Разработка Web приложений с Python и Flask

Дмитрий Купцов
2021

Оглавление

Введение

Веб приложения сегодня являются одним из самых распространённых способов создания сервисов, которые используются миллионами пользователей. Веб приложения легко обновлять, переносить, и распространять - они не требуют установки специальных средств на рабочей машине пользователей, кроме веб браузера.

В этой книге мы рассмотрим как создавать большие и структурированные веб приложения используя Python и библиотеку Flask. Мы рассмотрим такие вещи как безопасность веб приложений, базы данных, веб формы и структурирование больших веб приложений.

Данная книга предназначена для начинающих веб программистов, а также для всех тех, кто желает познакомиться с веб программированием. Мы предполагаем, что читатель уже ознакомился с нашей книгой об основах Python и алгоритмах. Данную книгу можно скачать бесплатно здесь [1].

ГЛАВА 1

Компоненты

Современное веб приложение состоит из множества компонентов: системы управления базами данных, или СУБД (как NoSQL, так и SQL), веб форм, контроллеров и моделей, клиентской части приложения написанной на JavaScript, подсистемы безопасности и многочисленных пользовательских библиотек. В данной главе мы рассмотрим основные компоненты, которые мы будем использовать при построение нашего приложения во второй части нашей книги.

1.1 Среда окружения

Часто на рабочей машине может быть запущено несколько проектов одновременно. Каждый проект должен иметь свои зависимости и установленные библиотеки. Для того, чтобы не было путаницы в библиотеках на рабочей машине, для разработки, устанавливают виртуальное окружение и устанавливают нужные библиотеки.

На дистрибутиве Linux Ubuntu 20.04 (мы предполагаем наличие Python 3.8.5) среда может быть установлена следующим образом.

Для начала необходимо установить пакет, который позволит работать с виртуальным окружением:

```
sudo apt-get install python3-venv
```

Далее создадим окружение:

```
$ python3 -m venv book_ml
```

После нужно активировать виртуальную среду разработки:

```
$ source book_ml/bin/activate
```

Далее можно устанавливать необходимые библиотеки и они не будут пересекаться с другими проектами:

```
$ pip3 install pycryptodome
```

1.2 Безопасность

Безопасность в веб приложениях является одним из основных вопросов. Популярность веб приложений и их повсеместность накладывает определённые требования к безопасности. В данной главе мы познакомим читателя с такими атаками как Cross Site Request Forgery (CSRF), неаутентифицированный и неавторизованный доступ к ресурсам, а также рассмотрим методы защиты от таких атак. И конечно, мы затронем тему шифрования канала от пользователя до веб сервера с помощью Secure Socket Layer (SSL).

1.2.1 CSRF

Очень часто можно послать запрос на сайт замаскированный как сторонний и произвести какую либо транзакцию так, чтобы пользователь этого не заметил. Например, пусть пользователь авторизовался на вашем сайте, получил куки и не вышел из системы. Позже злоумышленник может прислать пользователю на почту картинку с изображением, например, кошечки и попросил перейти по ссылке. Ссылка же на самом деле ведёт к вашему сайту и автоматически пошлёт все куки файлы с запросом.

Это опасно тем, что если система не защищена от CSRF атак, то запрос выполнится и атакующий сможет изменить содержимое базы данных. Для того чтобы этот тип атаки не смог быть реализован, необходимо в форме HTML вставлять некий токен безопасности. Вместе с тем, тот же токен необходимо хранить в зашифрованном виде в куке файле. Если на сервер придет куки файл, и токен в нем будет отличаться от токена, полученного в форме, то запрос стоит отвергнуть, так как он не безопасный.

Приведём пример того, как можно использовать CSRF защиту в Flask приложении.

Для начала установим библиотеку для работы с формами и сам Flask:

```
$ pip3 install flask
$ pip3 install flask_wtf
```

Далее нужно сконфигурировать наше Flask приложение для работы с CSRF защитой. Для этого вставим следие строки в наше приложение:

```
from flask_wtf.csrf import CSRFProtect

app = Flask()

csrf = CSRFProtect(app)
```

В веб форме прописываем скрытое поле, которое будет содержать наш CSRF токен:

```
<form method="post">
    {{ form.csrf_token }}
</form>
```

Если вы используете jQuery AJAX запросы то можно с запросом посылать и токен:

```
<script type="text/javascript">
    var csrf_token = "{{ csrf_token() }}";

    $.ajaxSetup({
        beforeSend: function(xhr, settings) {
            if (!/^(GET|HEAD|OPTIONS|TRACE)$/i.test(settings.type) && !
                                                    this.crossDomain) {
                xhr.setRequestHeader("X-CSRFToken", csrf_token);
            }
        }
    });
</script>
```

Если проверка верности токена будет нейдачной, Flask выбросит ошибку CSRFError. По умолчанию, Flask вернет HTTP с кодом 400 и объяснением причины ошибки. Если вы хотите отправить своё сообщение об ошибке то стоит зарегистрировать соответствующий обработчик ошибки:

```
from flask_wtf.csrf import CSRFError

@app.errorhandler(CSRFError)
def handle_csrf_error(e):
    return render_template('csrf_error.html', reason=e.description),
                                            400
```

Стоит заметить CSRF защита требует секретный ключ для подписи токена. По умолчанию Flask будет использовать SECRET_KEY переменную для этих целей. Если же вы захотите использовать отдельный ключ для этих целей, то можно установить переменную WTF_CSRF_SECRET_KEY. Требуется использовать достаточно большой ключ для шифрования - ключ, который будет содержать достаточно энтропии и его будет невозможно угадать. Мы рекомендуем использовать ключ не менее 256 бит (в реальности сложность будет где-то 192 бита). Для генерации такого ключа можно выполнить следующую команду в консоли Ubuntu:

```
$ apg -x 32 -m 32 -a 1
```

1.2.2 Аутентификация и авторизация

1.2.3 Проверка форм регулярными выражениями

1.2.4 Шифрование данных с SSL

1.3 Модель данных

1.3.1 MySQL и SQLAlchemy

Объектно-ориентированный подход к запросам

Миграции

1.3.2 NoSQL и MongoDB

1.4 Представления

1.4.1 Веб формы

1.4.2 Jinja

1.4.3 Bootstrap

1.5 Контроллеры и бизнес логика

1.5.1 Blueprints: Структурируем большое приложение

1.6 Клиентский код: Vue.js

ГЛАВА 2

Разрабатываем приложение на примере машинного обучения

Литература

[1] D. Kuptsov. Python in examples. 2020. https://github.com/dmitriykuptsov/python_in_examples/blob/master/book.pdf (visited 2021-05-14).