**Отражение**

Объясняет, как можно использовать рефлексию для повышения прозрачности и интерпретируемости RPC.

Обзор

Reflection — это протокол, который серверы gRPC могут использовать для объявления API, определённых протоколом protobuf, которые они экспортируют через стандартизированный RPC-сервис, включая все типы, на которые ссылаются сообщения запросов и ответов. Затем клиенты могут использовать эту информацию для кодирования запросов и декодирования ответов в удобочитаемом виде.

Отражение активно используется в таких инструментах отладки, как grpcurl и Postman. Те, кто знаком с миром REST, могут сравнить API отражения gRPC с документом OpenAPI, который размещается на HTTP-сервере и описывает REST API.

Прозрачность и Интерпретируемость

Одним из факторов, обеспечивающих высокую производительность gRPC, является использование Protobuf для сериализации — *двоичного* непрочитанного человеком протокола. Хотя это значительно ускоряет RPC, это также может усложнить взаимодействие с сервером вручную. Предположим, что для отправки запроса gRPC на сервер по HTTP/2 вручную с помощью curl вам пришлось бы:

1. Знайте, какие RPC-сервисы предоставлены серверу.
2. Знайте определение protobuf сообщения запроса и все типы, на которые оно ссылается.
3. Ознакомьтесь с определением protobuf для сообщения ответа и со всеми типами, на которые *оно* ссылается.

Затем вам пришлось бы использовать эти знания, чтобы вручную преобразовать сообщение (или сообщения) запроса в двоичный формат и кропотливо декодировать ответное сообщение (или сообщения). Это заняло бы много времени, было бы утомительным и чреватым ошибками. Вместо этого протокол отражения позволяет инструментам автоматизировать весь этот процесс, делая его незаметным.

Включение отражения на сервере gRPC

Отладка *не* включается автоматически на сервере gRPC. Автор сервера должен вызвать несколько дополнительных функций, чтобы добавить службу отладки. Эти вызовы API немного отличаются в зависимости от языка и в некоторых языках требуют добавления зависимости от отдельного пакета с названием вроде grpc-reflection

Перейдите по этим ссылкам ниже для получения подробной информации о вашем конкретном языке:

ЯзыкРуководствоJavaПример JavaВпередПерейти к примеруC++Пример C ++ПитонПример PythonJavascriptПример Javascript

Советы

Отладка работает настолько плавно с такими инструментами, как grpcurl , что зачастую люди даже не подозревают, что она происходит «под капотом». Однако, если отладка не отображается, всё будет работать не так плавно. Вместо этого клиент будет выдавать неприятные ошибки. Люди часто сталкиваются с этим при написании конфигурации маршрутизации для службы gRPC. Служба *отладки* должна быть направлена на соответствующий сервер, а также на основную RPC-службу приложения.

Если ваш API gRPC доступен для обычных пользователей, вы можете *не захотеть* раскрывать службу отражения, так как это может быть расценено как угроза безопасности. В конечном счёте вам нужно будет принять решение, которое обеспечит наилучший баланс между безопасностью и удобством использования для вас и ваших пользователей.

Последнее изменение 6 июня 2024 г.: добавлена ссылка на пример отражения в JavaScript (#1292) (de015fd)