Заголовки откликов

1. **ETag** или **entity tag** — один из регламентируемых спецификацией [RFC 7232](https://tools.ietf.org/html/rfc7232), служебных заголовков протокола [HTTP/1.1](https://ru.wikipedia.org/wiki/HTTP), который может быть установлен [веб-сервером](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) в фазе формирования ответа, на полученный от [клиента](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80#%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82%D1%8B) запрос. Содержимое заголовка ETag является идентификатором, значение которого прямо зависит от состояния загружаемого [клиентом](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80#%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82%D1%8B) ресурса. В дальнейшем, этот идентификатор, используется с целью актуализации состояния загруженного ресурса его оригиналу, расположенному на [Веб-сервере](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80). Что достигается путём отправки серверу [HTTP/1.1](https://ru.wikipedia.org/wiki/HTTP) запроса с указанием ETag идентификатора как значении заголовка - [If-None-Match](https://tools.ietf.org/html/rfc7232" \l "section-3.2" \o "rfc:7232). Сервер, обнаружив такой заголовок, на основании сравнения его значения с текущим состоянием ресурса сообщает клиенту о том, что копия, хранящаяся в кэше клиента, актуальна т.е. необходимости в повторной загрузке нет, или, в противном случае, необходима загрузка актуальной версии.
2. nnCoection – это тег connection, но это не опечатка, а он специально изменен, чтобы упрощать поднятие упавшего сервера
3. nel: Заголовок nel используется для настройки ведения журнала сетевых запросов.
4. P3P это [протокол](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB_%D0%BF%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D1%87%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), позволяющий [сайтам](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82) информировать [браузер](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) о предполагаемом получении личных данных пользователя. Протокол был разработан [Консорциумом Всемирной паутины](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%81%D0%BE%D1%80%D1%86%D0%B8%D1%83%D0%BC_%D0%92%D1%81%D0%B5%D0%BC%D0%B8%D1%80%D0%BD%D0%BE%D0%B9_%D0%BF%D0%B0%D1%83%D1%82%D0%B8%D0%BD%D1%8B) (W3C) для того, чтобы предоставить пользователям больший контроль над персональной информацией во время просмотра веб-сайтов.
5. Pragma Общий заголовок **Pragma** HTTP / 1.0 - это заголовок, зависящий от реализации, который может иметь различные эффекты в цепочке запрос-ответ. Он используется для обратной совместимости с кешами HTTP / 1.0, где заголовок Cache-Control HTTP / 1.1 ещё не присутствует.
6. Timing-Allow-origin позволяет origin видеть значения атрибутов полученных с помощью свойств Resourse timing ipa
7. x-frontend – как я понял это заголовок который возвращает тип используемого фронтэнда
8. **Strict-Transport-Security**  - заголовок ответа (часто используется аббревиатура [HSTS (en-US)](https://developer.mozilla.org/en-US/docs/Glossary/HSTS)), позволяющий web-сайтам уведомить браузер о том, что доступ к ним должен быть осуществлён только посредством HTTPS вместо HTTP.
9. Vary Заголовок Vary используется для отображения запрашиваемого содержимого страницы при ее кэшировании. Наиболее актуально использование данного заголовка при наличии нескольких версий динамических или статических страниц под одним URL.
10. Via заголовок который может быть и в откликах и в запросах нужен для отслеживания пути сообщений, чтобы избегать циклов и определять возможности протоколов запроса/отклика

Заголовки запросов

1)Downlink – показывает пропускную способность соединения клиента с сервером

2)ect – показывает эффективный тип соединения с сервером

3) rtt предоставляет приблизительное время приема-передачи на уровне приложения в миллисекундах, включает время обработки сервера

4) **Upgrade-Insecure-Requests** отправляет сигнал на сервер, выражающий предпочтение клиента в отношении зашифрованного и аутентифицированного ответа, и что он может успешно обработать директиву csp

5)sec-fetch-site указывает на взаимосвязь между происхождения на просьбу инициатора и происхождения запрашиваемого ресурса.**Sec-Fetch-Site** Другими словами, этот заголовок сообщает серверу, исходит ли запрос ресурса из того же источника, того же сайта, другого сайта или является запросом, инициированным пользователем. Затем сервер может использовать эту информацию, чтобы решить, следует ли разрешить запрос.

6) **Service worker** фактически выполняет роль прокси-сервера, находящегося между веб-приложением и браузером, а также сетью (если доступна). Он позволяет (кроме прочего) описывать корректное поведение веб-приложения в режиме офлайн, перехватывать запросы сети и принимать соответствующие меры, основываясь на доступности сети, и обновлять данные, находящиеся на сервере при доступе к нему. Также они имеют доступ к push-уведомлениям и API для фоновой синхронизации

7)**Accept language** Запрос **Accept-Language** сообщает серверу, какие языки клиент понимает и какая локаль предпочтительнее (имеются в виду естественные языки, такие как английский, а не языки программирования).

8)cookie хранит в себе куки которые ассоциируются с сервером

9) **x-csrf-token** Этот токен используется для проверки того, что именно авторизованный пользователь делает запрос в приложение.

10) **sec-ch-ua-platform** показывает серверу платформу пользователя