TMS GRPC API Package

Table of contents

> ⚠ **Attention**
>
> NVIDIA Triton Management Service (TMS) will reach the end of life on July 31, 2024. The version 1.4.0 is the last release.

With every release of Triton Management Service a zip-compressed archive of the gRPC IDL files is provided. These packages can be downloaded from NVIDIA's NGC Catalog.

Versions

It is important to use the version of the TMS gRPC API that matches the version of the TMS Server being communicated with. NVIDIA provides no guarantee of forward/backward compatibility for the programmatic interfaces of beta software. The TMS interface is still in development and expected to fluctuate.

Using the Package

1. Download the gRPC API Package, either using the web interface (provided above) or using the NGC CLI with the following command
   `ngc registry resource download-version "nvaie/triton-management-service_grpc-api-bundle:1.4.0"`
   . Notice that the desired version is the last component of the command, and can be adjusted to match the version of TMS as necessary.

2. Extract the contents of the downloaded package.

   Linux

   ```
   $ unzip ./files.zip
   Archive: ./files.zip
   inflating: files/triton-management-service_grpc-api-bundle-v1.4.0.zip
   $ unzip triton-management-service_grpc-api-bundle-v1.4.0.zip
   Archive: triton-management-service_grpc-api-bundle-v1.4.0.zip
   inflating: bespoke-triton.proto
   ```

```
inflating: pooled-triton.proto
inflating: lease-state.proto
inflating: triton-allowlist-service.proto
inflating: model.proto
inflating: model-state.proto
inflating: triton-pool-service.proto
inflating: triton.proto
inflating: lease-service.proto
inflating: lease-name-service.proto
inflating: common.proto
inflating: lease-duration.proto
inflating: error-code.proto
inflating: lease-event.proto
inflating: triton-state.proto
```

Windows Extracting the IDL on Windows will require a tool like 7-Zip which can handle compressed TAR files.

```
> 7z x .\files.zip
7-Zip 23.01 (x64) : Copyright (c) 1999-2023 Igor Pavlov : 2023-06-20
Scanning the drive for archives:
1 file, 18579 bytes (19 KiB)
Extracting archive: .\files.zip
--
Path = .\files.zip
Type = zip
Physical Size = 18579
Everything is Ok
Size: 19324
Compressed: 18579
> 7z x ./triton-management-service_grpc-api-bundle-v1.4.0.zip
7-Zip 23.01 (x64) : Copyright (c) 1999-2023 Igor Pavlov : 2023-06-20
Scanning the drive for archives:
1 file, 19324 bytes (19 KiB)
Extracting archive: triton-management-service_grpc-api-bundle-v1.4.0.zip
```

```
--
Path = triton-management-service_grpc-api-bundle-v1.4.0.zip
Type = zip
Physical Size = 19324
Everything is Ok
Files: 15
Size: 65633
Compressed: 19324
```

Once extracted you should have the following list of files:

```
bespoke-triton.proto
lease-duration.proto
lease-service.proto
model.proto
triton-pool-service.proto
common.proto
lease-event.proto
lease-state.proto
pooled-triton.proto
triton-state.proto
error-code.proto
lease-name-service.proto
model-state.proto
triton-allowlist-service.proto
triton.proto
```

3. Use the `protoc` compiler to generate language specific code files from the provided IDL (*.proto) files. The necessary compiler and tools can be downloaded from the Protocol Buffers Release Page on GitHub. The latest, release version is v23.4. Download the version of the tools that best suite your platform.

   Once all tools are downloaded, use them to generate code in your language of choice. Use Protocol Buffers Getting Started as a guide as needed.

   Example for JavaScript code generation:

```
$ protoc --proto_path=proto --
js_out=library=tms_model_state,binary:js_autogen proto/model-state.proto
```

Will create a JavaScript file named `tms_model_state.js` from `proto/model-state.proto`, and output the results to a folder named `js_autogen` (must exist before running `protoc`). Notice that the above assumes the *.proto file are contained in a folder named `proto` which is a child of the current working folder.