

# Документация к боту

## Общее описание

Этот телеграм-бот позволяет пользователям создавать PDF-файлы с подписями учеников. Основной функционал:

- Прием исходного файла (PDF, DOC или DOCX)
- Прием списка учеников
- Генерация отдельных PDF-файлов для каждого ученика с их подписями

## Структура проекта

### 1. config.py

Конфигурационный файл с настройками проекта.

**Константы:**

- BASE\_DIR - корневая директория проекта
- DEFAULT\_INPUT\_PDF - путь к стандартному шаблону PDF
- DEFAULT\_OUTPUT\_FOLDER - папка для готовых PDF
- FONTS\_FOLDER - папка со шрифтами
- ROBOTO\_FONT - путь к шрифту Roboto
- SUPPORTED\_FORMATS - словарь поддерживаемых форматов файлов
- STATES - состояния бота (ожидание файла или списка учеников)

### 2. converters.py

Модуль для конвертации файлов между форматами.

**Функции:**

- `convert_to_pdf(input_path, output_folder)` - конвертирует файл в PDF
- `cleanup_temp_files()` - очищает временные файлы

### 3. handlers.py

Обработчики сообщений телеграм-бота.

**Класс Handlers:**

- `handle_start(message)` - приветственное сообщение
- `handle_help(message)` - справка по использованию бота
- `handle_create_pdf(message)` - начинает процесс создания PDF
- `process_file_step(message)` - обрабатывает загруженный файл
- `process_students_step(message)` - обрабатывает список учеников
- `cleanup_user_data(chat_id)` - очищает временные данные пользователя

## 4. keyboards.py

Модуль для создания клавиатур бота.

**Функции:**

- `get_main_keyboard()` - основная клавиатура (старт/помощь)
- `get_cancel_keyboard()` - клавиатура с кнопкой "Отмена" (закомментирована)

## 5. main.py

Основной файл для запуска бота.

**Функционал:**

- Инициализация бота
- Регистрация обработчиков команд и сообщений
- Запуск бота в режиме polling

## 6. operations.py

Основной модуль для обработки файлов.

**Класс FileProcessor:**

- `convert_to_pdf(input_path, output_folder)` - конвертирует DOCX в PDF
- `create_pdfs_for_students(students_list, input_path, output_folder)` - создает PDF с подписями
- `process_single_student(student, input_pdf_path, output_folder)` - добавляет подпись ученика в PDF
- `register_fonts()` - регистрирует шрифты
- `create_watermark(text, width, height)` - создает подпись (водяной знак)

## Процесс работы

1. Пользователь запускает бота командой /start или нажимает кнопку "Создать PDF с подписями"
2. Бот переходит в состояние ожидания файла (WAITING\_FOR\_FILE)
3. Пользователь отправляет файл (PDF, DOC или DOCX)
4. Бот проверяет формат файла и сохраняет его
5. Бот переходит в состояние ожидания списка учеников (WAITING\_FOR\_STUDENTS)
6. Пользователь отправляет список учеников (каждый с новой строки)
7. Бот создает отдельные PDF-файлы для каждого ученика с их подписями
8. Бот отправляет готовые файлы пользователю

## Требования

- Python 3.7+
- Установленные зависимости (см. requirements.txt)
- Шрифт Roboto (опционально)