

О преподавателях



Алена Елизарова

a.elizarova@corp.mail.ru



Антон Кухтичев

a.kukhtichev@corp.mail.ru



Дмитрий Смаль

mialinx@gmail.com

О курсе

- Цель курса - разработать мессенджер (a-la Slack)
- В паре с курсом по Frontend
- Разработка - индивидуальная, без команд
- Задание и разбалловка для всех одна
- О творческом подходе...

Материалы

- Тех. задание: [task.md](#)
- Схема базы данных: [database.png](#)
- Описание API: TODO
- разбалловка: [points.md](#)
- Вообще: <https://github.com/mialinx/tt-fullstack>

О занятиях

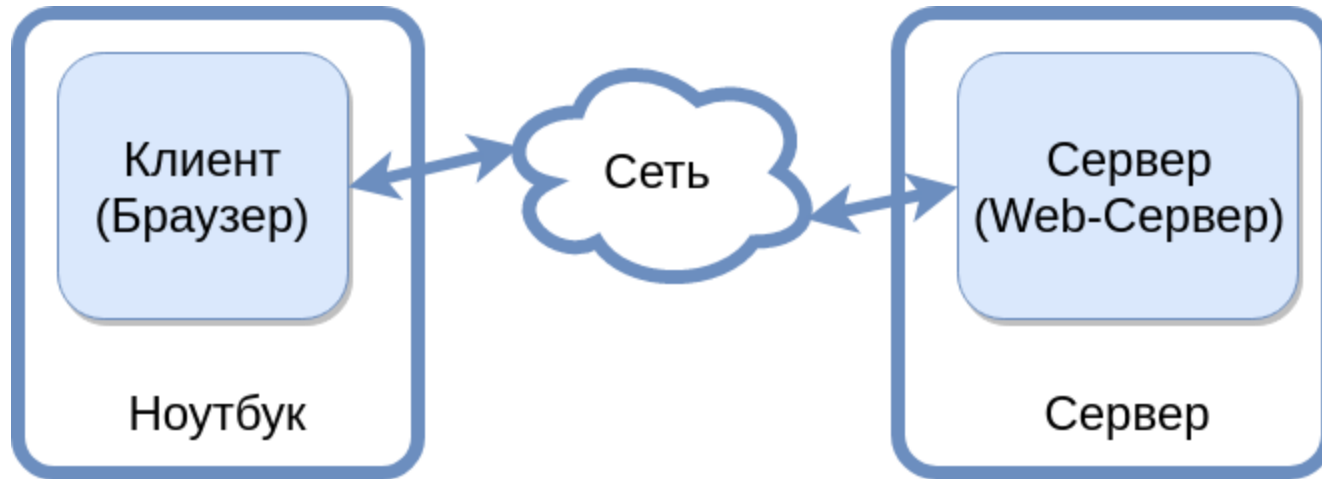
- Посещение - обязательное, check-in первым делом!
- Половина занятия - теория
- Вторая половина - мастер класс и помощь с ДЗ
- Оценка ставится на основе баллов
- Баллы - только за работающий функционал
- Любая разработка и сдача - через Git

Коротко о программе

- Интенсив по Python
- HTTP, Web сервера
- Flask, Application Server
- JSON, API, RPC
- Работа с СУБД
- Авторизация
- Real-Time сообщения

Архитектура Web

Клиент-серверная архитектура



Основное назначение браузера - отображение HTML страниц.
Однако, возможности современных браузеров огромны.
Существуют операционные системы и 3D-игры, работающие
внутри браузеров!

www.evolutionoftheweb.com

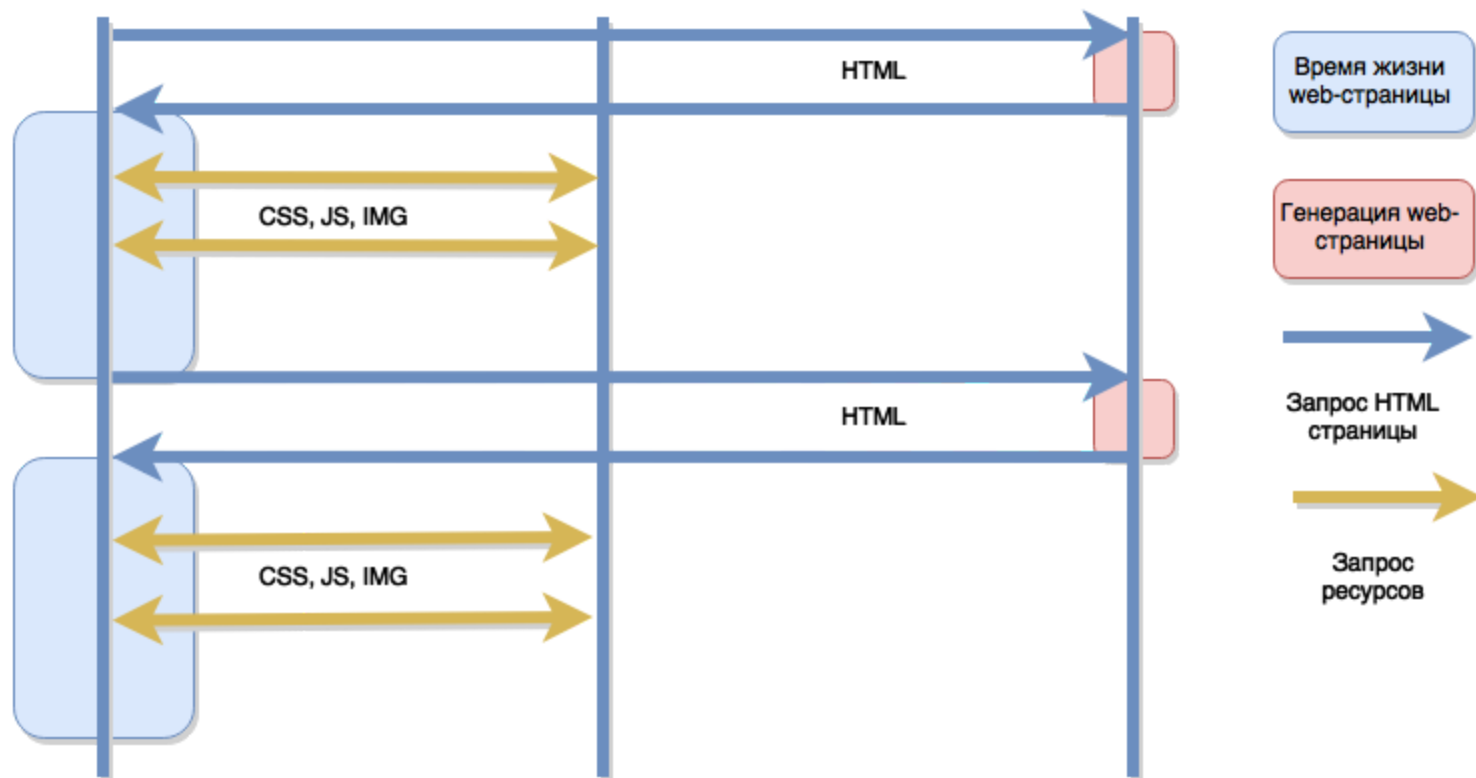
Сценарий работы классического web приложения

- Пользователь вводит URL
- Браузер загружает Web страницу - HTML документ
- Браузер анализирует (parse) HTML и загружает доп. ресурсы
- Браузер отображает (rendering) HTML страницу
- Пользователь переходит по гиперссылке или отправляет форму
- Цикл повторяется

Браузер

Web-Сервер

Application-сервер



Сценарий работы современного приложения

- Браузер загружает Web страницу, ресурсы и отображает ее
- JavaScript загружает данные с помощью AJAX запросов
- JavaScript обеспечивает полноценный UI на странице
- Пользователь взаимодействует с UI, что приводит к вызову JavaScript обработчиков
- JavaScript обновляет данные на сервере или загружает новые данные, используя AJAX

Браузер

Web-Сервер

Application-сервер



Особенности современных Web-приложений

- UI находится на 1 или нескольких страницах (one-page)
- UI полностью статичен: HTML, CSS, JS - статические файлы
- Логика UI полностью работает на стороне клиента
- Используется шаблонизация в JavaScript
- Application сервер возвращает чистые данные (JSON или XML, а не HTML)

Рабочее
окружение

Устанавливаем Ubuntu

Скачиваем VirtualBox

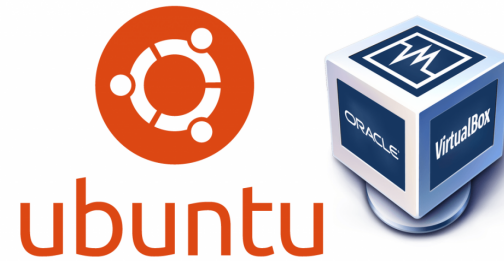
<https://www.virtualbox.org/wiki/Downloads>

Скачиваем образ Ubuntu

<https://www.ubuntu.com/download/desktop>

...

Profit!



Структура директорий Linux

```
/
|--home/
|   |--nuf/           # домашняя директория
|--usr
|   |--bin/           # исполняемые файлы (программы)
|   |   |--python
|   |--lib/           # библиотеки (например so)
|       |--python2.7/ # библиотеки *.py
|--etc/               # настройки серверов
|   |--nginx/         # настройки nginx
|--var/               # разное
|   |--lib/           # часто файлы баз данных
|--tmp/               # временные файлы
```


Установка программ в Linux

```
sudo apt install git      # установить известную программу
sudo apt search nodejs   # поискать среди доступных
sudo apt remove git      # удалить программу
sudo apt purge nginx     # удалить программу и все ее файлы
```

apt - пакетный менеджер Ubuntu (yum, pacman, emerge...)

sudo - временное повышение привелегий до root

В репозиториях ОС обычно не самые свежие программы.

Установка (например) NodeJS v8

Гуглим **nodejs8 download ubuntu**

Переходим на офф. сайт

<https://nodejs.org/en/download/package-manager/>

Там примерно следующее:

```
curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -  
sudo apt install -y nodejs
```

Отдельный репозиторий NodeJS

После этого Ubuntu будет знать откуда скачать и обновить NodeJS

```
[nuf@nuftop tmp]$ cat /etc/apt/sources.list.d/nodesource.list  
deb https://deb.nodesource.com/node_8.x xenial main  
deb-src https://deb.nodesource.com/node_8.x xenial main
```

Задание

Установить:

- Python 3
- NodeJS 8
- Nginx - подойдет любой
- Postgres - подойдет любой
- Git - подойдет любой



Структура директорий проекта

```
/home/nuf/quack      # корневая директория вашего проекта
|--public/           # директория с файлами для Frontend
|--node_modules/     # Библиотеки для JavaScript (*)
|--package.json      # Описание зависимостей для JavaScript
|
|--app/              # Директория с Python кодом приложения
|--tests/            # Тесты вашего приложения
|--venv/             # Виртуальное окружение, библиотеки Python (*)
|--requirements.txt  # Описание зависимостей для Python
|
|--sql/              # SQL скрипты, описание базы данных
|--.gitignore        # Настройки файлов для Git
```

Работа с Git

<code>git init</code>	<code># создать новый репозиторий</code>
<code># или</code>	
<code>git clone git@github.com:nuf/quack.git</code>	<code># клонировать</code>
<code>git status</code>	<code># посмотреть статус файлов</code>
<code>git add some_file some_dir</code>	<code># Добавить файлы в индекс</code>
<code>git commit</code>	<code># сформировать новый комит</code>
<code>git push</code>	<code># отправить изменения в github</code>

Содержимое .gitignore

node_modules/

venv/

*.рус

*.swo

*.swp

Установка библиотек Python (1)

PyPi - централизованный репозиторий библиотек для Python

<https://pypi.org/>

Для установки библиотек используется **pip**

Устанавливаем pip3

```
sudo apt install python3-pip
```

А затем уже библиотеки Python

```
sudo pip3 install flask      # В систему
```

```
pip3 install --user py.test  # или только для себя
```

Или ...

Использование VirtualEnv

virtualenv - программа для установки отдельного набора библиотек "для проекта"

Устанавливаем сам virtualenv

```
pip3 install --user virtualenv
```

Переходим в директорию проекта

```
cd /home/nuf/quack
```

Создаем Виртуальное окружение в директории venv

```
virtualenv venv
```

"Активируем" его

```
source ./venv/bin/activate
```

Установка библиотек Python (2)

Устанавливаем необходимые библиотеки в venv
`pip3 install flask pytest psycopg2`

"Запоминаем" Версии установленных библиотек
`pip3 freeze > requirements.txt`

Не забываем закомитить файл `requirements.txt`

Но не директорию venv! (она должна быть в `.gitignore`)

Задание

Установить:

- flask
- gunicorn
- psycopg2
- pylint
- pytest
- pytest-flask
- Flask-JSONRPC
- Authlib



Спасибо за внимание!

