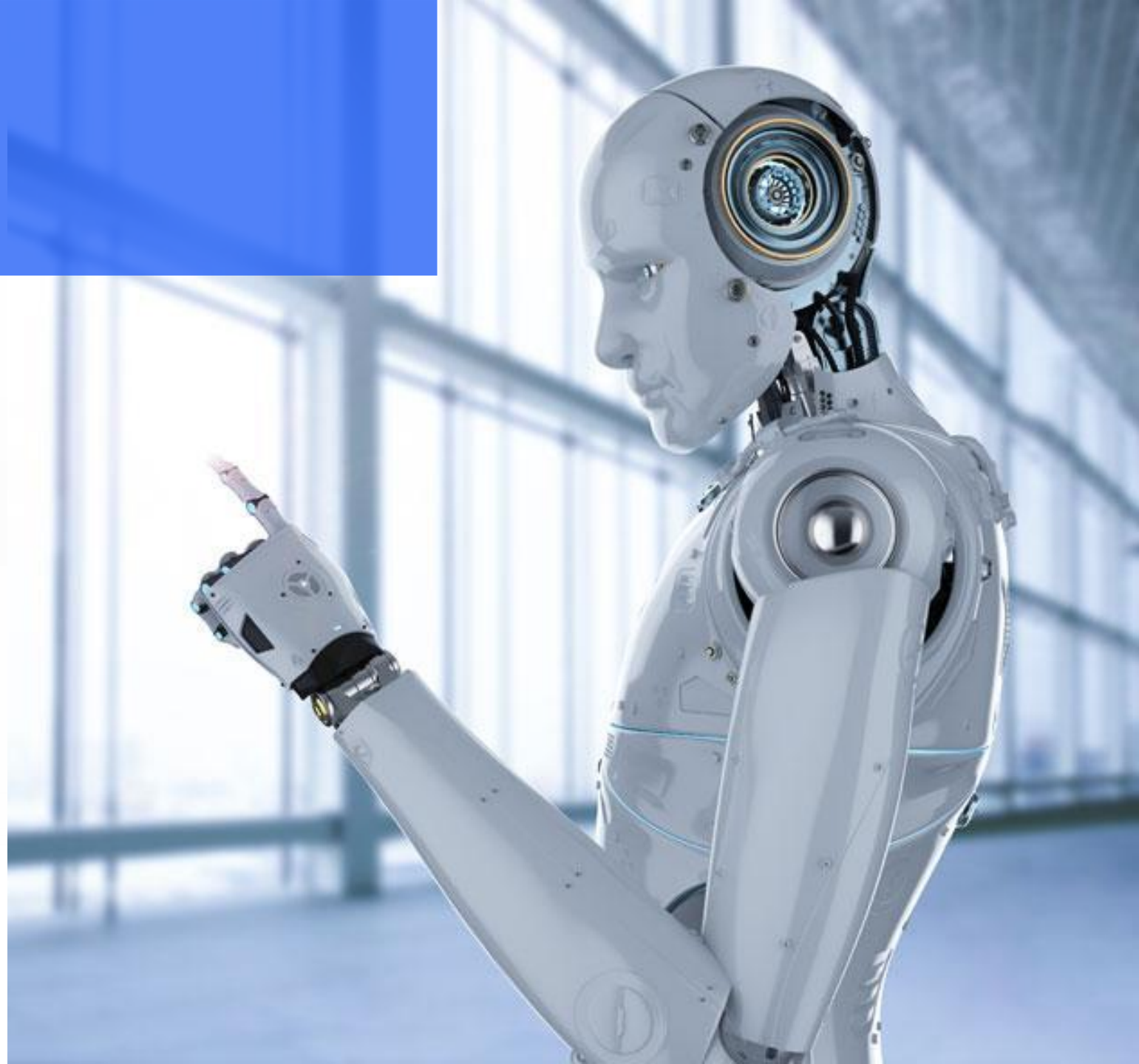
WEB-SCRAPPING

ПЛАН

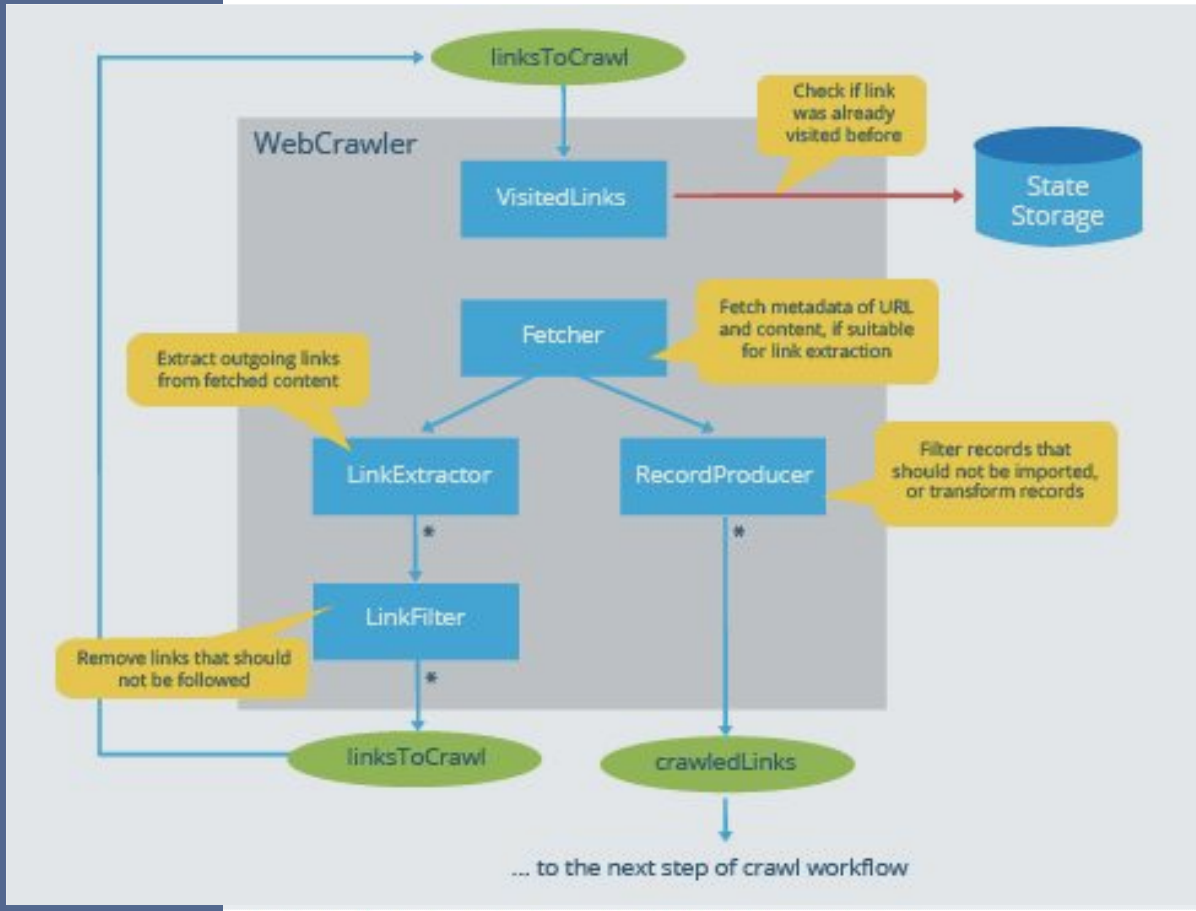
- Интернет и Данные
- Web Scrapping
- Web APIs
- Web scrapping vs API
- Requests + Beautifulsoup
- Scrappy



Данные и Интернет

- Более 474,000 Tweets в минуту в 2019!
- Youtube загружается > 400 hours видео в минуту! В 2019, пользователи в среднем просматривали 4,333,560 роликов в минуту.
- Instagram > 100 миллионов фото в день
- 1,209,600 новых данных производится в среднем в день медиа порталами.
- Большая 4-ка (Apple,FB,Google,Amazon) хранит более 1,200 Петабайт данных

Web – Crawling/Scrapping



Web Crawler («Bot» или «Spider») – программа для автоматизированного поиска документов и информации в интернете. Основная задача – автоматизация множественных переходов по ссылкам. Поисковики используют «Веб-Краулеры» для обхода новых документов и обновления поискового индекса

Самый известный Веб-Краулер на сегодняшний день это Googlebot.

Web APIs

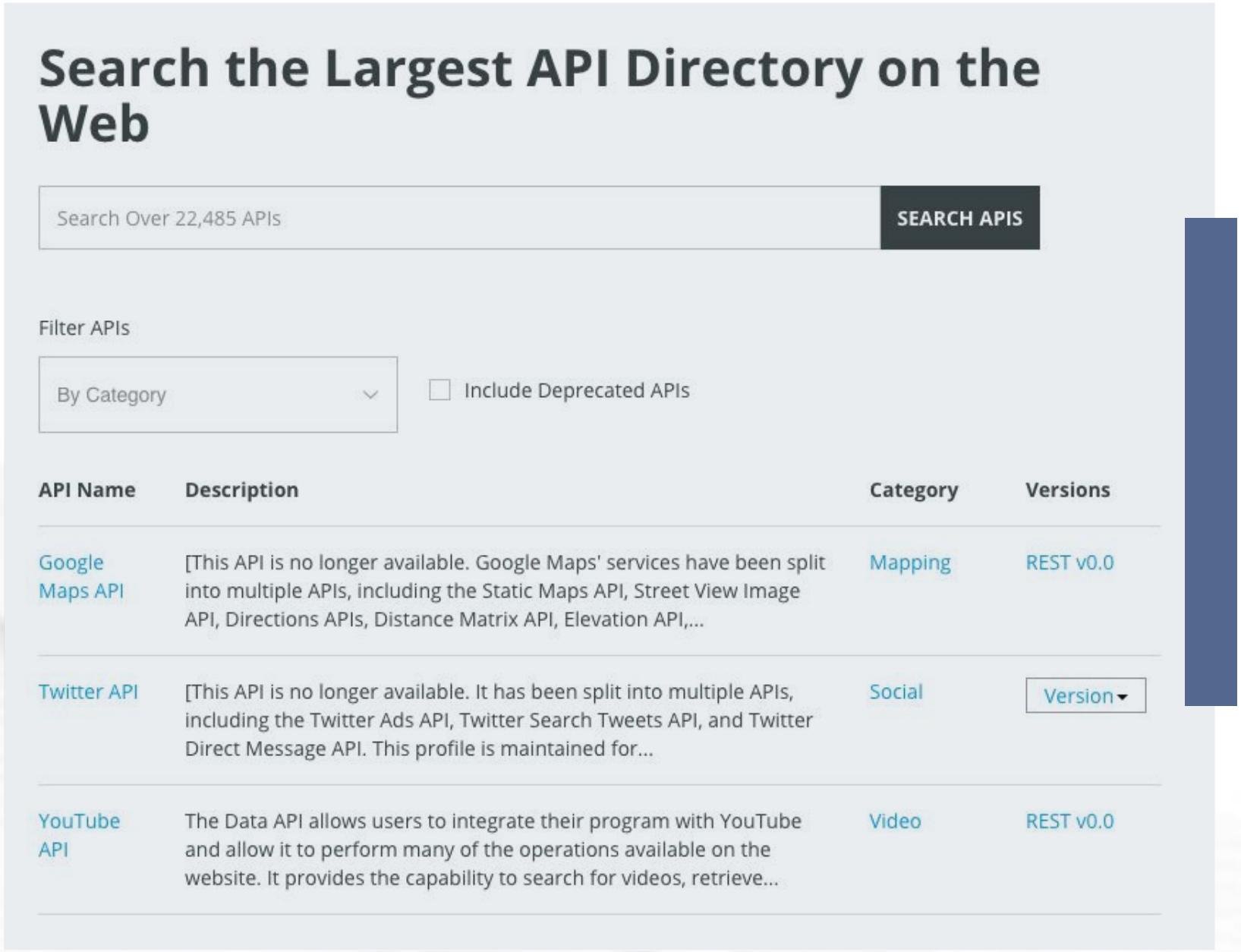
APIs - Набор правил, определяющих, как можно взаимодействовать с конкретным приложением (Веб-сайт)

Плюсы:

- Простота в использовании
- Легкость в интерпретации результата
- Легкость интегрирования
- Не зависят от изменений структуры Веб-сайта
- Хороши для прототипирования и проверки гипотез

Минусы:

- Могут быть отключены в любой момент времени
- Могут быть расхождения с информацией на Веб странице
- Часто существуют лимиты на использование
- Набор данных ограничен (на странице данных часто больше)



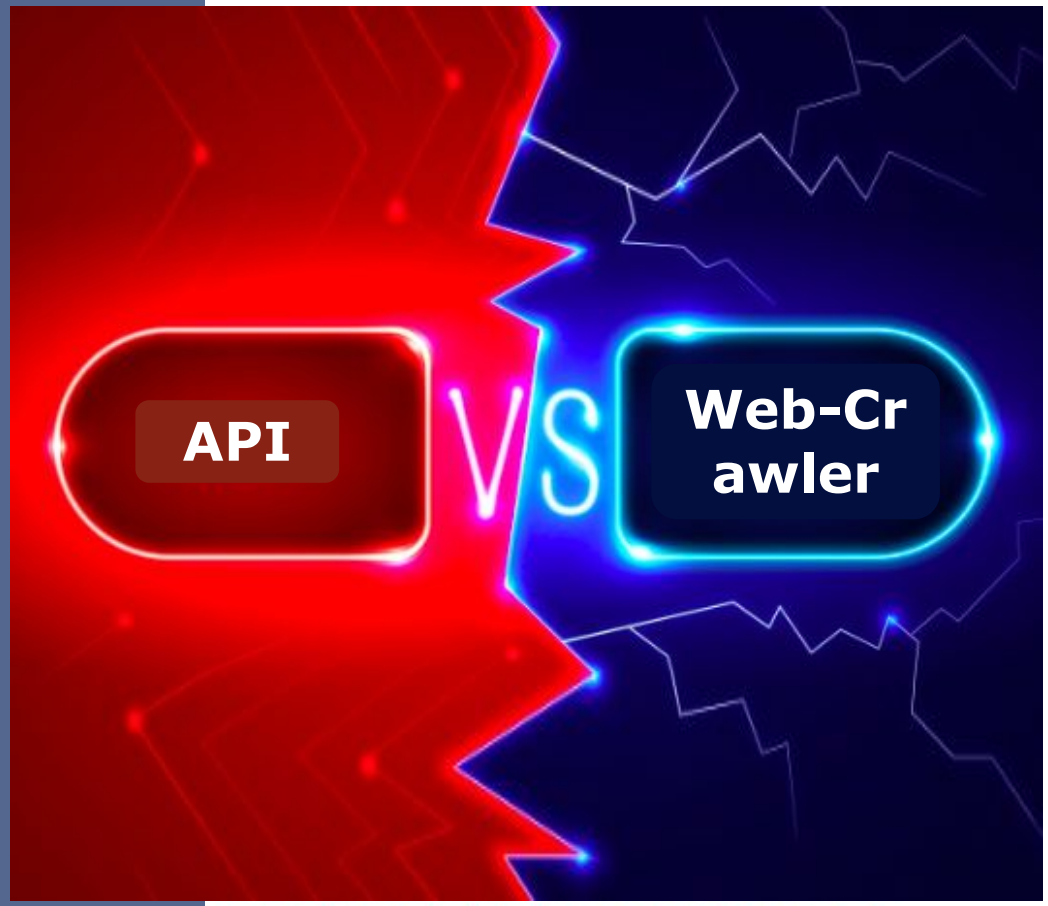
Web APIs

Ресурс содержит

более 16.000 APIs

https://www.programmableweb.com/category/all/apis

Web-Crawling vs API



Web scraping более надежен.

Доступен 24/7, и не требует поддержки со стороны кого-то еще кроме своего разработчика.

Web scraping как правило предоставляет более достоверные данные.

Web scraping не имеет ограничений на выдачу информации.

Большинство APIs имеют ограничения на выдачу

Web scraping часто выгружает более структурированную информацию.

Хотя API разработаны для предоставления структурированной информации, часто их схемы плохо поддерживаются

REQUESTS
Beautifulsoup

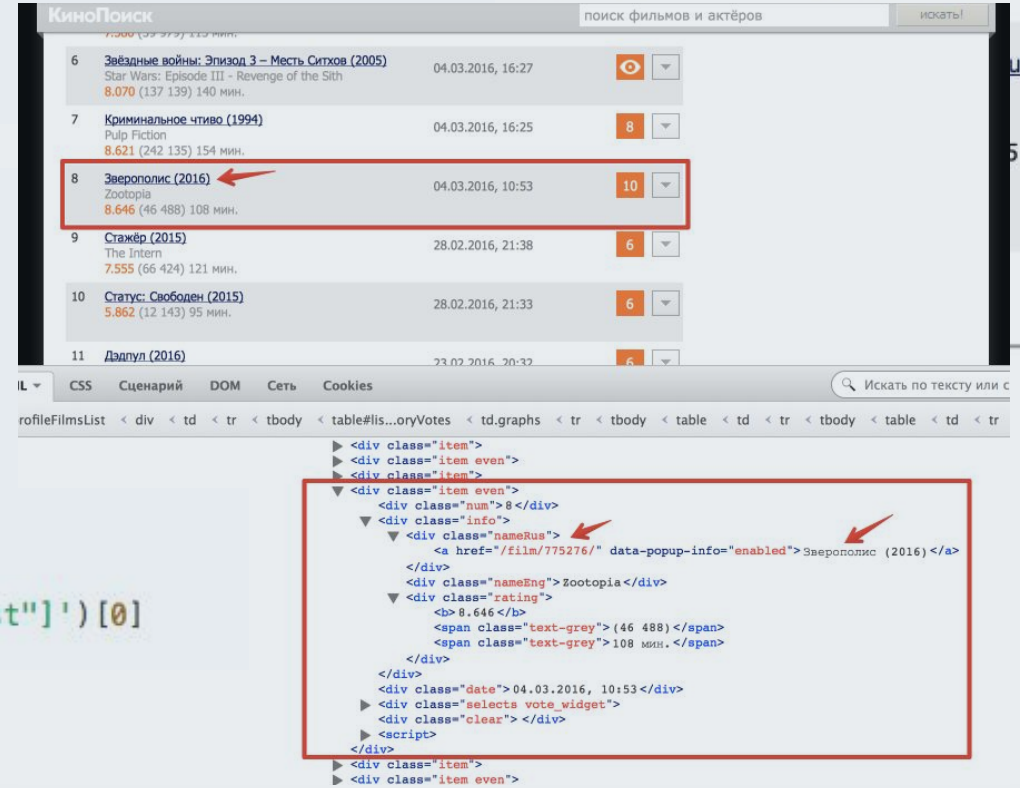
```python
import requests
user_id = 12345
url = 'http://www.kinopoisk.ru/user/%d/votes/list/ord/date/page/2/#list' % (user_id) #
url для второй страницы
r = requests.get(url)
with open('test.html', 'w') as output_file:
  output_file.write(r.text.encode('cp1251'))
```

```python
from bs4 import BeautifulSoup
from lxml import html

# Beautiful Soup
soup = BeautifulSoup(text)
film_list = soup.find('div', {'class': 'profileFilmsList'})

# lxml
tree = html.fromstring(text)
film_list_lxml = tree.xpath('//div[@class = "profileFilmsList"]')[0]
```

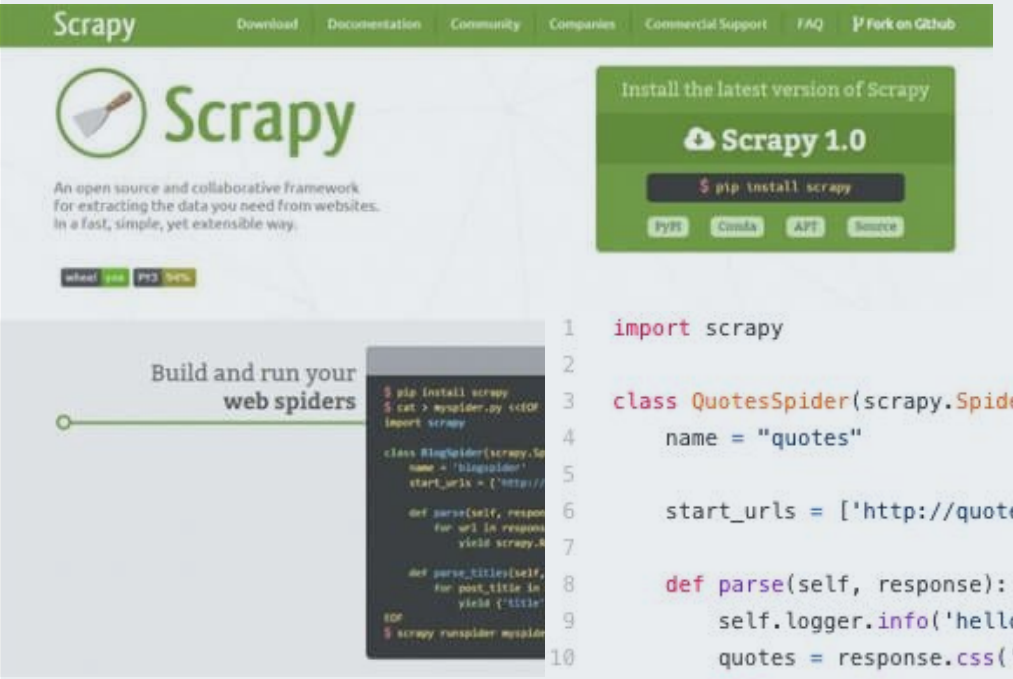
КиноПоиск
найди своё кино!

Если вы видите эту страницу, значит с вашего IP-адреса поступило необычно много запросов. Система защиты от роботов (СЗоР) решила, что вместо вас действует программа, и ограничила доступ.

Для автоматического получения рейтинга, пожалуйста используйте xml версию.



Scrappy




```python
import scrapy

class QuotesSpider(scrapy.Spider):
    name = "quotes"

    start_urls = ['http://quotes.toscrape.com']

    def parse(self, response):
        self.logger.info('hello this is my first spider')
        quotes = response.css('div.quote')
        for quote in quotes:
            yield {
                'text': quote.css('.text::text').get(),
                'author': quote.css('.author::text').get(),
                'tags': quote.css('.tag::text').getall(),
            }

        next_page = response.css('li.next a::attr(href)').get()

        if next_page is not None:
            next_page = response.urljoin(next_page)
            yield scrapy.Request(next_page, callback=self.parse)
```

Scrapy – Open Source платформа для совместной работы для извлечения данных с веб-сайтов с очень высокой производительностью.

Особенности:

- Имеет встроенную поддержку для извлечения данных из источников HTML с использованием XPath и CSS.
- Кроссплатформенность.
- Легко расширяемая платформа
- Быстрая.
- Потребляет намного меньше памяти и ресурсов процессора.
- Имеет хорошую поддержку за счет большого сообщества разработчиков.

APIs

API (Application programming interface) — контракт, который предоставляет программа. «Ко мне можно обращаться так и эдак, я обязуюсь делать то и это».

REST (Representational State Transfer) - согласованный набор архитектурных принципов для создания масштабируемой и гибкой сети, описывающий компоненты системы и их взаимодейтсвия)

RESTful —сеть, которая отвечает ограничениям Филдинга:

- Определение ресурсов
- Управление ресурсами через представления
- Гипермедиа

- Код по требованию
- Система слоёв
- Кэширование