



Занятие №9



**IT
Education
Academy**

WWW.ITEA.UA



www.itea.ua

Поляков Антон

◆ Python backend developer

Фото
инструктора

КОНТАКТНЫЕ ДАННЫЕ

Telegram: @polyakov1



API

API — Application programming interface.

REST (RESTful) - это общие принципы организации взаимодействия приложения/сайта с сервером посредством протокола HTTP.

JSON

```
import json
```

```
# a Python object (dict):
```

```
x = {  
    "name": "John",  
    "age": 30,  
    "city": "New York"  
}
```

```
# convert into JSON:
```

```
y = json.dumps(x)  
print(y)
```



JSON

```
import json
```

```
# some JSON:
```

```
x = '{ "name":"John", "age":30, "city":"New York"}'
```

```
# parse x:
```

```
y = json.loads(x)
```

```
# the result is a Python dictionary:
```

```
print(y["age"])
```

REST

Всё взаимодействие с сервером сводится к 4 операциям (4 - это необходимый и достаточный минимум, в конкретной реализации типов операций может быть больше):

1. получение данных с сервера (обычно в формате JSON, или XML)
2. добавление новых данных на сервер
3. модификация существующих данных на сервере
4. удаление данных на сервере



REST

Для каждого типа операции используется свой метод HTTP-запроса:

1. получение - GET
2. добавление - POST
3. модификация - PUT
4. удаление - DELETE

Mongoengine types

BooleanField — True/False

DateTimeField — Date and time

EmailField

FloatField

IntegerField

ListField — list of other types

ReferenceField — analog of foreign key

StringField

Field arguments **required** — field must be filled. **Default** — if field not filled there gonna be default value. **Choices, unique.**

Mongoengine examples

```

From mongoengine import *
class Object(Document):

    created_by = ReferenceField(User, reverse_delete_rule=CASCADE)
    name = StringField(max_length=250, required=True)
    radius = DecimalField()
    longitude = DecimalField()
    latitude = DecimalField()
    users = ListField(ReferenceField(User))

class User(Document):
    first_name = StringField()
    last_name = StringField()
    email = StringField()
    tel = StringField()
    password = StringField()
    active = BooleanField(default=True)

```

Запросы

```
# This will return a QuerySet that will only iterate over users whose  
# 'country' field is set to 'uk'  
uk_users = User.objects(country='uk')
```

```
# This will return a QuerySet that will only iterate over pages that have  
# been written by a user whose 'country' field is set to 'uk'  
uk_pages = Page.objects(author__country='uk')
```

Операторы

ne – not equal to

lt – less than

lte – less than or equal to

gt – greater than

gte – greater than or equal to

not – negate a standard check, may be used before other operators (e.g.

Q(age__not__mod=(5, 0)))

in – value is in list (a list of values should be provided)

nin – value is not in list (a list of values should be provided)

mod – value % x == y, where x and y are two provided values

all – every item in list of values provided is in array

size – the size of the array is

exists – value for field exists

Запросы

`num_users = User.objects.count()` - кол-во объектов

`yearly_expense = Employee.objects.sum('salary')` — сумма значений

`mean_age = User.objects.average('age')` — среднее значение

`Obj.objects.filter(id__in=set(1, 2, 3))` — Оператор in

`Obj.object.filter().field.append(new_value).save()` - добавляем новое значение

Marshmallow

```
from marshmallow import Schema, fields
```

```
class UserSchema(Schema):
    name = fields.String()
    surname = fields.String()
    id = fields.IntegerField()
```

Return serialized user:

```
Return UserSchema().dump(object_to_serialize)[0]
```

Практическая задача

- 1) Создать базу данных студентов (ФИО, группа, оценки, куратор студента, факультет). Написать CRUD ко всем полям. Описать методы для вывода отличников по каждому факультету. Вывести всех студентов определенного куратора.
- 2) Создать модуль, который будет заполнять базу данных случайными валидными значениями (минимум 100 студентов).

Домашнее задание

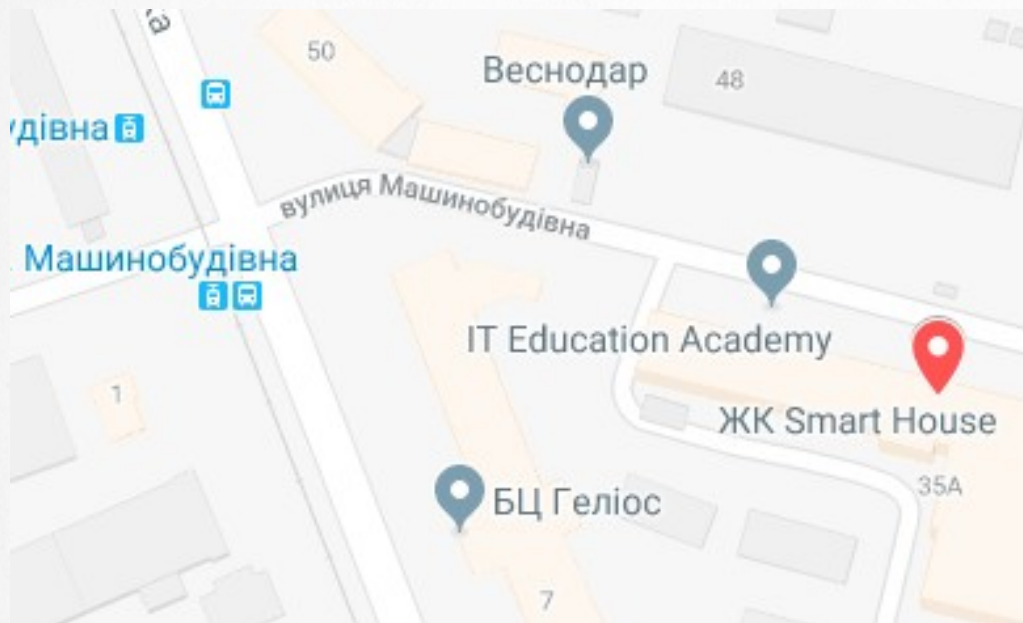
Основные методы http запросов:

GET — метод на чтение данных

POST — метод на запись данных

PUT/PATCH — метод на изменение данных

DELETE — метод на удаление данных



КОНТАКТНЫ Е ДАННЫЕ ITEA

ул. Машиностроительная, 41,
ЖК «Smart House», Киев

ул. Срибнокильская, 1, офис
269, Киев

пр. Академика Глушкова, 1,
корп.17, Киев

+38 (044) 599-01-79
facebook.com/itea
info@itea.ua
itea.ua