

Проект API. Работа с протоколом HTTP

План урока

- 1 Повторение
- 2 Задание CGI-параметров url
- 3 Поисковое приложение
- 4 API Поиска по организациям

Аннотация

В уроке продолжается рассказ про использование HTTP-API. Показан пример поискового приложения, формирующего запросы к HTTP-API на основании результатов других запросов. Рассматривается API поиска по организациям с примерами задач.

1. Повторение

На прошлом уроке мы начали разговор о разных видах API. Практическую часть мы посвятили использованию HTTP-API на примере Yandex.Maps.StaticAPI и Yandex.Maps.Geocoder. Давайте вспомним, как выглядит это взаимодействие с точки зрения python-приложения.

2. Задание CGI-параметров url

В наших примерах запрос к API был представлен константной строкой, поскольку этого достаточно для первого знакомства с возможностями API. В более сложных задачах запрос приходится формировать в ходе выполнения программы, основываясь на данных, введённых пользователем или полученных иным образом.

Для удобства формирования запросов в функции **requests.request()** есть именованный параметр **params**, представляющий собой словарь (dict). В нём содержится отображение из названий CGI-параметров в их значения. Давайте рассмотрим пример из предыдущей лекции:

```
map_request = \
    "http://static-maps.yandex.ru/1.x/?ll=37.530887,55.703118&spn=0.002,0.002&l=m"
response = requests.get(map_request)
```

Этот код можно переписать вот так:

```
import requests
api_server = "http://static-maps.yandex.ru/1.x/"

lon = "37.530887"
lat = "55.703118"
delta = "0.002"

params = {
    "ll": ",".join([lon, lat]),
    "spn": ",".join([delta, delta]),
    "l": "map"
}
response = requests.get(api_server, params=params)
```

Результаты выполнения двух вариантов кода будут идентичными. Понятно, что первый вариант использовать проще и быстрее, если нам нужно выполнить всего один константный запрос. Второй же вариант предпочтительнее, когда надо формировать запрос на лету. Например, если координаты нам ввёл пользователь или мы получили их, например, в ответе геокодера.

Про кодирование url-запросов (замены пробелов на «+», %-кодирование) можно прочитать [вот](#) [тут](#).

3. Поисковое приложение

Давайте напишем простое поисковое приложение. Пользователь печатает в командной строке запрос, а наша задача состоит в том, чтобы найти координаты запрошенного объекта и показать его на карте, выбрав соответствующий масштаб и позицию карты.

Часть кода, которая выполняет запросы к API, может выглядеть так:

```
import sys
import requests

# Пусть наше приложение предполагает запуск:
# python search.py Москва, ул. Ак. Королева, 12
# Тогда запрос к геокодеру формируется следующим образом:
toponym_to_find = " ".join(sys.argv[1:])

geocoder_api_server = "http://geocode-maps.yandex.ru/1.x/"

geocoder_params = {"geocode": toponym_to_find, "format": "json"}

response = requests.get(geocoder_api_server, params=geocoder_params)
if not response:
    # обработка ошибочной ситуации
    pass

# Преобразуем ответ в json-объект
json_response = response.json()
# Получаем первый топоним из ответа геокодера.
toponym = json_response["response"]["GeoObjectCollection"]["featureMember"][0]["GeoObject"]
# Координаты центра топонима:
toponym_coordinates = toponym["Point"]["pos"]
# Долгота и широта:
toponym_longitude, toponym_latitude = toponym_coordinates.split(" ")

delta = "0.005"

# Собираем параметры для запроса к StaticMapsAPI:
map_params = {
    "ll": " ".join([toponym_longitude, toponym_latitude]),
    "spn": " ".join([delta, delta]),
    "l": "map"
```

```
}

map_api_server = "http://static-maps.yandex.ru/1.x/"
# ... и выполняем запрос
response = requests.get(map_api_server, params=map_params)
```

4. API Поиска по организациям

Мы рассмотрели две части Yandex.Maps.API из трёх. Давайте познакомимся и с третьей, **Поиском по организациям**. Или Адресным справочником.

Страница API: <https://tech.yandex.ru/maps/geosearch/>

Прочтите описание/соглашение и найдите отличие от того, с чем нам приходилось работать раньше.

Для этого API нужен ключ. Для задач учебного курса ключ с необходимыми правами доступа получен заранее. Этот ключ позволяет выполнить лимитированное число запросов, поэтому пользоваться им разрешается только при решении задач этого курса. После курса ключ будет деактивирован, и запросы с ним перестанут работать. Если же для своих проектов вам понадобится доступ к API поиска по организациям, необходимо получить ключ самостоятельно. Для этого нужно заполнить форму, указав данные для связи, а также сайт, на котором предполагается использовать данное API. Мы говорили о том, что по условиям результаты надо обязательно использовать на сайте — обратите на это внимание. Исключение сделано только для наших учебных проектов.

Ключ на 2018-2019 года: **dda3ddba-c9ea-4ead-9010-f43fbc15c6e3**

Прочитайте о возможностях API поиска по организациям.

Формат диалога с Поиском по организациям похож на формат геокодера. Давайте найдём ближайшую аптеку к вашему дому. Будет похоже на написанную раньше программу поиска объекта по адресу.

Формат запроса описан на [странице](#).

```
search_api_server = "https://search-maps.yandex.ru/v1/"
api_key = "... "

address_ll = "37.588392,55.734036"

search_params = {
    "apikey": api_key,
    "text": "аптека",
```

```

        "lang": "ru_RU",
        "ll": address_ll,
        "type": "biz"
    }

    response = requests.get(search_api_server, params=search_params)
    if not response:
        #...
        pass

```

Формат ответа смотрите на странице:

https://tech.yandex.ru/maps/doc/geosearch/concepts/response_structure_business-docpage/

Продолжим разбор примера...

```

# Преобразуем ответ в json-объект
json_response = response.json()

# Получаем первую найденную организацию.
organization = json_response["features"][0]
# Название организации.
org_name = organization["properties"]["CompanyMetaData"]["name"]
# Адрес организации.
org_address = organization["properties"]["CompanyMetaData"]["address"]

# Получаем координаты ответа.
point = organization["geometry"]["coordinates"]
org_point = "{0},{1}".format(point[0], point[1])
delta = "0.005"

# Собираем параметры для запроса к StaticMapsAPI:
map_params = {
    # позиционируем карту центром на наш исходный адрес
    "ll": address_ll,
    "spn": ",".join([delta, delta]),
    "l": "map",
    # добавим точку, чтобы указать найденную аптеку
    "pt": "{0},pm2dgl".format(org_point)
}

map_api_server = "http://static-maps.yandex.ru/1.x/"
# ... и выполняем запрос
response = requests.get(map_api_server, params=map_params)

```

[Помощь](#)

© 2018 – 2019 ООО «Яндекс»