Оглавление

[**Запуск check-list backend приложения** 2](#_Toc156917799)

[**Облегченный запуск** 10](#_Toc156917800)

[**Настройка подключения к БД** 11](#_Toc156917801)

[**Работа с swagger** 12](#_Toc156917802)

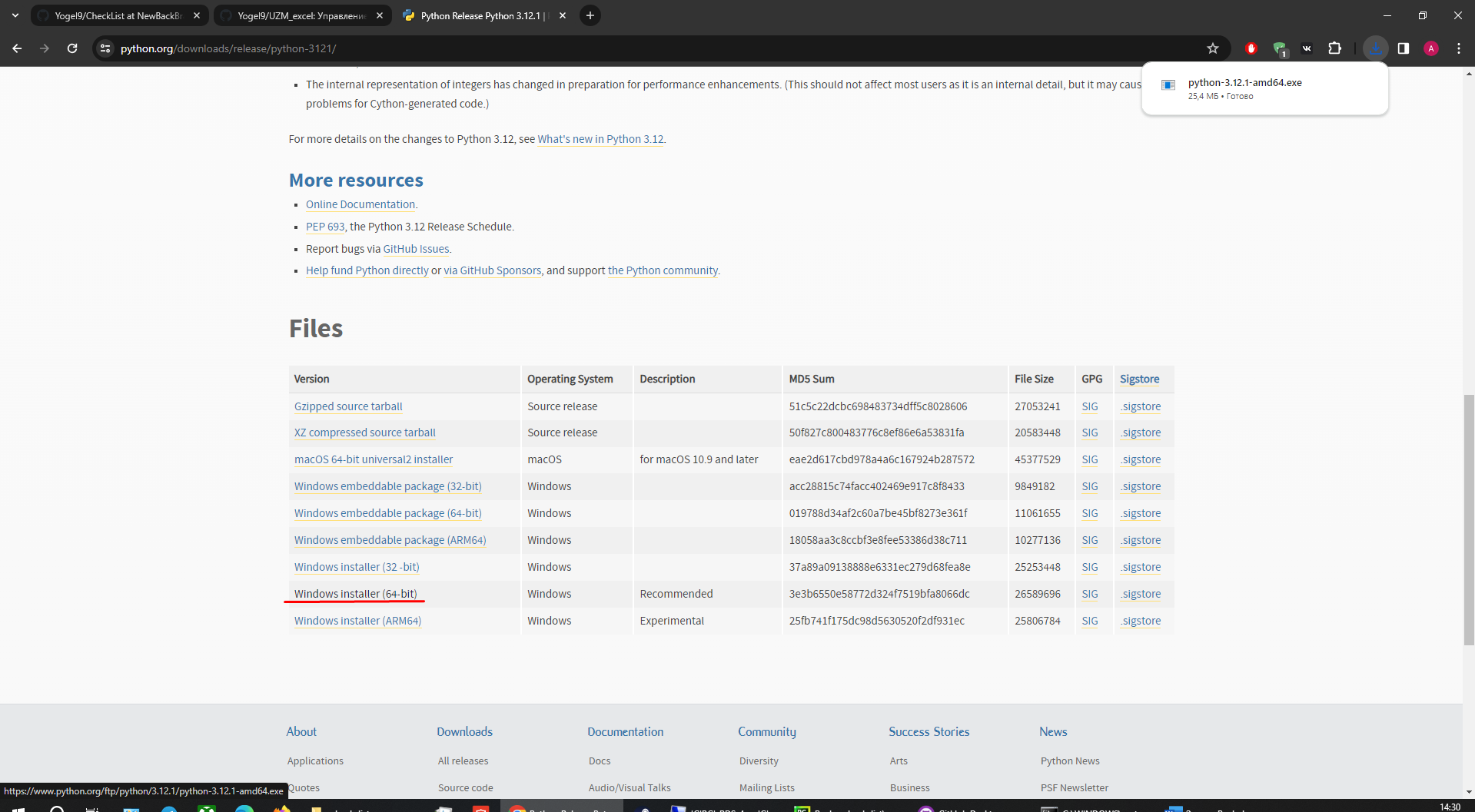
[**Возможные ошибки** 14](#_Toc156917803)

# **Запуск check-list backend приложения**

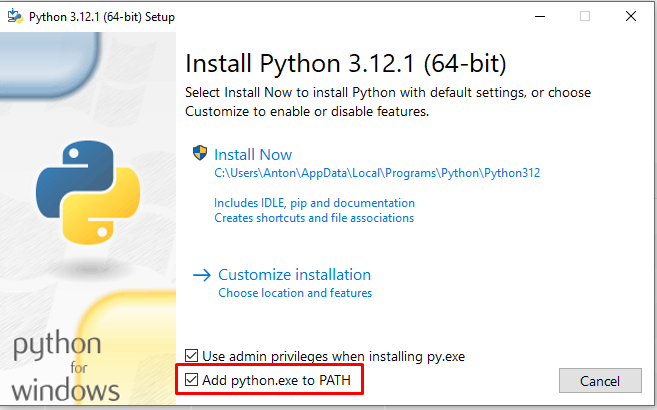
**Шаг 1. Установка Python интерпретатора** (пропустить, если уже установлен)

Скачиваем и устанавливаем любую версию Python 3 старше 3.10 с официального сайта

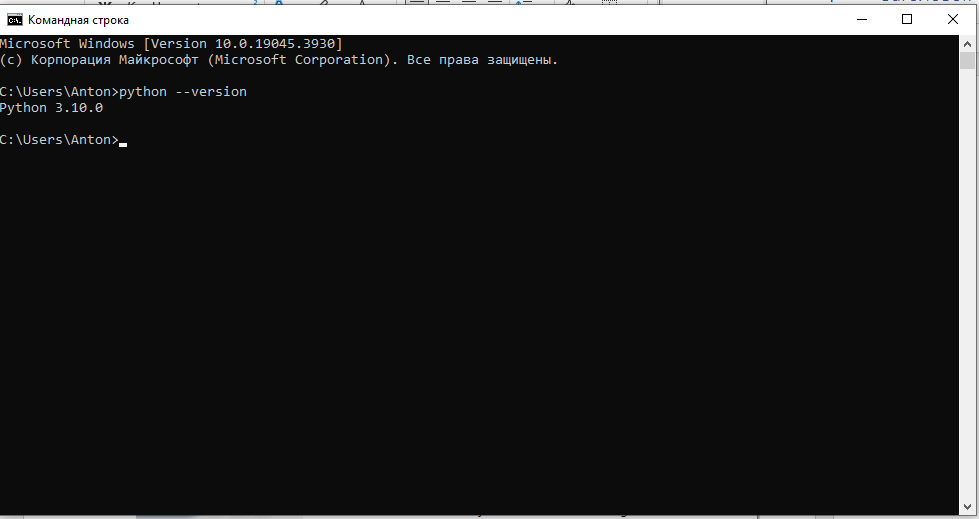
<https://www.python.org/downloads/windows/>



Во время установки обязательно добавляем python в системную переменную PATH:



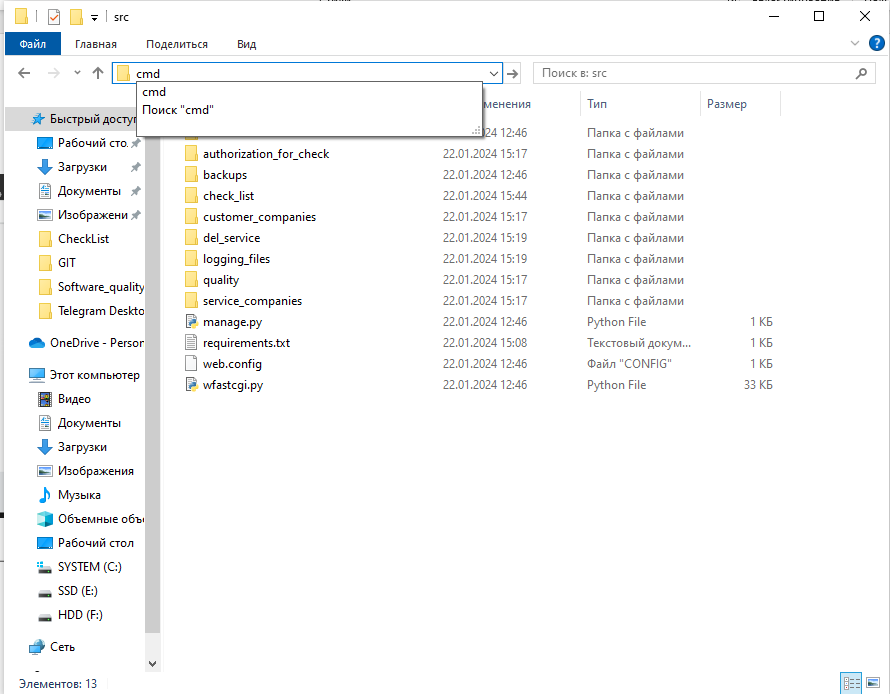
***Проверить*** установленный Python можно через командную строчку cmd командой python --version

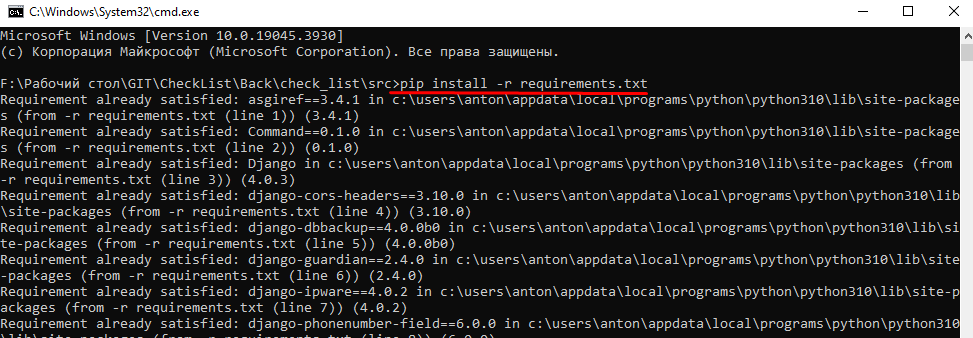


**Шаг 2. Установка библиотек** (пропустить, если уже установлены)

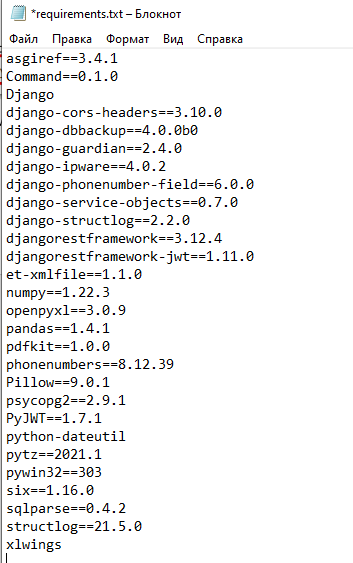
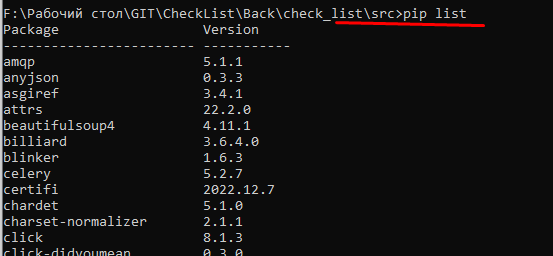
Переходим в папку CheckList\Back\check\_list\src.

Открываем cmd в данном репозитории(либо пишем cmd в строке системной навигации, либо в открытом терминале при помощи команды CD попадаем в src).



Команда pip install -r requirements.txt установит все необходимые python библиотеки из файла requirements.txt  
  


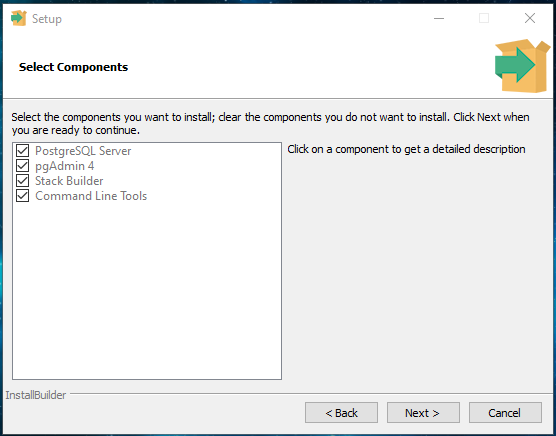
***Проверить*** установленные Python библиотеки можно через команду pip list. В списке должны быть названия из файла requirements.txt.



**Шаг 3. Установка БД** (пропустить, если уже установлен)

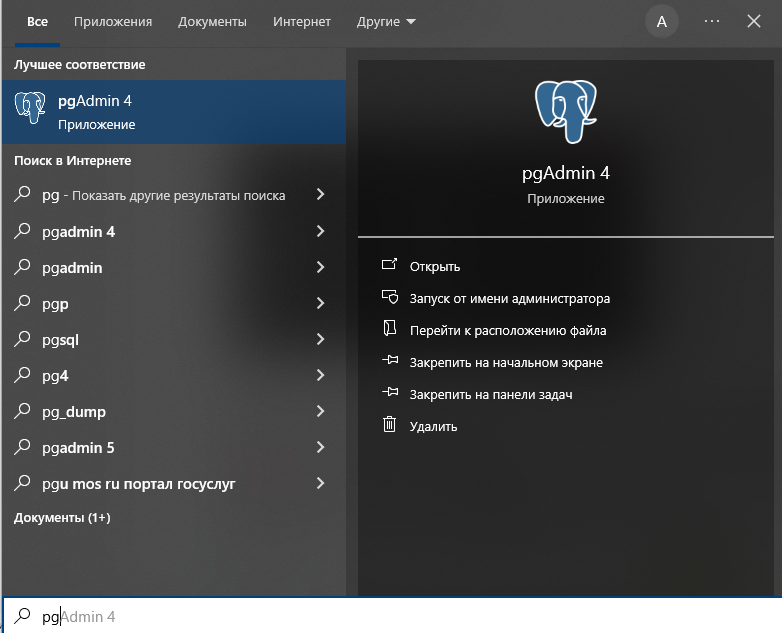
Для локального запуска необходимо иметь установленный PostgreSQL. Скачать с официального сайта https://www.enterprisedb.com/downloads/postgres-postgresql-downloads (проверялось на ***Version 16.1 версия для Windows x86-64***)

В инсталлере для установки обязательны pgAdmin, PostgreSQL Server остальные параметры опциональны. Далее по инструкции инсталлера.

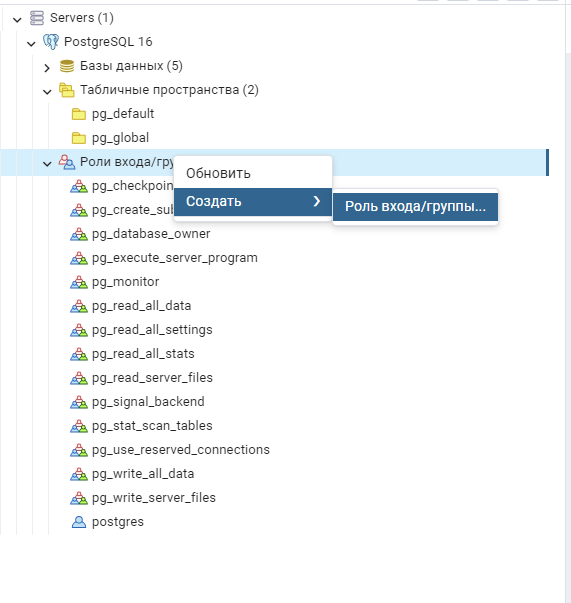


**Шаг 4. Настройка БД** (пропустить, если уже есть)

Заходим в pgAdmin

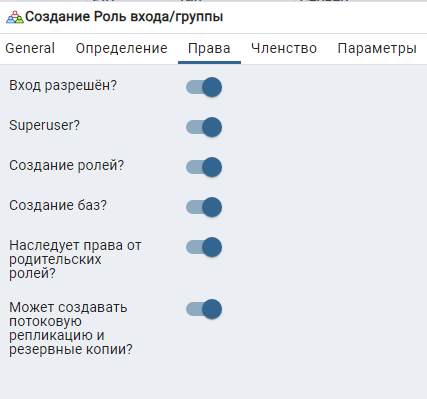


Ecли в списке ролей нет **postgres**, создаем новую роль.

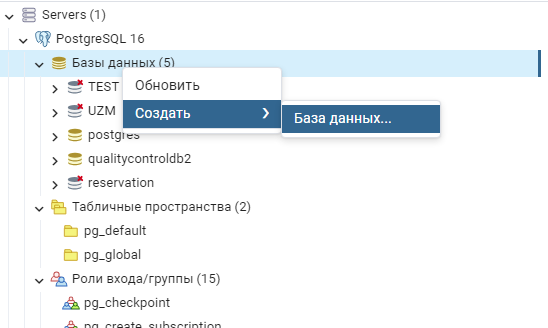


Для логина и пароля рекомендуется использовать данные: **postgres** / **sqlp@$$word.**

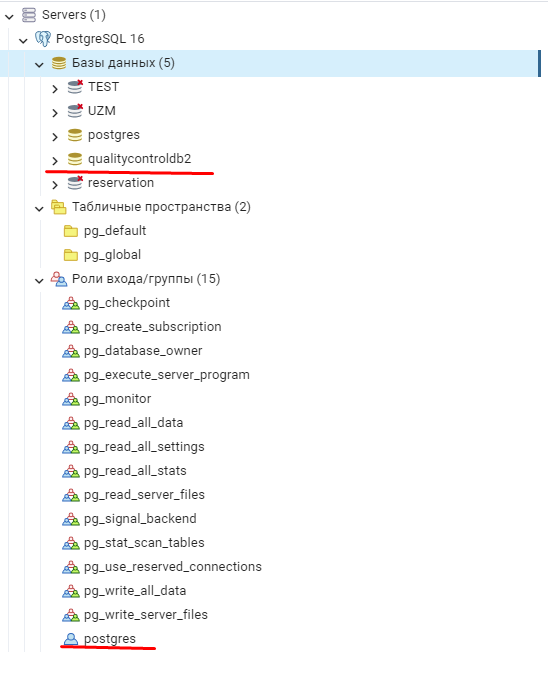
Также пользователю необходимо выставить следующие права:



Если в списке баз данных нет QualityControlDB2, создаем новую БД.



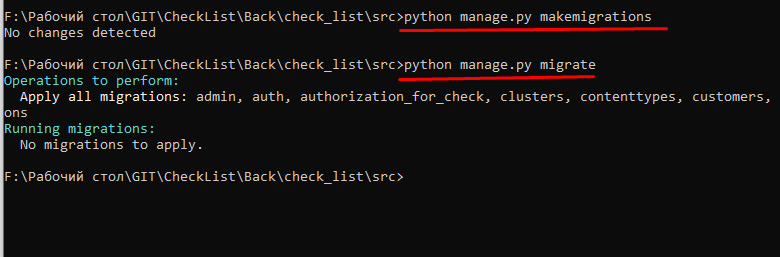
***По итогу*** мы должны иметь QualityControlDB2 в списке БД и postgres в Роли***:***



**Шаг 5. Миграция в БД** (пропустить, если уже проведена)

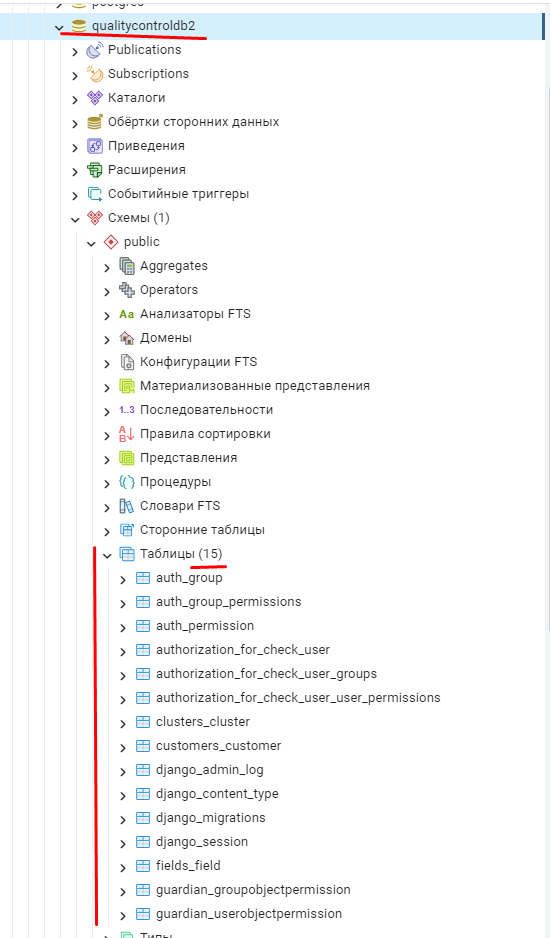
В консоли выполняем команды миграции, в репозиторииCheckList\Back\check\_list\src.

python manage.py makemigrations – Создаёт новые миграции

python manage.py migrate – Применяет текущие миграции

***Важно*** при первом запуске создается пуста БД для допуска к API рекомендую создать суперпользователя (см [Возможные ошибки. Создание пользователя.)](#_Возможные_ошибки)

***Проверить*** успех миграций можно в PgAdmin. В БД будет добавлено ~15 таблиц:

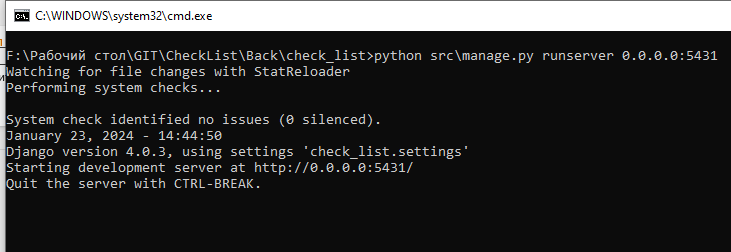


**Шаг 6. Запуск приложения**

Для запуска приложения в папке \CheckList\Back\check\_list есть check\_list\_start.bat , который запустит приложение по локальному адресу 127.0.0.1 с портом 5431 (Приложение будет доступно по адресу http://127.0.0.1:5431/).

Запуск приложения ***также можно*** осуществить через консоль, командой python manage.py runserver (мы должны находиться в репозитории с файлом manage.py, он лежит в CheckList\Back\check\_list\src). Тогда приложение будет доступно по адресу http://127.0.0.1:8000/.

Стандартный вариант запуска:



# **Облегченный запуск**

Шаги 1,2 обязательные, если на нашей машине нет Python

Шаги 3,4,5 нужны для обеспечения связи Python и БД.

Чтобы не устанавливать PostgreSQL и пропустить шаги 3,4 можем в файле settings.py установить в качестве БД Sqlite (см. [Настройка подключения к БД](#_Настройка_подключения_к)). Шаг 5 остаётся обязательным.

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.sqlite3',

'NAME': 'mydatabase', # This is where you put the name of the db file.

# If one doesn't exist, it will be created at migration time.

}

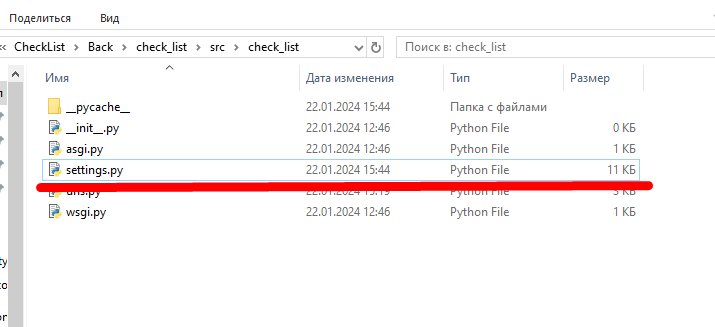
}

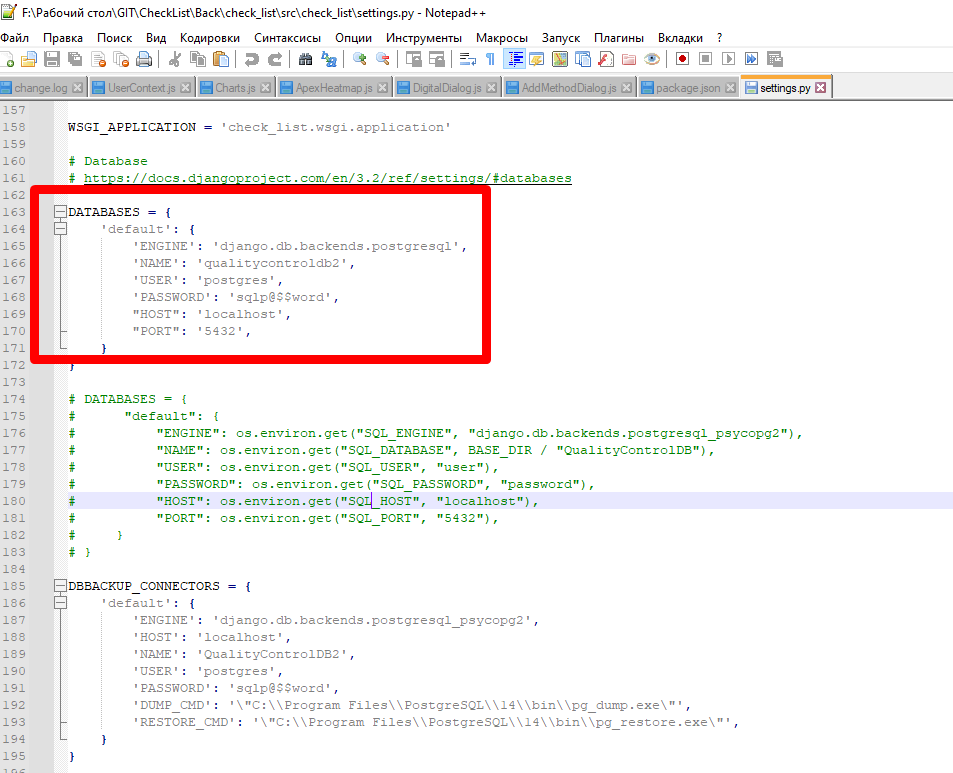
Шаг 6. Единственный обязательный шаг, если у нас настроена БД и установлен Python

# **Настройка подключения к БД**

Если мы хотим использовать свои роли или свои БД при работе с бэкэнд приложением, можем изменить текущие настройки подключения. Для этого:

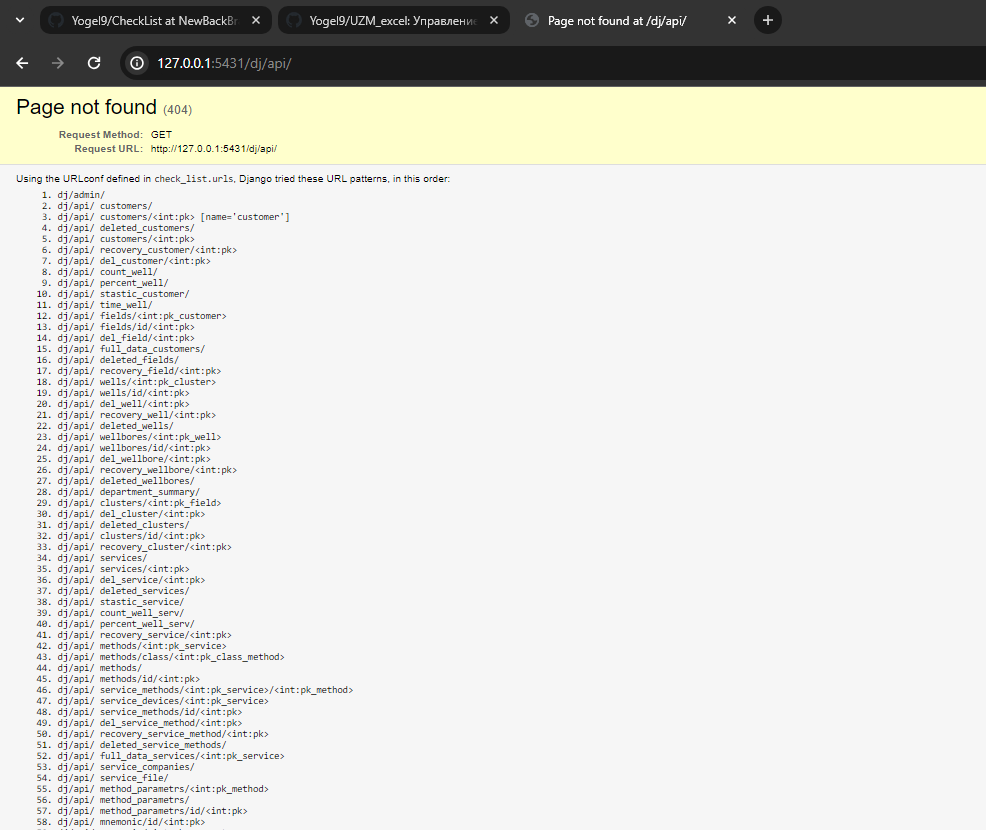
1. Переходим в settings.py, репозиторий CheckList\Back\check\_list\src\check\_list.
2. Находим DATABASES
3. Меняем параметры 'ENGINE', 'NAME', 'USER', 'PASSWORD',"HOST","PORT"





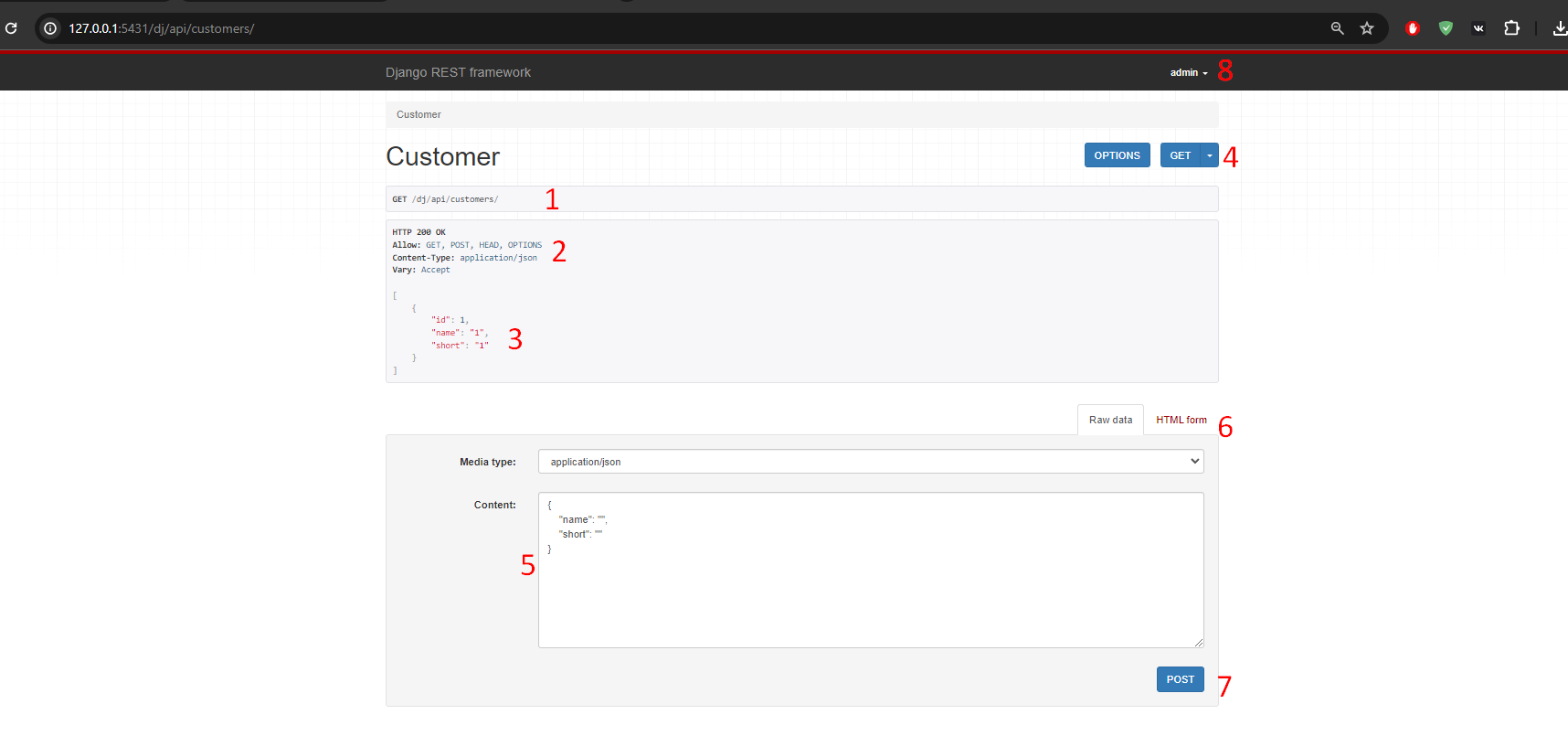
# **Работа с swagger**

В браузере при переходе напрямую по http://127.0.0.1:5431/dj/api/ можно увидеть список доступных адресов backend приложения.

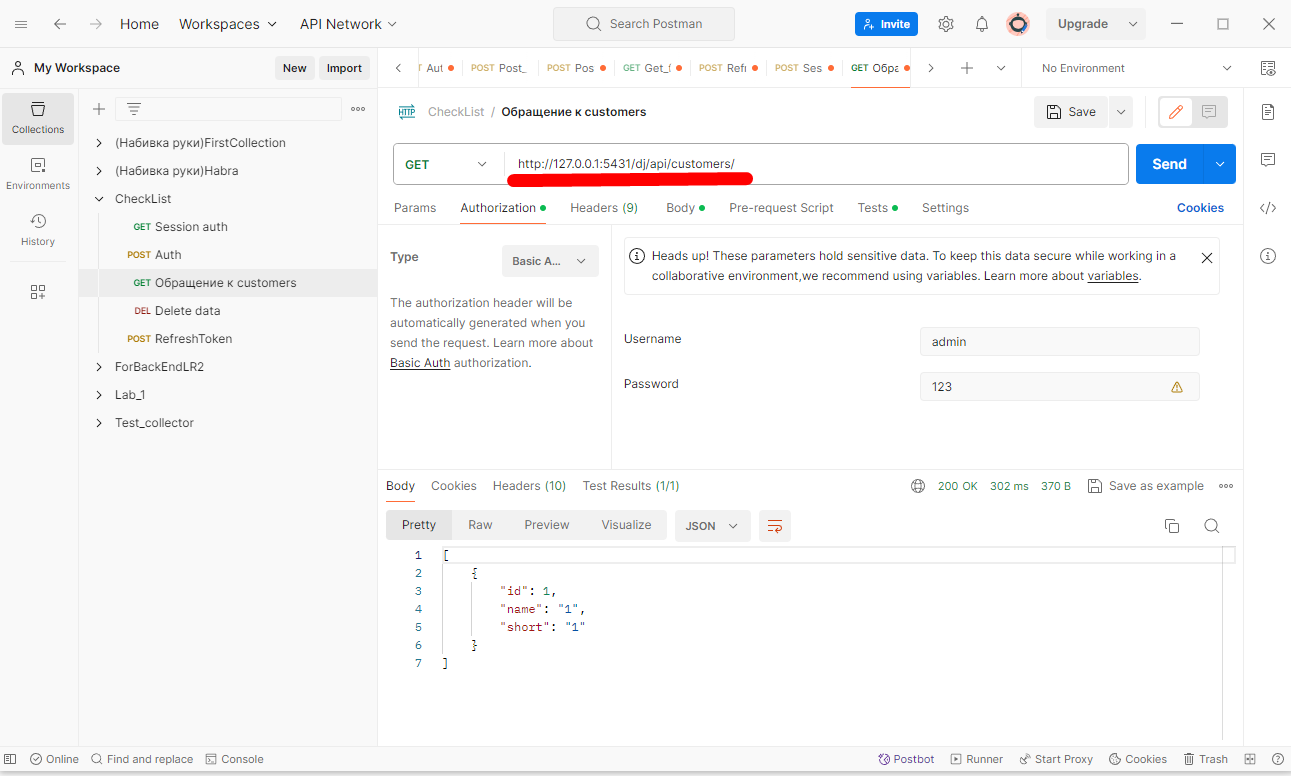


Возьмём в пример адрес <http://127.0.0.1:5431/dj/api/customers/>.

1. Текущий HTTP метод, по которому обращаемся к <http://127.0.0.1:5431/dj/api/customers/> .
2. Заголовок запроса и доступные для http://127.0.0.1:5431/dj/api/customers/ методы HTTP запросов.
3. Формат данных в JSON который получаем при GET обращении (Кнопка 4).
4. Выполнение GET запроса по адресу <http://127.0.0.1:5431/dj/api/customers/> .
5. Форма для POST запроса к <http://127.0.0.1:5431/dj/api/customers/> .
6. Переключение формата формы (Raw data – эмулирует отправку JSON с Frontend приложения, HTML form – эмулирует базовую html форму).
7. Отправка POST запроса с данными формы 5.
8. Для обращения к некоторым адресам требуется авторизоваться в системе.



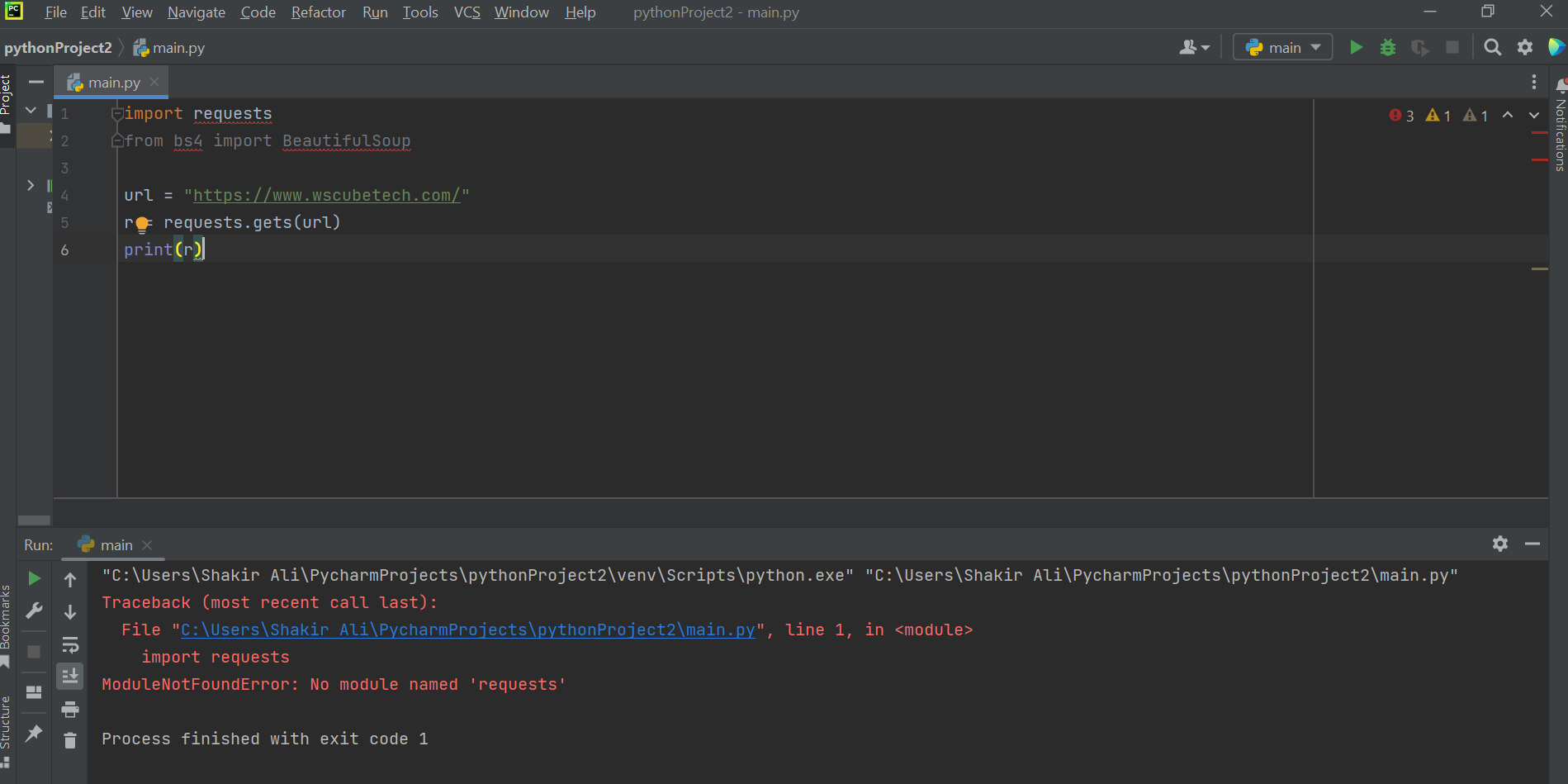
Все операции в swagger можно аналогично выполнять в Postman.



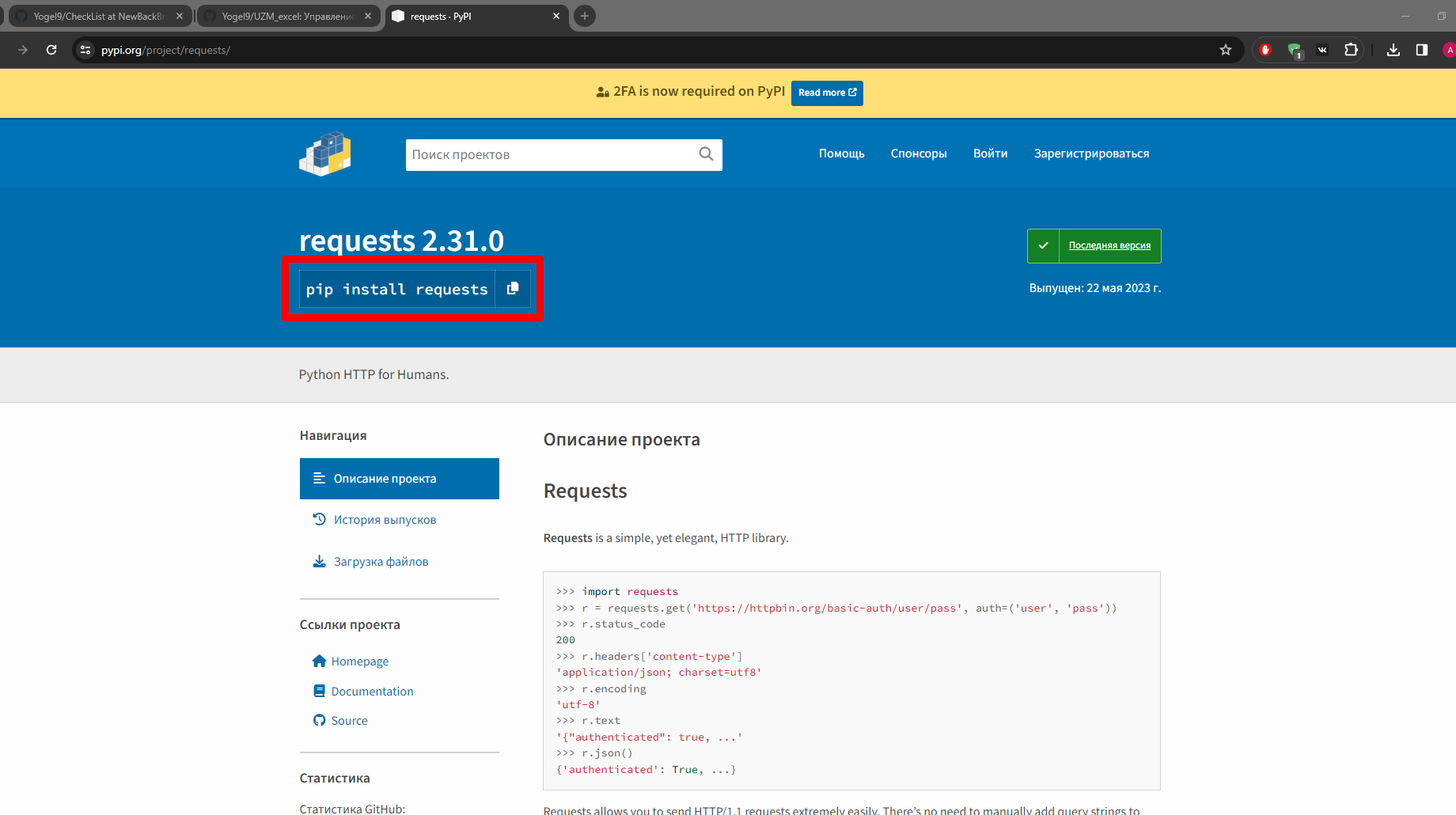
# **Возможные ошибки**

**Ошибка с отсутствием библиотек**

При запуске приложения ошибка ModuleNotFound решается установкой библиотеки при помощи команды pip install <module-name>

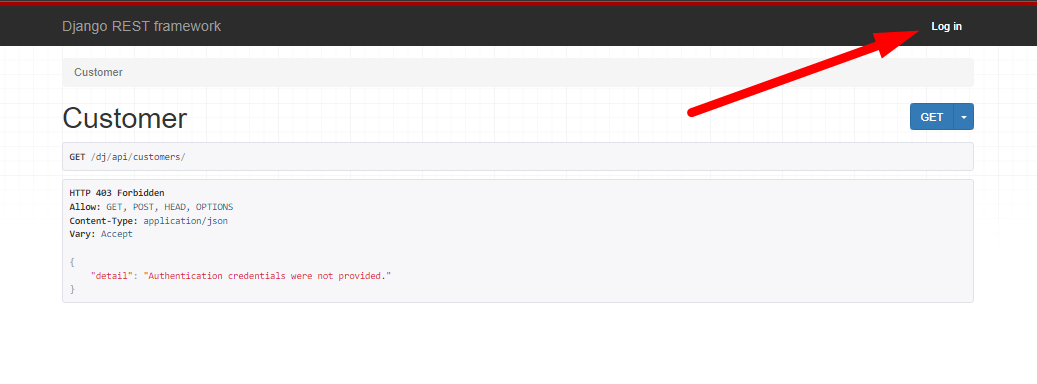


На официальном сайте pypi.org указаны все команды для установки любой библиотеки. Для перехода на pypi.org в поисковике достаточно указать python <module-name>.



**Ошибка с доступом к API**

Если пользователь не авторизовался на сайте он может это сделать через кнопку login или перейдя на <http://127.0.0.1:5431/dj/api-auth/login/>.



**Создание пользователя**

Чтобы добавить нового пользователя при локальном тестировании выполните команду python manage.py createsuperuser через cmd в \CheckList\Back\check\_list.

Пример создания нового пользователя (полe email необязательно):

