**Инструкция по beget для создания сайта и установки django**

1. ***Первое в разделе сайты создаём новый сайт, после создаём новое доменное имя*** ***привязываем его к созданному сайту***

Дальше на beget включаем доступ по ssh и подключаемся к нашему серверу через терминал с помощью ssh. Логин и пароль от хоста те, которые нам отправляли по смс и на почту

Подключитесь по SSH к серверу:

local@host:~$ ssh (адрес сервера)

Затем подключитесь к Docker-контейнеру:

django17@quasar:~ [0] $ ssh localhost -p222

Теперь перейдите во временный каталог на аккаунте:

(docker) django17@quasar:~ [0] $ cd ~/.beget/tmp/

Обратите внимание!

Если каталог не создан, ответ будет следующим:

-bash: cd: /home/i/idkfa/.beget/tmp: No such file or directory

Создать временный каталог и перейти в него можно следующей командой:  
mkdir -p ~/.beget/tmp && cd ~/.beget/tmp/

1. ***Если вам необходим Python версии 3.10 и выше, то необходимо дополнительно установить*** ***более свежую версию openssl.***

***Если вам необходим python версии ниже чем 3.10, то пропустите этот раздел и переходите к*** ***сборке непосредственно Python.***

Скачайте архив openssl версии 1.1.1:

(docker) django17@quasar:~/.beget/tmp [0] $ wget https://www.openssl.org/source/openssl-1.1.1l.tar.gz

Распакуйте архив и перейдите в папку с файлами:

(docker) django17@quasar:~/.beget/tmp [0] $ tar -xvzf openssl-1.1.1l.tar.gz && cd openssl-1.1.1l

С помощью утилиты configure настройте необходимые зависимости, префиксы и переменные, после чего будет сгенерирован Makefile:

(docker) django17@quasar:~/.beget/tmp/openssl-1.1.1l [0] $ ./config --prefix=$HOME/.local --openssldir=$HOME/.local/ssl '-Wl,--enable-new-dtags,-rpath,$(LIBRPATH)'

Запустите процесс компиляции и установки:

(docker) django17@quasar:~/.beget/tmp/openssl-1.1.1l [0] $ make -j$((`nproc`/4)) && make install

Проверить что openssl установлен корректно можно командой:

(docker) django17@quasar:~/.beget/tmp/openssl-1.1.1l [0] $ ~/.local/bin/openssl version

Если в ответ вы получили следующий вывод, то openssl установлен корректно:

OpenSSL 1.1.1l 24 Aug 2021

1. ***Перейдем к сборке самого Python.***

Обратите внимание!

В качестве примера установки используется версия Python 3.11.0.  
Если вы устанавливаете другую версию, то цифры в названиях папок и архивов будут отличаться.

Перейдите во временный каталог:

(docker) django17@quasar:~ [0] $ cd ~/.beget/tmp

Скачайте архив необходимой версии Python:

(docker) django17@quasar:~/.beget/tmp [0] $ wget

https://www.python.org/ftp/python/3.11.0/Python-3.11.0.tgz

Распакуйте архив и сразу перейдите в каталог с исходным кодом:

(docker) django17@quasar:~/.beget/tmp [0] $ tar -xvzf Python-3.11.0.tgz && cd Python-3.11.0

С помощью утилиты configure настройте необходимые зависимости, префиксы, переменные, после чего будет сгенерирован Makefile. Обратите внимание, что для Python меньше 3.10 команда будет отличаться.

Для Python 3.10 и выше:

(docker) django17@quasar:~/.beget/tmp/Python-3.11.0 [0] $ ./configure --prefix=$HOME/.local --with-openssl=$HOME/.local --with-openssl-rpath=auto --enable-optimizations --enable-loadable-sqlite-extensions LDFLAGS="-Wl,-rpath /usr/local/lib"

Для Python 3.9.x и ниже:

(docker) django17@quasar:~/.beget/tmp/Python-3.9.0 [0] $ ./configure --prefix=$HOME/.local --enable-optimizations --enable-loadable-sqlite-extensions LDFLAGS="-Wl,-rpath /usr/local/lib"

Теперь запустите процесс компиляции и установки:

(docker) django17@quasar:~/.beget/tmp/Python-3.11.0 [0] $ make -j$((`nproc`/4)) && make install

Проверить корректность установки можно следующей командой:

Input: (docker) django17@quasar:~ [0] $ python3 -V

Output: Python 3.11.0

Помимо Python, также устанавливается пакетный менеджер pip:

Input: (docker) django17@quasar:~ [0] $ pip3 -V

Output: pip 22.3 from /home/d/django17/.local/lib/python3.11/site-packages/pip (python 3.11)

ПОМЕТКА!

ПРИ ИСПОЛЬЗОВАНИИ КОМАНД ВАМ НУЖНО ПИСАТЬ ПЕРВЫЕ ДВЕ ЦИФРЫ В ВЕРСИИ ПИТОНА ИЛИ PIP, ВОТ ПРИМЕРЫ: python3.11 -V или pip3.11 -V

1. ***Настройка окружения virtualenv***

Если сайтов, использующих Python, на аккаунте несколько, то для каждого может потребоваться свое окружение и свой набор пакетов разных версий, которые могут конфликтовать между собой. Эту проблему можно решить настройкой виртуального окружения с помощью virtualenv.

Для начала перейдите в каталог с сервером (вместо «u» и «user» укажите первую букву логина и сам логин):

cd ~/home/u/user/директория\_сайта/public\_html

Если у вас собранный локально Python, для установки virtualenv укажите полный путь до pip:

(docker) django17@quasar:~/ директория\_сайта [0] $ ~/.local/bin/pip3.11 install virtualenv

Дальше следует создать директорию окружения.

Обратите внимание!

Лучше присвоить название директории с отсылкой к проекту, для которого она будет использоваться. Это нужно для того, чтобы в дальнейшем не путаться.

Команды будут отличаться в зависимости от версии Python.

Чтобы создать директорию окружения с названием venv, используя Python 2.7, выполните команду:

(docker) django17@quasar:~/ директория\_сайта [0] $ python -m virtualenv venv

Для создания директории окружения с названием venv, используя Python 3.6, выполните команду:

(docker) django17@quasar:~/ директория\_сайта [0] $ python -m virtualenv venv --python=/usr/bin/python3.6

Для локально собранного Python команда выглядит так:

(docker) django17@quasar:~/ директория\_сайта [0] $ ~/.local/bin/python3.11 -m virtualenv venv

Для входа в окружение выполните следующую команду:

(docker) django17@quasar:~/ директория\_сайта [0] $ source venv/bin/activate

Проверяем, что вход выполнен успешно:

Input: (docker) django17@quasar:~/ директория\_сайта [0] $ which python3.11

Output: /home/d/django17/django17.beget.tech/venv/bin/python

Теперь мы можем устанавливать все необходимые пакеты для данного окружения с помощью pip (при этом полный путь до него указывать уже не нужно):

(docker) django17@quasar:~/ директория\_сайта 0] $ pip3.11 install django

Чтобы выйти из окружения, выполните команду:

(docker) django17@quasar:~/ директория\_сайта [0] $ deactivate

ПОМЕТКА! ПОКА ЧТО ЛУЧШЕ НЕ ВЫХОДИТЬ ИЗ ВИРТУАЛЬНОГО ОКРУЖЕНИЕ ОНО НАМ ЕЩЁ ПРИГОДИТСЯ!

1. ***Установка и настройка Django***

(docker) django17@quasar:~/ директория\_сайта [0] $ pip3.11 install django

Затем необходимо создать новый проект:

(docker) django17@quasar:~/ директория\_сайта [0] $ django-admin.py startproject НАЗВАНИЕ ПРОЕКТА

Далее переходи в директорию проекта Django

~/ директория\_сайта /public\_html/ НАЗВАНИЕ ПРОЕКТА / НАЗВАНИЕ ПРОЕКТА

И создать файл passenger\_wsgi.py со следующим содержимым:

import os, sys

from django.core.wsgi import get\_wsgi\_application

site\_user\_root\_dir = '<полный\_путь\_до\_каталога\_с\_проектом>'

sys.path.insert(0, os.path.join(site\_user\_root\_dir, ‘НАЗВАНИЕ ПРОЕКТА’))

sys.path.insert(1,os.path.join(site\_user\_root\_dir,’<полный\_путь\_до\_Django>'))

os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'НАЗВАНИЕ ПРОЕКТА.settings')

application = get\_wsgi\_application()

Посмотреть версию и путь до Django можно так:

(docker) django17@quasar:~/django17.beget.tech [0] $ python3.6 -c "import django; print(django.get\_version()); print(django.\_\_file\_\_)"

3.0.5 ; /home/o/django17/.local/lib/python3.6/site-packages/django/\_\_init\_\_.py

Для последних версий Django также требуется явно задать домен, на котором работает сайт. Для этого нужно отредактировать список ALLOWED\_HOSTS в файле <название\_проекта>/<название\_проекта>/settings.py. Вместо указания определенного домена можно использовать символ подстановки \*, тогда проект будет работать на любом домене.

В нашем примере он будет выглядеть так:

ALLOWED\_HOSTS = ['django17.beget.tech']

Для корректной отдачи статического контента средствами Nginx необходимо создать символьную ссылку public, указывающую на public\_html:

(docker) django17@quasar:~/ директория\_сайта [0] $ ln -s public\_html public

Затем нужно создать файл htaccess в public\_html и указать в нем путь до Python:

PassengerEnabled On

PassengerPython /home/a/anesrnli/site.anesrnli.beget.tech/public\_html/venv/bin/python3.11

PassengerAppRoot /home/a/anesrnli/site.anesrnli.beget.tech/public\_html/ НАЗВАНИЕ ПРОЕКТА / НАЗВАНИЕ ПРОЕКТА/

Обратите внимание!

Путь может отличаться в зависимости от используемой версии Python.

После завершения настройки окружения необходимо создать каталог tmp, где будет храниться файл restart.txt:

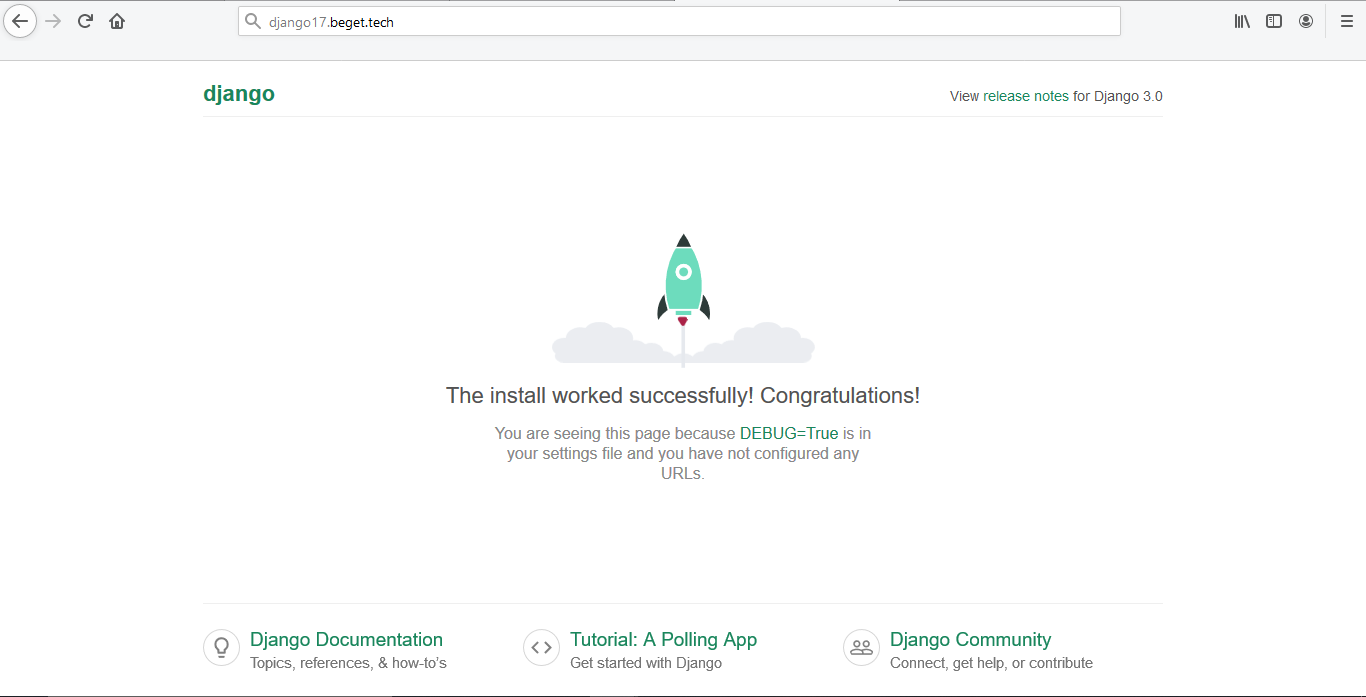
(docker) django17@quasar:~/ директория\_сайта [0] $ mkdir tmp; touch tmp/restart.txt

После каждого изменения файлов сервера нужно будет “нажимать” на пустой файл restart.txt для того, чтобы изменения вступили в силу. Это нужно делать в директории где вы создали этот файл, главное его надо создать в той же директории где находиться сам сайт. Вот команда для рестарта:

touch tmp/restart.txt

В целях безопасности на сервере установлена система разграничения прав доступа между сайтами, основанная на POSIX ACL. Сделано это для того, чтобы файлы одного сайта не могли обращаться к файлам других сайтов. Приложения, которые будут устанавливаться в каталоги .local, по умолчанию недоступны при попытке их запуска через сайт. Необходимо настроить общий доступ к этим каталогам через [Файловый менеджер](https://cp.beget.com/fm) (подробная [справка](https://beget.com/ru/kb/manual/fajlovyj-menedzher) по разделу).

Для проверки работоспособности нужно открыть сайт в браузере:



1. ***После того как вы проверили, что Django запустился можно приступать к загрузке самого*** ***сайта. Для этого вам необходимо зайти на beget в файловый менеджер и загрузить туда ваш*** ***локальный сайт на Django.***

Копируем на хостинг папки с вашими приложениями из вашего локального проекта, так же копируем файл urls.py для навигации в главную директорию проекта

В файл settings.py из вашего локального проекта добавляем в конец файла добавляем STATIC\_ROOT = /home/u/user/директория\_сайта/public\_html/staticl, а так же в строке ALLOWED\_HOSTS написать адрес сервера

ALLOWED\_HOSTS = [‘адрес сервера’]

Для начала перейдите в каталог с сервером(вместо “u” и “user” укажите первую букву логина и сам логин):

переходим cd ~/директория\_сайта /public\_html/ и создаем папку mkdir static

После переходим в эту только, что созданую папку cd ./staticl и выполняем команду

python3.11 ./home/a/anesrnli/ директория\_сайта /public\_html/ НАЗВАНИЕ ПРОЕКТА / НАЗВАНИЕ ПРОЕКТА /manage.py collectstatic

Переходим cd ~/ директория\_сайта /public\_html/ НАЗВАНИЕ ПРОЕКТА и выполняем миграцию проекта python3.11 manage.py migrate

***Проверяем, что наш сайт Django на хостинге работает так же, как и локальный сайт!***