

第87期-对比gRPC服务与HttpApi接口

2020年2月10日 17:06

任何技术都有优缺点，下表提供了 gRPC 和 HTTP Api 与 JSON 之间功能的高级比较。

功能	gRPC	具有 JSON 的 HTTP Api
协定	必需 (proto)	可选 (OpenAPI)
协议	HTTP/2	HTTP
Payload	<u>Protobuf (小, 二进制)</u>	JSON (大、可读)
Prescriptiveness	<u>严格规范</u>	松散. 任何 HTTP 都有效。
流式传输	<u>客户端、服务器、双向</u>	客户端、服务器
浏览器支持	<u>否 (需要 grpc-web)</u>	是
安全性	传输 (TLS)	传输 (TLS)
客户端代码生成	是	OpenAPI + 第三方工具

gRPC的优势

性能方面

使用 Protobuf 对消息进行格式化，宽带负载占用小，gRPC 使用 HTTP2 协议，性能更高，在一个 TCP 连接上进行多路复用，消除阻塞。

规范性更好

gRPC 在平台和实现中保持一致，接口规范无争议，与平台无关性。

<https://www.jianshu.com/p/f6ff6381a81a>

流式传输

HTTP2 为生存期较长的实时通信流奠定了基础。gRPC 为流式处理提供一流的支持。

gRPC的缺点

受限制的浏览器支持

现在无法从浏览器直接调用 gRPC 服务。gRPC 大量使用 HTTP2 功能，并且没有浏览器保

证实现 HTTP 2 的全部功能。

gRPC-Web 提供了一个JavaScript库，使浏览器客户端可以访问gRPC服务。

<https://github.com/grpc/grpc-web>

用户不可读

HTTP API 请求以文本的形式发送，人为可读性较好，使用 Protobuf 对 gRPC 消息进行编码，虽然高效，但不可读，调试起来就没那么容易了。

诸如server 反射和gRPC 命令行工具等功能可帮助进行二进制 Protobuf 消息。

gRPC的应用场景

- 微服务- gRPC 旨在实现低延迟和高吞吐量通信。 gRPC 非常适合轻型微服务，其中效率非常重要。
- 点到点实时通信- gRPC 对双向流式处理提供了极佳支持。 gRPC services 无需轮询即可实时推送消息。
- Polyglot 环境- gRPC 工具支持所有常用的开发语言，并为多语言环境提供 gRPC。
- gRPC 消息的网络约束环境使用 Protobuf（一种轻量消息格式）进行序列化。 GRPC 消息始终小于等效的 JSON 消息。

其它备用方案推荐

- gRPC 可以提供浏览器支持，但它具有**局限性**并引入了服务器代理。
- **广播**实时通信- gRPC 支持通过流式处理进行实时通信，但将消息广播到注册连接的概念并不存在， **SignalR** 是此方案的有用框架。 SignalR 具有持续连接的概念和广播消息的内置支持。
- **进程间**的通信- 必须托管 HTTP/2 服务器以接受传入的 gRPC 调用。 进程间通信**管道**是一种快速、轻量的通信方法。