Основы веб-технологий

1. W3С консорциум. Цели и задачи W3C
2. Инженерный совет Интернета IETF
3. Сеть Интернет и протоколы
4. RFC и стандартизация
5. Стек протоколов TCP/IP, TCP-соединение
6. Системы адресации. IPv4. IPv6
7. Система доменных имен DNS. Рекурсивная и нерекурсивная схемы. Структура полного доменного имени
8. Взаимодействие по протоколу HTTP, протокол HTTPS, HTTP-запрос
9. Методы HTTP запросов. Основные статусы. Содержание запросов и ответов
10. Принципы маршрутизации данных в Интернете. BGP
11. Принципы маршрутизации данных в Интернете. OSPF
12. Прокси-сервер: использование, виды
13. Веб-сервера. Статический и Динамический контент
14. Сеть доставки содержимого (CDN)
15. Современные веб-браузеры. Принципы работы
16. Хранение данных на клиенте. Cookie. Недостатки Cookie
17. Хранение данных на клиенте. Технология Web Storage. Преимущества веб-хранилищ
18. Особенности хранения данных на стороне клиента. LocalStorage и SessionStorage. Сохранение, извлечение и удаление данных

Верстка и программирование

1. Комплексные селекторы. Примеры
2. Псевдоклассы и псевдоэлементы CSS. Примеры
3. Адаптивный веб-дизайн (RWD). Медиа-запросы (media queries). Формула ширины элемента. Размер текста
4. Доступность веб-страниц (accessibility). Основные принципы
5. CSS-фреймворки (библиотеки), преимущества и недостатки
6. CSS-препроцессоры. Преимущества и недостатки
7. Методология БЭМ. Базовые принципы и именование классов
8. Особенности формата SVG. Плюсы и минусы векторной графики
9. Документ и объекты страницы. Окружение: DOM, BOM
10. Регулярные выражения. Примеры
11. Обработка событий в JavaScript
12. Поток документа. Скрытие элементов. Позиционирование элементов
13. Возможности инспектора элементов в браузерах, FireBug/WebInspector
14. JS-фреймворки (библиотеки), преимущества и недостатки
15. Формат JSON
16. Программный интерфейс приложения API
17. Работа в редакторе Vim. Демонстрация
18. Базовые команды оболочки Bash/PowerShell. Примеры
19. Принцип работы системы контроля версий Git
20. Основные команды Git. Примеры

1) W3С консорциум. Цели и задачи W3C

W3C — это международная. независимая, некоммерческая организация, созданная для разработки и внедрения единых стандартов работы сети Интернет.

W3C разрабатывает для Интернета единые принципы и стандарты (называемые «Рекоменда́циями», англ. W3C Recommendations), которые затем внедряются производителями программ и оборудования. Таким образом достигается совместимость между программными продуктами и аппаратурой различных компаний, что делает Всемирную сеть более совершенной, универсальной и удобной.

Миссия W3C: «Полностью раскрыть потенциал Всемирной паутины, путём создания протоколов и принципов, гарантирующих долгосрочное развитие Сети».

Основная цель стандартов W3C – создание так называемого единого интерфейса пользователя и совмести проектов всех разработчиков, работающих независимо.

Задачи:

· Интернационализация Сети и стремление сделать интернет максимально удобным для населения всего земного шара.

· Обеспечение удобства использования интернета людьми с ограниченными возможностями.

· обеспечение доступности ресурсов интернета для людей с ограниченными физическими возможностями.

· доступность информации на всех языках мира, а не только на английском.

· программы должны иметь несколько форм и режимов взаимодействия с пользователем.

· ресурсы Всемирной сети должны быть доступны для мобильных устройств.

2)Инженерный совет Интернета IETF

**IETF** ([англ.](https://ru.bmstu.wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) ***I****nternet****E****ngineering****T****ask****F****orce*) - организация (открытое международное сообщество), куда входят специалисты различного профиля. Занимаются развитием сети Интернет - новыми протоколами и вопросами архитектуры, в частности выпускают стандарты **RFC**.

Рабочее предложение (англ. **Request** for **Comments**, **RFC**) — документ из серии пронумерованных информационных документов Интернета, содержащих технические спецификации и стандарты, широко применяемые во всемирной сети.

Задачи IETF (в соответствии с [RFC 4677](https://tools.ietf.org/html/rfc4677)):

* Идентификация проблем и предложение решений в технических аспектах организации Интернета;
* Разработка спецификаций, стандартов и соглашений по общим архитектурным принципам протоколов Интернета;
* Вынесение рекомендаций относительно стандартизации протоколов на рассмотрение Internet Engineering Steering Group ([IESG](https://ru.wikipedia.org/wiki/IESG));
* Содействие широкому распространению технологий и стандартов, разрабатываемых в Internet Research Task Force ([IRTF](https://ru.wikipedia.org/wiki/IRTF));
* Организация дискуссии для обмена информации в сообществе Интернета между учёными, разработчиками, пользователями, производителями оборудования и услуг, сетевыми администраторами и т. д.

3) Сеть Интернет и протоколы

Сеть Интернет – это глобальная сеть, соединяющая сети различного размера по всему миру. Сеть Интернет – это информационное пространство, содержащее огромное количество информации, хранилище информационных ресурсов. Информационными ресурсами являются совокупности текстов, изображений и других данных, а также тематические связи между ними.

Сетевой протокол - **это** набор правил, позволяющий осуществлять соединение и обмен данными между двумя и более включёнными в сеть компьютерами.Фактически разные **протоколы** зачастую описывают лишь разные стороны одного типа связи; взятые вместе, они образуют так называемый стек протоколов.

MAC или (Media Access Control) - это протокол низкого уровня, который используется для идентификации устройств в локальной сети. У каждого устройства, подключенного к сети есть уникальный MAC адрес, заданный производителем. В локальных сетях, а все данные выходят из локальной сети и попадают в локальную сеть перед тем, как попасть к получателю, используются физические MAC адреса для обозначения устройств

IP ( Internet Protocol) - расположен уровнем выше, за MAC. Он отвечает за определение IP адресов, которые будут уникальными для каждого устройства и позволяют компьютерам находить друг друга в сети. Он относится к сетевому уровню модели TCP/IP. Сети могут быть связанны друг с другом в сложные структуры, с помощью этого протокола компьютеры могут определить несколько возможных путей к целевому устройству, причем во время работы эти пути могут меняться. Есть несколько реализаций протокола, но наиболее популярной на сегодняшний день является IPv4 и IPv6.

ICMP (Internet control message protocol) - используется для обмена сообщениями между устройствами. Это могут быть сообщения об ошибках или информационные сообщения, но он не предназначен для передачи данных. Такие пакеты используются в таких диагностических инструментах, как ping и traceroute. Этот протокол находится выше протокола IP;

TCP (Transmission control protocol) - это еще один основной сетевой протокол, который находится на том же уровне, что и ICMP. Его задача - управление передачей данных. Сети ненадежны. Из-за большого количества путей пакеты могут приходить не в том порядке или даже теряться. TCP гарантирует, что пакеты будут приняты в правильном порядке, а также позволяет исправить ошибки передачи пакетов. Информация приводится к правильному порядку, а уже затем передается приложению. Перед передачей данных создается соединение с помощью так называемого алгоритма тройного рукопожатия. Он предусматривает отправку запроса и подтверждение открытия соединения двумя компьютерами.

UDP (user datagram protocol) - это популярный протокол, похожий на TCP, который тоже работает на транспортном уровне. Отличие между ними в том, что здесь используется ненадежная передача данных. Данные не проверяются при получении, это может выглядеть плохой идеей, но во многих случаях этого вполне достаточно. Поскольку нужно отправлять меньше пакетов, UDP работает быстрее, чем TCP. Поскольку

соединение устанавливать не нужно, то этот протокол может использоваться для отправки пакетов сразу на несколько машин или IP телефонии.

HTTP (hypertext transfer protocol) - это протокол уровня приложения, который лежит в основе работы всех сайтов интернета. HTTP позволяет запрашивать определенные ресурсы у удаленной системы, например, веб страницы, и файлы;

FTP (file transfer protocol) - это протокол передачи файлов. Он работает на уровне приложений и обеспечивает передачу файла от одного компьютера к другому. FTP - не безопасный, поэтому не рекомендуется его применять для личных данных;

DNS (domain name system) - протокол того же уровня, используемый для преобразования понятных и легко читаемых адресов в сложные ip адреса, которые трудно запомнить и наоборот. Благодаря ему мы можем получить доступ к сайту по его доменному имени;

SSH (secure shell) - протокол уровня приложений, реализованный для обеспечения удаленного управления системой по защищенному каналу. Многие дополнительные технологии используют этот протокол для своей работы.

4) RFC и стандартизация

**“Тема для обсуждения”, “запрос на отзывы” (англ. *Request for Comments, RFC*) — документ из серии пронумерованных информационных документов Интернета, содержащих технические спецификации и стандарты, широко**

применяемые во всемирной сети. В настоящее время первичной публикацией документов RFC занимается IETF под эгидой открытой организации Общество Интернета (англ. *Internet Society, ISOC*).

https://ru.wikipedia.org/wiki/RFC

Несмотря на название, запросы на отзывы RFC сейчас рассматриваются как **стандарты Интернета**(а рабочие версии стандартов обычно называют *драфтами*, от англ. *draft*— проект). Согласно RFC 2026, жизненный цикл стандарта

выглядит следующим образом:

1.Выносится на всеобщее рассмотрение **интернет-проект**(*Internet Draft*). Проекты не имеют официального статуса и удаляются из базы через шесть месяцев после последнего изменения.

2.Если проект стандарта оказывается достаточно удачным и непротиворечивым, он получает статус **предложенного стандарта**(*Proposed Standard*), и свой номер RFC.

3.Следующая стадия — **проект стандарта**(*Draft Standard*) — означает, что предложенный стандарт принят сообществом, в частности, существуют две независимые по коду совместимые реализации разных команд разработчиков.

4.Высший уровень — **стандарт Интернета**(*Internet Standard*). Это спецификации с большим успешным опытом применения и зрелой формулировкой. Параллельно с нумерацией RFC они имеют свою собственную нумерацию STD.

Рабочее предложение (англ. **Request** for **Comments**, **RFC**) — документ из серии пронумерованных информационных документов Интернета, содержащих технические спецификации и стандарты, широко применяемые во всемирной сети.

Запросы на отзывы RFC сейчас рассматриваются как стандарты Интернета (а рабочие версии стандартов обычно называют драфтами). Согласно RFC 2026, жизненный цикл стандарта выглядит следующим образом:

Выносится на всеобщее рассмотрение интернет-проект (Internet Draft). Проекты не имеют официального статуса и удаляются из базы через шесть месяцев после последнего изменения.

Если проект стандарта оказывается достаточно удачным и непротиворечивым, он получает статус предложенного стандарта (Proposed Standard), и свой номер RFC. Наличие программной реализации стандарта желательно, но не обязательно.

Следующая стадия — проект стандарта (Draft Standard) — означает, что предложенный стандарт принят сообществом, в частности, существуют две независимые по коду совместимые реализации разных команд разработчиков. В проекты стандартов ещё могут вноситься мелкие правки, но они считаются достаточно стабильными и рекомендуются для реализации.

Высший уровень — стандарт Интернета (Internet Standard). Это спецификации с большим успешным опытом применения и зрелой формулировкой. Параллельно с нумерацией RFC они имеют свою собственную нумерацию STD. Список стандартов имеется в документе STD 1 (сейчас это RFC 5000, но нумерация может измениться). Из более чем трёх тысяч RFC этого уровня достигли только несколько десятков.

5) Стек протоколов TCP/IP, TCP-соединение

Стек TCP/IP – это набор иерархически упорядоченных сетевых протоколов. Название стек получил по двум важнейшим протоколам – TCP (Transmission Control Protocol) и IP (Internet Protocol). Помимо них в стек входят ещё несколько десятков различных протоколов. В настоящее время протоколы TCP/IP являются основными для Интернета, а также для большинства корпоративных и локальных сетей.

Стек протоколов TCP/IP обладает двумя важными свойствами:

· платформонезависимостью, т. е. возможна его реализация на самых разных операционных системах и процессорах;

· открытостью, т. е. стандарты, по которым строится стек TCP/IP, доступны любому желающему.

Набор протоколов TCP/IP — ведущий стандарт передачи данных по сетям и через Интернет. Он состоит из уровней, выполняющих необходимые функции по подготовке данных для передачи по сети.

1. Уровень приложений – работают протоколы верхнего уровня (HTTP, Telnet, FTP, SMTP, DNS и HTML);
2. Транспортный уровень – взаимодействие приложений через отправку ими запросов на указанные порты (TCP и UDP);
3. Уровень межсетевого взаимодействия – обеспечение IP-адресации и маршрутизации (IP и ICMP);
4. Уровень сетевого доступа – описание МАС-адресации и физических компонентов сети (CSMA/CD и 10BASE-T).

Протокол TCP повторно передает потерянные, удаляет дублированные и гарантирует правильные порядок передачи данных и их доставку в пункт назначения. Недостатки протокола проявляются в успешных атаках, при которых злоумышленник может получить доступ к передаваемым данным, выдать себя за другую сторону или привести систему в нерабочее состояние, а также TCP требует явного указания максимального размера сегмента.

Передача с использованием TCP аналогична отправке пакетов с трекингом, путь которых отслеживается от отправителя до получателя. Если заказ разбит на несколько частей, заказчик может зайти на веб-сайт транспортной компании и посмотреть порядок доставки.

TCP использует следующие три основные операции для обеспечения надежности:

· Отслеживание количества сегментов, отправленных на то или иное устройство тем или иным приложением.

· Подтверждение полученных данных.

· Повторная передача сегментов с неподтвержденными данными по истечении определенного времени ожидания.

6) Системы адресации. IPv4. IPv6

Для каждого компьютера в Интернет устанавливают два адреса: цифровой IP-адрес (IP – Internetwork Protocol – межсетевой протокол) и доменный адрес. Оба адреса могут применяться равноценно. Цифровой адрес удобен для обработки на компьютере, а доменный адрес – для восприятия пользователем.

Каждому компьютеру, подключенному к сети Интернет, присваивается числовой адрес, называемый IP-адресом.

IP-адрес используется в протоколах передачи данных. IP-адрес содержит полную информацию необходимую для идентификации узла в сети.

При сеансовом подключении к сети Интернет IP-адрес выделяется компьютеру только на время этого сеанса. Такое присвоение адреса компьютеру называется динамическим распределением IP-адресов. Динамическое распределение IP-адресов позволяет обслуживать большое количество пользователей, имея небольшое количество IP-адресов, так как один и тот же IP-адрес в разные моменты времени может быть выделен разным пользователям.

IP - адрес имеет числовой вид, так как его используют в своей работе компьютеры. Но он весьма сложен для запоминания, поэтому была разработана доменная система имен: DNS.

**DNS** - адрес включает более удобные для пользователя **буквенные сокращения**, которые также разделяются точками на отдельные информационные блоки (*домены*).

IPv4 (Интернет Протокол версии 4) является четвёртой версией Интернет Протокола (IP), и используется для идентификации устройств в сети через адресную систему, позволяя, так же, соединять устройства через веб. IPv4 использует 32-битную адресную схему, позволяя существование 2^32 (более 4 миллиардов) адресов. При этом вместе с ростом Интернета ожидается, что количество неиспользуемых IPv4 адресов достаточно быстро закончится, так как каждое устройство, включая компьютеры, смартфоны и игровые консоли при подключении к Интернету требует для себя IP-адрес.

Новый адресная система Интернет использующая Интернет-Протокол версии 6 (IPv6) разрабатывалась для того, чтобы полностью удовлетворить возрастающую потребность в необходимом числе свободных интернет-адресов.

Новый IPv6 создан чтобы обеспечить интернету устойчивый и надёжный рост, касающийся как номера наличных хостов, так и общего количества передаваемого траффика, поддерживая 2^128 адресов – намного больше устаревшего протокола IPv4.

7) Система доменных имен DNS. Рекурсивная и нерекурсивная схемы. Структура полного доменного имени

Полное доменное имя узла состоит из ряда отдельных доменов (от англ. *domain* — область), которые разделены точками. Например, полное доменное имя [google.com](http://google.com/) включает в себя домен второго уровня .**google**, который входит в домен первого уровня .**com**.

Существуют определенные ограничения на длину доменных имен: как правило, полное имя не должно состоять из более чем 63 символов. Тем не менее в некоторых государствах разрешается регистрация имен длиной до 127 знаков.

**DNS** (англ. Domain Name System — система доменных имён) — компьютерная распределённая система для получения информации о доменах. Чаще всего используется для получения IP-адреса по имени хоста (компьютера или устройства), получения информации о маршрутизации почты, обслуживающих узлах для протоколов в домене (SRV-запись).

DNS (domain name system) - это система, обеспечивающая работу привычных нам доменных имен сайтов. Связь между устройствами в сети Интернет осуществляется по IP адресам, например: "192.64.147.209". Однако, запомнить IP адреса сложно, поэтому были придуманы удобные для человека доменные имена, например: "google.com".

Компьютер / сервер не хранит таблицу соответствия доменов и их IP адресов. Точнее, не хранит всю таблицу, а временно запоминает данные для часто используемых доменов. Когда в браузере вводится домен сайта, компьютер автоматически узнает его IP адрес, и отправляет по нему запрос. Этот процесс называется «разрешение адреса домена» (domain resolving).

Все запросы, отправляемые DNS-клиентом DNS-серверу для разрешения имен, делятся на два типа:

* итеративные запросы (клиент посылает серверу DNS запрос, в котором требует дать наилучший ответ без обращений к другим DNS-серверам);
* рекурсивные запросы (клиент посылает серверу DNS запрос, в котором требует дать окончательный ответ даже если DNS-серверу придется отправить запросы другим DNS-серверам; посылаемые в этом случае другим DNS-серверам запросы будут итеративными).

При рекурсивном запросе Вы просто обращаетесь к серверу, а он, если не найдет у себя нужной записи, идет к другим серверам и спрашивает у них. Нерекурсивный dns сервер в данном случае просто говорит - "я не знаю, но спроси у этого сервера". И клиент будет слать ещё один запрос. Понятное дело, что при медленном интернете первый вариант лучше.

8) Взаимодействие по протоколу HTTP, протокол HTTPS, HTTP-запрос

HTTP — прикладной протокол передачи данный, используемый для получения информации с веб-сайтов

HTTPS — расширение протокола HTTP, поддерживающее шифрование по протоколам TLS и SSL

Различия:

· HTTPS не является отдельным протоколом передачи данных, а представляет собой расширение протокола HTTP с надстройкой шифрования;

· передаваемые по протоколу HTTP данные не защищены, HTTPS обеспечивает конфиденциальность информации путем ее шифрования;

· HTTP использует порт 80, HTTPS — порт 443.

Использование HTTPS

Обеспечение безопасной передачи данных необходимо на сайтах, где вводится и передается конфиденциальная информация (личные данные пользователей, детали доступа, реквизиты платежных карт) — на любых сайтах с авторизацией, взаимодействием с платежными системами, почтовыми сервисами. Шифрование таких данных позволит предотвратить их получение и использование третьими лицами.

HTTP

Этот протокол описывает взаимодействие между двумя компьютерами (клиентом и сервером), построенное на базе сообщений, называемых запрос (Request) и ответ (Response). Каждое сообщение состоит из трех частей: стартовая строка, заголовки и тело. При этом обязательной является только стартовая строка.

Стартовые строки для запроса и ответа имеют различный формат — нам интересна только стартовая строка запроса, которая выглядит так:

· METHOD URI HTTP/VERSION,

где METHOD — это как раз метод HTTP-запроса, URI — идентификатор ресурса, VERSION — версия протокола (на данный момент актуальна версия 1.1).

Заголовки — это набор пар имя-значение, разделенных двоеточием. В заголовках передается различная служебная информация: кодировка сообщения, название и версия браузера, адрес, с которого пришел клиент (Referrer) и так далее.

Тело сообщения — это, собственно, передаваемые данные. В ответе передаваемыми данными, как правило, является html-страница, которую запросил браузер, а в запросе, например, в теле сообщения передается содержимое файлов, загружаемых на сервер. Но как правило, тело сообщения в запросе вообще отсутствует.

Пример HTTP-взаимодействия

Запрос:

1. GET /index.php HTTP/1.1

2. Host: example.com

3. User-Agent: Mozilla/5.0 (X11; U; Linux i686; ru; rv:1.9b5) Gecko/2008050509 Firefox/3.0b5

4. Accept: text/html

5. Connection: close

Первая строка — это строка запроса, остальные — заголовки; тело сообщения отсутствует

Ответ:

1. HTTP/1.0 200 OK

2. Server: nginx/0.6.31

3. Content-Language: ru

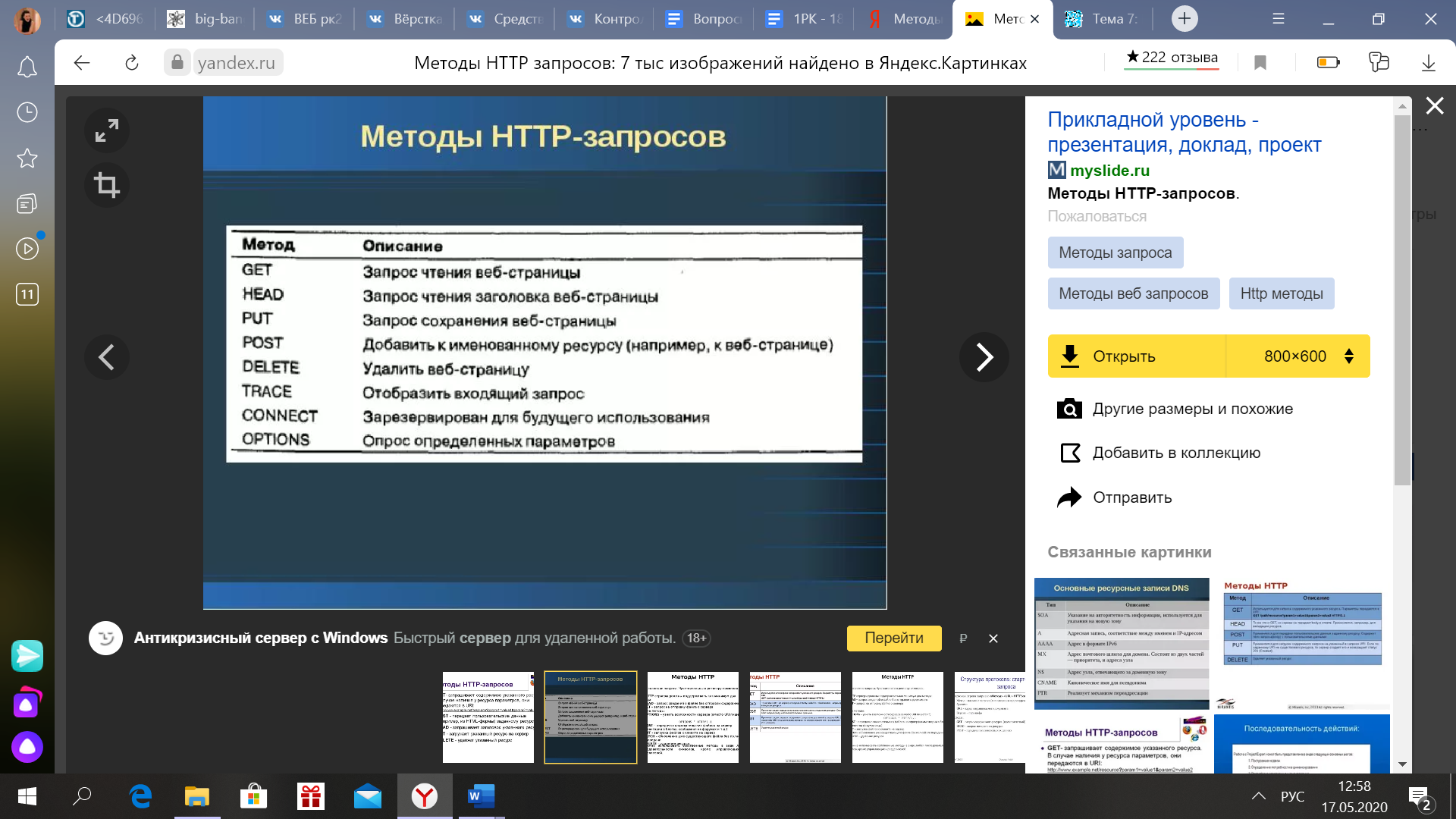
4. Content-Type: text/html; charset=utf-8

5. Content-Length: 1234

6. Connection: close

9) Методы HTTP запросов. Основные статусы. Содержание запросов и ответов

HTTP определяет множество **методов запроса**, которые указывают, какое желаемое действие выполнится для данного ресурса. Несмотря на то, что их названия могут быть существительными, эти методы запроса иногда называются HTTP глаголами. Каждый реализует свою семантику, но каждая группа команд разделяет общие свойства: так, методы могут быть [безопасными](https://developer.mozilla.org/ru/docs/%D0%A1%D0%BB%D0%BE%D0%B2%D0%B0%D1%80%D1%8C/safe), [идемпотентными](https://developer.mozilla.org/ru/docs/%D0%A1%D0%BB%D0%BE%D0%B2%D0%B0%D1%80%D1%8C/Idempotent) или [кэшируемыми](https://developer.mozilla.org/ru/docs/%D0%A1%D0%BB%D0%BE%D0%B2%D0%B0%D1%80%D1%8C/cacheable).



## **Список кодов статуса HTTP**

Существует более 40 различных кодов статуса сервера. Но тех, с которыми вы будете сталкиваться регулярно меньше дюжины. Ниже приводится список кодов статуса **HTTP**:

### **Код статуса 200**

**200:** «Все в порядке». Это код, который возвращается, когда веб-страница или ресурс действуют точно так, как ожидается.

### **Коды статуса 300**

**301:** «Запрошенный ресурс был перемещен навсегда». Этот код возвращается, когда веб-страница или ресурс заменяется другим ресурсом. Он используется для постоянного редиректа **URL-адресов**.

**302:** это **http-ошибка** «Запрошенный ресурс перемещен, но был найден». Этот код используется для указания того, что запрошенный ресурс был найден, но не в том месте, где это ожидалось. Он используется для временного редиректа **URL-адресов**.

**304:** «Запрошенный ресурс не был изменен с момента последнего обращения к нему». Сообщает, что ресурсы, хранящиеся в кэше браузера, не изменились. Он используется для ускорения доставки веб-страниц за счет повторного использования ранее загруженных ресурсов.

### **Коды статуса 400**

**401:** «Не авторизован». Возвращается сервером, когда для доступа к целевому ресурсу отсутствуют валидные учетные данные.

**http-ошибка 403:** «Доступ к этому ресурсу запрещен». Возвращается, когда пользователь пытается открыть ресурс, для которого у него нет прав доступа. Например, попытка просмотра неавторизованным пользователем контента, защищенного паролем, может привести к ошибке **403**.

**404:** «Запрошенный ресурс не найден». Наиболее распространенное сообщение об ошибке. Означает, что запрошенный ресурс не существует и сервер не знает, существовал ли он когда-либо.

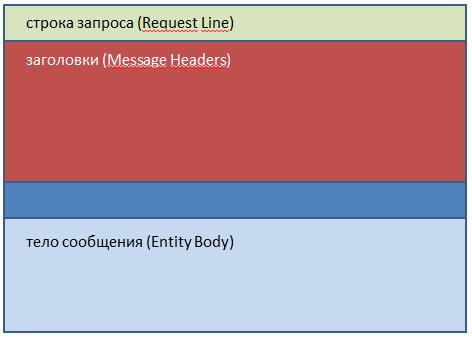
**405:** «Метод не разрешен». Генерируется, когда хостинг-сервер (исходный сервер) поддерживает полученный метод, но целевой ресурс отсутствует.

**406:** «Неприемлемый ответ». Запрошенный ресурс способен генерировать только контент, неприемлемый в соответствии с заголовками **Accept**, отправленными в запросе.

**408:** «Время ожидания сервером поступления остальной части запроса из браузера истекло». Генерируется, когда сервер прерывает обработку после истечения времени ожидания полного запроса от браузера. Другими словами, сервер не получил полный запрос, отправленный браузером. Одной из возможных причин может быть перегрузка сети, приводящая к потере пакетов между браузером и сервером.

**410:** «Запрошенный ресурс отсутствует и не будет возвращен». Подобен коду **404** «Не найден», за исключением того, что код статуса **410**, указывает, что данный статус ожидается на постоянной основе.

HTTP запрос состоит из трех основных частей, которые идут в нем именно в том порядке, который указан ниже. Между заголовками и телом сообщения находится пустая строка (в качестве разделителя), она представляет собой символ перевода строки.



1. строка запроса (Request Line)

2. заголовки (Message Headers)

Пустая строка (разделитель)

3. тело сообщения (Entity Body) – необязательный параметр

**Строка запроса** – указывает метод передачи, URL-адрес, к которому нужно обратиться и версию протокола HTTP.

**Заголовки** – описывают тело сообщений, передают различные параметры и др. сведения и информацию.

**тело сообщения**  - это сами данные, которые передаются в запросе.  Тело сообщения – это необязательный параметр и может отсутствовать.

Когда мы получаем ответный запрос от сервера, тело сообщения, чаще всего представляет собой содержимое веб-страницы. Но, при запросах к серверу, оно тоже может иногда присутствовать, например, когда мы передаем данные, которые заполнили в форме обратной связи на сервер.

10)Принципы маршрутизации данных в Интернете. BGP

**BGP** ([англ.](https://ru.bmstu.wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) ***B****order****G****ateway****P****rotocol* – протокол граничного шлюза) — динамический протокол маршрутизации. Относится к классу протоколов маршрутизации внешнего шлюза ([англ.](https://ru.bmstu.wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *EGP — External Gateway Protocol*). Данный протокол предназначен для обмена информацией о достижимости подсетей между автономными системами (АС, [англ.](https://ru.bmstu.wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *AS — autonomous system*). Передаваемая информация включает в себя список АС, к которым имеется доступ через данную систему. Выбор наилучших маршрутов осуществляется исходя из правил, принятых в сети.

Маршрутизаторы, использующие протокол BGP, обмениваются информацией о доступности сетей. Вместе с информацией о сетях передаются различные атрибуты этих сетей, с помощью которых BGP выбирает лучший маршрут и настраиваются политики маршрутизации.

Один из основных атрибутов, который передается с информацией о маршруте — это список автономных систем, через которые прошла эта информация. Эта информация позволяет BGP определять где находится сеть относительно автономных систем, исключать петли маршрутизации, а также может быть использована при настройке политик.

Маршрутизация осуществляется пошагово от одной автономной системы к другой. Все политики BGP настраиваются, в основном, по отношению к внешним/соседним автономным системам. То есть, описываются правила взаимодействия с ними.

Так как BGP оперирует большими объемами данных (текущий размер таблицы для IPv4 более 450 тысяч маршрутов), то принципы его настройки и работы отличаются от внутренних протоколов динамической маршрутизации (IGP).