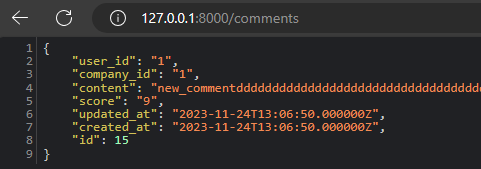


Рисунок 14 – Демонстрация функции создания комментария

Изменение комментария

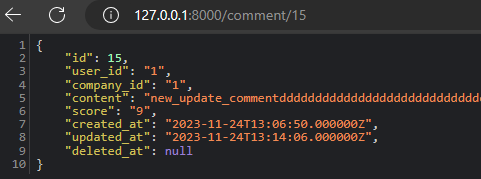


Рисунок 15 – Демонстрация функции изменения комментария

Удаление комментария



Рисунок 16 – Демонстрация функции удаления комментария

* 1. Функции статистики

Отображение всех комментариев компании

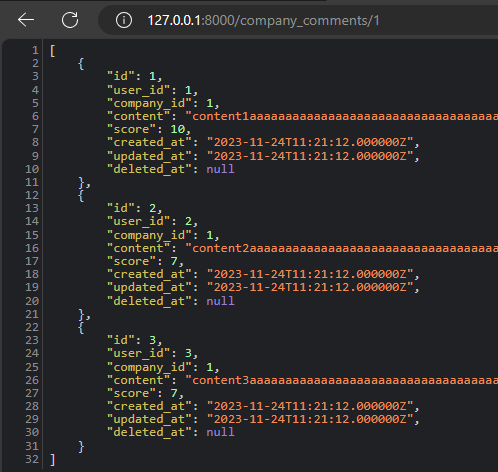


Рисунок 17 – Демонстрация функции отображения всех комментариев компании

Отображение общей оценки компании

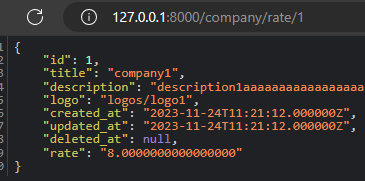


Рисунок 18 – Демонстрация функции отображения общей оценки компании

Отображение рейтинга компаний

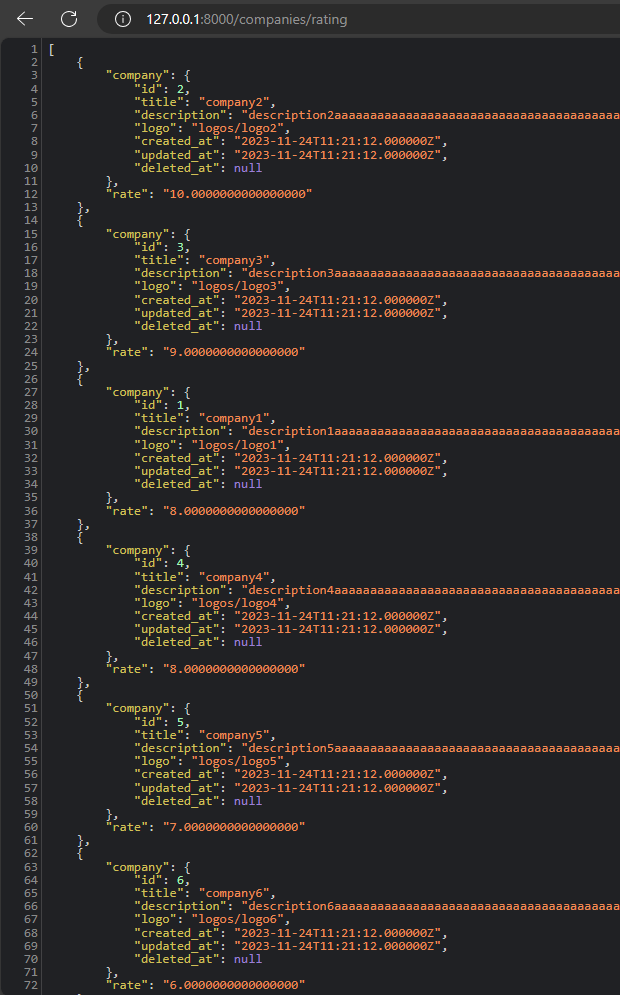


Рисунок 19- Демонстрация функции отображения рейтинга компаний