

HTTP, HTTPS, FTP



Протокол HTTP

HTTP — протокол прикладного уровня поверх TCP/IP, используемый для передачи гипертекста в WWW и локальных сетях. Помимо передачи содержимого веб-страниц, протокол используется приложениями для обмена информацией. В основе протокола лежат запросы, заголовки и коды результатов. В протоколе всегда выражены 2 стороны - сервер и клиент.

Клиент передает запрос в виде:
начальная строка, заголовок (или заголовки), тело сообщения.

Сервер возвращает результат в виде:
начальная строка с кодом результата (или ошибки), заголовок (или заголовки), тело сообщения.

HTTP и HTTPS

HTTP (HyperText Transfer Protocol — протокол передачи гипертекста) — это прикладной протокол передачи данных в сети. На текущий момент используется для получения информации с веб-сайтов. Протокол HTTP основан на использовании технологии «клиент-сервер»: клиент, отправляющий запрос, является инициатором соединения; сервер, получающий запрос, выполняет его и отправляет клиенту результат.

HTTPS (от англ. HyperText Transfer Protocol Secure — безопасный протокол передачи гипертекста) — это расширение протокола HTTP, поддерживающее шифрование посредством криптографических протоколов SSL и TLS.

HTTP и HTTPS

Отличия HTTP и HTTPS

1. В протоколах используются разные порты: в HTTP применяется порт 80, в HTTPS — 443.
2. В HTTPS применяется шифрование данных, которые затем расшифровываются с помощью ключей на сервере.
3. HTTPS обеспечивает сохранность данных. Если произойдет какое-то изменение передаваемых сведений, это будет зафиксировано.
4. Безопасный протокол гарантирует аутентификацию — попадание пользователей именно на тот ресурс, который необходим, это обеспечивает борьбу с мошенническими вмешательствами.
5. Специалисты Google **рекомендуют** использовать на любых сайтах HTTPS, это положительно **влияет** на позиции проекта в поисковой выдаче.
6. При переходе на сайты с HTTP браузер отображает сообщение о небезопасном подключении.
7. На шифрование в HTTPS тратится часть ресурсов сервера, поэтому загрузка сайта может немного замедлиться. Чтобы избежать проблем с долгой загрузкой страниц, стоит настроить кэширование и оптимизировать изображения.

Протокол HTTP

Клиентами в WWW являются браузеры, например Google Chrome или Microsoft Edge. В качестве серверов могут выступать HTTP-сервера, например Apache.

Начальная строка запроса клиента - это метод, запрашиваемый URI, и версия протокола (HTTP-version). HTTP-заголовки можно разделить на 3 группы - заголовки запроса, заголовки ответа и заголовки, которые встречаются и в запросе, и в ответе. Поля заголовка запроса позволяют клиенту передавать серверу дополнительную информацию о запросе и о самом себе.

Заголовки запросов

- **Accept Charset** — поддерживаемая кодировка. Имеет значение для сервера, который может выдавать один и тот же документ в разных кодировках
- **Accept Language** — поддерживаемый язык. Имеет значение для сервера, который может выдавать один и тот же документ в разных языковых версиях
- **Host** — имя хоста, с которого запрашивается ресурс
- **Referer** — URL, с которого перешли на этот ресурс
- **User Agent** — браузер

Коды ответов сервера

Начальная строка ответа сервера - это строка состояния (Status Line). Она состоит из версии протокола (HTTP-version), числового кода состояния (Status Code) и поясняющей фразы (Reason Phrase). В общей сложности есть пять категорий статуса HTTP, каждая из которых идет с шагом в 100 единиц.

Information (100 - 199)

Коды в этом диапазоне в буквальном смысле звучат как фраза "а дальше...". Это как первые свидания – вы пытаетесь получить как можно больше информации, чтоб знать что делать дальше.

Пример запроса и ответа

Пример HTTP-запроса:

```
GET /default.aspx HTTP/1.1
```

Пример HTTP-ответа:

```
HTTP/1.1 200 OK
Date: Wed, 11 Feb 2009 11:20:59 GMT
Server: Apache
X-Powered-By: PHP/5.2.4-2ubuntu5wm1
Last-Modified: Wed, 11 Feb 2009 11:20:59 GMT
Content-Language: ru
Content-Type: text/html; charset=utf-8
Content-Length: 1234
Connection: close
```

(далее следует запрошенная страница в HTML)

GET / POST

HTTP-методы GET и POST — самые распространённые способы отправить или получить данные с сервера. Но в разных случаях оба метода могут быть небезопасными или неудобными в использовании. В этой заметке рассмотрим, какой метод когда использовать.

GET — метод для чтения данных с сайта. Например, для доступа к указанной странице. Он говорит серверу, что клиент хочет прочитать указанный документ. На практике этот метод используется чаще всего, например, в интернет-магазинах на странице каталога. Фильтры, которые выбирает пользователь, передаются через метод **GET**.

```
:authority: htmlacademy.ru  
:method: GET  
:path: /tutorial/php/http-header
```

POST — метод для отправки данных на сайт. Чаще всего с помощью метода **POST** передаются формы.

```
URL-адрес запроса: https://htmlacademy.ru/consulting  
Метод запроса: POST  
Код состояния: 200
```

Получение HTML страницы

<https://git.io/JGCJO>

Получение файла с хостинга

<https://git.io/JGCUL>

FTP

FTP (File Transfer Protocol) — протокол передачи больших двоичных файлов по сети, появившийся в 1971 году задолго до HTTP и даже до TCP/IP, благодаря чему является одним из старейших прикладных протоколов. На сегодняшний день широко используется для распространения ПО и доступа к удалённым хостам. Гарантирует передачу файла (либо выдачу ошибки).

Протокол построен на архитектуре «клиент-сервер». Пользователи FTP могут пройти аутентификацию, передавая логин и пароль открытым текстом, или же, если это разрешено на сервере, они могут подключиться анонимно. Можно использовать протокол SSH для безопасной передачи, скрывающей (шифрующей) логин и пароль, а также шифрующей содержимое.

Практика

- Выполнить запрос на получение контента страницы с сайта gismeteo, и отобразить в консоли текущую температуру воздуха за окном по Одессе.

Практика

- Добавить прогноз погоды на завтра. Информация должна включать погодные условия (солнечно/облачно/туман), скорость ветра, влажность воздуха, температуру воды в Чёрном море.