HTTP-ЗАПРОСЫ

HTTP

Это протокол, который описывает взаимодействие между двумя компьютерами (клиентом и сервером), построенное на базе сообщений, называемых запрос (Request) и ответ (Response).

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

- Серверы как основные поставщики услуг хранения и обработки информации (обработка запросов).
- Клиенты — конечные потребители услуг сервера (отправка запроса).
- Прокси для выполнения транспортных служб.

СТРУКТУРА ПРОТОКОЛА

- Стартовая строка (*Starting line*) — обязательный элемент, определяющий тип сообщения;
- Заголовки (*Headers*) — характеризуют тело сообщения, параметры передачи и прочие сведения;
- Тело сообщения (*Message Body*) — непосредственно данные сообщения. Обязательно должно отделяться от заголовков пустой строкой.

СТАРТОВАЯ СТРОКА

Строка запроса:

- GET URI — для версии протокола 0.9.
- Метод URI HTTP/Версия — для остальных версий.

Стартовая строка ответа сервера:

- HTTP/Версия КодСостояния Пояснение,

- Метод (*Method*) — название запроса, одно слово заглавными буквами. В версии HTTP 0.9 использовался только метод GET.
- URI определяет путь к запрашиваемому документу.
- Версия (*Version*) — пара разделённых точкой арабских цифр. Например: 1.0
- КодСостояния (*Status Code*) — три арабские цифры. По коду статуса определяется дальнейшее содержимое сообщения и поведение клиента.
- Пояснение (*Reason Phrase*) — текстовое короткое пояснение к коду ответа для пользователя. Никак не влияет на сообщение и является необязательным.

Запрос:

GET /wiki/HTTP HTTP/1.0
Host: ru.wikipedia.org

Ответ:

HTTP/1.0 200 OK

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 т. д.

ТЕЛО СООБЩЕНИЯ

Это 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.

МЕТОДЫ

Метод (тип) HTTP— последовательность из любых символов, кроме управляющих и разделителей, указывающая на основную операцию над ресурсом. Обычно метод представляет собой короткое английское слово, записанное заглавными буквами. Обратите внимание, что название метода чувствительно к регистру.

Основные:

- GET — получение ресурса
- POST — создание ресурса
- PUT — обновление ресурса
- DELETE — удаление ресурса

ИНТЕРЕСНЫЕ ФАКТЫ

- Спецификация HTTP не обязывает сервер понимать все методы, кроме GET.
- Спецификация HTTP не указывает серверу, что он должен делать при получении запроса с тем или иным методом (на запрос GET /index.php HTTP/1.1 **не обязан** возвращать вам страницу index.php, он может ее удалять, например))

GET

- Используется для запроса содержимого указанного ресурса. С помощью метода GET можно также начать какой-либо процесс. В этом случае в тело ответного сообщения следует включить информацию о ходе выполнения процесса.
- Клиент может передавать параметры выполнения запроса в URI целевого ресурса после символа «?»:
  GET /path/resource?param1=value1&param2=value2 HTTP/1.1
- Кроме обычного метода GET, различают ещё условный GET и частичный GET.

- Conditional Get (условный) - когда в запросе используется поле If-Modified-Since. Метод Conditional Get означает, что определенный запросом объект будет передаваться в случае, если дата его модификации старше даты, указанной в поле If-Modified-Since;
- Partial Get (частный) - когда в запросе используется поле Range. Это позволяет определить, какую часть от определенного объекта требуется передать.

HEAD

Аналогичен методу GET, за исключением того, что в ответе сервера отсутствует тело. Запрос HEAD обычно применяется для извлечения метаданных, проверки наличия ресурса и чтобы узнать, не изменился ли он с момента последнего обращения.

POST

- Применяется для передачи пользовательских данных заданному ресурсу. Например, в блогах посетители обычно могут вводить свои комментарии к записям в HTML-форуму, после чего они передаются серверу методом POST и он помещает их на страницу. При этом передаваемые данные (в примере - текст комментария) включаются в тело запроса. Аналогично с помощью метода POST обычно загружаются файлы на сервер.
- При результате выполнения 200 (Ok) в тело ответа следует включить сообщение об итоге выполнения запроса. Если был создан ресурс, то серверу следует вернуть ответ 201 (Created) с указанием URI нового ресурса в заголовке Location.

PUT

- Применяется для загрузки содержимого запроса на указанный в запросе URI. Если по заданному URI не существовало ресурса, то сервер создаёт его и возвращает статус201 (Created). Если же был изменён ресурс, то сервер возвращает 200 (Ok) или 204 (No Content). Сервер не должен игнорировать некорректные заголовки Content-*передаваемые клиентом вместе с сообщением. Если какой-то из этих заголовков не может быть распознан или не допустим при текущих условиях, то необходимо вернуть код ошибки 501 (Not Implemented).
- Фундаментальное различие методов POST и PUT заключается в понимании предназначений URI ресурсов.
  Метод POST предполагает, что по указанному URI будет производиться обработка передаваемого клиентом содержимого. Используя PUT, клиент предполагает, что загружаемое содержимое

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.

LINK

- Устанавливает связь указанного ресурса с другими.

UNLINK

- Убирает связь указанного ресурса с другими.

OPTIONS

- метод запроса информации о параметрах соединения, используемых в цепочках запрос/ответ;

TRACE

- метод, применяющийся для проверки корректности соединения. Конечный адресат в сети должен вернуть сообщение, посланное программой-клиентом с сообщением ответа 200 (“Ok”). Метод Trace не должен содержать тело объекта и должен включать поле Content-Lenght (текущая длина) заголовка запроса со значением 0.

ТИПЫ СООБЩЕНИЙ HTTP

- Нулевой запрос (пустая строка) всегда должен игнорироваться. Программа-клиент не должна посылать нулевой запрос, но возможны ситуации ошибок и тестирования, в которых нулевой запрос может быть послан как ошибочный, и он не должен являться причиной ошибки на сервере. Нулевой запрос имеет вид: Null-Reqest: CRLF.

ТИПЫ СООБЩЕНИЙ HTTP

- Сообщение полного запроса, передаваемое от программы-клиента к серверу включает метод доступа к ресурсу, идентификатор ресурса и версию используемого протокола. Для совместимости с более простым протоколом HTTP/0.9 используются две формы запросов HTTP: полный запрос и полный запрос с нулевым запросом (Full-Reqest | Null-Reqest).

КОДЫ СОСТОЯНИЯ

Код состояния является частью первой строки ответа сервера. Он представляет собой целое число из трех арабских цифр. Первая цифра указывает на класс состояния. За кодом ответа обычно следует отделённая пробелом поясняющая фраза на английском языке, которая разъясняет человеку причину именно такого ответа:

201 Webpage Created

1XX: INFORMATIONAL (ИНФОРМАЦИОННЫЕ)

- В этот класс выделены коды, информирующие о процессе передачи. При работе через протокол версии 1.0 сообщения с такими кодами должны игнорироваться. В версии 1.1 клиент должен быть готов принять этот класс сообщений как обычный ответ, но серверу отправлять что-либо не нужно. Сами сообщения от сервера содержат только стартовую строку ответа и, если требуется, несколько специфичных для ответа полей заголовка. Прокси-сервера подобные сообщения должны отправлять дальше от сервера к клиенту.

2XX: SUCCESS (УСПЕШНО)

- Сообщения данного класса информируют о случаях успешного принятия и обработки запроса клиента. В зависимости от статуса сервер может ещё передать заголовки и тело сообщения.

3XX: REDIRECTION (ПЕРЕНАПРАВЛЕНИЕ)

- Коды этого класса сообщают клиенту, что для успешного выполнения операции необходимо сделать другой запрос, как правило, по другому URI. Из данного класса пять кодов 301, 302, 303, 305 и 307 относятся непосредственно к перенаправлениям. Адрес, по которому клиенту следует произвести запрос, сервер указывает в заголовке Location. При этом допускается использование фрагментов в целевом URI.
- По последним стандартам клиент может производить перенаправление без запроса пользователя только если второй ресурс будет запрашиваться методом GET или HEAD

4XX: CLIENT ERROR (ОШИБКА КЛИЕНТА)

- Класс кодов 4xx предназначен для указания ошибок со стороны клиента. При использовании всех методов, кроме HEAD, сервер должен вернуть в теле сообщения гипертекстовое пояснение для пользователя.

5XX: SERVER ERROR (ОШИБКА СЕРВЕРА)

- Коды 5xx выделены под случаи неудачного выполнения операции по вине сервера. Для всех ситуаций, кроме использования метода HEAD, сервер должен включать в тело сообщения объяснение, которое клиент отобразит пользователю.