

HTTP

HTTP (англ. *HyperText Transfer Protocol* — «протокол передачи гипертекста») — протокол прикладного уровня передачи данных, изначально — в виде гипертекстовых документов в формате HTM, в настоящее время используется для передачи произвольных данных.

Структура протокола определяет, что каждое HTTP-сообщение состоит из трёх частей, которые передаются в следующем порядке:

1. **Стартовая строка** (англ. *Starting line*) — определяет тип сообщения;
2. **Заголовки** (англ. *Headers*) — характеризуют тело сообщения, параметры передачи и прочие сведения;
3. **Тело сообщения** (англ. *Message Body*) — непосредственно данные сообщения. Обязательно должно отделяться от заголовков пустой строкой.

CRUD — это аббревиатура от Create, Read, Update и Delete.

Когда вы работаете с веб-службами, CRUD соответствует следующим HTTP-методам, которые используются, чтобы сообщить веб-серверу, как вы хотите взаимодействовать с веб-сайтом:

- Создать: POST.
- Читать: GET.
- Обновление: PUT.
- Удалить: DELETE.

Методы протокола

Метод HTTP — последовательность из любых символов, кроме управляющих и разделителей, указывающая на основную операцию над ресурсом.

Метод	Краткое описание
-------	------------------

Метод	Краткое описание
OPTIONS	Используется для определения возможностей веб-сервера или параметров соединения для конкретного ресурса. Предполагается, что запрос клиента может содержать тело сообщения для указания интересующих его сведений. Формат тела и порядок работы с ним в настоящий момент не определён. Сервер пока должен его игнорировать. Аналогичная ситуация и с телом в ответе сервера. Для того чтобы узнать возможности всего сервера, клиент должен указать в URI звёздочку — «*». Запросы «OPTIONS * HTTP/1.1» могут также применяться для проверки работоспособности сервера (аналогично «пингованию») и тестирования на предмет поддержки сервером протокола HTTP версии 1.1. Результат выполнения этого метода не кэшируется.
GET	Используется для запроса содержимого указанного ресурса. Получает ресурсы. 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¶m2=value2 HTTP/1.1 Согласно стандарту HTTP, запросы типа GET считаются идемпотентными[4] — многократное повторение одного и того же запроса GET должно приводить к одинаковым результатам (при условии, что сам ресурс не изменился за время между запросами). Это позволяет кэшировать ответы на запросы GET. Кроме обычного метода GET, различают ещё условный GET и частичный GET. Условные запросы GET содержат заголовки If-Modified-Since, If-Match, If-Range и подобные. Частичные GET содержат в запросе Range. Порядок выполнения подобных запросов определён стандартами отдельно.
HEAD	Аналогичен методу GET, за исключением того, что в ответе сервера отсутствует тело. Запрос HEAD обычно применяется для извлечения метаданных, проверки наличия ресурса (валидация URL) и чтобы узнать, не изменился ли он с момента последнего обращения. Заголовки ответа могут кэшироваться. При несовпадении метаданных ресурса с соответствующей информацией в кэше копия ресурса помечается как устаревшая.

Метод	Краткое описание
POST	<p>Применяется для передачи пользовательских данных заданному ресурсу. Создает ресурсы. Например, в блогах посетители обычно могут вводить свои комментарии к записям в HTML-форму, после чего они передаются серверу методом POST и он помещает их на страницу. При этом передаваемые данные (в примере с блогами — текст комментария) включаются в тело запроса. Аналогично с помощью метода POST обычно загружаются файлы. В отличие от метода GET, метод POST не считается идемпотентным[4], то есть многократное повторение одних и тех же запросов POST может возвращать разные результаты (например, после каждой отправки комментария будет появляться одна копия этого комментария). При результатах выполнения 200 (Ok) и 204 (No Content) 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. Сообщение ответа сервера на выполнение метода POST не кэшируется.</p>
PUT	<p>Применяется для загрузки содержимого запроса на указанный в запросе URI. Обновляет ресурсы. Если по заданному URI не существовало ресурса, то сервер создаёт его и возвращает статус 201 (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 соответствует находящемуся по данному URI ресурсу. Сообщения ответов сервера на метод PUT не кэшируются.</p>
DELETE	Удаляет указанный ресурс.
TRACE	Возвращает полученный запрос так, что клиент может увидеть, что промежуточные сервера добавляют или изменяют в запросе.
CONNECT	<p>Системный вызов connect при работе с TCP-сокетами служит для установления логического соединения со стороны клиента. Вызов connect скрывает внутри себя настройку сокета на выбранный системой порт и произвольный сетевой интерфейс (по сути дела, вызов bind с нулевым номером порта и IP-адресом INADDR_ANY).</p>

Ответы HTTP - сервера

Код ответа (состояния) HTTP показывает, был ли успешно выполнен определённый HTTP запрос. Коды сгруппированы в **5 классов**:

1. Информационные 100 - 199
2. Успешные 200 - 299
3. Перенаправления 300 - 399
4. Клиентские ошибки 400 - 499
5. Серверные ошибки 500 - 599

HTTP (англ. *HyperText Transfer Protocol* — «протокол передачи гипертекста») — протокол прикладного уровня передачи данных, изначально — в виде гипертекстовых документов в формате HTM, в настоящее время используется для передачи произвольных данных.

Ответы HTTP - сервера

Следующая таблица содержит список всех кодов и их значения:

Код ответа	Название	Описание	Версия HTTP
100	Continue	"Продолжить". Этот промежуточный ответ указывает, что запрос успешно принят и клиент может продолжать присылать запросы либо проигнорировать этот ответ, если запрос был завершён.	Только HTTP/1.1
101	Switching Protocol	"Переключение протокола". Этот код присылается в ответ на запрос клиента, содержащий заголовок Upgrade: , и указывает, что сервер переключился на протокол, который был указан в заголовке. Эта возможность позволяет перейти на несовместимую версию протокола и обычно не используется.	Только HTTP/1.1

Код ответа	Название	Описание	Версия HTTP
102	Processing	"В обработке". Этот код указывает, что сервер получил запрос и обрабатывает его, но обработка ещё не завершена.	Только HTTP/1.1
103	Early Hints	"Ранние подсказки". В ответе сообщаются ресурсы, которые могут быть загружены заранее, пока сервер будет подготавливать основной ответ. RFC 8297 (Experimental) .	Только HTTP/1.1
200	OK	"Успешно". Запрос успешно обработан. Что значит "успешно", зависит от метода HTTP, который был запрошен: <ul style="list-style-type: none"> • GET: "ПОЛУЧИТЬ". Запрошенный ресурс был найден и передан в теле ответа. • HEAD: "ЗАГОЛОВОК". Заголовки переданы в ответе. • POST: "ПОСЫЛКА". Ресурс, описывающий результат действия сервера на запрос, передан в теле ответа. • TRACE: "ОТСЛЕЖИВАТЬ". Тело ответа содержит тело запроса полученного сервером. 	HTTP/0.9 и выше
201	Created	"Создано". Запрос успешно выполнен и в результате был создан ресурс. Этот код обычно присылается в ответ на запрос PUT "ПОМЕСТИТЬ".	HTTP/0.9 и выше

Код ответа	Название	Описание	Версия HTTP
202	Accepted	"Принято". Запрос принят, но ещё не обработан. Не поддерживаемо, т.е., нет способа с помощью HTTP отправить асинхронный ответ позже, который будет показывать итог обработки запроса. Это предназначено для случаев, когда запрос обрабатывается другим процессом или сервером, либо для пакетной обработки.	HTTP/0.9 и выше
203	Non-Authoritative Information	"Информация не авторитетна". Этот код ответа означает, что информация, которая возвращена, была предоставлена не от исходного сервера, а из какого-нибудь другого источника. Во всех остальных ситуациях более предпочтителен код ответа 200 ОК.	HTTP/0.9 и 1.1
204	No Content	"Нет содержимого". Нет содержимого для ответа на запрос, но заголовки ответа, которые могут быть полезны, присылаются. Клиент может использовать их для обновления кешированных заголовков полученных ранее для этого ресурса.	HTTP/0.9 и выше
205	Reset Content	"Сбросить содержимое". Этот код присылается, когда запрос обработан, чтобы сообщить клиенту, что необходимо сбросить отображение документа, который прислал этот запрос.	Только HTTP/1.1
206	Partial Content	"Частичное содержимое". Этот код ответа используется, когда клиент присылает заголовок диапазона, чтобы выполнить загрузку отдельно, в несколько потоков.	Только HTTP/1.1

Код ответа	Название	Описание	Версия HTTP
300	Multiple Choice	"Множественный выбор". Этот код ответа присылается, когда запрос имеет более чем один из возможных ответов. И User-agent или пользователь должен выбрать один из ответов. Не существует стандартизированного способа выбора одного из полученных ответов.	HTTP/1.0 и выше
301	Moved Permanently	"Перемещён на постоянной основе". Этот код ответа значит, что URI запрашиваемого ресурса был изменён. Возможно, новый URI будет предоставлен в ответе.	HTTP/0.9 и выше
302	Found	"Найдено". Этот код ответа значит, что запрошенный ресурс <i>временно изменён</i> . Новые изменения в URI могут быть доступны в будущем. Таким образом, этот URI, должен быть использован клиентом в будущих запросах.	HTTP/0.9 и выше
303	See Other	"Просмотр других ресурсов". Этот код ответа присылается, чтобы направлять клиента для получения запрашиваемого ресурса в другой URI с запросом GET.	HTTP/0.9 и 1.1
304	Not Modified	"Не модифицировано". Используется для кеширования. Это код ответа значит, что запрошенный ресурс не был изменён. Таким образом, клиент может продолжать использовать кешированную версию ответа.	HTTP/0.9 и выше

Код ответа	Название	Описание	Версия HTTP
305	Use Proxy	"Использовать прокси". Это означает, что запрошенный ресурс должен быть доступен через прокси. Этот код ответа в основном не поддерживается из соображений безопасности.	Только HTTP/1.1
306	Switch Proxy	Больше не использовать. Изначально подразумевалось, что "последующие запросы должны использовать указанный прокси."	Только HTTP/1.1
307	Temporary Redirect	"Временное перенаправление". Сервер отправил этот ответ, чтобы клиент получил запрошенный ресурс на другой URL-адрес с тем же методом, который использовал предыдущий запрос. Данный код имеет ту же семантику, что код ответа 302 Found , за исключением того, что агент пользователя не должен изменять используемый метод HTTP: если в первом запросе использовался POST , то во втором запросе также должен использоваться POST .	Только HTTP/1.1
308	Permanent Redirect	"Перенаправление на постоянной основе". Это означает, что ресурс теперь постоянно находится в другом URI, указанном в заголовке Location: HTTP Response. Данный код ответа имеет ту же семантику, что и код ответа 301 Moved Permanently , за исключением того, что агент пользователя не должен изменять используемый метод HTTP: если POST использовался в первом запросе, POST должен использоваться и во втором запросе.	Только HTTP/1.1

Код ответа	Название	Описание	Версия HTTP
400	Bad Request	"Плохой запрос". Этот ответ означает, что сервер не понимает запрос из-за неверного синтаксиса.	HTTP/0.9 и выше
401	Unauthorized	"Неавторизованно". Для получения запрашиваемого ответа нужна аутентификация. Статус похож на статус 403, но, в этом случае, аутентификация возможна.	HTTP/0.9 и выше
402	Payment Required	"Необходима оплата". Этот код ответа зарезервирован для будущего использования. Первоначальная цель для создания этого кода была в использовании его для цифровых платёжных систем(на данный момент не используется).	HTTP/0.9 и 1.1
403	Forbidden	"Запрещено". У клиента нет прав доступа к содержимому, поэтому сервер отказывается дать надлежащий ответ.	HTTP/0.9 и выше
404	Not Found	"Не найден". Сервер не может найти запрашиваемый ресурс. Код этого ответа, наверно, самый известный из-за частоты его появления в вебе.	HTTP/0.9 и выше
405	Method Not Allowed	"Метод не разрешён". Сервер знает о запрашиваемом методе, но он был деактивирован и не может быть использован. Два обязательных метода, GET и HEAD , никогда не должны быть деактивированы и не должны возвращать этот код ошибки.	Только HTTP/1.1

Код ответа	Название	Описание	Версия HTTP
406	Not Acceptable	Этот ответ отсылается, когда веб сервер после выполнения <u>server-driven content negotiation</u> , не нашёл контента, отвечающего критериям, полученным из user agent.	Только HTTP/1.1
407	Proxy Authentication Required	Этот код ответа аналогичен коду 401, только аутентификация требуется для прокси сервера.	Только HTTP/1.1
408	Request Timeout	Ответ с таким кодом может прийти, даже без предшествующего запроса. Он означает, что сервер хотел бы отключить это неиспользуемое соединение. Этот метод используется все чаще с тех пор, как некоторые браузеры, вроде Chrome и IE9, стали использовать <u>HTTP механизмы предварительного соединения</u> для ускорения сёрфинга (смотрите bug 634278 , будущей реализации этого механизма в Firefox). Также учитывайте, что некоторые серверы прерывают соединения не отправляя подобных сообщений.	Только HTTP/1.1
409	Conflict	Этот ответ отсылается, когда запрос конфликтует с текущим состоянием сервера.	Только HTTP/1.1
410	Gone	Этот ответ отсылается, когда запрашиваемый контент удалён с сервера.	Только HTTP/1.1
411	Length Required	Запрос отклонён, потому что сервер требует указание заголовка Content-Length , но он не указан.	Только HTTP/1.1
412	Precondition Failed	Клиент указал в своих заголовках условия, которые сервер не может выполнить	Только HTTP/1.1

Код ответа	Название	Описание	Версия HTTP
413	Request Entity Too Large	Размер запроса превышает лимит, объявленный сервером. Сервер может закрыть соединение, вернув заголовок <code>Retry-After</code>	Только HTTP/1.1
414	Request-URI Too Long	URI запрашиваемый клиентом слишком длинный для того, чтобы сервер смог его обработать	Только HTTP/1.1
415	Unsupported Media Type	Медиа формат запрашиваемых данных не поддерживается сервером, поэтому запрос отклонён	Только HTTP/1.1
416	Requested Range Not Satisfiable	Диапазон указанный заголовком запроса <code>Range</code> не может быть выполнен; возможно, он выходит за пределы переданного URI	Только HTTP/1.1
417	Expectation Failed	Этот код ответа означает, что ожидание, полученное из заголовка запроса <code>Expect</code> , не может быть выполнено сервером.	Только HTTP/1.1
500	Internal Server Error	"Внутренняя ошибка сервера". Сервер столкнулся с ситуацией, которую он не знает как обработать.	HTTP/0.9 и выше
501	Not Implemented	"Не выполнено". Метод запроса не поддерживается сервером и не может быть обработан. Единственные методы, которые сервера должны поддерживать (и, соответственно, не должны возвращать этот код) - <code>GET</code> и <code>HEAD</code> .	HTTP/0.9 и выше
502	Bad Gateway	"Плохой шлюз". Эта ошибка означает что сервер, во время работы в качестве шлюза для получения ответа, нужного для обработки запроса, получил недействительный (недопустимый) ответ.	HTTP/0.9 и выше

Код ответа	Название	Описание	Версия HTTP
503	Service Unavailable	"Сервис недоступен". Сервер не готов обрабатывать запрос. Зачастую причинами являются отключение сервера или то, что он перегружен. Обратите внимание, что вместе с этим ответом удобная для пользователей(user-friendly) страница должна отправлять объяснение проблемы. Этот ответ должен использоваться для временных условий и Retry-After: HTTP-заголовок должен, если возможно, содержать предполагаемое время до восстановления сервиса. Веб-мастер также должен позаботиться о заголовках, связанных с кешем, которые отправляются вместе с этим ответом, так как эти ответы, связанные с временными условиями, обычно не должны кешироваться.	HTTP/0.9 и выше
504	Gateway Timeout	Этот ответ об ошибке предоставляется, когда сервер действует как шлюз и не может получить ответ вовремя.	Только HTTP/1.1
505	HTTP Version Not Supported	"HTTP-версия не поддерживается". HTTP-версия, используемая в запросе, не поддерживается сервером.	Только HTTP/1.1