

ttrpc

build **passing**

GRPC for low-memory environments.

The existing grpc-go project requires a lot of memory overhead for importing packages and at runtime. While this is great for many services with low density requirements, this can be a problem when running a large number of services on a single machine or on a machine with a small amount of memory.

Using the same GRPC definitions, this project reduces the binary size and protocol overhead required. We do this by eliding the `net/http`, `net/http2` and `grpc` package used by grpc replacing it with a lightweight framing protocol. The result are smaller binaries that use less resident memory with the same ease of use as GRPC.

Please note that while this project supports generating either end of the protocol, the generated service definitions will be incompatible with regular GRPC services, as they do not speak the same protocol.

Usage

Create a gogo vanity binary (see [cmd/protoc-gen-gogottrpc/main.go](https://github.com/protocolbuffers/protobuf-go/blob/master/cmd/protoc-gen-gogottrpc/main.go) for an example with the ttrpc plugin enabled).

It's recommended to use [protobuild](https://github.com/protocolbuffers/protobuf-go/blob/master/cmd/protoc-gen-gogottrpc/main.go) to build the protobufs for this project, but this will work with protoc directly, if required.

Differences from GRPC

- The protocol stack has been replaced with a lighter protocol that doesn't require http, http2 and tls.
- The client and server interface are identical whereas in GRPC there is a client and server interface that are different.
- The Go stdlib context package is used instead.
- No support for streams yet.

Status

Very new. YMMV.

TODO:

- ☒ Plumb error codes and GRPC status
- ☒ Remove use of any type and dependency on typeurl package
- ☒ Ensure that protocol can support streaming in the future
- ☐ Document protocol layout
- ☐ Add testing under concurrent load to ensure
- ☐ Verify connection error handling

Project details

ttrpc is a containerd sub-project, licensed under the [Apache 2.0 license](#). As a containerd sub-project, you will find the:

- [Project governance](#),
- [Maintainers](#),
- and [Contributing guidelines](#)

information in our [containerd/project](#) repository.