

Руководство по установке

Для использования Django необходимо его установить. Этот раздел содержит описание быстрой установкой, которая подойдет для ознакомления с Django.

Установка Python

Будучи веб-фреймворком Python, Django требует Python.

Какие версии Python можно использовать с Django?

Версия Django	Версия Python
1.8	2.7, 3.2, 3.3, 3.4, 3.5
1.9, 1.10	2.7, 3.4, 3.5

Для каждой версии Python, только последний микро-релиз (А.В.С) поддерживается официально. Последние релизы вы можете найти на [странице загрузок Python](#).

Обычно мы поддерживаем версию Python, срок поддержки которой заканчивается до срока поддержки Django LTS релиза. Например, поддержка Python 3.3 заканчивается в сентябре 2017, а поддержка Django 1.8 LTS заканчивается в апреле 2018. Следовательно, Django 1.8 последняя версия, которая поддерживает Python 3.3.

Если вы начинаете новый проект и зависимые библиотеки поддерживают Python 3, вам следует использовать Python 3. Если нет, можно помочь в их переносе на Python 3, или использовать Python 2.

Так как последние версии Python обычно быстрее, содержат больше возможностей и лучше поддерживаются, мы советуем использовать последние релизы 2.x.y или 3.x.y.

Вы ничего не потеряете, используя Django со старой версией, но вы не получите улучшений и оптимизаций новых версий Python. Сторонние приложения Django могут иметь свою политику поддержки версий Python.

Использовать ли стабильную версию?

Если ваш код уйдёт в продакшн, то используйте стабильную версию. Разработчики Django выпускают стабильный релиз примерно каждые девять месяцев, а уж мелкие исправления выходят постоянно. Стабильные релизы содержат API, которое обратно совместимо с предыдущими версиями, так что при обновлении на следующий стабильный релиз никаких проблем возникнуть не должно.

Проверить установлен ли Python можно выполнив команду `python` в терминале, вы должны увидеть приблизительно следующее:

```
Python 3.4.x
[GCC 4.x] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Установка базы данных

Этот шаг необходим, если вы хотите использовать “большую” базу данных, такую как PostgreSQL, MySQL, или Oracle.

Создание рабочей базы данных

Если вы планируете использовать функционал API баз данных Django, вам нужно удостовериться, что сервер баз данных запущен. Django поддерживает много различных СУБД. В частности, к ним относятся PostgreSQL, MySQL, Oracle и SQLite.

Если вы заняты разработкой простого проекта или чего-то, что вы не планируете развернуть в производственной среде, SQLite в целом является наипростейшим вариантом, поскольку он вообще не требует установки отдельного веб-сервера. Однако, SQLite очень отличается от других баз данных, поэтому, если вы работаете над чем-то существенным, рекомендуется выбрать ту же базу данных, которая будет использована на “боевом” сервере.

В дополнение к официально поддерживаемым базам данных есть бэкэнды, поставляемые третьими сторонами, которые позволяют использовать другие БД с Django.

В дополнение к установке необходимой БД вы также должны убедиться, что вами выбран и установлен соответствующий модуль Python.

- При использовании PostgreSQL вам понадобится пакет `psycopg2`;
- При использовании MySQL вам понадобится DB API драйвер;
- Если вы используете SQLite, вам следует прочитать про особенности использования SQLite;
- При использовании Oracle вам понадобится копия `cx_Oracle`.

Если вы планируете использовать команду `manage.py migrate` для автоматической генерации таблиц БД для ваших моделей (после первого запуска Django и создания проекта), вам нужно убедиться, что Django имеет разрешение на создание и изменение таблиц БД; если же вы хотите создавать таблицы вручную, вам надо просто предоставить Django разрешения `SELECT`, `INSERT`, `UPDATE` и `DELETE`. После создания пользователя БД с указанными привилегиями, вам понадобится изменить настройки вашего проекта.

Удалите старые версии Django

Если вы обновляете версию Django, вам следует удалить старые версии Django перед установкой новой.

Удаление предыдущей версии Django

Вам нужно полностью удалить предыдущую версию Django перед установкой новой версии.

В случае, если Django предварительно установлен с помощью `pip` или `easy_install`, вам не понадобится удалять предыдущую версию вручную: просто запустите `pip` или `easy_install` снова, это приведёт к обновлению текущей версии на более новую.

В случае, если вы устанавливали Django, используя `python setup.py install`, его удаление будет заключаться в удалении самой директории `django` из `site-packages`. Чтобы найти нужную директорию вы должны запустить следующую команду в командной оболочке (не в интерактивной оболочке Python):

```
$ python -c "import django; print(django.__path__)"
```

Установка Django

У вас есть варианта установки Django:

- Установить версию Django *предоставленную дистрибутивом вашей операционной системой*. Это самый простой способ для операционных систем, дистрибутив которых содержит Django;

Распространение Django

- ❖ Существует много сторонних дистрибьюторов, которые предоставляют Django через свои системы управления пакетами. Это упрощает процесс установки и обновления Django т.к. эти системы устанавливают все необходимые зависимости (такие как библиотеки для работы с базами данных);
- ❖ Обычно эти пакеты содержат последнюю стабильную версию Django. Если вам необходима разрабатываемая версия Django, следуйте *инструкции*, которая описывает как установить версию из нашего Git репозитория;

Инструкция: Установка разрабатываемой версии

Если вы хотите обновить Django до разрабатываемой версии, чтобы иметь возможность использовать последние, возможно, нестабильные изменения и улучшения, следуйте этой инструкции:

1. Убедитесь, что у вас установлен **Git** и вы можете запустить его из командной оболочки. (Введите `git help` в командной оболочке, чтобы проверить это.)
2. Склонируйте разрабатываемую ветку Django вот так:

```
$ git clone git://github.com/django/django.git
```

Эта команда создаст каталог `django` в вашей текущей директории;

3. Убедитесь, что интерпретатор Python может загрузить код Django. Самый удобный способ сделать это – использовать **virtualenv**, **virtualenvwrapper** и **pip**;
4. После того, как вы создали и активировали виртуальное окружение, выполните в консоли:

```
pip install -e django/
```

Это сделает код Django импортируемым, а также сделает доступным запуск `django-admin.py`. Иначе говоря, всё готово!

Когда вам понадобится обновить вашу копию Django, просто запустите команду `git pull` в директории `django`. После того, как вы сделаете это, Git автоматически загрузит изменения.

- ❖ Если вы используете Linux или Unix систему, например, OpenSolaris, проверьте предоставляет ли ваш дистрибутив уже упакованную версию Django. Если вы используете дистрибутив Linux, и не знаете где искать такую информацию, обратитесь к **википедии Django**.
- Установить *официальный релиз*. Это самый лучший вариант для большинства пользователей;

Установка официальной версии с помощью pip

Это рекомендуемый способ установки Django.

1. Установите [pip](#). Самым простым способом сделать это является использование [автономного установщика pip](#). Если ваш дистрибутив уже включает установленную версию pip, возможно, вам потребуется обновить её до более свежей. (Если версия устарела, вы сразу поймёте это, поскольку установка не будет работать;
2. Взгляните на [virtualenv](#) и [virtualenvwrapper](#). Эти программы предоставляют изолированное окружение Python, что намного более практично, чем установка общесистемных пакетов. Такой подход также позволяет провести установку без административных прав;
3. После того, как вы создали и активировали виртуальное окружение, выполните в консоли `pip install Django`.

Проверка

Чтобы проверить что Django доступен для Python, выполните `python` в терминале. Теперь в консоли Python выполните импорт Django:

```
>>> import django
>>> print(django.get_version())
1.9
```

У вас может быть установлена другая версия Django.

Вот и все!

Вот и все, вы можете переходить к главе: *«Создаём своё первое приложение с Django, часть I»*.