

# Разработка интеграции TeamFlame с мессенджерами

Kurenkov AI | misis

## Введение

Наша задача состояла в создании интеграции между платформой TeamFlame и одним из популярных мессенджеров, такими как Slack, Microsoft Teams или Telegram. Интеграция должна позволять пользователям взаимодействовать с задачами и досками TeamFlame, не выходя из мессенджера

## Требования

1. Создание бота или приложения, которое будет взаимодействовать с API TeamFlame и API выбранного мессенджера
2. Возможность создания новых задач, изменения статуса задачи, добавления комментариев и просмотра информации о задачах и досках
3. Удобный и интуитивно понятный интерфейс взаимодействия с задачами через мессенджер

*Дополнительные бонусные задания*

4. Уведомления о событиях в TeamFlame (например, новые задачи, изменения статуса) через мессенджер
5. Возможность назначения ответственных лиц и управления сроками выполнения задач
6. Расширение функциональности интеграции, например, добавление возможности создания досок и списков задач

## Продукт

Рабочее приложение или бот, способное интегрироваться с выбранным мессенджером и позволяющее пользователям управлять задачами и досками в TeamFlame через мессенджер

## Краткий путь пользователя

Исполненный нами продукт -бот в самом начале в нем происходит безопасная авторизация. Далее пользователю дается возможность выбрать доску из любого пространства или проекта. Просмотр доски с задачами обеспечивается в отдельном окне. Там же можно смотреть задачи, добавлять их, оставлять комментарии

## Решение

Решение сейчас это комплекс сервисов, которые можно протестировать в телеграмм @TeamFlame\_bot

## Особенности и архитектура

Особенность нашего решения в том, что оно состоит из двух независимых блоков. Бот в телеграмме и backend часть, связанная с базой данных и TeamFlame API. Services - названные для удобства функции и классы, которые отправляют requests к api

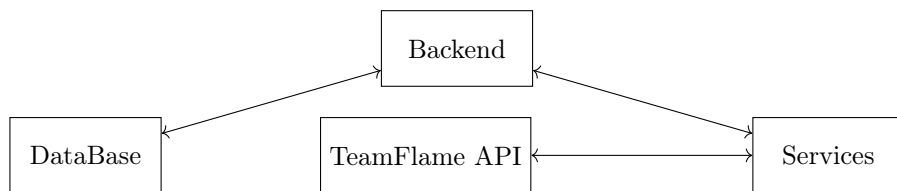


Рис. 1: Block Diagram Representation of Backend

Вторая часть - интегрируемая, это боты(в нашем случае телеграмм бот). Наш построен на библиотеке *telebot*

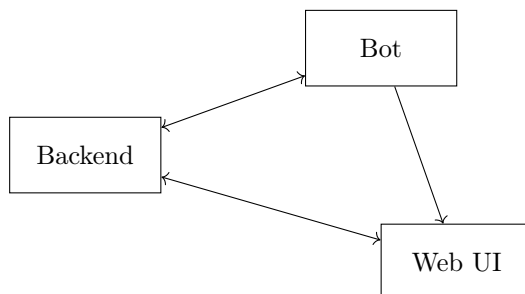


Рис. 2: Block Diagram Representation of Bot

Как они взаимодействуют? Бот отправляет запросы к бекенду. При достижении точки входа в доску, человек проваливается в интерфейс *telegram mini app*

## Описание серверной части

### Services

Services - каталог классов

global-inno-hack/services

Files root system:

services

|-auth.py

|-boards.py

|-columns.py

|-comments.py

|-projects.py

|-spaces.py

|-tasks.py

|-users.py

Первое, с чем придется столкнуться, это авторизация. С помощью почты и пароля вызывается *auth.AuthService.sign\_in* на выходе получаем json файл с токеном. Он записывается для доступа к апи TeamFlame

Вот пример класса для досок. По сути, это имплементация исходного апи нужных для нас методов. Другие классы устроены примерно также

```
class BoardsService():
    API_URL = 'https://api.teamflame.ru/board'
    def get_board_by_id(self,
                       email: str,
                       board_id: str):
        ...
        return board.json()

    def get_board_by_project_id(self,
                               email: str,
                               project_id: str):
        ...
        return boards.json()
    def create_board(self,
                    email: str,
                    name: str,
                    space_id: str,
                    project_id: str):
        ...
        return board.json()
```

## FastAPI

Серверная часть описана в папке

`global-inno-hack/api`

Files root system:

```
server.py
api
|-main.py
|-auth.py
|-boards.py
|-columns.py
|-comments.py
|-projects.py
|-spaces.py
|-tasks.py
|-users.py
```

Файловая система и смысл повторяет Services, но теперь это апи, к которому можно обратиться извне и подключать любые мессенджеры. *main.py* менеджерит роутеры. Главный файл, в котором создатся `app()` - *server.py*

## База данных

В основе лежит реляционная база данных PostgreSQL. Мы храним три таблицы:

*tokens*: Таблица для нашего бота. Связывает почту пользователя с его токенами мессенджера

*users*: Таблица, которая соединяет `user_id` из мессенджера с почтой

С помощью такого инструмента как Alembic в базе данных автоматически проходят миграции

## Масштабируемость

Для взаимодействия с другими мессенджерами достаточно написать бота, который будет обращаться к бекенду нашего приложения

### Бот

Бот написан на библиотеке *telebot*. Все методы обращаются к написанному нами апи. С помощью кнопок пользователь выбирает доску из любого пространства или проекта. Далее у него есть возможность перейти в *telegram mini app*

### Telegram Mini App

Написанное на *React* веб приложение которое обеспечивает работу с доской. В том числе: просмотр, создание задач, добавление комментариев, назначение людей