

سوال: چارچوب gRPC به چه صورت است و چه مزایایی نسبت به Rest API دارد؟

پاسخ: gRPC و API Rest هر دو روش‌هایی برای ارتباط بین سرویس‌ها در برنامه‌نویسی وب هستند، اما با رویکردهای مختلفی به این کار می‌پردازند.

gRPC یک فرمت مبتنی بر RPC (Remote Procedure Call) است که برای تبادل داده بین سرویس‌ها استفاده می‌شود. این فرمت از protobuf (Protocol Buffers) به عنوان فرمت پیش‌فرض برای توصیف داده‌ها استفاده می‌کند که به صورت باینری و کم حجم است. gRPC از HTTP/۲ برای انتقال داده استفاده می‌کند که باعث بهبود کارایی و کاربردی‌تر شدن ارتباطات میان سرویس‌ها می‌شود. همچنین gRPC از ویژگی‌هایی مانند توزیع کد و آپشن‌های امنیتی پیشرفته پشتیبانی می‌کند.

Rest API به عنوان یک روش معمول برای تبادل داده در اینترنت مورد استفاده قرار می‌گیرد. در این روش، منابع (معمولاً با استفاده از URL) توسط عملیات HTTP مختلفی مانند GET، POST، PUT و DELETE مورد دسترسی قرار می‌گیرند. در مقابل، Rest API از فرمت‌هایی مانند JSON یا XML برای نمایش داده‌ها استفاده می‌کند که برای برنامه‌نویسان خوانایی بیشتری دارند.

از آنجا که gRPC و Rest API هر دو روش‌های معتبر برای ارتباط بین سرویس‌ها هستند، انتخاب بین آن‌ها بسته به نیازهای پروژه و محیط برنامه‌نویسی مختلف است. در مواردی که نیاز به کارایی بالا و کاربردی‌ترین روش ارتباط بین سرویس‌ها وجود دارد، gRPC ممکن است انتخاب مناسبی باشد. اما در موارد دیگر که نیاز به سادگی و قابلیت استفاده از هر زبان برنامه‌نویسی و فرمت داده‌ای معمولی‌تر است، Rest API می‌تواند گزینه مناسب‌تری باشد.

سوال: در gRPC از چه پروتوکلی برای انتقال داده استفاده می‌شود؟

پاسخ: پروتکل HTTP/۲، یک پروتکل ارتباطی است که برای ارتباط بین کلاینت و سرور در gRPC استفاده می‌شود. این پروتکل در اساس یک نسخه بهبود یافته از HTTP/۱.۱ است که طراحی شده است تا کارایی و کاربردی بودن ارتباطات وب را افزایش دهد. در gRPC، از HTTP/۲ به عنوان پروتکل انتقال داده استفاده می‌شود به دلیل ویژگی‌های زیر:

- موازی و دوطرفه بودن: HTTP/۲ امکان ارسال و دریافت داده‌ها به صورت موازی و دوطرفه را فراهم می‌کند، که بهبود کارایی و پاسخگویی سریع‌تر را در مقایسه با HTTP/۱.۱ ایجاد می‌کند.
- فشرده‌سازی هدرها: HTTP/۲ از فشرده‌سازی هدرها استفاده می‌کند که حجم داده‌های ارسالی را کاهش می‌دهد و پهنای باند را بهبود می‌بخشد.
- مدیریت جریان: HTTP/۲ امکان مدیریت بهتر جریان داده‌ها را فراهم می‌کند که بهبود کارایی ارتباطات را ایجاد می‌کند و از تبادل داده‌های بی‌اعتبار جلوگیری می‌کند.
- توانایی ارسال همزمان چندین درخواست و پاسخ: HTTP/۲ امکان ارسال همزمان چندین درخواست و پاسخ را بر روی یک اتصال TCP ایجاد می‌کند، که بهبود عملکرد و کارایی ارتباطات را فراهم می‌کند.
- به طور کلی، استفاده از HTTP/۲ در gRPC بهبود کارایی، پایداری و امنیت ارتباطات را بهبود می‌بخشد و تجربه استفاده از خدمات وب را بهبود می‌دهد.

سوال: تفاوت پروتوکول gRPC و فرمت JSON در چه مواردی است؟

پاسخ: gRPC و JSON هر دو ابزاری هستند که برای ارتباطات بین سیستم‌های مختلف در برنامه‌نویسی استفاده می‌شوند، اما دارای ویژگی‌ها و موارد استفاده متفاوتی هستند. در ادامه تفاوت‌های اصلی بین gRPC و JSON را بررسی می‌کنیم:

فرمت JSON	پروتوکل gRPC	موارد مورد بررسی
از پروتکل‌های ارتباطی متنوعی مانند HTTP، HTTPS، TCP و ... استفاده می‌کند. عموماً در ارسال و دریافت داده‌ها از HTTP/HTTPS استفاده می‌شود.	از پروتکل HTTP/۲ برای ارتباطات استفاده می‌کند که یک پروتکل دوطرفه و موازی است که بهبود کارایی و امکانات ارتباطات را نسبت به HTTP/۱.۱ ارائه می‌دهد.	پروتوکل ارتباطی
از فرمت JSON یا JavaScript Object Notation برای ارسال داده‌ها استفاده می‌کند که یک فرمت متنی است و به صورت خوانا برای انسان‌ها قابل فهم است.	از پروتکل Buffers جهت ارسال داده‌ها استفاده می‌کند که بر پایه protobuf یا Proto- Buffers است. این فرمت داده سریع، کم‌حجم و ساختاردهی شده است.	فرمت داده
ارسال و دریافت داده‌ها به صورت متنی و استفاده از پروتکل‌های HTTP/HTTPS ممکن است باعث کاهش کارایی و افزایش حجم داده شود.	به دلیل استفاده از پروتکل HTTP/۲ و فرمت Buffers، دارای کارایی بالا، فشرده‌سازی داده، مدیریت جریان و پشتیبانی از ارتباطات موازی است.	کارایی
پشتیبانی گسترده‌تری از زبان‌ها و فریم‌ورک‌ها دارد و به عنوان یک فرمت متن‌باز و استاندارد در برنامه‌نویسی وب مورد استفاده قرار می‌گیرد.	پشتیبانی از زبان‌های برنامه‌نویسی مختلفی مانند ++C، Java، Python، Go و ... و فریم‌ورک‌های متنوعی مانند .NET Core، Node.js و ... دارد.	پشتیبانی از زبان‌ها و فریم‌ورک‌ها