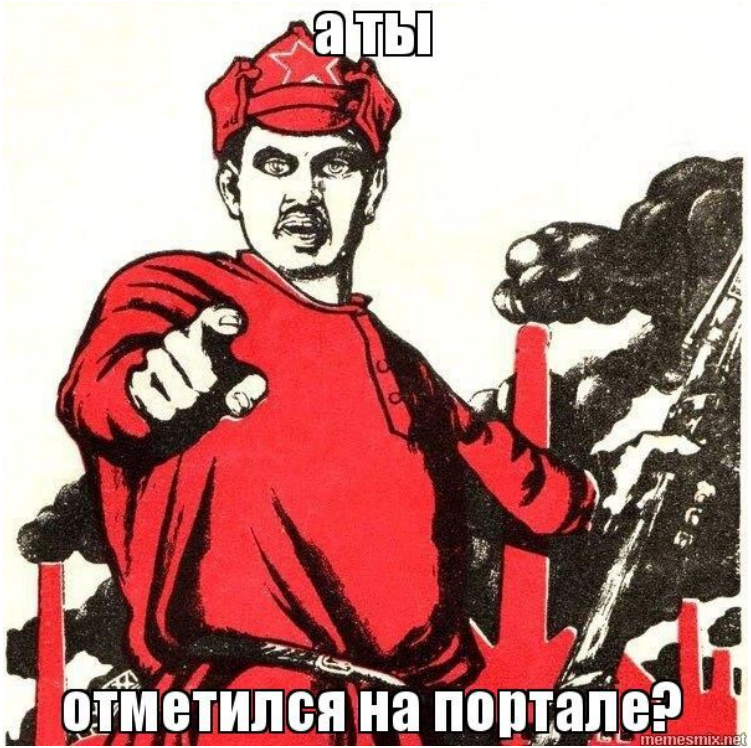
Python

Лекция 4: Парсинг данных. Регрессия

Отметься на портале!



План занятия

- Введение в веб
- Парсинг
- Парсинг данных
- Фласк
- Введение в линейную регрессию
- Нелинейная регрессия

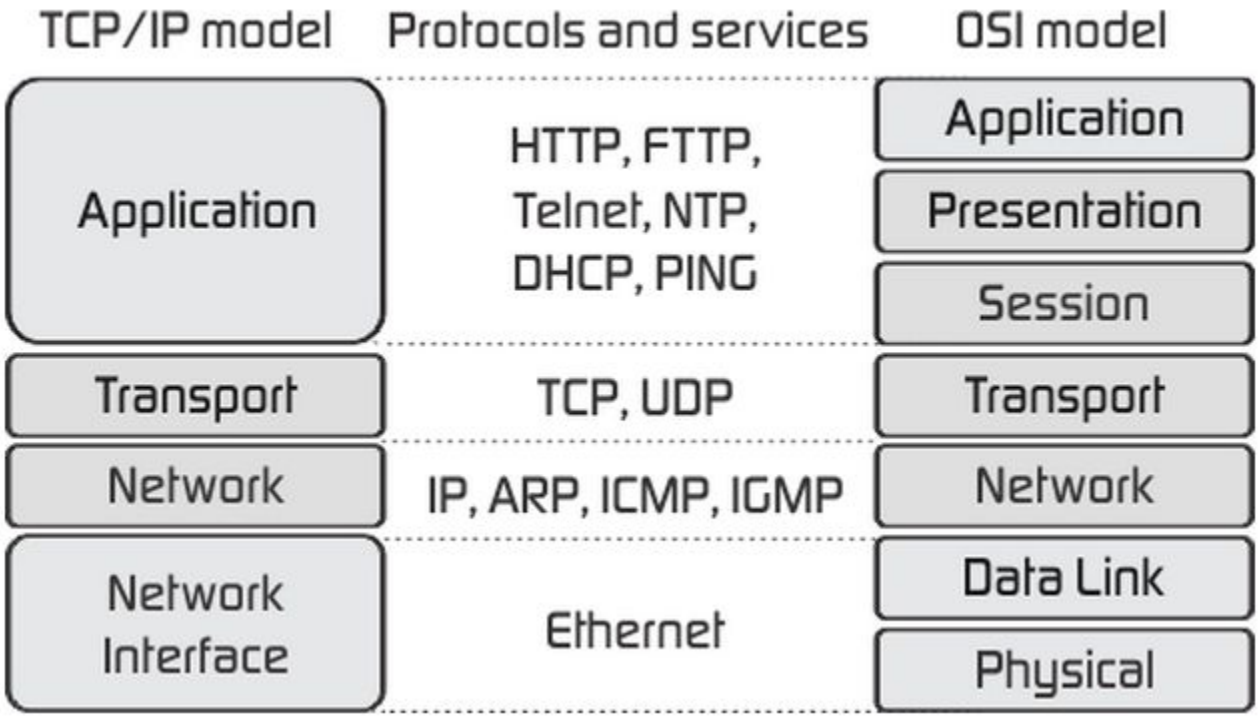
Подготовимся к лекции

1. Переходим в директорию вашего форка
2. Открываем в ней терминал
3. git checkout master
4. git pull upstream master
5. git push -u origin master

Сетевые протоколы

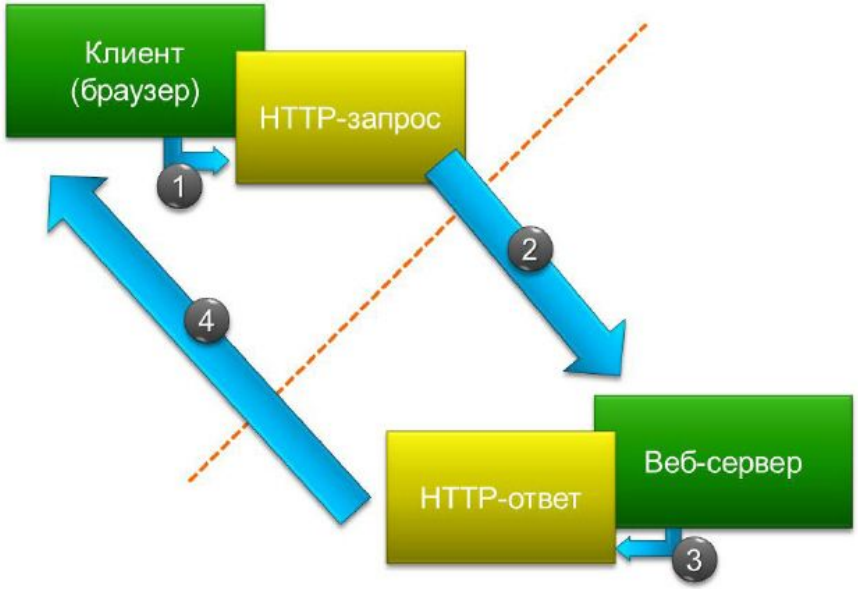
- Сеть -- совокупность устройств, подключенных друг к другу (физически или логически) и общающихся между собой
- Сетевой протокол -- набор правил и действий для “общения” >2 устройств, подключенных к сети
- Модель OSI
- Модель TCP/IP

Модель OSI & TCP/IP



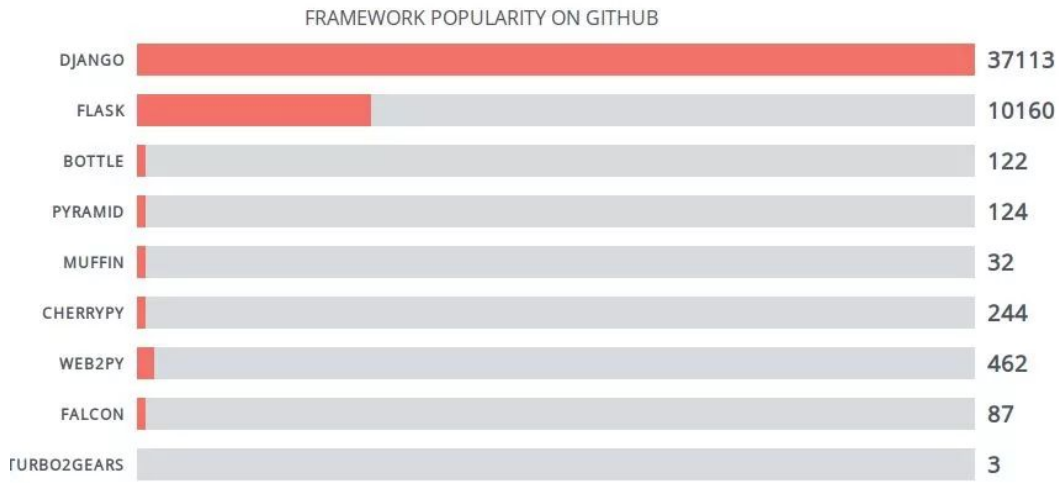
Протокол HTTP(S)

- HTTP (HyperText Transfer Protocol) -- протокол передачи гипертекста
- Клиент-серверное общение:
  - Клиент формирует (http) запрос
  - Сервер обрабатывает запрос и отвечает



Flask

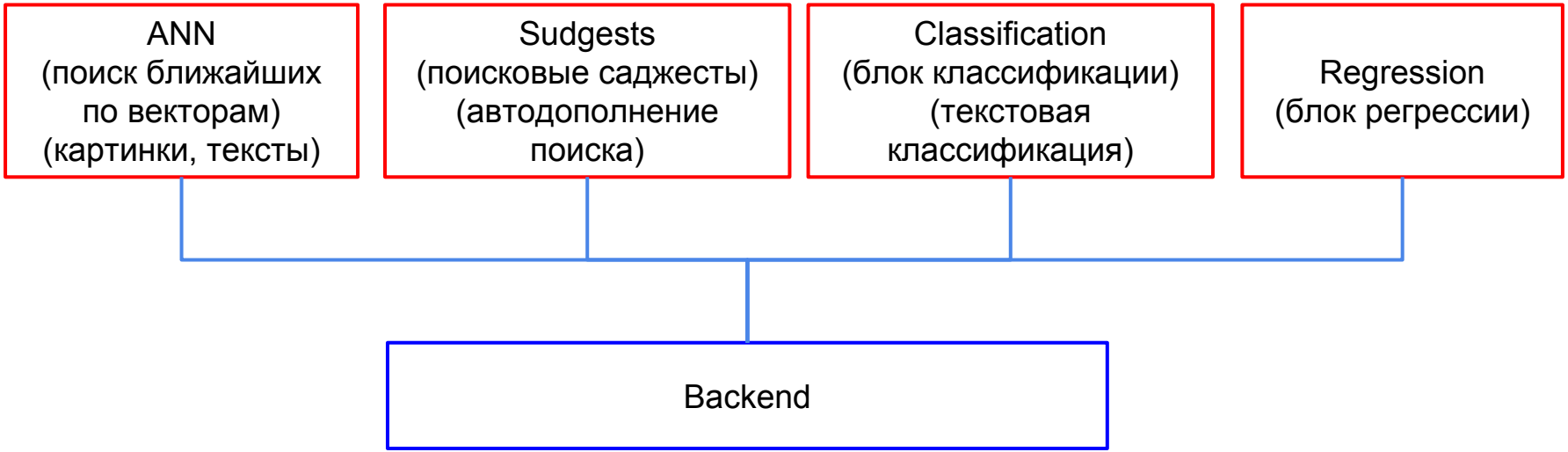
Микро фреймворк для создания веб приложений



http://flask.pocoo.org/docs/1.0/quickstart/#quickstart

http://flask.pocoo.org/docs/1.0/tutorial/

Flask


ANN
(поиск ближайших
по векторам)
(картинки, тексты)
Sudgests
(поисковые саджесты)
(автодополнение
поиска)
Classification
(блок классификации)
(текстовая
классификация)
Regression
(блок регрессии)
Backend

Flask

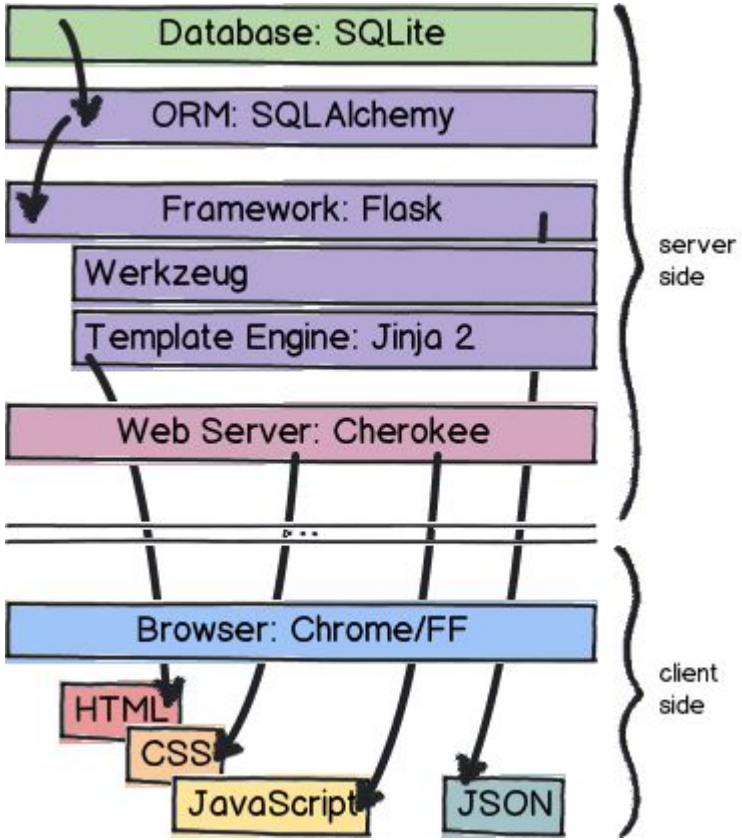
```
from flask import Flask

app = Flask(__name__) #app is the name of the object here
```

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!"

if __name__ == "__main__":
    app.run()
```



Взгляд с другой стороны. Парсинг

Зачем вообще нам что-то парсить?

Алгоритм парсинга

1) Можно не парсить - не парсить
2) Есть возможность получить api-ключ - проще получить
3) Контент динамический?
    a) да: Понять какой запрос идет на сервер
    b) нет: Запрашивать страницу, понять структуру страницы
4) Есть ли защита от парсинга?
    a) нет: Просто грузим страницы
    b) да: подмена headers и user-agent, использование сетей прокси, selenium, эмуляция поведения человека

Можно не парсить - не парсить

1) Погуглить известные хранилища открытых данных
   a) https://data.mos.ru/
   b) https://www.kaggle.com/
   c) https://docs.google.com/spreadsheets/d/1ZSLP1McnXv0FtOd9t7dMp3AfaiusvaGwWV0F9g2pbho/edit#gid=0

Простейший случай. Requests

Статический контент, защиты от парсинга нет.

```
r = requests.get("https://lenta.ru/2018/06/16/")
r.ok
r.cookies
r.content
r.status_code
r.url
r.text
```

Спарсили, и что делать?

Простейший случай. BeautifulSoup

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(html_doc, 'html.parser')
```

```
soup = BeautifulSoup('<b class="boldest">Extremely bold</b>', 'html.parser')
tag = soup.b
type(tag)
```

```
bs4.element.Tag
```

Тэг, у тега есть атрибуты

```
tag.attrs
```

```
{'class': ['boldest']}
```

Динамический контент

Динамический контент, защиты от парсинга нет.

[Пример сайта с динамическим контентом]

```
</div>
    <div class="item-view-similars">
   <div class="similars js-similars similars_column-4"
data-show-more-btn="1"> <div class="similars-inner similars-inner_hidden js-similars-inner"> <div class="similars-title">Похожие объявления</div>
```

Что делать?

1. Консоль разработчика
2. Найти запрос, по которому возвращается контент
3. Эмулировать запросы от клиента

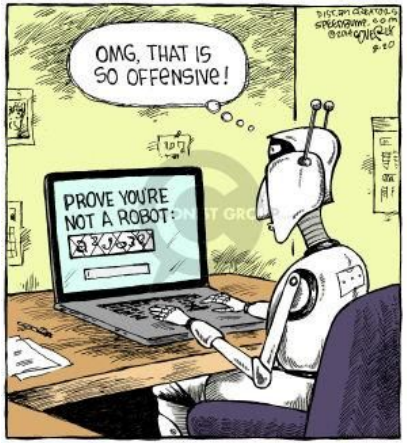
Защита от парсинга

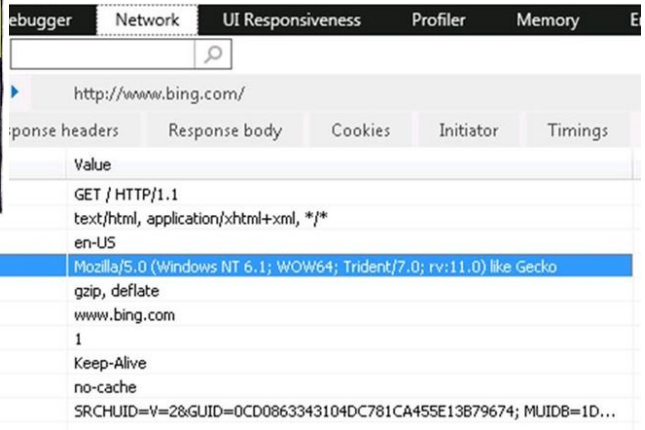
Способы:

- капча
- бан
- левые статические страницы

Что делать?

1. Подстановка user-agent, headers
2. Таймауты
3. Прокси, тор
4. Парсинг из точек с большим лимитом (общага =))





Тяжелая артиллерия. Selenium

Более сложный случай - сильная защита, контент типа саджестов, который сложно эмулировать запросами итд итп

Что делать?

Заставим компьютер быть похожим на человека

XPath:

```
/wikimedia/projects/project/editions/*[2]
```

XML document:

```
<?xml version="1.0" encoding="utf-8"?>
<wikimedia>
 <projects>
  <project name="Wikipedia" launch="2001-01-05">
   <editions>
    <edition language="English">en.wikipedia.org</edition>
    <edition language="German">de.wikipedia.org</edition>
    <edition language="French">fr.wikipedia.org</edition>
    <edition language="Polish">pl.wikipedia.org</edition>
   </editions>
  </project>
  <project name="Wiktionary" launch="2002-12-12">
   <editions>
    <edition language="English">en.wiktionary.org</edition>
    <edition language="French">fr.wiktionary.org</edition>
    <edition language="Vietnamese">vi.wiktionary.org</edition>
    <edition language="Trukish">tr.wiktionary.org</edition>
   </editions>
  </project>
 </projects>
</wikimedia>
```

Давайте поиграем

Хочется с https://fred.stlouisfed.org/ спарсить все временные ряды, относящиеся к США.

1) Какой тип контента?
2) Как будем действовать?

Вернемся к yahoo.finance

1) Как можно по-другому спарсить данные?
2) Давайте попробуем сделать это

Домашнее задание

Подготовка данных для проекта. С каждого PR с парсером (по парсеру на человека). Ветка homework_04

Источники

Документа супа

Flask мануал

Объемный обзор фласка

Хаброучебник фласка

Краткий и емкий мануал по requests