grpcurl curl for gRPC

CONCEPTS

> Flags must come before the address and method; the address goes first, then list, describe or the method name.

grpcurl flags + JSON
↓ HTTP/2
gRPC server host:port
↓ schema from
Reflection or -proto / -protoset

> Without -plaintext grpcurl uses TLS; a plain server then fails the handshake. Reflection must be enabled on the server, or give a schema file.

DISCOVER

`grpcurl <host:port> list`
List services via reflection

`grpcurl <host:port> list <service>`
List a service's methods

`grpcurl <host:port> describe`
Describe every service

`grpcurl <host:port> describe <service>`
Describe a service

`grpcurl <host:port> describe <service>.<method>`
Describe one method

`grpcurl <host:port> describe <message>`
Describe a message type

`grpcurl -msg-template <host:port> describe <message>`
Show a JSON template for a message

`grpcurl -protoset <file> list`
List from a descriptor set

`grpcurl -import-path <dir> -proto <file> list`
List from a .proto file

CALL

`grpcurl <host:port> <service>/<method>`
Call with an empty request

`grpcurl -d '<json>' <host:port> <service>/<method>`
Call with a JSON body

`grpcurl -d @ <host:port> <service>/<method>`
Read the body from stdin

`grpcurl -emit-defaults ...`
Print zero-valued fields

`grpcurl -max-time <secs> ...`
Limit total time

CONNECTION AND TLS

`grpcurl -plaintext <host:port> ...`
Connect without TLS

`grpcurl -insecure <host:port> ...`
Skip server cert verification

`grpcurl -cacert <file> <host:port> ...`
Trust a custom CA

`grpcurl -cert <file> -key <file> <host:port> ...`
Present a client certificate (mTLS)

`grpcurl -authority <name> ...`
Set :authority and TLS name

METADATA

`grpcurl -H '<name>: <value>' ...`
Send a header (repeatable)

`grpcurl -v ...`
Show headers, trailers and details

SCHEMA SOURCES

`grpcurl -proto <file> -import-path <dir> ...`
Use .proto source (repeatable)

`grpcurl -protoset <file> ...`
Use a compiled descriptor set