Retrieval API Notes - DB Node Web Service

Thursday, January 15, 2026 4:29 PM

Web service for DB Nodes

Options to consider for web service framework:

- ~~REST API~~
- ~~OData~~
- ~~GData~~
- ~~Twirp~~
- **gRPC**
- ConnectRPC (potential future option)
- ~~Apache Thrift~~

GData vs Odata

Rejecting both.

- They are protocols for creation of WebAPIs
- They are JSON based, which may not be suitable for bytes blob data
- GData is older, by Google
- OData has library for C/C++/C# and js

https://developers.google.com/gdata/
https://www.wikiwand.com/en/articles/GData
https://learn.microsoft.com/en-us/odata/
https://www.odata.org/
https://www.wikiwand.com/en/articles/Open_Data_Protocol

REST API vs protobuf based frameworks

Rejecting REST API.

- Our requirements:
  - compatibility with blob type files, and
  - fast retrieval of large time series data for repeated queries
  - Need something like RPC (Remote procedure call) to invoke database retrieval method on the DB server side
- JSON based HTTP protocols will not be sufficient
- Alternative is to use combination of JSON and FTP-like POST calls
- Or, use protobuf IDL (Interface definition language) which is 2-10x performant.

- That's where Twirp, gRPC, ConnectRPC, Apache Thrift come in.
- Twirp/gRPC/ConnectRPC are basically based on **protobuf** IDL
- Apache Thrift defines its own IDL, "thrift"

Thrift vs Protobuf

Rejecting Thrift.

- Both support different programming languages, code, generation, and call remote procedures

- Thrift has built in RPC framework (Apache thrift), whereas for Protobuf, one has to use gRPC/Twirp/ConnectRPC
- Thrift supports various transport protocols, like HTTP/1, making it easy for web integration
  - Protobuf via gRPC is only supported on HTTP/2. Twirp and ConnectRPC however make it compatible with HTTP/1
- Protobuf is smaller
- **Main sell for Protobuf:** Supports streaming.
  - This can be helpful in future for real-time retrieval applications.

https://jsontotable.org/blog/protobuf/protobuf-vs-thrift
https://github.com/apache/thrift

Twirp vs gRPC

Rejecting Twirp.

- Twirp
  - enables protobuf for HTTP/1. So easier web integration
  - minimalist requests/response
  - supports JSON payloads
  - easy debugging due to HTTP/1 support and JSON
- Twirp is new. Development led by Twitch
- Language support for C/C++ doesn't exist.
- Python support is new (0.0.7). No stable release.

https://twitchtv.github.io/twirp/docs/intro.html
https://github.com/twitchtv/twirp
https://github.com/verloop/twirpy
https://leapcell.io/blog/grpc-vs-twirp-in-go-a-practical-guide-for-internal-service-communication

ConnectRPC vs gRPC

Slightly rejecting ConnectRPC.

- ConnectRPC supports:
  - gRPC
  - gRPC-Web
  - Connect protocol (which works with HTTP/1.1, 2, 3)
- ConnectRPC development led by Buf, a company which has tooling/software around protobuf
- Python support is in beta
- Easier server/client setup as compared to gRPC

- But ConnectRPC is new
  - It is a Sandbox project under CNCF
  - Started in 2022
- Can consider ConnectRPC if
  - it has a stable release for Python
  - It is a fully CNCF project
- It is easy to switch from gRPC to ConnectRPC since the latter supports gRPC.

https://connectrpc.com/docs/introduction
https://github.com/connectrpc
https://github.com/orgs/connectrpc/discussions/17