45. **Протокол HTTP. Формат запроса**

* Протокол передачи файлов. Но данные могут передаваться по разному, как передача объектов или гипертекстовой информации
* HTTP/1.0 RFC 1945, 1996 г.
* HTTP/1.1 RFC 2068, 1997 г.
* Действующий стандарт HTTP/1.1 RFC 2616, 1999г.

HTTP запрос формируется как запрос состоящий из заголовков и тела запроса следующего формата



Сначала идет некоторое колисчетсво заголовков потом пустая строка и потом тело. Все что <> кроме первой строчки это целый список возможных заголовков. Особенность HTTP в том, что у него может быть очень много заголовков при помози которых мы можем управлять тем, что происходит в сети

<Request-line>- строка запроса. Используется только в запросе

<General-header>- общий заголовок Используется только в запросе

<Request-header>- заголовок запроса Может быть и в запросе и в ответе

<Entity-header>- заголовок сообщения Может быть и в запросе и в ответе

<Body>- тело



HTTPзапрос начинается со строки запроса. Имеет 3 параметра

Формат: < METHOD > < URL > < HTTP-VERSION >

< METHOD > - что мы хотим сделать с ресурсом

< URL > - это ресурс который мы запрашиваем. В большинстве случаев ресурс HTTP-шный поэтому это скорее всего адрес веб страницы или ресурса на ней

< HTTP-VERSION > -поддерживаемая версия HTTP. 1.0, 0.9, 1.1, 2.0

Методы:

* GET – запросить с помощью HTTP ресурс
* POST – передать в ресурс какие-то параметры в теле запроса. Например для передачи данных для поискового сервера
* HEAD – запросить только заголовок ресурса, например если ресурс не изменился и находиться в кэше
* PUT - поместить ресурс
* DELETE – удалить ресурс

PUT и DELETE используются в системах публикации

* OPTIONS – передача опций
* и т.п.

URL: ◦ [method://][user[:pass]@]host[:port][/path][?name=val{&name=val}][#anchor]

URL – стандартизированный по RFC формат доступа к ресурсу

Версия: ◦ HTTP/1.0, HTTP/1.1 или HTTP/2 (new)



Общий заголовок (General-header)

Присутствует, когда есть тело сообщения

* Connection – должны ли сохранять соединение. В отличии от всех протоколов которые поддерживают состояние соединения штатный вариант работы HTTP это соединились, послали запрос, получили ответ и разорвали соединение, то есть это протокол без сохранения соединения чтобы не использовать ресурсы сервера. В Connection можно передать опцию которая соохранит соединение но это внештатный режим работы
* Data
* Pragma
* Transfer-encoding
* Upgrade
* no-cache
* И т.д.

**Заголовок запроса (Request-header)**

* Accept: принимаемый контент
* Accept-Charset: принимаемый набор символов
* Accept-Encoding: compress, zip
* Accept-Language: da, ru

Заголовки Accept указывают на те параметры контента которые приемлмы для клиента. Например чтобы передавали страницу на русском языке. Так работает что одну страничку выдают разным пользователям на разном языке

* Authorization: basic xxx=\*\*\*\*\*\*. Один из способов передачи логина и пароля на сервер. После Authorization используется ключевое слово как здесь basic
* From:
* Host:
* If-modified-since:… -выдать стриницу если она модифицированна только с какого-то времени
* Referrer: - для построение цепоцек связи. Показывает откуда мы пришли на текущую страницку
* User-agent: - что за пользовательский агент у нас используется это тип браузера, но при этом всегда првый текст что они предают это что браузер mozila
* И т.д.

**Заголовок сообщения (Entity-header)**

* Allow: GET, POST, HEAD – какие команды поддерживает
* Content-Encoding: x-zip
* Content-Language:
* Content-Length: 1245
* Content-Type: ...text/html; charset=win-1251

Content описывает то, что находиться в теле

* Expires:
* Last-Modified: