

# MTG Project Documentation

**Версия:** 1.0

**Автор:** MrVladikk

**Дата:** Март 2025

## 1. Обзор проекта

**MTG Project** – это веб-приложение, разработанное на основе Django для управления коллекцией карт Magic: The Gathering. Приложение предоставляет следующие возможности:

- **Импорт данных** о карточках из CSV/Excel-файлов.
- **Скачивание изображений** карт с API Scryfall.
- **Обновление ссылок** на изображения в базе данных.
- **Управление колодами** (добавление, редактирование, удаление) через административную панель.

## 2. Стек технологий

- **Python 3.12**
- **Django** – основной веб-фреймворк
- **SQLite** – база данных (по умолчанию)
- **Requests** – для выполнения HTTP-запросов
- **Pandas** – для обработки CSV/Excel файлов
- **tqdm** – для отображения прогресс-бара при загрузке изображений

## 3. Структура проекта

```
mtg_project/
├── db.sqlite3
├── manage.py
├── add_deck.py
├── update_image_urls.py
├── mtg_app/
│   ├── admin.py
│   ├── models.py
│   ├── parser_mtg/
│   │   └── my_cards.csv
│   └── ...
└── mtg_project/
    ├── settings.py
    └── urls.py
```

## 4. Установка и настройка

### 4.1. Клонирование репозитория

```
git clone https://github.com/MrVladikk/mtg_project.git
cd mtg_project
```

## 4.2. Создание виртуального окружения и установка зависимостей

```
python -m venv venv
source venv/bin/activate # Linux/macOS
venv\Scripts\activate # Windows
pip install -r requirements.txt
```

## 4.3. Применение миграций и запуск сервера

```
python manage.py makemigrations
python manage.py migrate
python manage.py runserver
```

Перейдите в браузере по адресу: <http://127.0.0.1:8000>.

# 5. Модели и администрирование

## 5.1. Основные модели

Определены в `mtg_app/models.py`:

- **Set** – хранит информацию о наборах карт.
- **Card** – содержит данные о каждой карте.
- **Deck** – позволяет создавать колоды.

## 5.2. Административная панель

В `mtg_app/admin.py` зарегистрированы модели **Set**, **Card**, **Deck**. Создайте суперпользователя:

```
python manage.py createsuperuser
```

Затем зайдите в админку по адресу <http://127.0.0.1:8000/admin>.

# 6. Парсеры и скрипты

## 6.1. Парсер изображений

Находится в `mtg_app/parser_mtg/your_parser.py`:

- **download\_image** – получает изображение по Scryfall ID.
- **is\_image\_downloaded** – проверяет, существует ли файл.

## 6.2. Скрипт обновления ссылок

Файл `update_image_urls.py` обновляет поле `image_url` в базе, проверяя наличие файлов изображений.

## 6.3. Скрипт добавления колод

Файл `add_deck.py` позволяет добавлять колоды в базу данных.

## 7. Эксплуатация

### 7.1. Запуск сервера

```
python manage.py runserver
```

### 7.2. Импорт данных

```
python manage.py import_cards
```

### 7.3. Скачивание изображений

```
python mtg_app/parser_mtg/your_parser.py
```

### 7.4. Обновление ссылок

```
python update_image_urls.py
```

## 8. Развертывание

### 8.1. Настройки для продакшена

- Установите `DEBUG=False` в `settings.py`.
- Добавьте домены в `ALLOWED_HOSTS`.
- Выполните сбор статических файлов:

```
python manage.py collectstatic
```

- Используйте Gunicorn и Nginx для запуска в продакшене.

## 9. Заключение

**MTG Project** – это удобное Django-приложение для коллекционеров MTG, позволяющее хранить, обрабатывать и управлять данными о картах. Благодаря гибкости Django, его можно расширять, добавлять новые функции и интегрировать с внешними сервисами.