

Еще один способ оптимистической блокировки - через **ETag** и **If-Match**.

Хеши ETag (Entity Tag) генерируются сервером на основе содержимого ресурса. Когда клиент запрашивает ресурс у сервера, сервер вычисляет хэш ETag на основе текущего состояния ресурса и включает его в заголовки ответа HTTP. Затем клиент может сохранить это значение ETag и использовать его в последующих запросах к тому же ресурсу для обнаружения изменений.

Вот как реализуется оптимистическая блокировка на основе ETag:

- Клиент запрашивает ресурс: Клиент посылает HTTP GET запрос на сервер для получения определенного ресурса.

GET /resource/1 HTTP/1.1

Хост: example.com

- Сервер вычисляет ETag и отправляет ресурс: сервер генерирует хэш ETag на основе содержимого запрашиваемого ресурса и включает его в заголовки HTTP-ответа вместе с ресурсом.

HTTP/1.1 200 OK

Content-Type: application/json

ETag: "5a1c9d4e9b56e"

```
{
  "id": 1,
  "name": "Образец ресурса",
  // другие поля ресурса...
}
```

- Клиент обновляет ресурс: клиент вносит изменения в ресурс и отправляет HTTP PUT или PATCH запрос для обновления ресурса на сервере. В этом запросе клиент включает ранее полученное значение ETag в заголовок If-Match, чтобы указать, что обновление должно быть применено только в том случае, если текущий ETag на сервере совпадает с предоставленным.

PUT /resource/1 HTTP/1.1

Хост: example.com

Content-Type: application/json

If-Match: "5a1c9d4e9b56e"

```
{  
  "id": 1,  
  
  "name": "Обновленный ресурс",  
  
  // обновленные поля ресурса...  
}
```

- Сервер проверяет ETag и обрабатывает обновление: сервер сравнивает значение ETag в заголовке If-Match с текущим значением ETag для ресурса.
 - Если значения ETag совпадают, сервер продолжает обновление и возвращает успешный ответ.
 - Если значения ETag не совпадают, это означает, что ресурс был изменен другим клиентом, и сервер возвращает статус 412 Precondition Failed, указывающий на конфликт.

Пример использования: вы создаете приложение для совместного редактирования документов, в котором несколько пользователей могут одновременно редактировать разные части документа, и детальное разрешение конфликтов имеет большое значение. Оптимистическая блокировка на основе ETag может помочь в этом сценарии, предоставляя более точную информацию о состоянии ресурса. Использование хэшей ETag позволяет более детально контролировать обновление ресурсов, поскольку значения ETag напрямую связаны с содержимым ресурса. В случае конфликта клиент может получить последнюю версию ресурса, разрешить конфликт и повторить обновление с обновленным значением Etag. Также вы легко можете использовать кеширование с помощью тех же условных запросов (мы изучали это в разделе про кеширование).