

Использование **транзакций**.

Транзакции позволяют сгруппировать несколько операций с данными в один атомарный блок. Это гарантирует, что либо все операции будут успешными, либо ни одна из них, и помогает поддерживать согласованность данных.

Использование транзакций - это техника, которая может быть использована для обеспечения согласованности данных в API путем группирования нескольких операций в одну атомарную транзакцию.

Ниже пример того, как транзакции могут быть реализованы в API с использованием методов запроса HTTP. Допустим, у нас есть API, который позволяет пользователям переводить средства между счетами. Чтобы обеспечить последовательность и атомарность перевода, мы можем сгруппировать операции снятия и внесения средств в одну транзакцию.

- Клиент начинает транзакцию, отправляя POST-запрос к API с параметром запроса, указывающим на то, что начинается транзакция:

POST /transactions?start=true HTTP/1.1

- Сервер отвечает идентификатором транзакции, который клиент может использовать для идентификации транзакции в последующих запросах:

HTTP/1.1 200 OK

Content-Type: application/json

```
{  
  
  "transactionId": "abc123"  
  
}
```

- Затем клиент отправляет запрос на снятие средств в API, включая в тело запроса идентификатор счета, сумму для снятия и идентификатор транзакции:

POST /accounts/123/withdrawals HTTP/1.1

Content-Type: application/json

```
{  
  
  "сумма": 100,
```

```
"transactionId": "abc123"
```

```
}
```

- Сервер проверяет, достаточно ли средств на счете, и если да, то снимает указанную сумму со счета:

HTTP/1.1 200 OK

- Затем клиент отправляет запрос на пополнение счета в API, включая в тело запроса идентификатор счета, сумму для пополнения и идентификатор транзакции:

POST /accounts/456/deposits HTTP/1.1

Content-Type: application/json

```
{
```

```
"сумма": 100,
```

```
"transactionId": "abc123"
```

```
}
```

- Сервер вносит указанную сумму на указанный счет.

HTTP/1.1 200 OK

- Наконец, клиент фиксирует транзакцию, отправляя POST-запрос к API с идентификатором транзакции и параметром запроса, указывающим, что транзакция должна быть зафиксирована:

POST /transactions?transactionId=abc123&commit=true HTTP/1.1

- Сервер проверяет, все ли операции в транзакции были успешными, и если это так, фиксирует транзакцию и возвращает ответ об успехе:

HTTP/1.1 200 OK

- Если какая-либо из операций в транзакции завершилась неудачно, клиент может прервать транзакцию, отправив запрос DELETE с идентификатором транзакции и параметром запроса, указывающим на необходимость отката транзакции:

DELETE /transactions?transactionId=abc123&rollback=true HTTP/1.1