

Изоморфные приложения и Python

Виталий Глибин

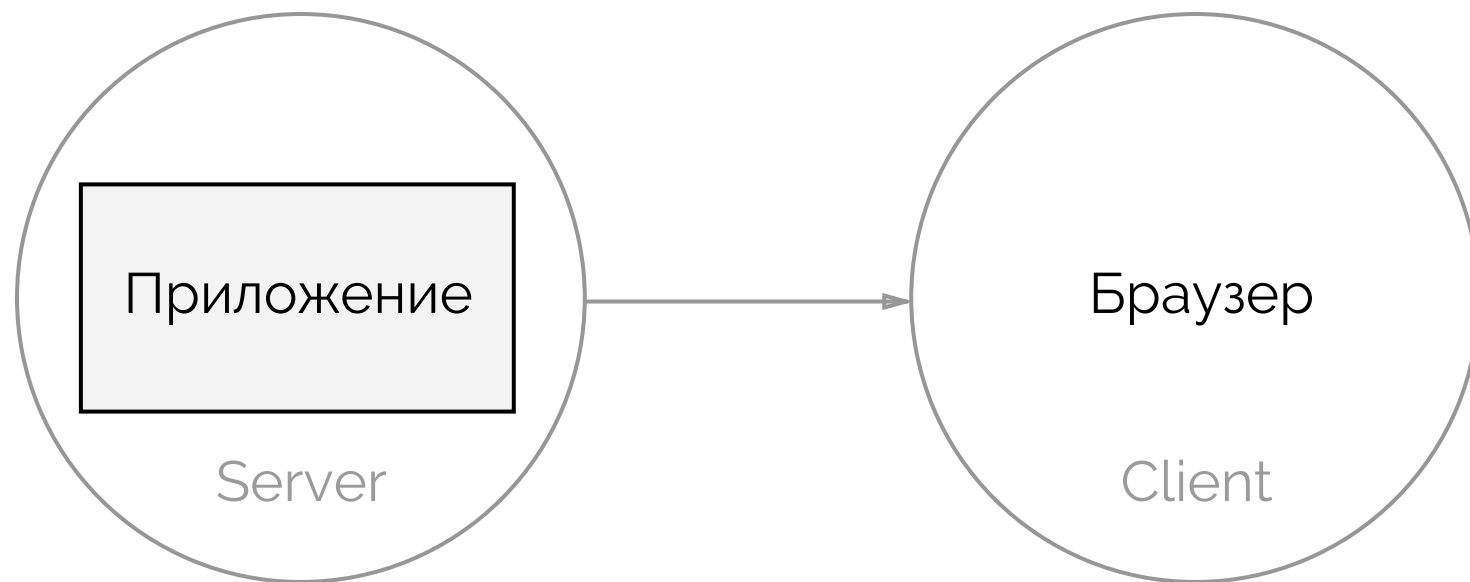
CTO @ Huntflow

Frontend Team Lead @ Программа Главбух

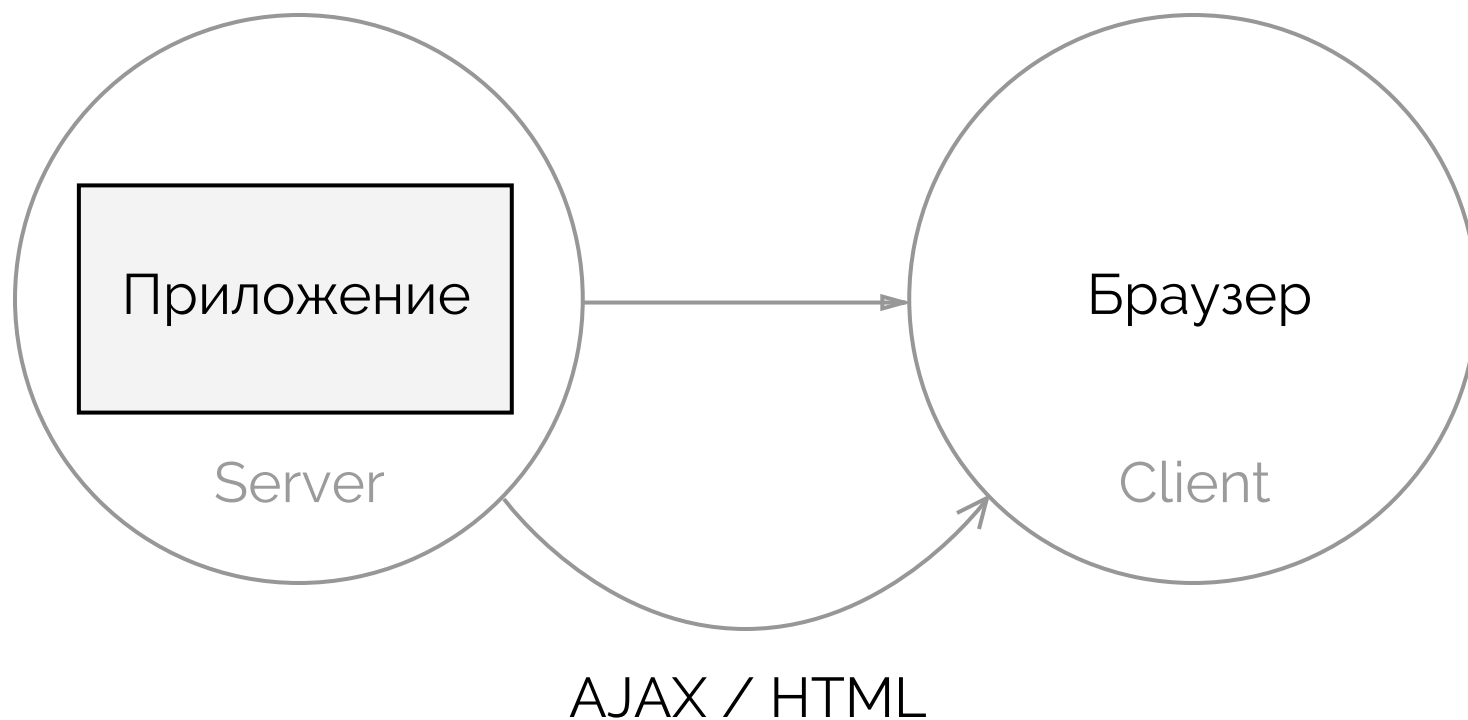
ex. Team Lead @ HeadHunter

**Причем тут
Python?**

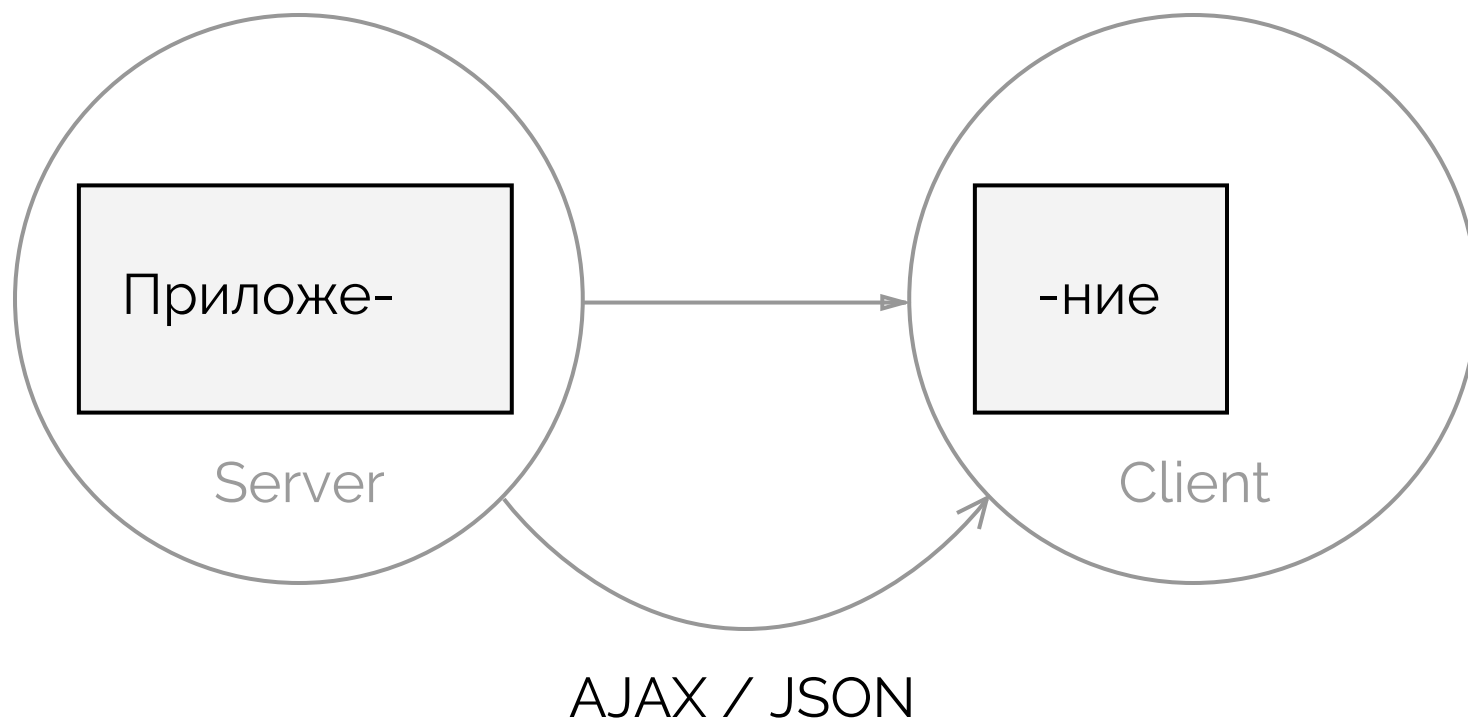
Когда-то давно...



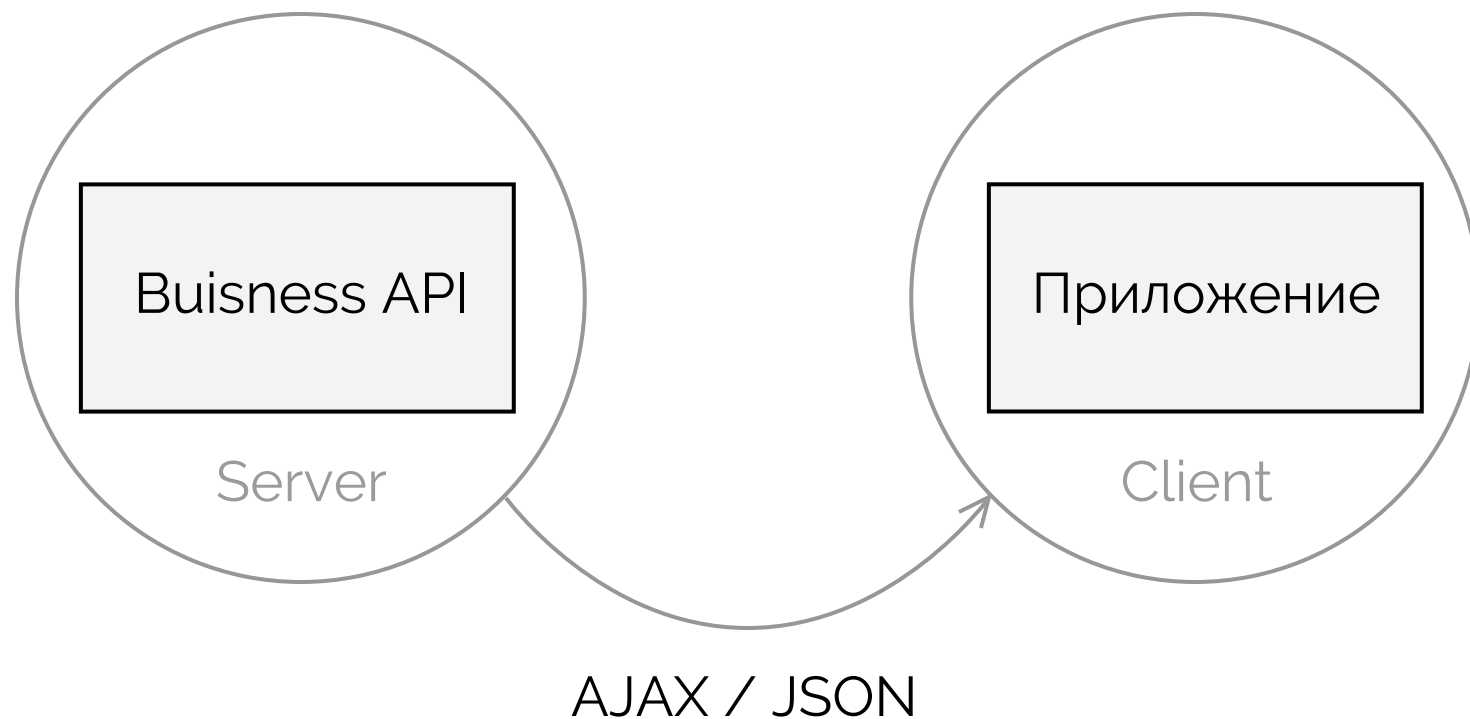
Через какое-то время появился **AJAX**



Потом решили еще ускориться



Single Page Application



Проблемы

Поисковые системы

```
<html>
```

```
  <head>...</head>
```

```
  <body>
```

```
    <div id="app"></div>
```

```
    <script src="all-app-is-here.js"></script>
```

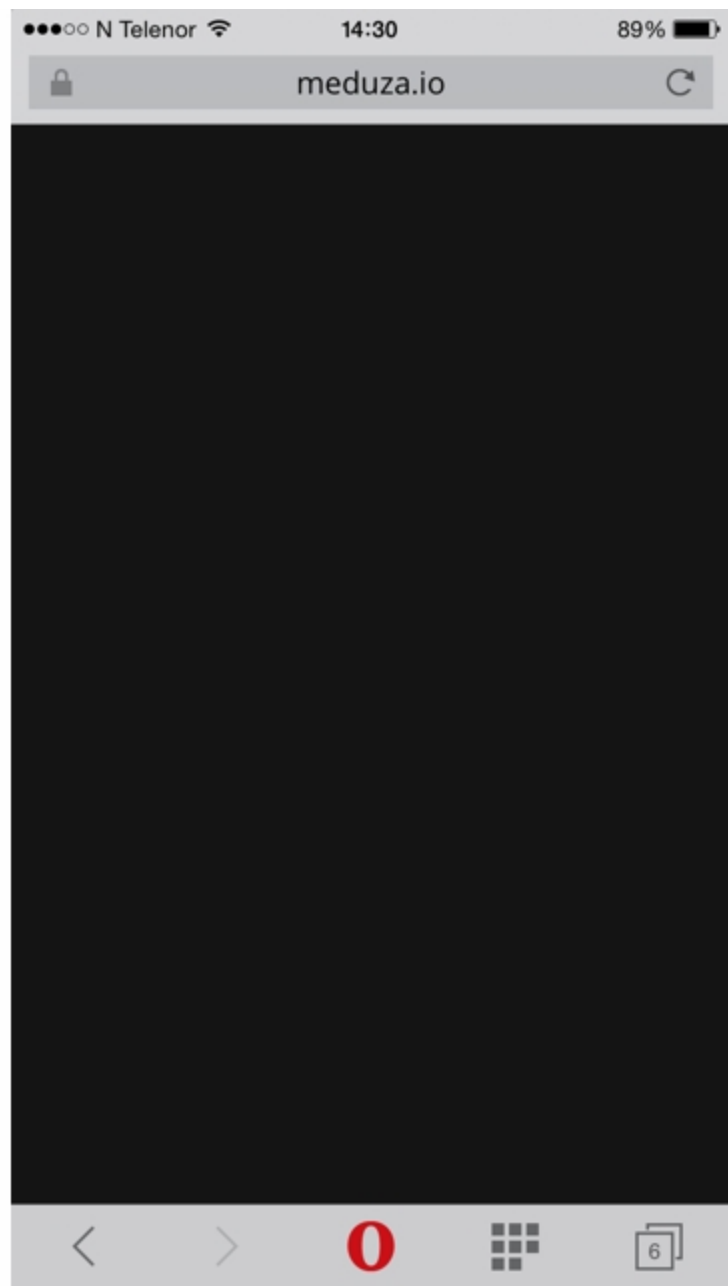
```
  </body>
```

```
</html>
```


Люди

1. Устройства без Javascript
2. Прокси-браузеры (Opera Mini, UC, Puffin)

<http://sokr.me/lbr>



Ajax crawling

#! => _escaped_fragment_

<http://prerender.io>

Долгая первоначальная загрузка

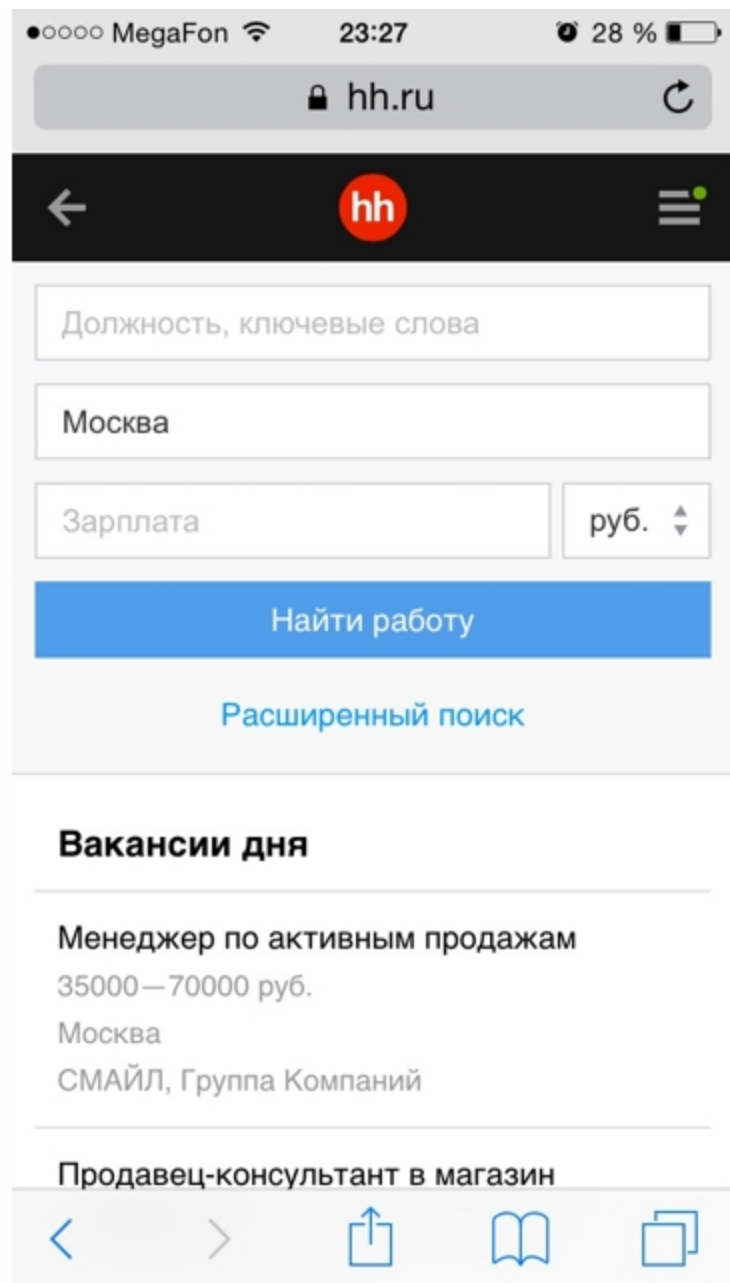
1. Пользователь какое-то время видит или белый экран, или спиннер загрузки
2. 40% людей закрывают страницу, если она загружается более трех секунд
3. Амазон за каждые 100мс загрузки теряет 1% прибыли

**Как сделать
хорошо?**

Новый мобильный сайт HeadHunter (2013–2014)

1. Python на сервере (frontik/tortik)
2. Основной функционал должен работать и без JavaScript
3. При этом сайт должен быть быстрым, модным, современным

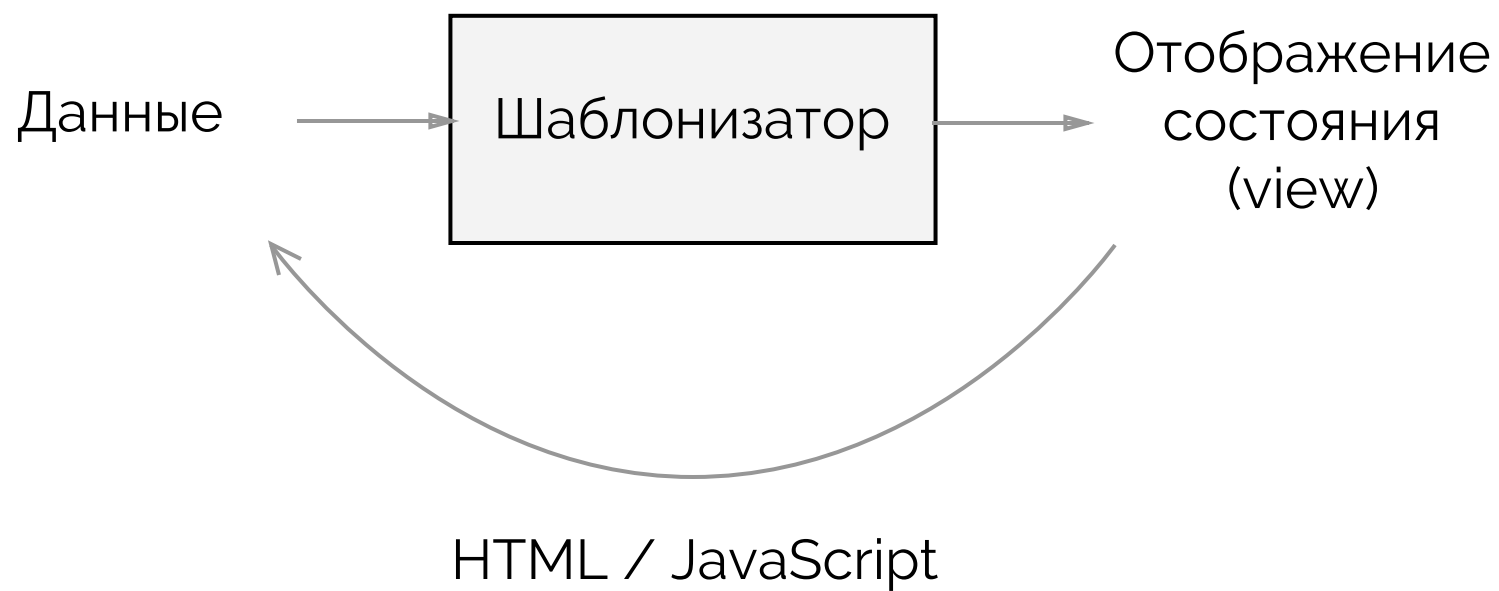
<https://m.hh.ru>



Меню

```
<div class="menu-button"><a href="
    {%- if data.is_menu %}
        {{- data.backurl -}}
    {% else -%}
        /menu?backurl=...
    {%- endif %}"
    onclick="toggleMenu();" ...></a></div>
```

Единая шаблонизация



Nunjucks – «Heavily inspired by jinja2» ©

<http://mozilla.github.io/nunjucks/>

Список резюме (/applicant/resumes)

```
class ResumesPage(page.PageHandler):  
    def get(self):  
        data = fetch_data_from_backends(...)  
        if is_json(self):  
            self.finish(data)  
        else:  
            self.render('pages/resumes.html', **data)
```

На клиенте

```
route('/applicant/resumes').then(function () {  
    var collection = new Resumes.Collection({  
        url: '/applicant/resumes'});  
    collection.fetch().then(function (data) {  
        render('blocks/resumes.html', data);  
    });  
});
```

Сервис для ведения вакансий Huntflow (2015)

1. Python на сервере (tortik)
2. Начали использовать такой же подход, как с мобильным HH

<https://huntflow.ru>

На сервере (pages/resumes.html)

```
{% extends "layout.html" %}
{% block content %}
    {% include 'blocks/resumes.html' %}
{% endblock %}
{% block scripts %}<script>
    var ResumeListData = {{ data.resumes|json|safe }};
</script>{% endblock %}
```

blocks/resumes/item.html

```
<a data-resumes-id="{{ item.id }}">
```

```
...
```

```
</a>
```


blocks/resumes/item.js

```
var el = $('[data-resumes-id="' + item.id + '"]')[0]
var view = new ResumeView({el: el, model: item});
if (!el) {
    view.render();
}
```

Новый сайт GameMiner (2015–...)

1. Python на сервере (tortik)
2. Решили попробовать что-нибудь новое, модное, современное

<http://gameminer.ru>

2007



2009



2012



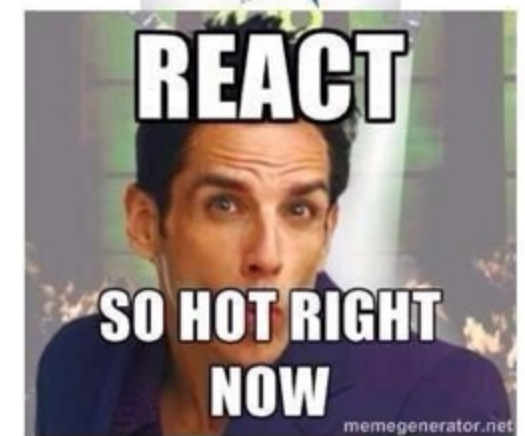
2013



2014



2015





React

React

1. Шаблонизатор (view)
2. Virtual DOM
3. Сам отвечает за восстановление состояния
4. Только на Javascript / NodeJS

<https://facebook.github.io/react/>

React as a service

1. Поднимаем отдельный сервис рендеринга на NodeJS
2. Если нужно отрисовать на сервере, отправляем данные в сервис, результат отдаем на клиент
3. На клиенте все работает

React as a service

```
class ResumesPage(page.PageHandler):  
    def get(self):  
        data = fetch_data_from_backends(...)   
        if is_json(self):  
            self.finish(data)  
        else:  
            html = react_render('App.jsx', data)  
            self.finish(html)
```

React as a service

```
app.post('/render', (request, response) => {  
  let props = request.body.props || {};  
  let uri = request.body.uri;  
  render(props, uri, output =>  
    response.send(output));  
});
```


React as a service

1. Если ломается сервис, на клиенте продолжит работать
2. Есть небольшая задержка на tcp/http

Итого

Спасибо!

Виталий Глибин

@glibin, glibin.ru

Этот доклад: <http://glibin.github.io/lections/pycon2015/>

Tortik: <http://glibin.github.io/lections/pycon2014/>