

НЕДЕЛЯ_5: REQUESTS JSON CRUD



СЕГОДНЯ В МЕНЮ

СОДЕРЖАНИЕ: INCOMPLETE

JSON Модуль

Requests Модуль

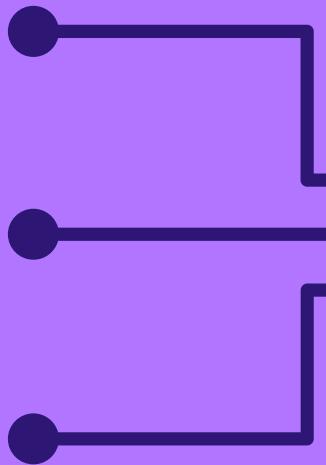
Получить Данные из Requests

HTTP Методы

CRUD – Create, Read, Update, Delete

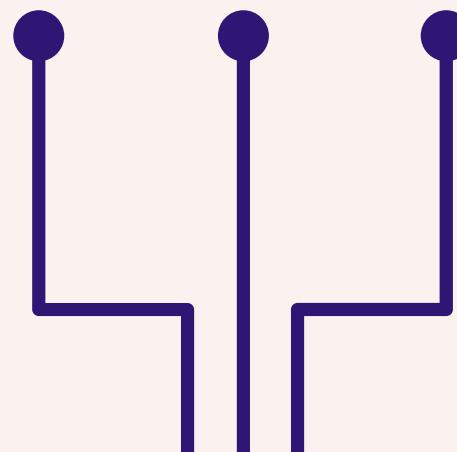
TimeOut

Environment



JSON – ЭТО ТИП ДАННЫХ

Json – является одним из главных форматов общения компьютеров в сети. Сходство json и dictionary 95%.



DUMP

ЗАПИСАТЬ В JSON

Если вы хотите что-то записать в json файл это нужно dump.

ПРИМЕР:

```
with open('file.json', 'w') as file:  
    json.dump(<что dump>, <куда dump>)
```

LOAD

ПРОЧИТАТЬ JSON

Если вы хотите выгрузить json файл в переменную это нужно load.

ПРИМЕР:

```
with open('file.json', 'r') as file:  
    data = json.load(<что load>)
```

1. Создайте 3 dictionary с ключами: `login`, `email`, `age`, `phone_number`. Добавьте их в лист и запишите его в JSON файл. После чего выгрузите все данные из JSON, добавьте к каждой dictionary ключ `language` а значение Python и снова запишите в JSON.

1. Возьмите JSON по <IP Address>/get_json_adobe, добавьте к каждому элементу `{"company": "Adobe"}` и сохраните у себя на рабочем столе.

GET

Запрос на получение данных

POST

Запрос на создание данных

PUT

Запрос на изменение данных

DELETE

Запрос на удаление данных

МОДУЛЬ REQUESTS

Requests даёт возможность делать HTTP запросы через Python Файл.

ПОЛУЧИТЬ КОНТЕНТ КОТОРЫЙ ПРИХОДИТ В
RESPONSE

```
data = requests.get('https://itc.kg')
print(data.content)
print(data.text)
print(data.json())
```



POST

КАК СДЕЛАТЬ POST ЗАПРОС В REQUESTS

```
data = requests.post(  
    url='https://postman-echo.com/post',  
    json={"name": "Azatot", "lang": "Python"}  
)
```

ПОЛУЧИТЬ КОНТЕНТ КОТОРЫЙ ПРИХОДИТ В
RESPONSE

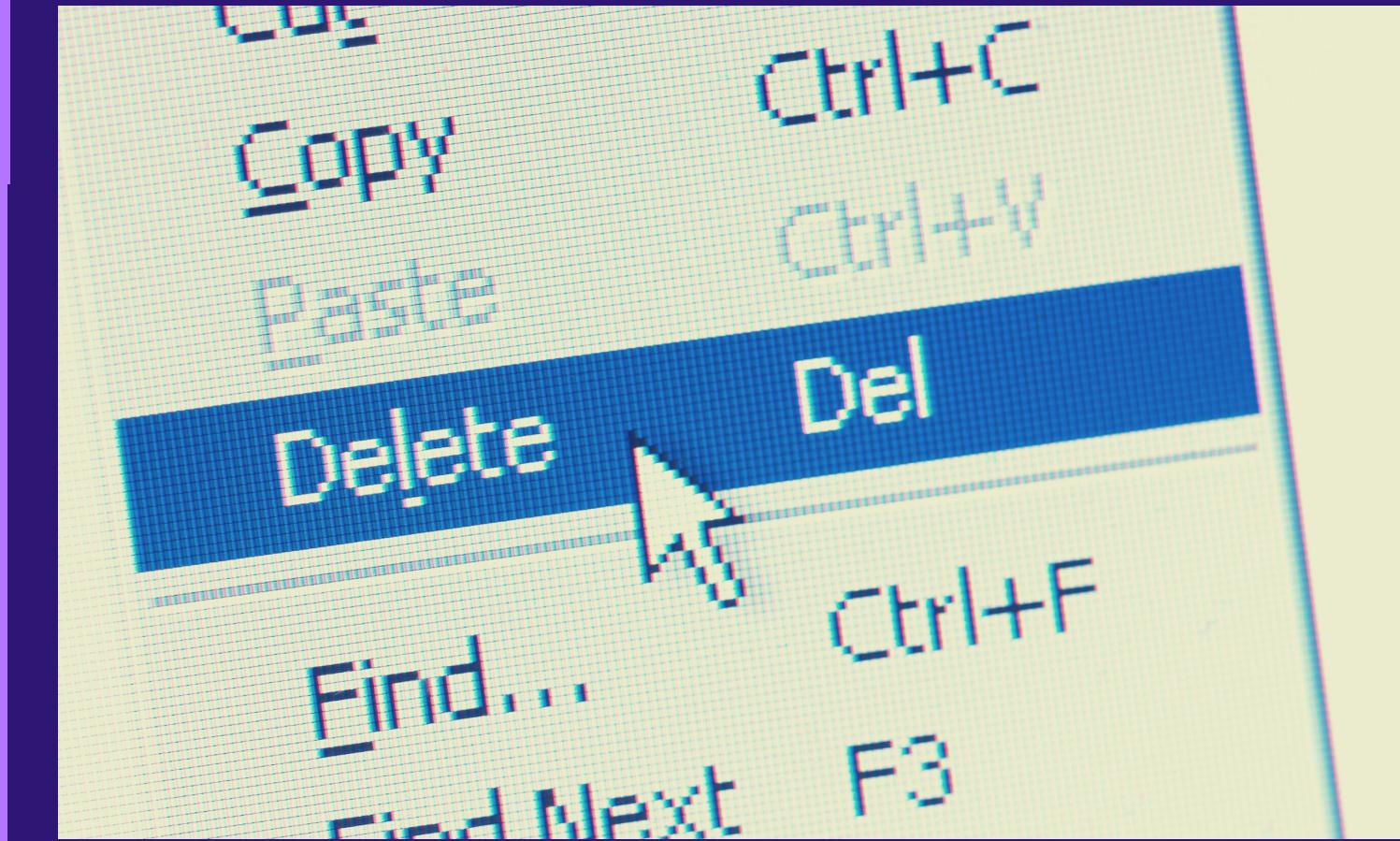
Чтобы получить чистый JSON из ответа от
сервера можете сразу написать:

```
print(data.json())
```



HTTP МЕТОД UPDATE(PUT, PATCH)

```
requests.put(  
    'https://postman-echo.com/put',  
    data={  
        "name": "Azatot",  
        "password": "qwerty",  
        "new_name": "Ktulhu"  
    })
```



HTTP МЕТОД DELETE

```
requests.delete(  
    'https://postman-echo.com/put',  
    data={  
        "name": "Azatot",  
        "password": "qwerty",  
        "old_name": "Ktulhu"  
    })
```

TIMEOUT – ВРЕМЯ ВЫШЛО . . .

TimeOut используется для отмены запроса если время ответа от сервера слишком долгое.



ПРИМЕР:

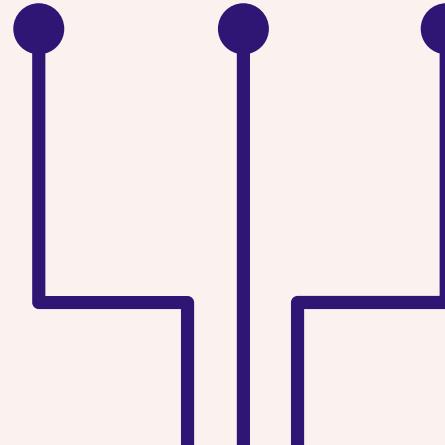
```
requests.get(<url>,timeout=40)  
requests.post(<url>,timeout=60)  
requests.put(<url>,timeout=50)  
requests.delete(<url>,timeout=20)
```

VIRTUALENV

```
sudo apt-get install virtualenv
```

```
virtualenv my_venv --python=python3.8
```

virtualenv – программа для создания
виртуального пространства защищённого
из вне.



1. С помощью `requests` получите данные по адресу `<Server IP>/get_json_users/` и вытащите `.json()` объект
2. Сделайте POST запрос по адресу `<Server IP>/send_back_data/` и отправьте туда json ОБЪЕКТ
`{"name": <Ваше Имя>, "model":<Модель Вашего Ноутбука>}.`
3. Через `psycopg2` подключитесь к удалённой БД, параметры подключениясмотрите по адресу `<Server IP>/itc_db/`
4. Сделайте POST запрос на `<Server IP>/upload_to_database`. В POST Вам нужно ОБЯЗАТЕЛЬНО передать
`{"name":<Имя Вашего Linux пользователя!>, "date_created":<Настоящее время в формате TIMESTAMP>}`
5. После того как Вы сделали успешный запрос на `<Server IP>/upload_to_database/` и получили в ответе то что отправляли зайдите через `psycopg2` в БД на сервере и посмотрите записи в Таблице `pythoniers`. Сделайте обычный `SELECT` из ТАБЛИЦЫ посмотрите в БД она там одна:)'
6. Сделайте GET запрос на `<Server IP>/all_users/` и посмотрите кто из пользователей сейчас есть на сервере.
7. Обновите `email` у своего пользователя сделав PUT запрос на `<Server IP>/update_me/`
Пример: {
 "`name`":<Ваше Имя в БД> ,
 "`date_created`":<Настоящее Время в формате TIMESTAMP> ,
 "`email`":<Ваш Email>
}
8. Удалите себя из БД по `email` с помощью DELETE запроса на `<Server IP>/delete_me/`:
Пример: {"`email`": <Ваш Email>}

1. Посмотрите кто есть в нашей группе http://<IP Server>/get_group/

2. Добавьте себя в группу через POST. Пример: {

```
"people_id": "",  
"experience_in_years": "",  
"skills": "",  
"created_at": ""
```

}

3. Зарегистрируйте себя через POST запрос по адресу: http://<IP Server>/register_member/

4. Обновите свою зарплату в полях salary_three_years и salary_five_years через

PUT: http://<IP Server>/update_member/

Пример для изменения: {

```
"people_id": <ID>,  
"salary_three_years": <Amount of Money>,  
"salary_five_years": <Amount of Money>
```

}