

## 1. Обзор

HTTP (англ. *Hypertext Transfer Protocol*) — прикладной протокол семейства TCP/IP, используемый для передачи произвольных данных. Передача HTTP-сообщений организована посредством TCP — протокола транспортного уровня, по умолчанию используется порт 80. Последней актуальной версией протокола является [HTTP/2](#).

HTTP базируется на сетевой архитектуре «клиент-сервер». Как правило, в качестве клиента выступает браузер, в качестве сервера — хостинг с развернутым веб-сайтом. Клиент отправляет HTTP-сообщение с запросом серверу, а сервер, который предоставляет ресурсы, возвращает ответное HTTP-сообщение клиенту.

Ресурс представляет из себя файл (например, HTML-страницу) или что-то абстрактное (например, услугу по некоторой обработке информации). Чтобы указать на необходимый ему ресурс, клиент в HTTP-запросе указывает [URI](#) (англ. *Uniform Resource Identifiers*).

## 2. Формат сообщений

В общем случае структура HTTP-сообщения имеет следующий вид:

1. **стартовая строка** (англ. *start line*) указывает на ресурс и операцию над ним;
2. **заголовки** (англ. *headers*) определяют параметры передачи и характеризуют тело сообщения;
3. пустая строка;
4. **тело сообщения** (англ. *message body*) данные, которые необходимо передать.

### 2.1. Стартовая строка

Как видно из таблицы, стартовые строки отличаются для запроса и ответа:

№		1	2	3
Запрос	Формат	METHOD	URI	HTTP/Version
	Пояснение	Метод. Операция, которую необходимо выполнить над ресурсом.	Идентифицирует ресурс.	Версия используемого протокола. Всегда добавляется к сообщению, начиная с версии 1.0.
Ответ	Формат	HTTP/Version	STATUS CODE	REASON
	Пояснение	Аналогично сообщению запроса.	Код состояния — это трехзначное целое число. Указывает на состояние запроса.	Текстовое пояснение от сервера о состоянии запроса. Необязательное поле.

#### 2.1.1. Методы

[Спецификацией HTTP](#) определены 9 методов. По соглашению эти методы определяются заглавными ASCII-символами. Кроме того, допускается реализация собственных методов. При этом все методы регистрозависимы. Наиболее частоиспользуемые методы — это:

- GET — используется, чтобы запросить указанный URI ресурс;
- HEAD — аналогичен GET за исключением того, что ответ сервера не будет содержать тело. Используется для извлечения информации о доступности ресурса, дате его изменения, а также прочей информации о нём.
- POST — просит сервер принять тело запроса в качестве ресурса с указанным URI.

Все серверы обязаны поддерживать методы GET и HEAD, все остальные методы опциональны.

### 2.1.2. Коды состояния

Также спецификацией [определено](#) множество кодов состояния. В зависимости от кода состояния клиент определяет, какие действия ему необходимо предпринять. Первая цифра кода указывает на класс ответа: 1xx — *информационный* (не содержит тело), 2xx — *успех*, 3xx — *перенаправление*, 4xx — *ошибка клиента*, 5xx — *ошибка сервера*.

### 2.2. Заголовки

Заголовки, по своей сути, уточняют контекст запросов и ответов, а синтаксически выглядят как пара имя и значение, разделенные двоеточием. В отличие от наименований методов, к регистру не чувствительны. Примеры некоторых стандартных заголовков:

- *Content-Type* — позволяет определить формат и способ обработки тела сообщения;
- *Content-Encoding* — указывает на кодировку тела сообщения;
- *Content-Length* — указывает на длину тела в байтах;
- *Accept-Ranges* — в ответе сервера указывает на возможность запрашивать фрагменты ресурса, указывая их смещения от начала файла в байтах;
- *Range* — используется для получения фрагмента ресурса: в сообщении указывается запрашиваемый диапазон байт. GET-запрос, в котором присутствует заголовок Range, называют частичным.
- *If-Modified-Since* — значение данного заголовка содержит дату. Сервер присылает ресурс в ответ на запрос только в том случае, если ресурс изменился после указанной даты. GET-запрос, если в него включен данный заголовок, называют условным.
- *ETag* — данный заголовок содержит идентификатор определенной версии ресурса. Наряду с *If-Modified-Since* может использоваться для актуализации кэшированных данных.

## 3. HTTP-сессии и cookie

Группу последовательно отправленных сообщений «запрос-ответ» называют HTTP-сессией. HTTP является протоколом без сохранения состояния: это означает, что сервер не обязан сохранять информацию о том, какие запросы выполнял клиент ранее в течение сессии. Тем не менее, существует механизм, который позволяет сохранять персональные настройки, аутентифицировать пользователя и отслеживать период действия его авторизации — куки (англ. *cookie*). Куки представляют собой небольшой фрагмент данных, отправляемых с сервером и сохраняемых браузером для дальнейшего использования. При дальнейших посещениях сервера браузер пересылает этот фрагмент данных обратно.