

Flask



КИРИЛЛ ТАБЕЛЬСКИЙ



КИРИЛЛ ТАБЕЛЬСКИЙ

Lightmap

 malik24ster@gmail.com



План занятия

1. [Что такое Flask](#)
2. [Простейшее приложение](#)
3. [Работа с БД](#)

Посмотрим рейтинг фреймворков в сети.





Что такое Flask

Flask — это микрофреймворк, предназначенный для создания легковесного http-сервера.

«Микро» означает, что простое приложение может поместиться в один файл и далее его можно развивать, постепенно добавляя функционал

Flask vs Django

Плюсы Flask:

- гибкость;
- скорость;
- мало магии.

Плюсы Django:

- все включено;
- скорость разработки.

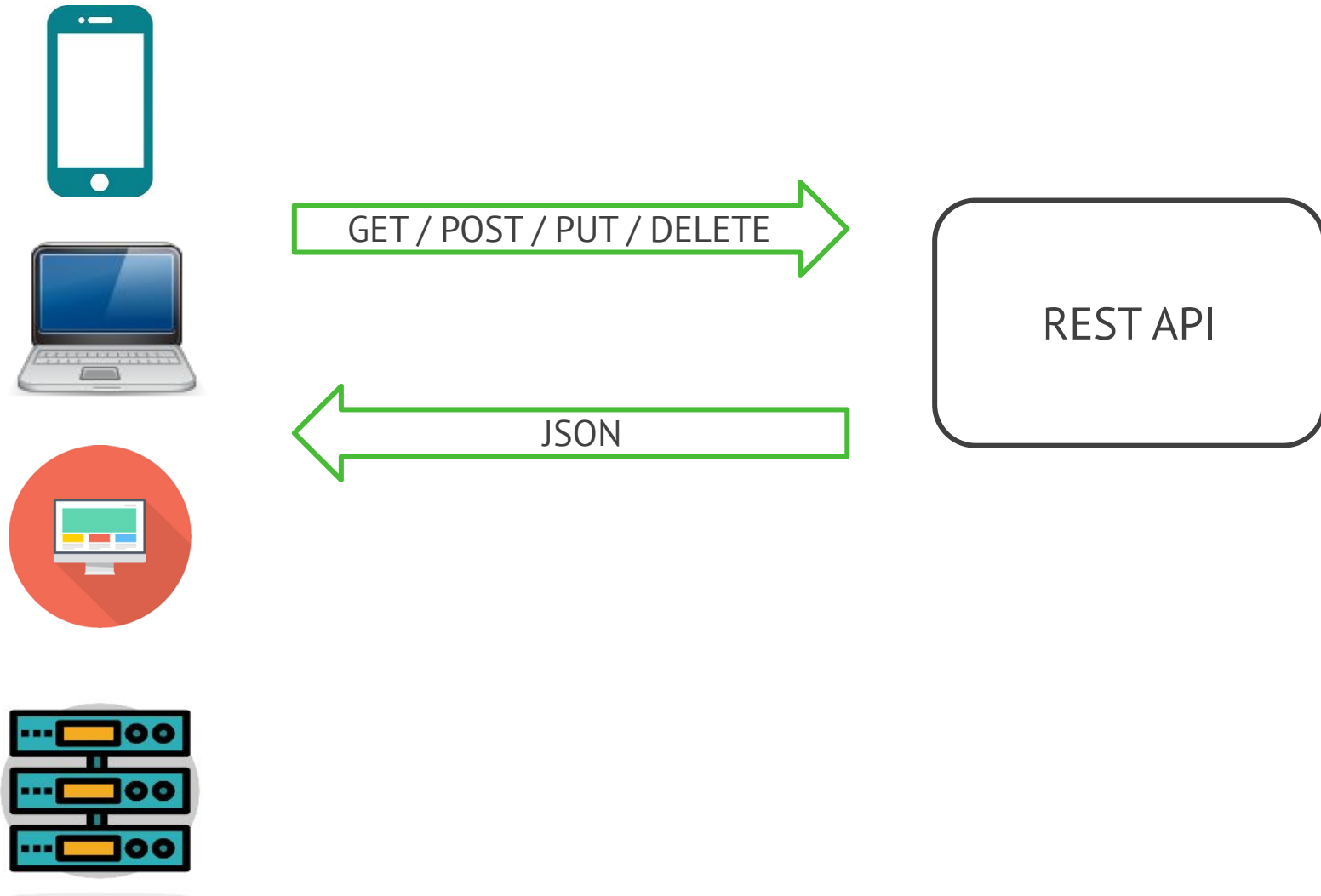


django

Сценарии использования Flask

- Микросервисы
- Небольшие API
- MOCK API

REST API





HTTP методы

Метод	Действие
GET	Получить данные
POST	Вставить данные в базу
PUT	Полностью изменить запись в базе
PATCH	Частично изменить запись в базе
DELETE	Удалить запись из базы



HTTP статусы

Статус	Описание
200	ОК
400	Неправильный формат запроса
401	Ошибка авторизации
404	Нет такой записи в базе
500	Внутренняя ошибка

Из чего состоит запрос

- uri (query string);
- body
- headers
- cookies

Простейшее приложение

```
from flask import Flask, jsonify

app = Flask(__name__)

@app.route('/api/v1/health_check', methods=['GET', ])
def health_check():

    return jsonify({'status': 'ok'})

app.run()
```

Парсинг запроса

```
from flask import request
@app.route('/api/v1/user/<int:user_id>', methods=['GET', ])
def health_check(user_id):
    app.logger.debug(
        json.dumps(
            {
                'user_id': user_id,
                'headers': dict(request.headers), # http заголовки
                'qs_params': request.args, # query string параметры
                'body_text': request.get_data().decode(), # тело запроса в байтах
                'body_json': request.get_json() # тело запроса json
            },
            indent=3
        )
    )
```



Валидация

Данные от клиента **всегда** нужно валидировать перед запросом.

В отличие от DRF во Flask нет встроенной валидации.

Мы будем использовать JSON Schema.

Валидация

```
POST = {
  'type': 'object',
  'properties':
    {
      'text': {
        'type': 'string'  # проверка типа
      },
      'visible_for': {
        'type': 'string',
        'enum': ['subsribers', 'friends', 'all']  # проверка вхождения
      },

      'priority': {
        'type': 'number',
        'minimum': 0,  # диапазон
        'maximum': 23  # диапазон
      },

      'expire_date': {
        'type': 'string',
        'pattern': '^\\d{4}-\\d{2}-\\d{2}$'  # соответствие регулярному выражению
      }
    }
}
```

Валидирующий декоратор

```
def validate(req_schema):
    """
    Валидатор входящих запросов
    :param req_schema – json схема с предыдущего слайда
    """

    def decorator(func):
        def wrapper(*args, **kwargs):
            try:
                jsonschema.validate(
                    request.get_json(), schema=req_schema,
                )
            except jsonschema.ValidationError as er:

                resp = flask.make_response(jsonify({'success': False,
                                                    'description': er.message}))

                resp.set_status = 401
                return resp
            result = func(*args, **kwargs)

            return result
        return wrapper
    return decorator
```


Валидация в действии

```
@app.route('/api/v1/post', methods=['POST', ])  
@validate(schema.POST)  
def make_post():  
  
    return jsonify({'status': 'ok'})
```



Работа с БД

Во Flask нет ORM.

Используем:

1. SQL Alchemy
2. MongoEngine
3. Голый язык запросов
4. **Не забываем хешировать пароли**

Middleware

```
@app.before_request
def middleware():
    # Действие перед каждым запросом, например, проверка авторизации
    pass
```

Дополнительные материалы

- <https://flask.palletsprojects.com>
- <https://json-schema.org/understanding-json-schema/>
- <https://blog.miguelgrinberg.com/post/restful-authentication-with-flask>
— примеры авторизации



Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задаём в чате Slack!
- Задачи можно сдавать по частям.
- Зачёт по домашней работе проставляется после того, как приняты **все задачи**.

Задавайте вопросы и напишите отзыв о лекции!

КИРИЛЛ ТАБЕЛЬСКИЙ



malik24ster@gmail.com