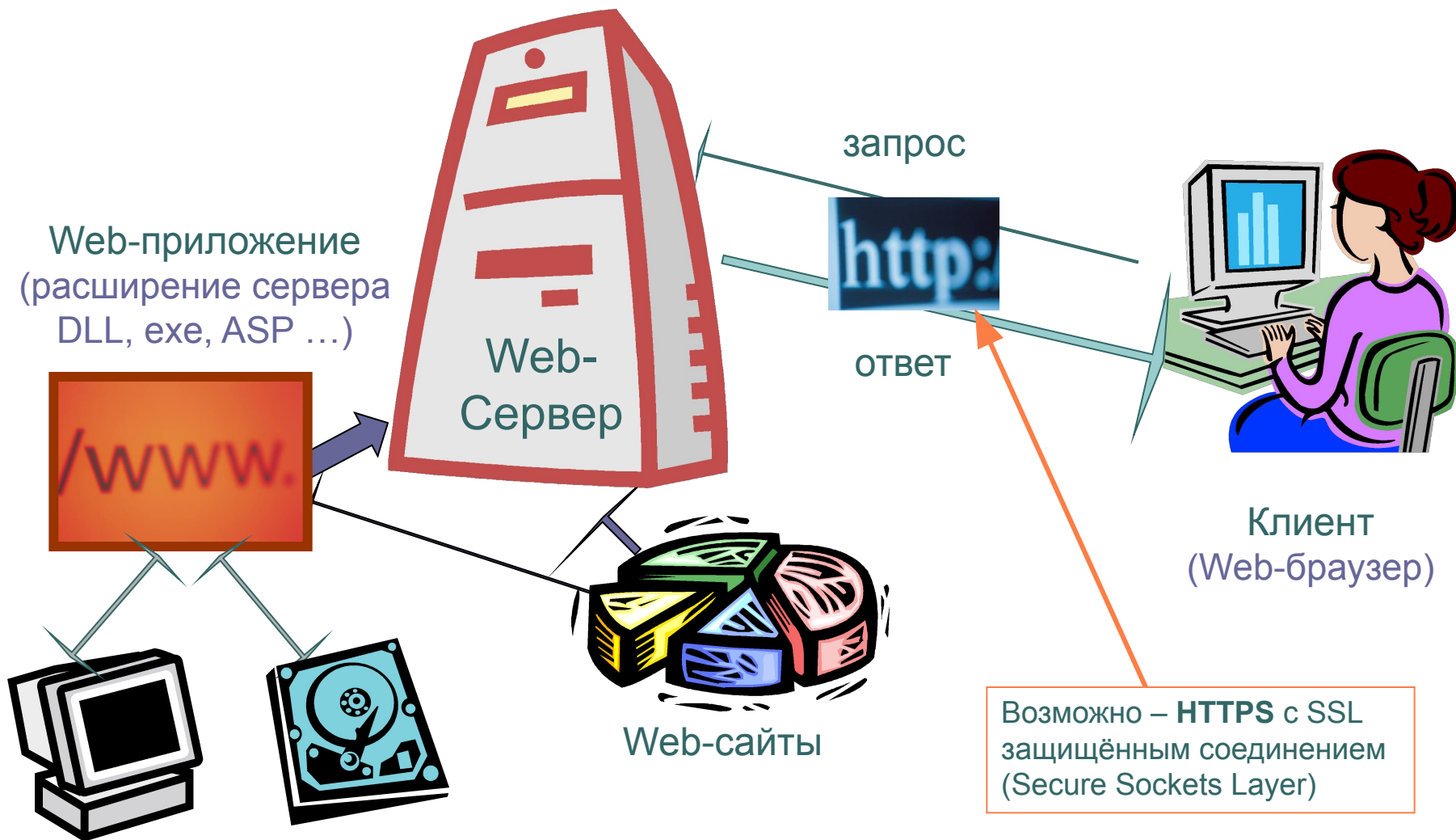




Технологии разработки **Internet-** приложений

Введение в Web-технологии

Взаимодействие Клиент – Web-сервер



Эволюция – Web-серверов

- HTML** 1. Возврат клиенту существующих статических HTML-страниц. 1990
Через несколько лет добавлена технология **SSI** (Server Side Include).
- exe** 2. Добавлен общий интерфейс **CGI** (Common Gateway Interface) для запуска внешних приложений (*расширений*) и обменом данными через STDIN и STDOUT.
- DLL** 3. Добавлен интерфейс **ISAPI** (MS Internet Server API для IIS), Apache Shared Module... для работы *расширений* в одном адресном пространстве с сервером (часть процесса Web-сервера). Не надо ничего дополнительно запускать, обмениваться данными через файлы или STDIN/STDOUT. Теоретически они готовы одновременно обрабатывать любое число запросов. При ошибках в таких расширениях может «вылететь» весь Web-сервер.



Скриптовые технологии **ASP, JSP, PHP** – 1997

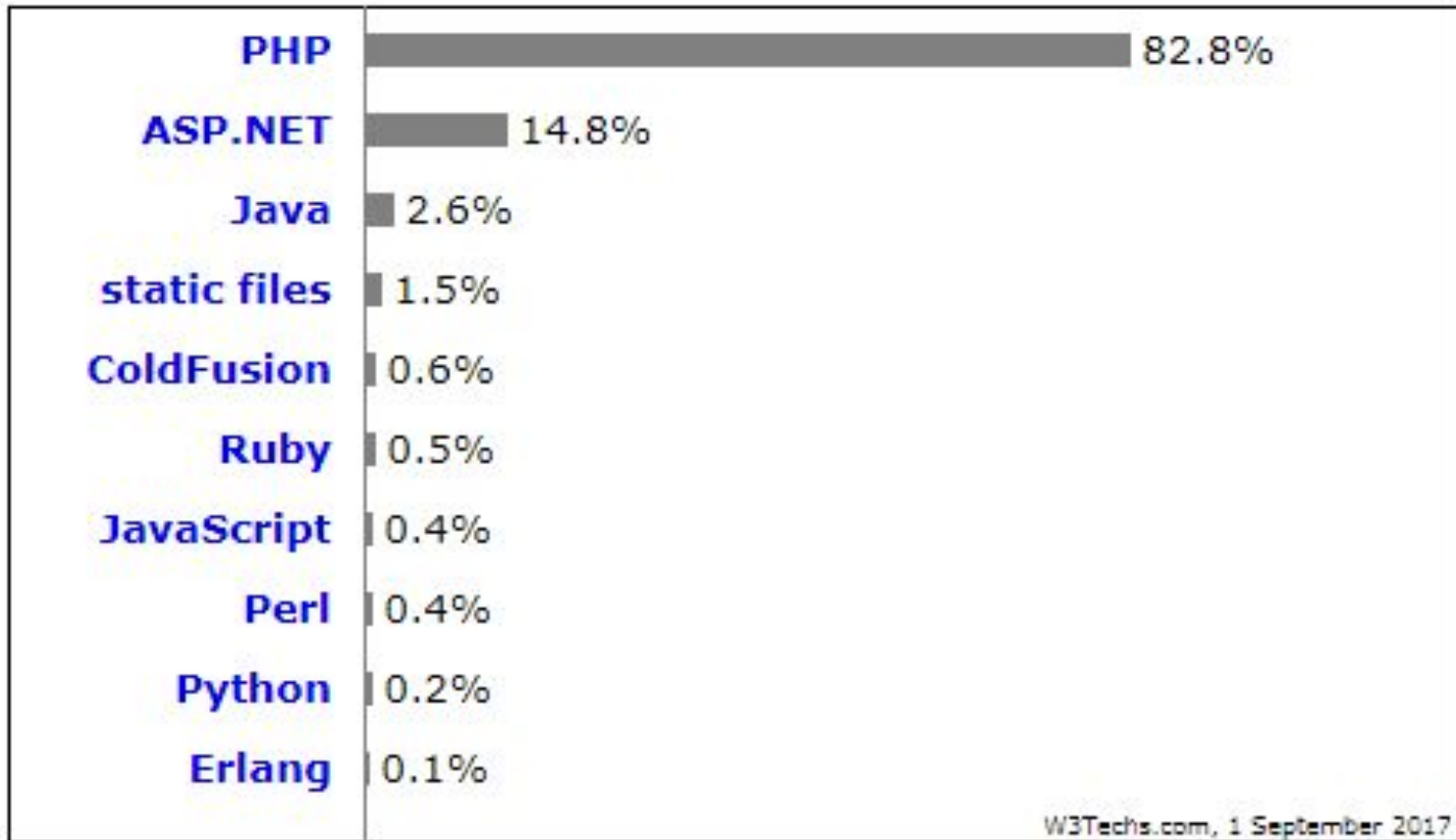
MS **.NET** –технологии (конкуренция **Java**):

ASP.NET	FORMS	2005
	MVC	2010
	CORE	2016

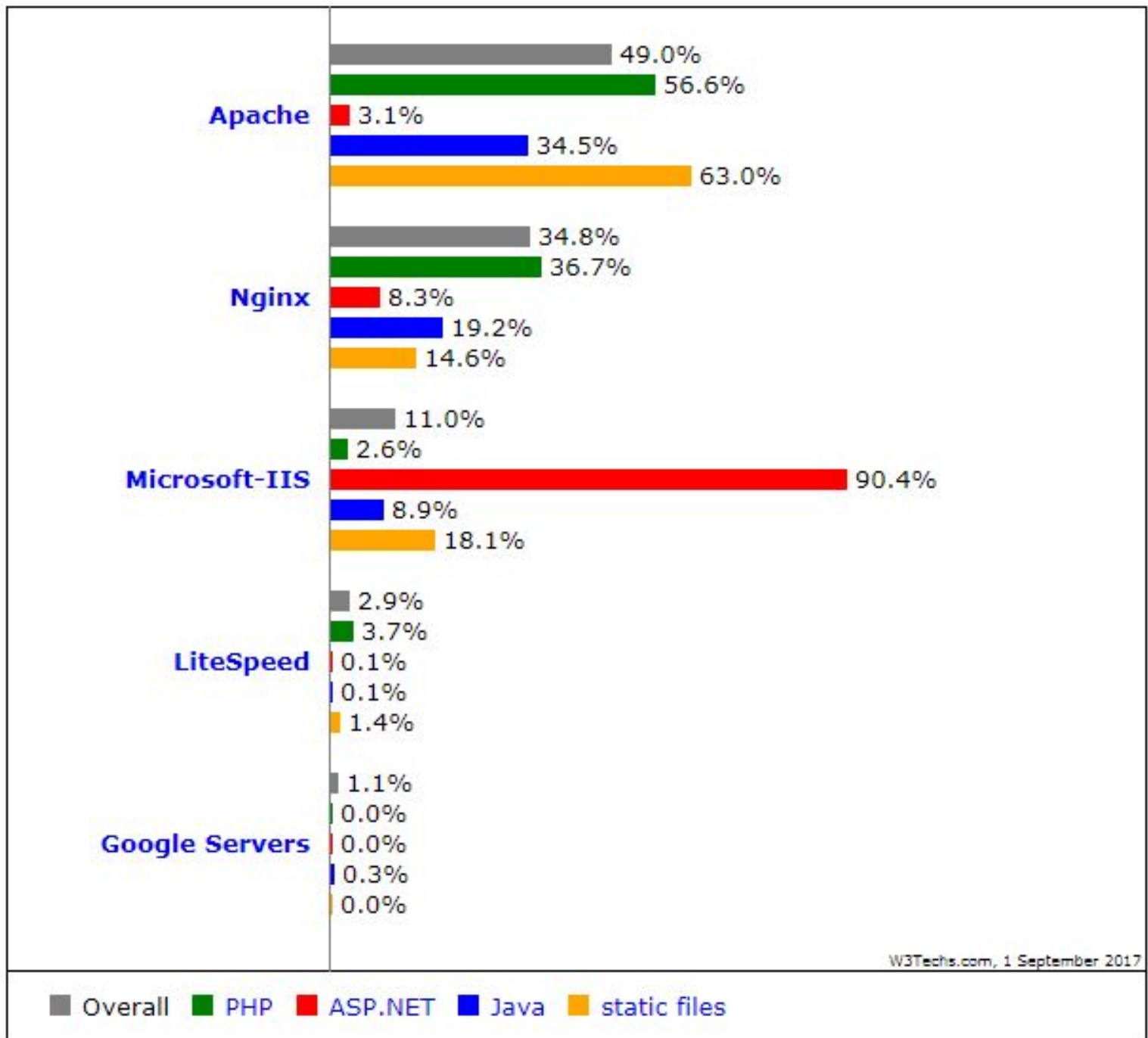
Usage of server-side programming languages for websites

W³Techs

Web Technology Surveys



We count the top 10 million websites



Стандарт URL

протокол://пользователь:пароль@сервер:порт/путь/файл/путь в файле?параметры

Например,

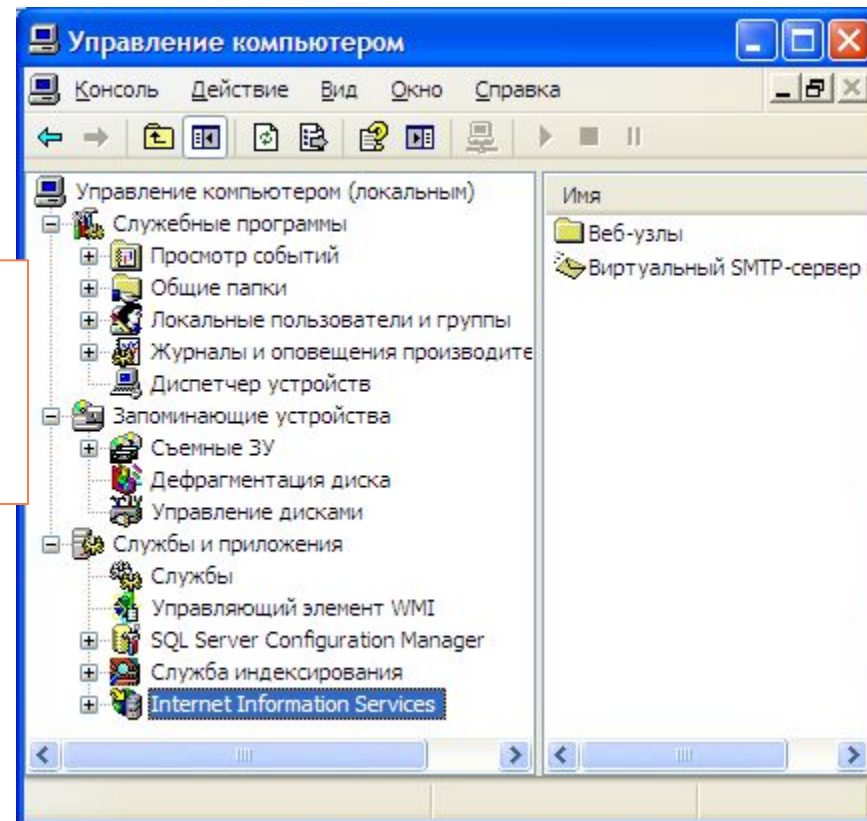
<http://vova:12345@www.donnu.edu.ua:81/hotdog/Scripts/x.exe/go?n1=20&n2=55>

Локальный сервер IIS

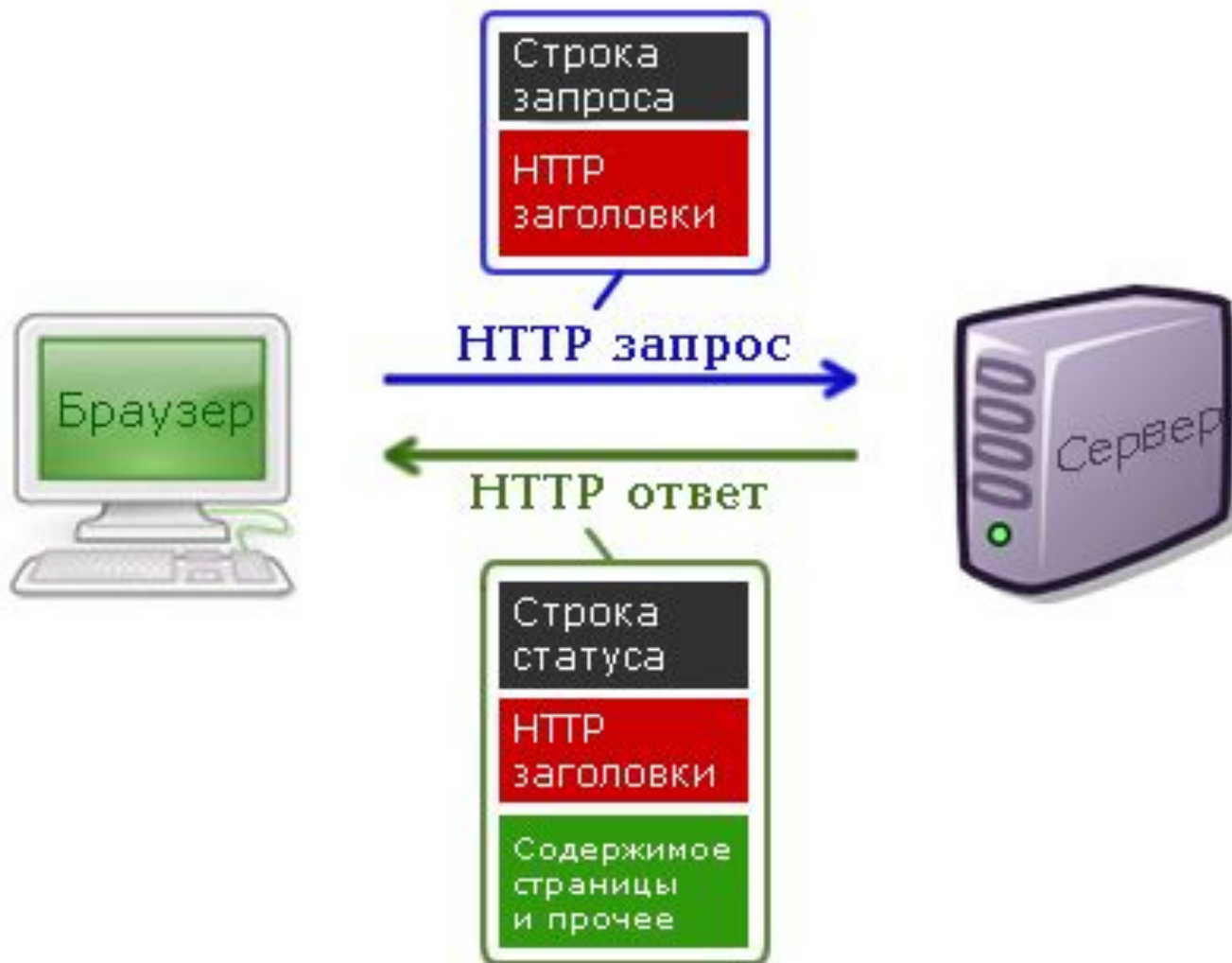


C:\inetpub\wwwroot

<http://localhost>



HTTP



HTTP – примеры запросов клиента

Пример GET-запроса

GET http://oak.oakland.edu/ HTTP/1.0?name1=value&name2=value2
Connection: Keep-Alive
User-Agent: Mozilla/4.04 [en] (Win95; I)
Host: oak.oakland.edu
Accept: image/gif, image/x-bitmap, image/jpeg, image/jpg, image/png, */*

Строка запроса

Параметры запроса – 2КБ

Связь закрывается если кто-то сообщит Connection: close

По стандарту MIME (см. далее)

Пример POST-запроса

POST http://oak.oakland.edu/ HTTP/1.0
Connection: Keep-Alive
User-Agent: Mozilla/4.04 [en] (Win95; I)
Host: oak.oakland.edu
Accept: image/gif, image/x-bitmap, image/jpeg, image/jpg, image/png, */*

Строка запроса

name1=value&name2=value2

Тело запроса

Параметры запроса – 8 МБ

Заголовки



MIME:

(Multipurpose
Internet Mail
Extensions)

Тип/подтип	Расширение	Описание файла
application/pdf	.pdf	Документ, предназначенный для обработки Acrobat Reader
application/msexcel	.xls	Документ в формате Microsoft Excel
application/postscript	.ps, .eps	Документ в формате PostScript
application/x-tex	.tex	Документ в формате TeX
application/msword	.doc	Документ в формате Microsoft Word
application/rtf	.rtf	Документ в формате RTF, для MS Word
image/gif	.gif	Изображение в формате GIF
image/jpeg	.jpeg, .jpg, .jpe	Изображение в формате JPF
image/tiff	.tiff, .tif	Изображение в формате TIFF
image/x-xbitmap	.xbm	Изображение в формате XBitmap
text/plain	.txt	ASCII-текст
text/html	.html, .htm	Документ в формате HTML
audio/midi	.midi, .mid	Аудиофайл в формате MIDI
audio/x-wav	.wav	Аудиофайл в формате WAV
message/rfc822		Почтовое сообщение
message/news		Сообщение в группы новостей
video/mpeg	.mpeg, .mpg, .mpe	Видеофрагмент в формате MPEG
video/avi	.avi	Видеофрагмент в формате AVI

Код возврата строки статуса

Код возврата характеризует успешность выполнения операции на сервере. Все коды возврата разделены на несколько классов – 1xx, 2xx, 3xx, 4xx и 5xx.

Подробнее см. [Коды статуса HTTP](#)

Класс кода возврата	Описание	Пример
1xx	Информационные коды возврата. Обычно сообщения с такими кодами игнорируются.	
2xx	Запрос выполнен успешно.	200 – ОК
3xx	Перенаправление. Означают, что для успешного выполнения операции клиенту необходимо выполнить еще один запрос по другому адресу.	302 – ресурс перемещен или временно удален
4xx	Ошибка клиента.	403 – обращение к запрещенному ресурсу 404 – обращение к несуществующему ресурсу
5xx	Ошибка сервера. Код на сервере отработал некорректно и привел к сбою на сервере.	500 – ошибка на сервере.

[Пример](#) сеанса браузера и сервера



HTTP 1 и 2



HTTP/2 разрабатывался для транспортировки контента с низким временем задержки.

1. http/2 – это бинарный протокол вместо текстового. Бинарные сообщения быстрее разбираются автоматически, но, в отличие от HTTP/1.x, не удобны для чтения человеком.
2. Мультиплексирование потоков – передача нескольких асинхронных HTTP-запросов по одному TCP-соединению. Основное время связи клиент-сервер занимает создание TCP-соединения. В HTTP/1 каждый элемент (файл) страницы загружается в новом TCP-соединении. Могут осуществляться и несколько параллельных TCP-соединений.
3. Сжатие заголовков (включая куки) методом HPACK.
4. Server Push — несколько ответов на один запрос. То есть сервер может «предвидеть» новые запросы и заранее загружать ресурсы клиенту без его непосредственного запроса.
5. Приоритизация запросов. Клиент может назначить приоритет потоку.
6. Безопасность – встроен протокол **TLS** (*Transport Layer Security*), вместо предшественника SSL (*Secure Sockets Layer*). Поверх него работает HTTP/2.

