

# Введение. Веб как программная платформа

Кулаков Кирилл Александрович

# Структура курса

- Продолжительность
  - 15 часов лекций
  - 30 часов практических занятий
- Контроль успеваемости
  - Выполнение практических заданий
  - Мини-проект
- Помощь
  - Сайт курса (<https://cs.petrSU.ru/~kulakov/courses/web-arch/>)
  - График консультаций кафедры ИМО
  - Электронная почта ([kulakov@cs.petrSU.ru](mailto:kulakov@cs.petrSU.ru))

# О чем этот курс?

- основополагающие принципы работы и стандарты Веб.
- Архитектурные принципы и подходы к разработке серверной части (back-end) веб-приложений.
- Инструментальные средства поддержки процесса разработки веб-приложений.
- Современные проблемы разработки веб-приложений: обеспечение качества, безопасность, высокие нагрузки.

# Необходимые базовые навыки

- Web-технологии (HTML, CSS).
- Сетевые технологии (TCP, IP, DNS).
- Управление реляционными БД (SQL).
- Знание любого языка программирования (Python, Ruby, C, Java, JavaScript, PHP, Pascal, ...)
- Работа с командной оболочкой UNIX/Linux (Bash).

# Требования к мини-проекту

- Функционально простое веб-приложение. Предполагаемый набор функций должен быть предоставлен в виде краткой спецификации требований.
- Приложение должно быть разработано с использованием существующего веб-фреймворка (любой язык, любой фреймворк).
- Приложение должно использовать БД (классическую реляционную, NoSQL, ...)
- Доступ к приложению должен быть обеспечен посредством одного из веб-серверов общего назначения (Apache WebServer, Nginx, Microsoft IIS, OpenBSD httpd, lighttpd, ...)

# Требования к мини-проекту

- В приложении должно быть реализовано разделение доступа (аутентификация и авторизация).
- Существенная часть кода должна быть покрыта unit-тестами.
- Должны быть учтены базовые рекомендации по обеспечению информационной безопасности.

# История развития идей гипермедиа

- Идеи по созданию интерактивных источников информации:
  - Хорхе Луис Борхес — рассказ «Сад расходящихся тропок», 1941 г.
  - Вэнивар Буш — эссе «Как мы можем мыслить» (гипотетическая система «Мемекс»), 1945 г.
- Гипермедиа — нелинейный носитель информации, содержащий изображения, аудио, видео, простой текст и гиперссылки.

# История развития идей гипермедиа

- Цифровые технологии дают возможность реализовать накопленные идеи:
  - Тэд Нельсон: Project Xanadu, 1960–2016 гг. HES (Hypertext Editing System), 1967 г.
  - Дуглас Энгельбарт: NLS (oN-Line System), 1968 г. См. также The Mother of All Demos.
  - Apple Inc.: HyperCard, 1987 г.
  - Тим Бернерс-Ли: WorldWideWeb, средство простого обмена информацией для физиков CERN, 1990 г.



## Становление «Всемирной паутины»

*«Мне всего-лишь нужно было взять идею гипертекста, соединить ее с идеями TCP (Transmission Control Protocol) и доменной системы имен (DNS) и — та-да! — получилась Всемирная паутина (the World Wide Web)...»*

— Тим Бёрнерс-Ли

# Причины успеха проекта WorldWideWeb

- Благоприятные условия появления World Wide Web
  - наработки в области гипермедиа,
  - стандартизирован язык разметки SGML (Standard Generalized Markup Language),
  - широкое распространение набора протоколов TCP/IP.
- Простота и открытость технологии, а также свободно-распространяемая реализация Web-сервера от CERN делают World Wide Web «killer feature» сети Интернет.
- Основное применение: обмен информацией, знаниями и ресурсами в академическом сообществе.

# Базовые компоненты технологии Веб

- Язык разметки для форматирования гипертекстовых документов. Простой текст с добавленной в него информацией о способах его представления и строении.
  - На основе SGML был разработан более простой язык HTML.
- Унифицированная нотация для адресации ресурсов, доступных по сети.

Концепция URI (Uniform Resource Identifier):

`scheme://host[:port]/path/.../[;url-params][?query-string]  
[#anchor]`

# Компоненты URI

`scheme://host[:port]/path/. . . /[?query-string][#anchor]`

- `scheme` — схема обращения к ресурсу, обычно выражается в виде сетевого протокола, по которому доступен ресурс (`http`, `file`, `mailto` и др).
- `host` — идентификатор компьютера, на котором находится ресурс (может быть задан в виде IP адреса или доменного имени).
- `port` — номер порта, по которому слушаются запросы на ресурс.
- `path` — путь до ресурса в иерархии директорий на компьютере, на котором находится ресурс.
- `query-string` — параметры запроса к ресурсу. Задаются в формате название параметра = значение параметра и разделены знаком `&`.
- `anchor` — ссылка на часть документа. В документе могут быть установлены

# Базовые компоненты технологии Веб

- Протокол для транспортировки гипертекстовых документов по сети.

Протокол прикладного уровня HTTP (HyperText Transfer Protocol), работающий поверх протокола TCP:

- взаимодействие в форме запрос/ответ,
- формат сообщений: заголовки + пустая строка + тело,
- протокол без сохранения состояния.

## От веб-сайтов к веб-приложениям

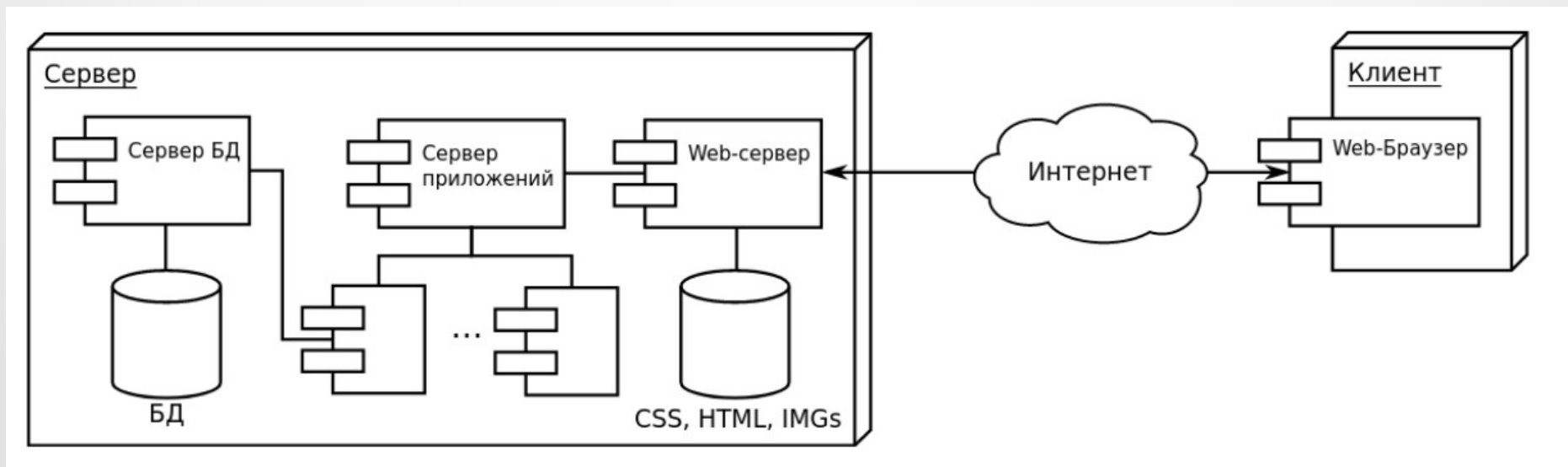
- Отдельные веб-страницы размещались на веб-серверах больших исследовательских и образовательных организаций.
- С ростом сети Интернет у большего количества пользователей появляется возможность разворачивать свои веб-серверы, появляются веб-сайты — наборы веб-страниц с единой тематикой и оформлением.

# От веб-сайтов к веб-приложениям

- Переход от понимания Web как системы обмена статическими ресурсами к Web как динамическому информационному сервису:
  - поисковые сервисы (сначала преимущественно по FTP и Gopher),
  - информация о погоде,
  - сколько воды осталось в автомате с газировкой,
  - телефонный справочник (одно из первых Web-приложений, продемонстрированных Т. Бернерсом-Ли в CERN).
- К середине 90-х Интернет получает все большее распространение среди обычных людей. Бизнес понимает, что Web — это новая платформа для продвижения своих товаров и услуг.

# Классическая высокоуровневая архитектура веб-приложения

- Web-приложение — клиент/серверное приложение, использующее Веб-браузер как реализацию программы-клиента и предоставляющее некоторый интерактивный сервис, взаимодействуя с одним или несколькими Веб-серверами в Интернет (или локальной сети).





# Особенности клиент-серверной архитектуры

- Клиент-серверная архитектура — одна из моделей организации распределенных приложений.
- Преимущества:
  - Централизация данных (проще обеспечить информационную безопасность)
  - Единая реализация приложения-сервера (нет необходимости учитывать особенности пользовательской платформы)
  - Независимая и простая реализация приложений-клиентов (проще разработать кросс-платформенное решение)
  - Менее требовательная к ресурсам реализация приложений-клиентов

# Особенности клиент-серверной архитектуры

- Недостатки и ограничения:
  - Сервер — единая точка отказа
  - Высокая зависимость от характеристик сети
  - Невозможность обеспечения согласованности данных, доступности и устойчивости к разделению одновременно (Теорема CAP)
  - Необходимость учета большого числа внешних негативных факторов при разработке приложений (Fallacies of distributed computing)

# «Обогащение» клиента

- 1992 г., ViolaWWW — один из первых графических браузеров (для X Window System).
  - первая реализация скриптового языка, встраиваемого в HTML-страницу.
- 1993 г., NSCA Mosaic — первый веб-браузер в современном понимании этого слова. World Wide Web становится интересен (и доступен) простым пользователям.
  - простой интерфейс
  - закладки, печать документа,
  - изображения встраиваются в страницу, а не показываются иконкой.

# «Обогащение» клиента

- 1995 г., первая реализация языка JavaScript в браузере Netscape Navigator.
  - изначальная идея — встроить Scheme в браузер.
  - результат: динамическая типизация, синтаксис языка C, семантика на основе идей из языков Self и Scheme.
  - Выдержка из пресс-релиза: «A multimedia weather forecast applet written in Java can be scripted by JavaScript to display appropriate images and sounds based on the current weather readings in a region».
- 1995 г., начало борьбы за господство на рынке браузеров, также известной как «Война браузеров»: Netscape vs. Microsoft.
  - Идея Netscape — Веб как распределенная операционная система (webtop vs. desktop).

# Расширение набора интерактивных возможностей на стороне клиента

- Тэг <object>, Java-апплеты, ActiveX, Adobe (Macromedia) Flash.
- DHTML (Dynamic HTML) — подход к разработке интерактивных клиентских приложений: HTML + JavaScript + CSS + DOM.

– Пример использования DHTML.

```
var element =  
document.getElementById("square");  
element.style.background = "blue";  
element.innerHTML = "Test";  
element.style.left = (parseInt(element.style.left)  
|| 0) + 20 + "px";
```



```
var element = document.getElementById("square");  
element.style.background = "blue";  
element.innerHTML = "Test";  
element.style.left = (parseInt(element.style.left) || 0) + 20 + "px";
```

Try

# Расширение набора интерактивных возможностей на стороне клиента

- AJAX (Asynchronous JavaScript And XML) — подход к разработке асинхронных Веб-приложений: дополнительные данные могут запрашиваться от сервера и отображаться в рамках текущей страницы без перезагрузки всей страницы.

- Пример использования AJAX.

```
if (window.XMLHttpRequest) { httpRequest = new XMLHttpRequest(); // Mozilla, Safari, ...  
} else if (window.ActiveXObject) { // IE 8 and older  
    httpRequest = new ActiveXObject("Microsoft.XMLHTTP"); }  
httpRequest.onreadystatechange = responseReadyCallback;  
httpRequest.open('GET', 'http://www.cs.petrSU.ru/~vadim/st/' + group + '.txt');  
httpRequest.send();
```

Список группы

```
1,22505,Балакирева Мария Сергеевна,mbalakir@cs.petrSU.ru  
2,22505,Беляков Сергей Ильич,belyakov@cs.petrSU.ru  
3,22505,Бурдин Григорий Олегович,burдин@cs.petrSU.ru  
4,22505,Вилаева Алина Романовна,vilaeva@cs.petrSU.ru  
5,22505,Куренчук Сергей Сергеевич,kurenchu@cs.petrSU.ru  
6,22505,Ларионов Дмитрий Дмитриевич,larionov@cs.petrSU.ru  
7,22505,Лукашов Тимофей Сергеевич,lukashov@cs.petrSU.ru  
8,22505,Меньшикова Екатерина Андреевна,menshiko@cs.petrSU.ru  
9,22505,Пашкин Данил Александрович,pashkin@cs.petrSU.ru  
10,22505,Потес Артем Сергеевич,potes@cs.petrSU.ru  
11,22505,Радивилов Артем Борисович,radivilo@cs.petrSU.ru  
12,22505,Сысоев Максим Федорович,msysoev@cs.petrSU.ru  
13,22505,Шилович Денис Вячеславович,shilovic@cs.petrSU.ru
```

# HTML5

- HTML5 — стандартизирует большое количество возможностей браузера, позволяющих использовать его как платформу для создания интерактивных клиент-серверных приложений.
  - тэги `<canvas>`, `<video>`, `<audio>`;
  - обработка ошибок при обработке HTML-документов;
  - разнообразные API и DOM являются частью стандарта HTML5;
  - рабочая группа WHATWG рассматривает HTML5 как живой стандарт (Living Standard).

# Высокоуровневая архитектура современных веб-приложений

- Браузер — программная платформа, позволяющая не просто отображать представления, сгенерированные на стороне сервера, но и брать на себя всю работу по взаимодействию с пользователем на клиентской стороне.

