Информатика

введение в web технологии

© Марченко Антон Александрович 2017 г.
Абрамский Михаил Михайлович

О чём семестр

- Web приложения
  - Как работают
  - Как разрабатываются
  - Какие технологии в основе

Зачем?

- Программный инженер должен хорошо представлять web разработку
- Web разработчики наиболее востребованы на рынке труда
- Понять нравится вам web разработка или нет

Что такое Web?

Web, он же WWW (World Wide Web)

- Система связанных ссылками документов, расположенных на различных компьютерах сети Интернет (Internet)
- Работает на основе Интернета, является основным способом доступа к информации

Интернет

- Глобальная сетевая инфраструктура
- Сеть сетей
- Позволяет любым компьютерам в ней общаться друг с другом
  - с помощью различных протоколов
    - Web использует HTTP

Предки Интернета

- 1950-ые: разработка концепции глобальных компьютерных сетей (WAN)
- 1960-ые: ввод WAN в эксплуатацию
- 1967-декабрь 1969: разработка DARPA экспериментальной сети ARPANET для военных целей

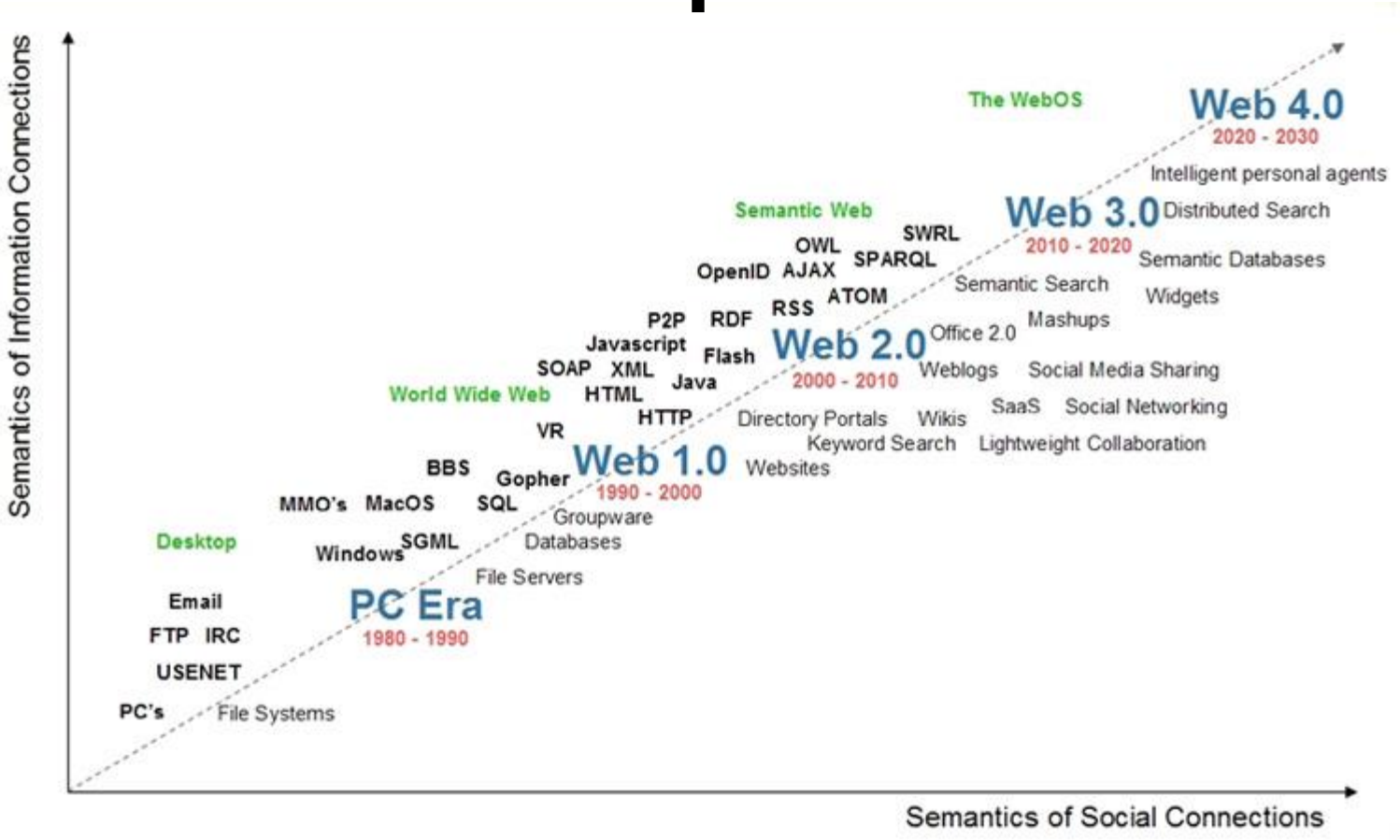
История Интернета

- 1969: сеанс связи ARPANET
- 1971: отправка первого Email
- 1973: интернационализация
- 1983: переход на TCP/IP
- 1984: запуск DNS
- 1989: появление WWW, HTTP, HTML
- 1990: первый браузер WorldWideWeb
- 1993: NCSA Mosaic
- 1995: Yahoo, Hotmail, Amazon
- 1998: Google, Mail.ru
- 2000: Yandex

Браузерные войны

- После появления NCSA Mosaic началась борьба браузеров за господство на рынке
- Положить конец которой был призван W3C

Эволюция Web

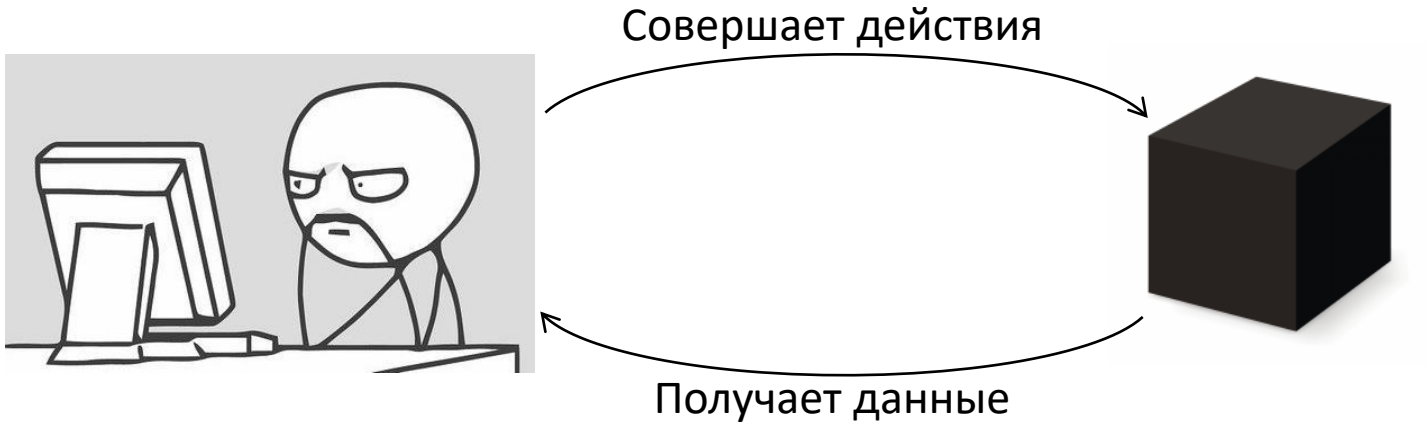


Web приложения

- Традиционные сайты: новости, блоги, wiki, визитки
- Глобальные приложения: почтовые сервисы, поиск, социальные сети
- E-commerce: магазины, бронирования, цифровая дистрибуция
- Замена десктоп-приложений: корпоративный софт, CRM, банковские клиенты

Как работают web-приложения

Пользователь через браузер



Какие действия?

- Переход по ссылке,
- Ввод адреса,
- Заполнение и отправка формы,
- Прокрутка,
- …

Какие данные получает?

- Обрабатываемые браузером:
  - Текст
  - Картинки
  - …
- Не обрабатываемые:
  - Файлы для скачивания

Больше деталей

- WWW – часть интернета, глобальной сети, в которую объединяются компьютеры для получения и отправки данных
- Сети:
  - Локальные (сетевой принтер, диск, игра по сетке)
  - Глобальные (интернет)
- Компьютеры объединяются в сеть множеством разных способов (google: Топология)

Клиент (Client)

- Участник (**узел**) сети
  - Компьютер, принтер, смартфон, …
- Запрашивает и получает данные
  - Иногда только получает
- Клиент просит и получает, а отдаёт кто?

Сервер (Server)

- Участник (**узел**) сети
  - Достаточно мощный компьютер
  - Возможно один из клиентов
  - А можно и вовсе без него
    google: «топологии сети», «точка-точка», «общая шина»
- Получает запросы клиентов
- Обрабатывает запросы
- Отправляет ответ клиенту

Вопросы

- Откуда клиент/сервер знают куда посылать запрос/ответ?
- Что они отправляют друг другу?
- Как сервер понимает какие данные генерировать для клиента?

Модель OSI

- Open Systems Interconnection basic reference model (ЭМВОС)
- 7 уровней взаимодействия компьютеров в сетях
  - Со своими форматами данных
  - Со своими правилами передачи (**протоколами**)

Сетевой протокол

- *Правила, действия, форматы, регламентирующие обмен данными между узлами сети*

Модель OSI

	Уровень (layer)	Тип данных (PDU)	Функции	Примеры
Host layers	7. Прикладной (application)		Доступ к сетевым службам	HTTP, FTP, SMTP, RDP, SNMP
	6. Представления (presentation)	Данные	Представление и шифрование данных	ASCII, EBCDIC, JPEG
	5. Сеансовый (session)		Управление сеансом связи	RPC, PAP
	4. Транспортный (transport)	Сегменты (segment)/ Дейтаграммы (datagram)	Прямая связь между конечными пунктами и надёжность	TCP, UDP, SCTP, PORTS
	3. Сетевой (network)	Пакеты (packet)	Определение маршрута и логическая адресация	IPv4, IPv6, IPsec, AppleTalk
Media layers	2. Канальный (data link)	Биты (bit)/ Кадры (frame)	Физическая адресация	PPP, IEEE 802.22, Ethernet, DSL, ARP, L2TP, Network Cards
	1. Физический (physical)	Биты (bit)	Работа со средой передачи, сигналами и двоичными данными	USB, витая пара, коаксиальный кабель, оптический кабель

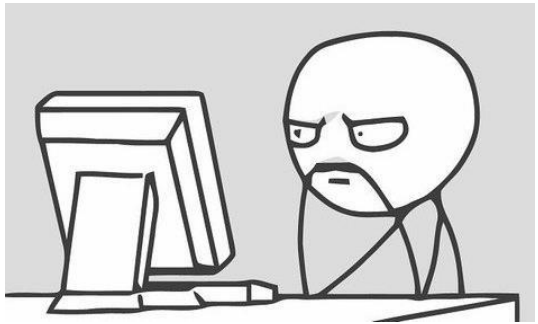
Протокол для web приложений

- Hyper Text Transfer Protocol (HTTP)
- 1991, Tim Berners-Lee, CERN
- 1996, HTTP/1.0
  https://tools.ietf.org/html/rfc1945
- 1999, HTTP/1.1
  https://tools.ietf.org/html/rfc2616
- 2015, HTTP/2
  https://tools.ietf.org/html/rfc7540

Модель OSI

	Уровень (layer)	Тип данных (PDU)	Функции	Примеры
	7. Прикладной (application)		Доступ к сетевым службам	**HTTP**, FTP, SMTP, RDP, SNMP
Host layers	6. Представления (presentation)	**Данные**	Представление и шифрование данных	ASCII, EBCDIC, JPEG
	5. Сеансовый (session)		Управление сеансом связи	RPC, PAP
	4. Транспортный (transport)	Сегменты (segment)/ Дейтаграммы (datagram)	Прямая связь между конечными пунктами и надёжность	TCP, UDP, SCTP, PORTS
	3. Сетевой (network)	Пакеты (packet)	Определение маршрута и логическая адресация	IPv4, IPv6, IPsec, AppleTalk
Media layers	2. Канальный (data link)	Биты (bit)/ Кадры (frame)	Физическая адресация	PPP, IEEE 802.22, Ethernet, DSL, ARP, L2TP, Network Cards
	1. Физический (physical)	Биты (bit)	Работа со средой передачи, сигналами и двоичными данными	USB, витая пара, коаксиальный кабель, оптический кабель

Уточняем схему



Клиент

Отправляет запрос (request)

Посылает ответ (response)

Сервер

Куда слать запрос/ответ?

- **URL (Uniform Resource Locator)**
  - Содержит информацию о местонахождении https://music.yandex.ru/artist/12674
- **URI (Uniform Resource Identifier)**
  - Может быть относительным путем /relative/URI/with/absolute/path/to/resource.txt
- Есть еще URN (google: URN)
- Нужно определять адрес узла по домену (3 уровень OSI) (google: DNS)

HTTP-request

```
GET /wiki/java 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
```

HTTP-request. Структура

- Стартовая строка
- Заголовки
- Тело

```
GET /wiki/java 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
```

HTTP методы

- Описывают действие
- Самые используемые:
  - GET
  - POST
- Другие:
  - PUT, PATCH, DELETE…

HTTP GET

- Самый используемый запрос
  - Ввод в адресную строку
  - Нажатие на ссылку
- Ничего не меняет на сервере (идемпотентный)

HTTP GET. Параметры

Хранятся в заголовке

https://market.yandex.ru/catalog/54553/list?hid=91012&rt=9&suggest_text=Servers&suggest=1&suggest_type=category&local-offers-first=0&deliveryincluded=0&onstock=1

- Host: market.yandex.ru
- Path: /catalog/54553/list
- Параметры: hid со значением 91012, rt со значением 9…

- Параметры запроса – основное средство передачи информации от клиента к серверу (вместе с URL)

HTTP POST

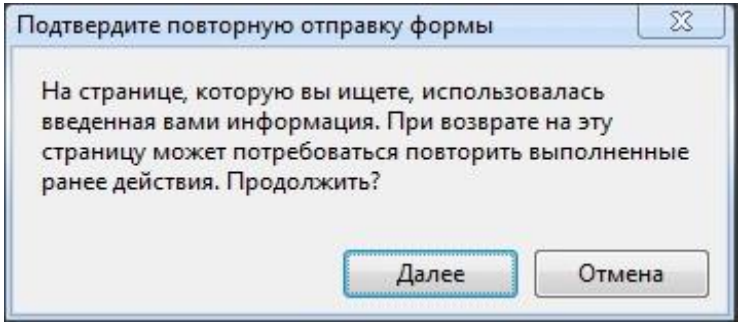
- Метод для изменения чего-то на сервере, обычно добавления
  - Требующего предварительной обработки
- Часто используется для отправки данных с веб-форм
- Хранит параметры в теле запроса
  - Поэтому пароли и другую конфиденциальную информацию посылаем POST-запросов

HTTP POST. Параметры

- **В отличие от GET-запроса, по URL нельзя понять, был отправлен POST-запрос или нет**
- GET-запрос на super.com с параметром id=1 возвратит страницу с адресом
  super.com/?id=1
- POST-запрос на super.com с параметром id=1 возвратит страницу с адресом
  super.com
- GET-запрос на super.com тоже возвратит страницу
  super.com

HTTP POST. Параметры

- Но в случае POST-запроса при попытке обновить страницу увидим сообщение



HTTP-response

Структура:

- Протокол КодОтвета
- Заголовки
- Содержимое

```
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
Далее реальное содержимое
```

HTTP-response. Код ответа

- **2xx Success**
  - 200 – OK
- **3xx Redirection**
  - 301 – Moved permanently
  - 302 – Found / Moved temporarily
- **4xx Client Error**
  - 403 – Forbidden
  - 404 – Not Found
  - 451 – Unavailable For Legal Reasons
- **5xx Server Error**
  - 500 – Internal Server Error
  - 502 – Bad Gateway
  - 503 – Service Unavailable

HTTP-response. Тип содержимого

https://tools.ietf.org/html/rfc2046

Multipurpose Internet Mail Extensions (MIME)

Базовые:

- application
- audio
- example
- image
- message
- model
- multipart
- text
- video

HTTP-response. Типы

Интересующие нас типы

- text/html
- text/css
- text/plain
- text/xml
- application/json
- application/javascript

Hyper Text Markup Language

- 1990, Tim Berners-Lee

Исправляет ошибки разработчика

http://htmlbook.ru/html5

  - ! Теги: html, body, head, title, a, h1-h6, ul, ol, li, div, img, br, p, form, input
  - ! Общие атрибуты тегов: class, id
  - ! Doctype + соответствие MIME-типа

Пример HTML

```
<!DOCTYPE html>
<html>
    <body>
        <h1>Some Heading</h1>
        <p>Some paragraph.</p>
    </body>
</html>
```

Смотрите исходный код веб-страниц

HTML. Формы

```
<form action="/login" method="?">
<input type="text" name="username"/>
<input type="password" name="password"/>
<input type="submit" value="OK"/>
</form>
```

Каким запросом отправлять форму?

HTML. Формы

Конечно, POST

<form action="/login" method=“POST">

Если отправить форму GET-запросом, в истории браузера сохранится:

…/login?user=username&password=qwerty

XML

- eXtensible Markup Language
- Формат хранения объектов
- Система *произвольных* тегов
- Строгий! Не прощает ошибок
  - google: XHTML

XML. Пример

```
<institute>
   <title>Высшая школа ИТИС</title>
   <address>
      <streetAddress>Кремлевская, д. 35</streetAddress>
      <city>Казань</city>
      <postalCode>420008</postalCode>
   </address>
   <phoneNumbers>
      <phoneNumber>(843) 221-34-33</phoneNumber>
      <phoneNumber>(843) 221-34-33 (доб.25)</phoneNumber>
   </phoneNumbers>
</institute>
```

Кстати, этот XML невалиден. Google: DTD for XML

XML. Пример 2

```
<institute>
   <title>Высшая школа ИТИС</title>
   <address streetAddress="Кремлевская, д. 35" city="Казань"
      postalCode="420008"/>
   <phoneNumbers>
      <phoneNumber>(843) 221-34-33</phoneNumber>
      <phoneNumber>(843) 221-34-33 (доб.25)</phoneNumber>
   </phoneNumbers>
</institute>
```

JSON

- Javascript Object Notation
  - Объект
  - Массив
  - Строка
  - Значение

JSON. Пример

```
{ "title": "Высшая школа ИТИС",
    "address": {
        "streetAddress": "Кремлевская, д.35",
        "city": "Казань",
        "postalCode": "420008"
    },
    "phoneNumbers": [
        "(843) 221-34-33",
        "(843) 221-34-33 (доб.25)"
    ]
}
```

Генерация содержимого

- Как сервер по запросу клиента определяет что нужно генерировать?

Статические сайты

- URI связан с файловой системой напрямую
- Пусть сайт расположен на сервере в папке srv/www/mysite

GET-запрос на www.mysite.ru/index.html

Вернёт файл /srv/www/mysupersite/index.html

Проблемы статических сайтов

- Одно содержимое для всех.
  - А интернет становится персональным

```
GET / HTTP/1.1
Host: vk.com
```

  - Для каждого из нас вернется разное содержимое. Но ведь запрос одинаковый!

- Как обновлять содержимое страниц?

Динамическая генерация

- Решение проблемы персонализации контента
- Содержимое ответа генерируется сервером при обработке запроса

PHP

- Hypertext Preprocessor
- 1994: Personal Home Page Tools, Rasmus Lerdorf

```
<html>
<head>
<title> Тестируем PHP </title>
</head>
<body>
        <?php echo 'Hello, world!'; ?>
</body>
</html>
```

PHP. Пример условия

```
<?php
if (strpos($_SERVER['HTTP_USER_AGENT'], 'MSIE') !== FALSE) {
?>
<h3>strpos(), должно быть, вернул не false</h3>
<p>Вы используете Internet Explorer</p>
<?php
} else {
?>
<h3>strpos() вернул false</h3>
<p>Вы не используете Internet Explorer</p>
<?php
}
?>
```

Web-фреймворки

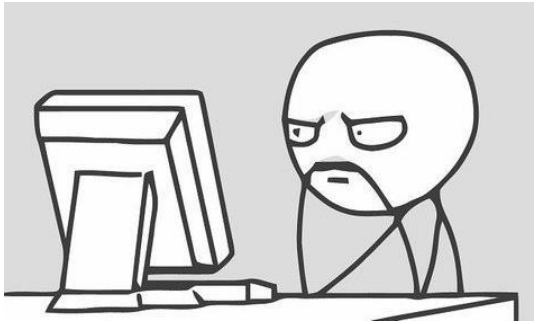
- Python
  - Django
- Ruby
  - Ruby on Rails
- PHP
  - Zend, Symphony
- C#
  - ASP.NET MVC

! google: framework

WEB MVC

- MODEL – данные
- VIEW – внешний вид (шаблоны)
- CONTROLLER – выполнение действий

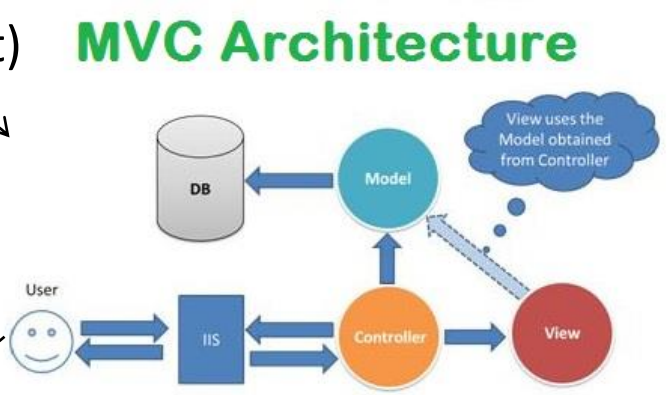
WEB MVC



Клиент

Отправляет запрос (request)

Посылает ответ (response)



Сервер

Почитать

- http://www.lib.ru/WEBMASTER/rfc2068/ (HTTP/1.1 на русском)
- http://htmlbook.ru/html5 (rus)
- http://www.w3schools.com/ (eng – html, css, js)

Вопросы?

e-mail: marchenko@it.kfu.ru

© Марченко Антон Александрович 2017 г.
Абрамский Михаил Михайлович