

Теория 3

HTTP

Сетевой протокол прикладного уровня, который изначально предназначался для получения с серверов гипертекстовых документов в формате HTML, а с течением времени стал универсальным средством взаимодействия между узлами как всемирной паутины, так и изолированных веб-инфраструктур.

Спецификации

RFC 124, RFC 1945, RFC 2616, RFC 2617, RFC 6266, RFC 7230, RFC 7240, RFC 8446, RFC 9110

Основой HTTP является технология клиент-сервер, то есть предполагается существование:

- Потребителей (клиентов), которые инициируют соединение и посылают запрос;
- Поставщиков (серверов), которые ожидают соединения для получения запроса, производят необходимые действия и возвращают обратно сообщение с результатом.

Основным объектом манипуляции в HTTP является ресурс, на который указывает URI (Uniform Resource Identifier) в запросе клиента.

Обмен сообщениями идёт по обыкновенной схеме «запрос-ответ».

В отличие от многих других протоколов, HTTP не сохраняет своего состояния.

HTTP устанавливает отдельную TCP-сессию на каждый запрос.

В более поздних версиях HTTP было разрешено делать несколько запросов в ходе одной TCP-сессии, но браузеры обычно запрашивают только страницу и включённые в неё объекты (картинки, каскадные стили и т. п.), а затем сразу разрывают TCP-сессию.

Для поддержки авторизованного (не анонимного) доступа в HTTP используются cookies; причём такой способ авторизации позволяет сохранить сессию даже после перезагрузки клиента и сервера.

HTTP перед тем, как передать сами данные, передаёт заголовок «Content-Type: тип/подтип», позволяющий клиенту однозначно определить, каким образом обрабатывать присланные данные.

Небольшие сноски:

- URI - унифицированный идентификатор ресурса.

Структура HTTP - сообщения

✓ 1. Стартовая строка

Определяет тип сообщения.

Стартовые строки для запроса и ответа различаются.

Строка запроса:

GET URI - для версии протокола 0.9

Метод URI HTTP/Версия - Для остальных версий

- Метод - тип запроса, одно слово заглавными буквами (в версии 0.9 использовался только метод GET).
- URI - путь к запрашиваемому документу.
- Версия - пара разделённых точкой цифр. Например: 1.0.

Строка ответа:

HTTP/Версия КодСостояния Пояснение

Версия - пара разделённых точкой цифр.

Код состояния - три цифры.

Пояснение - текстовое пояснение к коду ответа. Необязательно.

✓ 2. Заголовки

Характеризует тело сообщения, параметры для передачи и прочие сведения

Заголовок - пара параметр: значение .

Заголовки делятся на 4 основные группы:

1. General Headers (Основные заголовки) - могут включаться в любое сообщение клиента и сервера.

- Cache-control
- Connection
- Date
- Pragma
- Transfer-Encoding
- Upgrade
- Via

2. Request Headers (Заголовки запроса) - используются только в запросах клиента.

- Accept
- Authorization
- From
- Host - Имя домена ресурса RFC 2068 14.23

- **Range** - Какую часть документа следует вернуть (принимает в аргументы: тип - единица измерения, начало отчёта, конец отчёта, в формате тип=начало_первой_части-конец_первой_части, <другие части для возврата>) RFC 2068 14.36
- **Referer** - Полная ссылка к ресурсу RFC 2068 14.37

3. Response Headers (Заголовки ответа) - только для ответов от сервера.

- Age
- Location
- Public
- Server
- **Vary** - Определяет, как сопоставить будущие заголовки запроса, чтобы решить, можно ли использовать кешированный ответ, а не запрашивать новый с исходного сервера RFC 2068 14.43
- Warning

4. Entity Headers (Заголовки сущности) - сопровождают каждую сущность сообщения.

- Allow
- Content-Base
- Content-Language
- Content-Type
- ETag
- **Expires** - содержит дату/время, по истечении которой ответ сервера считается устаревшим. Прошедшая или некорректная дата, например 0, обозначает, что ресурс уже устарел. RFC 2068 14.21
- Last-Modified

Примеры заголовков взяты из RFC 2068, что соответствует HTTP 1.1

Все стандартные заголовки описаны в RFC, также можно вводить свои заголовки.

✓ 3. Тело сообщения

Данные сообщения. Обязательно должно отделяться от заголовков пустой строкой. Может отсутствовать.

- OPTIONS - Используется для определения возможностей веб-сервера или параметров соединения для конкретного ресурса. В ответ серверу следует включить заголовок `Allow` со списком поддерживаемых методов. Также в заголовке ответа может включаться информация о поддерживаемых расширениях.
- GET - Используется для запроса содержимого указанного ресурса.
- HEAD - Аналогичен `GET`, но в ответе отсутствует тело.
- POST - Применяется для передачи пользовательских данных заданному ресурсу.
- PUT - Применяется для загрузки содержимого запроса на указанный в запросе URI.
- PATCH - Аналогично PUT, но применяется только к фрагменту ресурса.
- DELETE - Удаляет указанный ресурс.
- TRACE - Возвращает полученный запрос так, что клиент может увидеть, какую информацию промежуточные серверы добавляют или изменяют в запросе.
- CONNECT - Преобразует соединение запроса в прозрачный TCP/IP-туннель, обычно чтобы содействовать установлению защищённого SSL-соединения через нешифрованный прокси.

⚡ Различие PUT и PATCH

📖 Цитата RFC 2068 19.6.1.1

The PATCH method is similar to PUT except that the entity contains a list of differences between the original version of the resource identified by the Request-URI and the desired content of the resource after the PATCH action has been applied. The list of differences is in a format defined by the media type of the entity (e.g., "application/diff") and MUST include sufficient information to allow the server to recreate the changes necessary to convert the original version of the resource to the desired version.

Т.е. PATCH содержит список изменений после исправления, такой, что следуя ему, можно повторно получить из оригинала измененную версию.

☰ Коды состояний

- 1xx - Информационное сообщение.
- 2xx - Успех выполнения запроса.
- 3xx - Ожидание другого запроса, как правило другому URI.
- 4xx - Ошибка со стороны клиента.
- 5xx - Ошибка со стороны сервера.

На месте 'x' ставятся другие цифры, которые делают код более информативным. Например, по `RFC 2068` :

- 200 - Запрос успешно выполнен.
- 302 - Запрошенному ресурсу был присвоен новый постоянный URI, и любые будущие запросы должны выполняться по нему.
- 304 - Если клиент выполнил условный запрос GET и доступ разрешен, но документ не был изменен, сервер ДОЛЖЕН ответить этим кодом состояния. Ответ НЕ ДОЛЖЕН содержать текст сообщения.
- 401 - Запрос требует аутентификации пользователя.
- 402 - Код зарезервирован на будущее. По RFC 2068 !!!
- 403 - Сервер понял запрос, но отказывается его выполнять. Авторизация не поможет и не стоит повторять этот запрос.

HTTP/2 и HTTP/3 были разработаны для улучшения производительности веб-сайтов, уменьшения задержки, улучшения безопасности и обеспечения более эффективного использования сетевых ресурсов.

HTTP/2 вводит многопоточность, что позволяет одному соединению обрабатывать множество запросов одновременно, устраняя проблему блокировки очереди (HOL) в HTTP/1. HTTP/3 использует протокол QUIC вместо TCP для улучшения производительности в условиях пакетной потери и для уменьшения задержки соединения.

Небольшие сноски:

- HOL - Проблема блокировки головы очереди в HTTP/1 и TCP происходит, когда пакет, заблокированный в начале очереди, блокирует все последующие пакеты, даже если последующие пакеты могут быть обработаны.
- QUIC - протокол, позволяющий мультиплексировать несколько потоков данных. Содержит возможности шифрования, эквивалентные TLS и SSL. Имеет более низкую задержку соединения и передачи, чем TCP. Лучшая переносимость потери пакетов.
- Мультиплексирование - создание в одном канале несколько подканалов с меньшей пропускной способностью.
- SSL/TSL - криптографические протоколы.

Использование HTTP/1 и base64 увеличивает размер исходных данных примерно на 33%, в связи со способом кодировки.

FTP

Протокол передачи файлов.

Особенность протокола FTP в том, что он использует множественное (как минимум — двойное) подключение. При этом один канал является управляющим, через который поступают команды серверу и возвращаются его ответы (обычно через TCP-порт 21), а через остальные происходит собственно передача данных, по одному каналу на каждую передачу. Поэтому в рамках одной сессии по протоколу FTP можно передавать

одновременно несколько файлов, причём в обоих направлениях.

Для каждого канала данных открывается свой TCP порт, номер которого выбирается либо сервером, либо клиентом, в зависимости от режима передачи.

☰ Команды

- ABOR - Прервать передачу файла;
- CDUP - Сменить каталог на вышестоящий;
- CWD - Сменить каталог;
- DELE - Удалить файл;
- EPSV - Войти в расширенный пассивный режим;
- HELP - Вывод списка команд;
- LIST - Список файлов каталога через соединение данных;
- MDTM - Время модификации файла;
- MKD - Создать каталог;
- NLST - Список файлов в более кратком формате, чем `LIST` ;
- NOOP - Пустая операция;
- PASS - Пароль;
- PASV - Войти в пассивный режим;
- PORT - Войти в активный режим;
- PWD - Возвращает текущий каталог;
- QUIT - Отключиться;
- REIN - Реинициализировать подключение;
- RETR - Скачать файл;
- RMD - Удалить каталог;
- RNFR - Что переименовать;
- RNT0 - Во что переименовать;
- SIZE - Размер файла;
- STOR - Закачать файл;
- SYST - Возвращает тип системы;
- TYPE - Установить тип передачи файла (бинарный, текстовый);
- USER - Имя пользователя для входа.

❓ Отличие пассивного и активного режимов

В пассивном клиент подключается к серверу.

В активном сервер подключается к клиенту.

❓ Возврат на PASV

Сервер на команду PASV возвращает IP адрес и порт, чтобы клиент мог по ним подключиться для последующего получения данных.

❓ Проблемы с NAT в активном режиме

В активном режиме FTP сервер пытается подключиться к клиенту. Если клиент находится за NAT, активный режим не будет работать, потому что клиент использует IP-адрес, недоступный для сервера. Кроме того, FTPS никогда не работает в активном режиме, если вовлечен NAT, поскольку NAT не может анализировать зашифрованные команды, которые отправляются.

☰ Коды состояний

Первая цифра - один из трёх исходов:

- 2xx - Успешный ответ.
- 1xx/3xx - Ошибка или неполный ответ.
- 4xx/5xx - Команда не может быть выполнена.

Вторая - тип ошибки.

Третья - окончательно специфицирует ошибку.

⚡ FTP не использует base64

NAT

Механизм в сетях TCP/IP, позволяющий преобразовывать IP-адреса транзитных пакетов. Позволяет не выдавать каждому устройству свой публичный IP-адрес, что экономит адреса. Также имеет названия IP Masquerading, Network Masquerading и Native Address Translation.

Наиболее популярным является SNAT, суть механизма которого состоит в замене адреса источника при прохождении пакета в одну сторону и обратной замене адреса назначения в ответном пакете. Наряду с адресами источник/назначение могут также заменяться номера портов источника и назначения.

Принимая пакет от локального компьютера, роутер смотрит на IP-адрес назначения. Если это локальный адрес, то пакет пересылается другому локальному компьютеру. Если нет, то пакет надо переслать наружу в интернет. Но ведь обратным адресом в пакете указан локальный адрес компьютера, который из интернета будет недоступен. Поэтому роутер «на лету» транслирует (подменяет) обратный IP-адрес пакета на свой внешний (видимый из интернета) IP-адрес и меняет номер порта (чтобы различать ответные пакеты, адресованные разным локальным компьютерам). Комбинацию, нужную для обратной подстановки, роутер сохраняет у себя во временной таблице. Через некоторое время после того, как клиент и сервер закончат обмениваться пакетами, роутер сотрёт у себя в таблице запись об n-м порте за сроком давности.

Существует 3 базовых концепции трансляции адресов:

- Статический NAT — отображение незарегистрированного IP-адреса на зарегистрированный IP-адрес на основании один к одному. Особенно полезно, когда устройство должно быть доступным снаружи сети.
- Динамический NAT — отображает незарегистрированный IP-адрес на зарегистрированный адрес из группы зарегистрированных IP-адресов. Динамический NAT также устанавливает непосредственное отображение между незарегистрированными и зарегистрированными адресами, но отображение может меняться в зависимости от зарегистрированного адреса, доступного в пуле адресов, во время коммуникации.
- Перегруженный NAT (NAPT, NAT Overload, PAT) — форма динамического NAT, который отображает несколько незарегистрированных адресов в единственный зарегистрированный IP-адрес, используя различные порты. Известен также как PAT (Port Address Translation). При перегрузке каждый компьютер в частной сети транслируется в тот же самый адрес, но с различным номером порта.

Типы NAT:

- Симметричный NAT (Symmetric NAT) — трансляция, при которой каждое соединение, инициируемое парой «внутренний адрес: внутренний порт» преобразуется в свободную уникальную случайно выбранную пару «публичный адрес: публичный порт». При этом инициация соединения из публичной сети невозможна.
- Cone NAT, Full Cone NAT — однозначная (взаимная) трансляция между парами «внутренний адрес: внутренний порт» и «публичный адрес: публичный порт». Любой внешний хост может инициировать соединение с внутренним хостом (если это разрешено в правилах межсетевого экрана).
- Address-Restricted cone NAT, Restricted cone NAT — постоянная трансляция между парой «внутренний адрес: внутренний порт» и «публичный адрес: публичный порт». Любое соединение, инициированное с внутреннего адреса, позволяет в дальнейшем получать ему пакеты с любого порта того публичного хоста, к которому он отправлял пакет(ы) ранее.
- Port-Restricted cone NAT — трансляция между парой «внутренний адрес: внутренний порт» и «публичный адрес: публичный порт», при которой входящие пакеты проходят на внутренний хост только с одного порта публичного хоста — того, на который внутренний хост уже посылал пакет.

Теоретически, без ограничений оборудования, предоставляющего NAT, NAT могут поддерживать *Диапазон_адресов_IPv4 * Диапазон_портов_TCP* на одном устройстве.

Примеры адресов для преобразования из одного диапазона в другой:

☐ 192.168.0.0 - 192.168.255.255

Префикс - /16

Диапазон адресов, которые используются в локальных сетях.

NAT может преобразовать эти адреса в публичные.

☰ **172.16.0.0 - 172.31.255.255**

Префикс - /12

Диапазон адресов, которые используются в корпоративных сетях.

NAT может преобразовать эти адреса в публичные.

☰ **10.0.0.0 - 10.255.255.255**

Префикс - /8

Диапазон адресов, которые используются в больших корпоративных сетях.

NAT может преобразовать эти адреса в публичные.

☰ **100.0.0.0**

Диапазон публичных адресов.

NAT не используется, так как это уже публичные адреса

📘 **Небольшие сноски:**

- Префикс сети - обозначает длину сетевой части IP - адреса. Помогает определить, какие IP - адреса принадлежат одной и той же подсети. Также с помощью префиксов NAT может определять, какие адреса должны быть определены.