

Сетевое взаимодействие

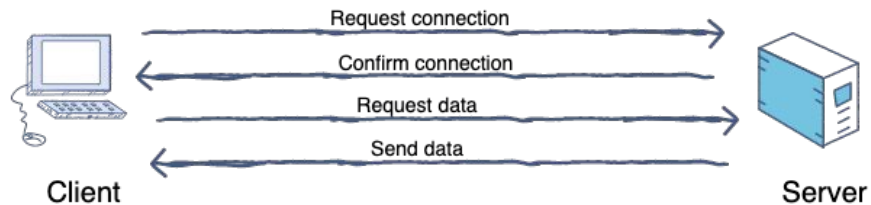
Лекция 7

Алена Елизарова

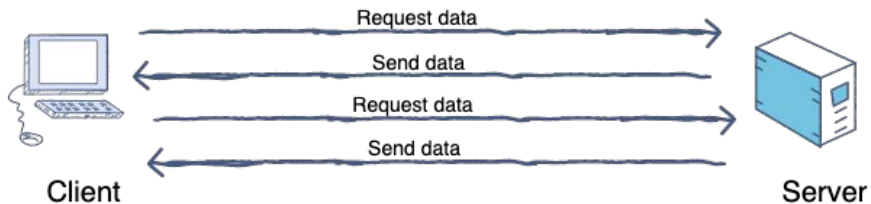
1. стек протоколов
2. Сокеты
3. Программа клиент-сервер
4. Полезные библиотеки
5. Создание веб-фреймворка

1. Физический
2. Канальный
3. Сетевой
4. Транспортный
5. Сеансовый
6. Представления
7. Прикладной

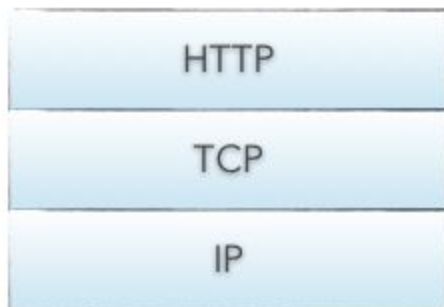
TCP



UDP

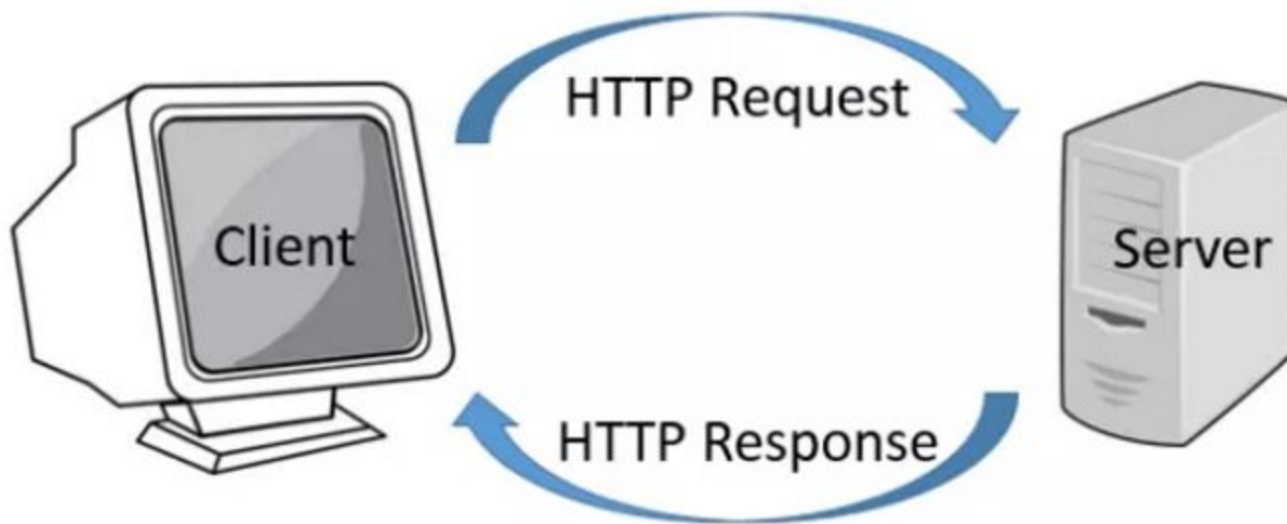


http



https





Методы

GET, POST, PUT, HEAD, OPTIONS, PATCH, DELETE

Пример запроса

GET /wiki/страница HTTP/1.1

Host: ru.wikipedia.org

User-Agent: Mozilla/5.0 (X11; U; Linux i686; ru; rv:1.9b5) Gecko/2008050509 Firefox/3.0b5

Accept: text/html

Connection: close

Пример ответа

HTTP/1.1 200 OK

Date: Wed, 11 Feb 2009 11:20:59 GMT

Server: Apache

X-Powered-By: PHP/5.2.4-2ubuntu5wm1

Last-Modified: Wed, 11 Feb 2009 11:20:59 GMT

Content-Language: ru

Content-Type: text/html; charset=utf-8

Content-Length: 1234

Connection: close

```
import socket

sock = socket.socket()
sock.bind(('', 10001))
sock.listen()
conn, addr = sock.accept()

print('connected:', addr)

while True:
    data = conn.recv(1024)
    if not data:
        break
    conn.send(data.upper())

conn.close()
```



```
import socket

sock = socket.socket()
sock.connect(('127.0.0.1', 10001))
sock.sendall('hello, world!'.encode('utf-8'))

data = sock.recv(1024)
sock.close()

print(data.decode('utf-8'))
```

```
# сервер
with socket.socket() as sock:
    sock.bind(('', 10001))
    sock.listen()
    conn, addr = sock.accept()
    with conn:
        while True:
            data = conn.recv(1024)
            if not data:
                break
            print(data.decode('utf-8'))

#клиент
with socket.create_connection(('127.0.0.1', 10001)) as sock:
    sock.sendall('ping'.encode('utf-8'))
```

```
# сервер
with socket.socket() as sock:
    sock.bind(('', 10001))
    sock.listen()
    conn, addr = sock.accept()
    conn.settimeout(5) # None | 0 | gt 0
    with conn:
        while True:
            try:
                data = conn.recv(1024)
            except socket.timeout:
                print('close connection by timeout')
                break
            if not data:
                break
        print(data.decode('utf-8'))
```

<https://github.com/django/django/blob/master/django/core/management/commands/runserver.py>

<https://medium.com/@arpitgupta820/what-happens-when-you-hit-runserver-in-django-962a295f06ea>

```
from smtplib import SMTP
from_addr = "any@mail.ru" # От кого
to_addr = "some@mail.ru" # Кому
message = """From: <f"{from_addr}">
To: <f"{to_addr}">
Subject: Some subject
MIME-Version: 1.0
Content-Type: text/plain;
Content-Transfer-Encoding: 8bit
Приветикки
"""
connect = SMTP('mail.smtp.ru')
connect.set_debuglevel(1)
connect.sendmail(from_addr, to_addr, message)
connect.quit()
```

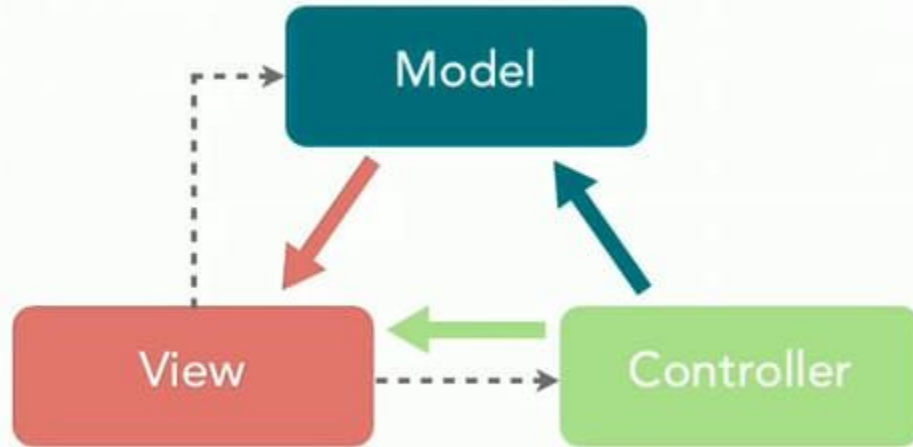
```
>>> r = requests.get('https://api.github.com/user', auth=('user',  
'pass'))  
>>> r.status_code  
200  
>>> r.headers['content-type']  
'application/json; charset=utf8'  
>>> r.encoding  
'utf-8'  
>>> r.text  
'{"type": "User" ...'  
>>> r.json()  
{ 'private_gists': 419, 'total_private_repos': 77, ... }
```

```
import urllib
doc = urllib.urlopen("https://www.python.org/").read()
print(doc[:50])

data = {"search": "Python"}
enc_data = urllib.urlencode(data)

# метод GET
f = urllib.urlopen("http://searchengine.com/search" + "?" +
enc_data)
print(f.read())

# метод POST
f = urllib.urlopen("http://searchengine.com/search", enc_data)
print(f.read())
```



Model

contains business logic

Controller

interacts with **Model** to create data for the **View**

View

renders content to the user and relays user commands to the **Controller**

Домашнее задание

- Написать сервер
- Написать консольную утилиту для парсинга веб-страницы, такую, что она будет отправлять запрос на ваш сервер, который в свою очередь получит тело страницы, распарсит ее, очистит от html-тегов, и сформирует топ 10 самых частых слов на странице и частоту их использования. При этом вы должны сделать функцию или класс, принимающий объект `HttpRequest` и возвращающий `HttpResponse` в формате json

Спасибо
за внимание!