

Снова про **идемпотентность**.

Мы уже знакомились с этим понятием и вы знаете как это работает. Но это еще более важно в контексте асинхронного API – потому что риск возникновения повторных событий, да еще и в разное время, куда более высокий чем при работе с простым синхронным API.

Как эффективно использовать это?

- Начнем с основного. Идемпотентные операции API: разработайте свои API-операции таким образом, чтобы они были идемпотентными, то есть чтобы выполнение нескольких одинаковых запросов имело тот же эффект, что и выполнение одного запроса.

Что делать? При проектировании API убедитесь, что повторные обращения к одной и той же конечной точке с одними и теми же параметрами имеют одинаковый результат. Этого можно добиться с помощью уникальных идентификаторов, условных запросов или версионирования элементов.

- Используйте уникальные идентификаторы: присвойте уникальные идентификаторы каждому ресурсу или запросу в вашем API, чтобы отслеживать статус ресурса и обеспечить согласованность при многократных вызовах.

Что делать? Для каждого ресурса или запроса создайте уникальный идентификатор (например, UUID) и включите его в запросы и ответы API, чтобы помочь в отслеживании и согласованности.

- Условные запросы: реализуйте условные запросы, чтобы избежать многократной обработки одной и той же операции, проверяя, соответствует ли текущее состояние ресурса заданным условиям.

Что делать? Используйте HTTP-заголовки типа If-Match, If-None-Match или ETag, чтобы задать условия для обработки запроса. Сервер должен обрабатывать запрос только в том случае, если условия выполнены, предотвращая дублирование.

- Версионирование элементов: интегрируйте версионирование ресурсов в свой API для отслеживания изменений и обеспечения согласованности между состоянием сервера и клиента.

Что делать? Присваивайте номера версий ресурсам и включайте их в ответы API. При обновлении или изменении ресурса требуйте от клиента предоставить текущий номер версии, чтобы убедиться, что он работает с последними данными.