**Готовность к ожиданию**

Объясняет, как настроить RPC, чтобы они ждали готовности сервера перед отправкой запроса.

Обзор

Это функция, которую можно использовать на заглушке, чтобы RPC-запросы ожидали, пока сервер станет доступным, прежде чем отправлять запрос. Это позволяет создавать надёжные пакетные рабочие процессы, поскольку временные проблемы с сервером не приведут к сбоям. Срок выполнения запроса по-прежнему действует, поэтому ожидание будет прервано, если срок выполнения запроса истечёт.

Когда при подключении канала к серверу возникает ошибка RPC, без параметра Wait-for-Ready он немедленно вернёт сообщение об ошибке; с параметром Wait-for-Ready он просто будет поставлен в очередь до тех пор, пока соединение не станет готовым. По умолчанию используется **без** параметра Wait-for-Ready.

Подробную семантику смотрите Здесь.

Как использовать функцию Ожидания готовности

Вы можете указать для заглушки, следует ли использовать Wait-for-Ready, который будет автоматически передаваться при создании RPC.

Примечание

RPC может завершиться ошибкой и по другим причинам, помимо того, что сервер не готов, поэтому обработка ошибок всё равно необходима.

Ниже показана последовательность событий, которые происходят, когда клиент отправляет сообщение на сервер, в зависимости от состояния канала и от того, установлен ли параметр «Ожидание готовности».

ApplicationRPCChannelServerCreate RPC using stubInitiate CommunicationSend messageWait for state changeWait for channelbecoming READY(or a permanent failure)Send messageFailurealt[with Wait-for-Ready][without Wait-for-Ready]Failurealt[channel state: READY][Channel state: IDLE or CONNECTING][Channel state: TRANSIENT\_FAILURE][Channel state is a Permanent Failure]ApplicationRPCChannelServer

Ниже приведено представление, основанное на состоянии

Check Channel StateFrom IDLE /  
CONNECTINGFrom Transient  
Failure with  
Wait-for-ReadyState ChangeFrom READYFrom Permanent failure or  
Transient Failure without  
Wait-for-ReadyInitiating CommunicationChannel Statewithout  
wait-for-readyIDLECONNECTINGREADYTRANSIENT\_FAILUREPermanent FailureMSG sentRPC FailedWAIT

Альтернативы

* Цикл (с экспоненциальным замедлением) до тех пор, пока RPC не перестанет возвращать временные сбои.
  + Для повышения эффективности это можно комбинировать с реализацией onReady обработчика *(для языков, которые это поддерживают)*.
* Принимайте сбои, которых можно было бы избежать, подождав, потому что вы хотите быстро завершить работу

Языковая Поддержка

ЯзыкПримерJavaПример JavaВпередПерейти к примеруПитонПример Python

Последнее изменение 22 августа 2023 г.: исправление: ошибка с примером URL-адреса на странице документации Wait-for-Ready (#1179) (007035b)