

Пессимистическая блокировка.

Это техника, которая может быть использована для обработки параллелизма в API путем блокировки ресурса во время его обновления, не позволяя другим пользователям или клиентам изменять его до тех пор, пока блокировка не будет снята.

Давайте рассмотрим пример, как пессимистическая блокировка может быть реализована в API с использованием методов запроса HTTP. Допустим, у нас есть API, который позволяет пользователям обновлять записи о клиентах. Чтобы реализовать пессимистическую блокировку, мы добавим блокировку к записи клиента, когда она будет обновляться, и снимем блокировку после завершения обновления. Вот как это можно сделать.

- Сначала клиент запрашивает блокировку записи клиента, отправляя POST-запрос к API с параметром запроса, указывающим, что запрашивается блокировка:

POST /customers/123?lock=true HTTP/1.1

- Сервер проверяет, не заблокирована ли запись клиента другим пользователем или клиентом. Если запись уже заблокирована, сервер возвращает ответ 409 Conflict, указывающий на то, что ресурс в настоящее время заблокирован:

HTTP/1.1 409 Conflict

- Если запись еще не заблокирована, сервер приобретает блокировку записи и возвращает ответ success с маркером блокировки, который клиент может использовать для снятия блокировки позже:

HTTP/1.1 200 OK

Content-Type: application/json

```
{  
  "lockToken": "abc123"  
}
```

- Теперь клиент может обновить запись клиента, отправив запрос PUT к API, включив обновленную запись в тело запроса вместе с маркером блокировки:

PUT /customers/123 HTTP/1.1

Content-Type: application/json

X-Lock-Token: abc123

```
{  
  "id": 123,  
  "name": "John Doe",  
  "email": "j.doe@example.com"  
}
```

- Сервер обновляет запись клиента и снимает блокировку, удаляя маркер блокировки:

HTTP/1.1 200 OK

Используя пессимистическую блокировку, мы можем обеспечить предотвращение конфликтов, не позволяя нескольким пользователям или клиентам одновременно изменять один и тот же ресурс. Однако этот подход может привести к снижению производительности и пропускной способности, если многие пользователи пытаются получить блокировки на один и тот же ресурс одновременно, поэтому его следует использовать при необходимости и аккуратно.