**Способы ускорения обработки запросов (отправка данных авторизации с запросом)**

HTTP authentification

Fротокол, описанный в стандартах HTTP 1.0/1.1, существует давно и до сих пор активно применяется в корпоративной среде. Применительно к веб-сайтам работает следующим образом:

1. Сервер, при обращении неавторизованного клиента к защищенному ресурсу, отсылает HTTP статус “401 Unauthorized” и добавляет заголовок “WWW-Authenticate” с указанием схемы и параметров аутентификации;
2. Браузер, при получении такого ответа, автоматически показывает диалог ввода username и password. Пользователь вводит детали своей учетной записи;
3. Во всех последующих запросах к этому веб-сайту браузер автоматически добавляет HTTP заголовок “Authorization”, в котором передаются данные пользователя для аутентификации сервером;
4. Сервер аутентифицирует пользователя по данным из этого заголовка. Решение о предоставлении доступа (авторизация) производится отдельно на основании роли пользователя, ACL или других данных учетной записи.

**В чем преимущества WebSocket перед HTTP запросами.**

WebSockets позволяет как серверу, так и клиенту отправлять сообщения в любое время без какого-либо отношения к предыдущему запросу. WebSocket решает несколько проблем с HTTP:

1. Двунаправленный протокол. Любой клиент / сервер может отправить сообщение другой стороне (в HTTP запрос всегда инициируется клиентом, а ответ обрабатывается сервером, что делает HTTP однонаправленным протоколом);
2. Полнодуплексная связь. Клиент и сервер могут одновременно разговаривать друг с другом независимо;
3. Единое TCP-соединение. Клиент и сервер обмениваются данными через одно и то же TCP-соединение (постоянное соединение) на протяжении всего жизненного цикла соединения WebSocket.

Плюсы WebSocket

1. WebSocket — это протокол, управляемый событиями, что означает, что вы можете использовать его для общения в реальном времени. В отличие от HTTP, где вам нужно постоянно запрашивать обновления, с веб-сокетами изменения отправляются немедленно, как только они станут доступны;
2. WebSockets поддерживает единое постоянное соединение открытым, устраняя проблемы с задержкой, возникающие при использовании методов HTTP-запросов / ответов;
3. WebSocket обычно не использует XMLHttpRequest, и поэтому заголовки не отправляются каждый раз, когда нам нужно получить дополнительную информацию с сервера. Это, в свою очередь, снижает объем данных, отправляемых на сервер.

**Каким образом реализуется передача данных асинхронно в рамках HTTP протокола.**

HTTP Polling

Клиентские устройства отправляют запросы к серверу через фиксированные интервалы. Однако сервер по-прежнему будет немедленно отвечать на каждый запрос, либо предоставляя новые данные, либо отправляя «пустой» ответ, если новых данных нет.

HTTP Long Polling

Сервер предпочитает держать клиентское соединение открытым как можно дольше, доставляя ответ только после того, как данные становятся доступным или истек порог тайм-аута. Недостатком являются дополнительные расходы, связанные с обработкой HTTP-запросов, которые могут создать множество проблем при большом масштабе

HTTP Streaming

Передача данных в стиле push, которая позволяет веб-серверу непрерывно отправлять данные клиенту по одному HTTP-соединению, которое остается открытым бесконечно. По сути, клиент делает HTTP-запрос, а сервер отправляет ответ неопределённой длины. Основная проблема заключается в том, что посредник может прервать соединение.