Фреймворк Django

Что такое Django?

Django - это высокоуровневый фреймворк для веб-приложений на языке Python. Он был создан в 2005 году и с тех пор активно развивается и обновляется сообществом разработчиков по всему миру.

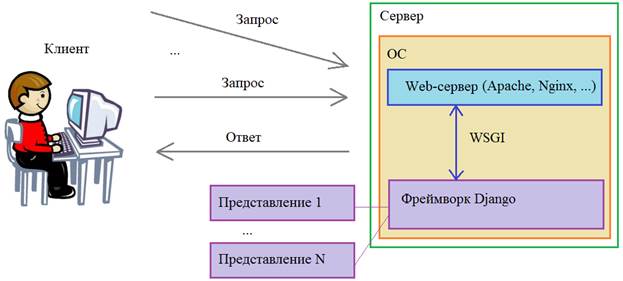
Но первый вопрос, почему именно Django? Наверное, этих фреймворков очень много и это действительно так. Однако именно Django завоевал огромную популярность благодаря простоте разработки даже очень сложных сайтов, например, таких как youtube, google search, dropbox и instagram. Все они сделаны с использованием фреймворка Django. И давайте теперь разберемся, для чего он вообще нужен. Если очень кратко, то для обеспечения работы сайтов на стороне сервера. То есть, это инструмент для бэкенд разработчиков. А если немного подробнее, то процесс взаимодействия пользователя с сайтом можно представить следующим образом:

Рисунок \_. Взаимодействие пользователя с сайтом.

Сначала от клиента поступает запрос, например, он вводит в браузере адрес https://youtube.com. Информационные пакеты начинают двигаться в сторону сервера с сайтом Youtube. Вопрос о том, как они это делают, мы оставим в стороне, это называется маршрутизацией в сети Интернет. Главное, что они доходят до сервера (то есть, компьютера), где расположен Youtube. На этом компьютере установлено специальное программное обеспечение – Web-сервер. Очень часто используют уже готовые разработки с открытым кодом: Apache и Nginx. Реже какие-либо свои программные продукты. Так вот, этот Web-сервер постоянно «слушает» каналы связи и в момент поступления запроса от пользователя должен перенаправить его на обработку соответствующему сайту (так как на одном сервере может находиться множество сайтов). Сайты способны по-разному обрабатывать входящие запросы, например, используя PHP и CGI скрипты, или же могут использоваться фреймворки, значительно упрощающие обработку запросов от клиентов сайта. В частности, Django – это и есть такой фреймворк, значительно упрощающий написание скриптов на языке Python. А взаимодействие с сервером происходит по интерфейсу WSGI, который передает обработку запроса в Django. При этом отрабатывает определенное представление, отвечающее за текущий запрос, и результатом обработки, как правило, является HTML-страница, которая сначала передается Web-серверу также по WSGI-интерфейсу, а он уже передает страницу конечному пользователю. Страница отображается в браузере клиента, и он видит заветный сайт Youtube. Вот так в очень упрощенном виде происходит взаимодействие между пользователем, сервером и фреймворком. WSGI (Web Server Gateway Interface) — стандарт взаимодействия между Python-программой, выполняющейся на стороне сервера, и самим веб-сервером.

Далее, мы установим интегрированную среду программирования PyCharm, разработанную специально для написания программ на Python. Она довольно удобна для разработки программ с использованием Django, поэтому все дальнейшие действия мы будем производить в PyCharm.

\_ Установка и настройка Django

Так как разработка велась на операционной системе MacOS, все команды, используемые в терминале, будут записанные в этом проекте для данное ОС.

Python помогают решить проблему управления зависимостями проектов. Если у вас есть несколько проектов на Python, то вероятность того, что вам придется работать с разными версиями библиотек или самого Python, очень высока. Но использование виртуальных сред позволяет создавать независимые группы библиотек для каждого проекта, что предотвращает конфликты между версиями и не дает одному проекту повлиять на другой. В Python уже есть модуль venv для создания виртуальных сред, который можно использовать как в разработке, так и в производстве.

Создаём виртуальное окружение в Linux или MacOS:

mkdir project

cd project

python3 -m venv venv

Далее активируем виртуальное окружение, чтобы все дальнейшие действия выполнялись внутри него: venv/bin/activate

Для установки Django рекомендуется использовать менеджер пакетов pip. Для этого необходимо открыть терминал и выполнить команду: pip3 install Django.

Помимо самого фреймворка будет установлено несколько обязательных зависимостей. В прочем сам Django содержит в себе около 15 пакетов, позволяющих решать большинство задач разработки без установки дополнительных пакетов и модулей.

\_Создание проекта

Первым шагом создания проекта мы определи папку, в которой будем вести разработку данного проекта. После выполним все действия, связанные с установкой и активации виртуального окружения и установкой самого Django.

Используя ядро Django, мы можем создавать множество разных сайтов под управлением Django в рамках текущего виртуального окружения. Чтобы посмотреть список команд ядра, достаточно в терминале записать: django-admin.

Для создания нового проекта в Django необходимо выполнить команду djangoadmin startproject <project\_name>. Здесь «project\_name», обычно, является доменным именем. Например, если мы собираемся располагать сайт на домене cinemasite, то пропишем: django-admin startproject cinemasite. Эта команда создаст структуру проекта, которая будет содержать все необходимые файлы и папки для работы с фреймворком.