**Перехватчики**

Объясняет, как перехватчики можно использовать для реализации универсального поведения, применимого ко многим методам RPC.

Обзор

В основе создания сервисов gRPC лежит реализация методов RPC. Но некоторые функции не зависят от выполняемого метода и должны применяться ко всем или большинству RPC. Для этой задачи хорошо подходят перехватчики.

Когда следует использовать Перехватчики

Возможно, вы уже знакомы с концепцией перехватчиков, но привыкли называть их «фильтрами» или «промежуточным программным обеспечением». Перехватчики очень хорошо подходят для реализации логики, которая не привязана к одному методу RPC. Их также легко использовать на разных клиентах или серверах. Перехватчики — важный и часто используемый способ расширения gRPC. Возможно, вы обнаружите, что некоторые нужные вам функции уже доступны в качестве перехватчиков в более широкой экосистеме gRPC.

Вот некоторые примеры использования перехватчиков:

* Обработкаметаданных
* Ведение журнала
* Впрыскивание неисправности
* Кэширование
* Показатели
* Обеспечение соблюдения Политики
* Проверка подлинности на стороне сервера
* Авторизация на стороне сервера

Примечание

Хотя аутентификация *на стороне клиента* может быть выполнена с помощью перехватчика, gRPC предоставляет специализированный API «вызова учётных данных», который лучше подходит для этой задачи. Подробнее об аутентификации на стороне клиента см. в Руководстве по аутентификации.

Как использовать перехватчики

Перехватчики можно добавлять при создании канала или сервера gRPC. Перехватчик вызывается для каждого RPC в этом канале или на этом сервере. API-интерфейсы перехватчиков отличаются для клиентской и серверной частей, поэтому перехватчик может быть либо «клиентским», либо «серверным».

Перехватчики по своей сути предназначены для каждого вызова; они не подходят для управления TCP-соединениями, настройки TCP-порта или TLS. Хотя они являются подходящим инструментом для большинства настроек, их нельзя использовать для всего.

Приказ Перехватчика

При использовании нескольких перехватчиков их порядок имеет значение. Вам нужно понимать, в каком порядке они будут выполняться в вашей реализации gRPC. Полезно представлять перехватчики как находящиеся на линии между приложением и сетью. Некоторые перехватчики будут «ближе к сети» и смогут лучше контролировать отправляемые данные, а другие будут «ближе к приложению» и смогут лучше отслеживать его поведение.

Предположим, у вас есть два клиентских перехватчика: перехватчик кэширования и перехватчик ведения журнала. В каком порядке они должны быть? Возможно, вы захотите разместить перехватчик ведения журнала ближе к сети, чтобы лучше отслеживать взаимодействие вашего приложения и игнорировать кэшированные RPC:

Cache missApplicationCaching  
InterceptorLogging  
InterceptorNetwork

Или вы можете разместить его ближе к приложению, чтобы понять его поведение и увидеть, какую информацию оно загружает:

Cache missApplicationLogging  
InterceptorCaching  
InterceptorNetwork

Вы можете выбирать между этими вариантами, просто меняя порядок перехватчиков.

Языковая Поддержка

ЯзыкПримерC++Пример C ++ВпередПерейти к примеруJavaПример JavaПитонПример Python

Последнее изменение 29 февраля 2024 года: руководство: добавить перехватчики (#1267) (d9bbae5)