

## Описание

PyDataAPI - это API инструмент, основанный на библиотеке FastAPI, для взаимодействия с базами данных и конечными интерфейсами (Web/Desktop).

## Установка

Для развертывания проекта PyDataAPI необходимо скачать **архив** с проектом. После скачивания необходимо:

1. Создать виртуальное окружение для проекта:

```
C:\Users\user\proj_folder>py -m venv env
```

2. После создания виртуального окружения, необходимо выполнить команду `pip install`, в качестве аргумента указав файл `requirements.txt` из архива с проектом:

```
(env) C:\Users\user\proj_folder>pip install -r requirements.txt
```

3. Для запуска проекта выполняем команду:

```
(env) C:\Users\user\proj_folder>uvicorn main:app --reload
```

## Принцип работы

API - инструмент для передачи данных между клиентскими приложениями и серверной частью. В качестве строки обращения принимает в себя HTTP запрос формата:

```
http://host:8000/api/request?[query]
```

где `host` - имя вашего локального (`localhost/127.0.0.1`) или глобального (`my.coolAPI.com`) сервера.

## Форматы обращений к API:

Тип запроса	Строка запроса	От
index	http://host:8000/api	HTML вид стр
create	http://host:8000/api/request? table=&method=create&id=&username=&uniq_key=&role=	JSON {"re res
read	http://host:8000/api/request? table=&method=read&id=&username=&uniq_key=&role=	JSON {"re [[da [da
delete	<a href="http://127.0.0.1:8000/api/request?table=&amp;method=delete&amp;id=&amp;username=&amp;uniq_key=&amp;role=">http://127.0.0.1:8000/api/request? table=&amp;method=delete&amp;id=&amp;username=&amp;uniq_key=&amp;role=</a>	JSON {"re res
update	<a href="http://127.0.0.1:8000/api/request?table=users&amp;method=update&amp;id=2&amp;username=&amp;uniq_key=&amp;role=1:tochange">http://127.0.0.1:8000/api/request? table=users&amp;method=update&amp;id=2&amp;username=&amp;uniq_key=&amp;role=1:tochange</a>	JSON {"re

## Форматы строки запроса:

Параметр	Содержимое	Необходимость
table	Имя SQL таблицы в формате строки. Прим.: ...table=users&...	create: ++ read: ++ delete: ++ update: ++
method	Тип выполняемой операции. Прим.: ...method=read&...	create: ++ read: ++ delete: ++ update: ++
schemas	Параметры соответствующие схеме таблицы (из параметра table строки запроса). Прим.: ...table=users&method=read&id=1&username=&...	create: + read: +- delete: +- update: +

## Как подключить свою таблицу к API?

1. Убедитесь что таблица существует в подключенной базе данных.
2. Откройте файл `schemas.json` и укажите схему вашей базы в виде имени таблицы и набора столбцов.

```
#schemas.json
{
  ...
  "users" : [
    "id",
    "username",
    "uniq_key",
    "role"
  ],
  "posts" : [
    "id",
    "title",
    "text",
    "short_description",
    "long_description",
    "date",
    "author"
  ]
  ...
}
```

3. Проверьте подключение к вашей таблице с помощью строки запроса (для безопасности, делайте это в режиме `read` ).

```
http://host:8000/api/request?
table=posts&method=read&id=&title=test&text=&short_description=&long_descripti
on=&date=&author
```