

ChannelOptions

August 17, 2026

ChannelOptions

Kind: Interface

Source: `packages/microservices/external/grpc-options.interface.ts`

Part of: [Microservices](#)

An interface that contains options used when initializing a Channel instance.

This listing is incomplete. Full reference:

https://grpc.github.io/grpc/core/group__grpc__arg__keys.html

`ChannelOptions` defines gRPC channel arguments used when creating or configuring a client connection. It provides typed access to common transport, message-size, TLS authority, user-agent, service configuration, and reconnection settings passed to the underlying gRPC implementation.

Properties

Property	Type
<code>'grpc.max_send_message_length'</code>	number
<code>'grpc.max_receive_message_length'</code>	number
<code>'grpc.max_metadata_size'</code>	number
<code>'grpc.ssl_target_name_override'</code>	string
<code>'grpc.primary_user_agent'</code>	string
<code>'grpc.secondary_user_agent'</code>	string
<code>'grpc.default_authority'</code>	string
<code>'grpc.service_config'</code>	string
<code>'grpc.max_concurrent_streams'</code>	number
<code>'grpc.initial_reconnect_backoff_ms'</code>	number

Property	Type
'grpc.max_reconnect_backoff_ms'	number
'grpc.use_local_subchannel_pool'	number
'grpc-node.max_session_memory'	number

Diagram

```
graph LR
  Client[gRPC Client Configuration] --> Options[ChannelOptions]
  Options --> Limits[Message and Metadata Limits]
  Options --> TLS[TLS Target and Authority]
  Options --> Identity[User-Agent Settings]
  Options --> Connection[Streams and Reconnect Settings]
  Options --> Service[Service Configuration]
  Limits --> Channel[gRPC Channel]
  TLS --> Channel
  Identity --> Channel
  Connection --> Channel
  Service --> Channel
```

Usage

```
import type { ChannelOptions } from './grpc-options.interface';

const channelOptions: ChannelOptions = {
  'grpc.max_send_message_length': 4 * 1024 * 1024,
  'grpc.max_receive_message_length': 4 * 1024 * 1024,
  'grpc.max_metadata_size': 8 * 1024,
  'grpc.ssl_target_name_override': 'grpc.example.com',
  'grpc.primary_user_agent': 'my-service/1.0.0',
  'grpc.default_authority': 'grpc.example.com',
  'grpc.max_concurrent_streams': 100,
  'grpc.initial_reconnect_backoff_ms': 1_000,
};

// Pass the options to the gRPC client/channel factory used by your application.
const client = createGrpcClient('grpc.example.com:443', channelOptions);
```

AI Coding Instructions

- Use the exact gRPC argument names, including the `grpc.` prefix; these keys are consumed by the underlying gRPC runtime.

- Specify message and metadata limits in bytes, and ensure configured limits match expected payload sizes across clients and servers.
- Set `grpc.ssl_target_name_override` and `grpc.default_authority` only when required for TLS, proxy, or custom authority scenarios.
- Treat `grpc.service_config` as serialized service configuration data expected by gRPC, not as an arbitrary application configuration object.
- Consult the gRPC channel argument reference before adding new options, since this interface represents only a subset of supported arguments.