protogen

fischor

Jul 14, 2023

CONTENTS:

1 Installation 1

2 protogen module 3

3 Indices and tables 21

Python Module Index 23

Index 25

CHAPTER

ONE

INSTALLATION

` pip install protogen `

CHAPTER

TWO

PROTOGEN MODULE

Package protogen makes writing protoc plugins easy.

Working with the raw protobuf descriptor messages can be cumbersome. *protogen* resolves and links the dependencies and references between the raw Protobuf descriptors and turns them into their corresponding *protogen* classes that are easier to work with. It also provides mechanisms that are espacially useful to generate Python code like dealing with Python imports.

Most classes in *protogen* are simply replacements of their corresponding Protobuf descriptors: *protogen.File* represents a FileDescriptor, *protogen.Message* a Descriptor, *protogen.Field* a FieldDescriptor and so on. They should be self explanatory. You can read their docstrings for more information about them.

The classes *protogen.Options*, *protogen.Plugin* and *protogen.GeneratedFile* make up a framework to generate files. You can see these in action in the following example plugin:

```
#!/usr/bin/env python
"An example plugin."

import protogen

def generate(gen: protogen.Plugin):
    for f in gen.files_to_generate:
        g = gen.new_generated_file(
            f.proto.name.replace(".proto", ".py"),
            f.py_import_path,
        )
        g.P("# Generated code ahead.")
        g.P()
        g.print_imports()
        g.P()
        for m in f.message:
            g.P("class ", m.py_ident, ":")
            for ff in m.fields:
                # ...
        for s in f.services:
            g.P("class ", s.py_ident, ":")
            for m in f.methods:
                g.P("  def ", m.py_name, "(request):")
                g.P("    pass")

if __name__ == "__main__":
    opts = protogen.Options()
    opts.run(generate)
```

class protogen.**Cardinality**(*value*)
Cardinality specifies whether a field is optional, required or repeated.

class protogen.**CodeGeneratorResponse**(*proto: google.protobuf.compiler.plugin_pb2.CodeGeneratorResponse*)
A code generator response.

This is the `protogen` equivalent to a protobuf CodeGeneratorResponse.

proto
The raw CodeGeneratorResponse.

Type google.protobuf.descriptor_pb2.CodeGeneratorResponse

__init__(*proto: google.protobuf.compiler.plugin_pb2.CodeGeneratorResponse*) → None

file_content(*filename*) → Tuple[str, bool]
Returns the content of a file from the CodeGeneratorResponse.

Parameters **filename** (*str*) – Name of the file to get the content for.

Returns **content** – Returns *True* and the content of the file if a file with that name exists in the CodeGeneratorResponse. Otherwise *False* and the empty string is returned.

Return type Tuple[bool, str]

has_file(*filename: str*) → bool
Checks if a file in the CodeGeneratorResponse.

Parameters **filename** (*str*) – Name of the file to check.

Returns **ok** – *True*, if the file is contained in the response, *False* otherwise.

Return type bool

class protogen.**Enum**(*proto: google.protobuf.descriptor_pb2.EnumDescriptorProto*, *parent_file:* protogen.File, *parent: Optional[*protogen.Message*]*, *path: List[int]*)
A proto enum.

This is the `protogen` equivalent to a protobuf EnumDescriptor. The enums attributes are obtained from the EnumDescriptor it is derived from and references to other `protogen` classes that have been resolved in the resolution process. It represents a Protobuf enum defined within an *.proto* file.

proto
The raw EnumDescriptor of the enum.

Type google.protobuf.descriptor_pb2.EnumDescriptorProto

py_ident
Python identifier for the Python class of the enum.

Type *PyIdent*

full_name
Full proto name of the enum.

Type str

parent_file
The File the enum is declared in.

Type *File*

parent
For nested enums, the message the enum is declared in. `None` otherwise.

Type *Message* or None

values
Values of the enum.

Type List[*EnumValue*]

location
Comments associated with the enum.

Type *Location*

__init__(*proto: google.protobuf.descriptor_pb2.EnumDescriptorProto*, *parent_file:* protogen.File, *parent: Optional[*protogen.Message*]*, *path: List[int]*)

class protogen.**EnumValue**(*proto: google.protobuf.descriptor_pb2.EnumValueDescriptorProto*, *parent:* protogen.Enum, *path: List[int]*)

A proto enum value.

This is the `protogen` equivalent to a protobuf EnumValueDescriptor. The enum values attributes are obtained from the EnumValueDescriptor it is derived from and references to other `protogen` classes that have been resolved in the resolution process. It represents a Protobuf enum value declared within an Protobuf enum definition.

proto
The raw EnumValueDescriptor of the enum value.

Type google.protobuf.descriptor_pb2.EnumValueDescriptorProto

py_ident
Python identifier for the Python attribute of the enum value.

Type *PyIdent*

full_name
Full proto name of the enum value. Note that full names of enum values are different: All other proto declarations are in the namespace of their parent. Enum values however are within the namespace of ther parent file. An enum value named `FOO_VALUE` declared within an enum `proto.package.MyEnum` has a full name of `proto.package.FOO:VALUE`.

Type str

number
The enum number.

Type int

parent
The enum the enum value is declared in.

Type *Enum*

location
Comments associated with the enum value.

Type *Location*

__init__(*proto: google.protobuf.descriptor_pb2.EnumValueDescriptorProto*, *parent:* protogen.Enum, *path: List[int]*)

protogen.**Extension**
A protobuf extension.

Protobuf extensions are described using FieldDescriptors. See *Field*.

class protogen.**Field**(*proto: google.protobuf.descriptor_pb2.FieldDescriptorProto*, *parent: Optional[protogen.Message]*, *parent_file: protogen.File*, *oneof: Optional[protogen.OneOf]*, *path: List[int]*)

A proto field.

This is the `protogen` equivalent to a protobuf FieldDescriptor. The fields attributes are obtained from the FieldDescriptor it is derived from and references to other `protogen` classes that have been resolved in the resolution process. It represents a Protobuf field declared within a Protobuf message definition. It is also used to describe protobuf extensions.

proto
The raw FieldDescriptor of the field.

Type google.protobuf.descriptor_pb2.FieldDescriptorProto

py_name
Python name of the field. This is a sanatized version of the original proto field name.

Type str

full_name
Full proto name of the field.

Type str

parent
The message the field is declared in. Or `None` for top-level extensions.

Type *Message* or None

parent_file
The file the field is declared in.

Type *File*

oneof
The oneof in case the field is contained in a oneof. `None` otherwise.

Type *OneOf* or None

kind
The field kind.

Type *Kind*

cardinality
Cardinality of the field.

Type *Cardinality*

enum
The enum type of the field in case the fields *kind* is `Kind.Enum`. `None` otherwise.

Type *Enum* or None

message
The message type of the field in case the fields *kind* is `Kind.Message`. `None` otherwise.

Type *Message* or None

extendee
The extendee in case this is a top-level extension. `None` otherwise.

Type *Message* or None

location
Comments associated with the field.

Type *Location*

__init__(*proto: google.protobuf.descriptor_pb2.FieldDescriptorProto*, *parent: Optional[*protogen.Message*]*, *parent_file:* protogen.File, *oneof: Optional[*protogen.OneOf*]*, *path: List[int]*)

is_list() → bool
Whether the field is a list field.

A list fields has a *cardinality* of `Cardinality.REPEATED` and is not a map field.

Returns `True` if the field is a list field. `False` otherwise.

Return type bool

is_map() → bool
Whether the field is a map field.

Returns `True` if the field is a map field. `False` otherwise.

Return type bool

map_key() → Optional[*protogen.Field*]
Return the map key if the field is a map field.

Returns The field of the map key if *is_map()* is `True`. `None` otherwise.

Return type *Field* or None

map_value() → Optional[*protogen.Field*]
Return the map value if the field is a map field.

Returns The field of the map value if *is_map()* is `True`. `None` otherwise.

Return type *Field* or None

class protogen.**File**(*proto: google.protobuf.descriptor_pb2.FileDescriptorProto*, *generate: bool*, *py_import_func: Callable[[str, str],* protogen.PyImportPath*]*)
A proto file.

This is the `protogen` equivalent to a protobuf FileDescriptor. The files attributes are obtained from the FileDescriptor it is derived from and references to other `protogen` classes that have been resolved in the resolution process. It represents a Protobuf file (*.proto* file).

proto
The raw FileDescriptor of the file.

Type google.protobuf.descriptor_pb2.FileDescriptorProto

generated_filename_prefix
Name of the original proto file (without `.proto` extension).

Type str

py_package_name
Name of the proto package the file belongs to. This is the result of the proto package name of the proto file applied to the `py_import_function` of the `Plugin` that is used to read the file.

Type str

py_import_path
Import path for the file.

Type *PyImportPath*

generate
Whether Python code should be generated for the file.

Type bool

dependencies
Files imported by the file.

Type List[*File*]

enums
Top-level enum declarations.

Type List[*Enum*]

messages
Top-level message declarations.

Type List[*Message*]

services
Service declarations.

Type List[*Service*]

extensions List[Extension]
Top-level extension declarations.

__init__(*proto: google.protobuf.descriptor_pb2.FileDescriptorProto*, *generate: bool*, *py_import_func: Callable[[str, str], protogen.PyImportPath]*)

class protogen.**GeneratedFile**(*name: str*, *py_import_path:* protogen.PyImportPath)
An output buffer to write generated code to.

A generated file is a buffer. New lines can be added to the output buffer by calling *P()*.

Additionally, the generated file provides mechanism for handling Python imports. Internally it maintains a list of *PyImportPath* s that are requested to be imported. Use `print_imports()` to mark the position in the output buffer the imports will be printed at.

To create a new instance of a generated file use *Plugin.new_generated_file()*. *Plugin.new_generated_file()* requires a `filename` and a `py_import_path` as parameter. The `filename` is obviously the name of the file to be created. The `py_import_path` is used for *import resolution*. It specifies the Python module the generated file is representing.

When calling *qualified_py_ident()* the generated files import path is compared to the import path of the Python identifier that is passed as an argument. If they refer to different Python modules, the *PyImportPath* of the argument is added to the list of imports of the generated file. Note that also *P()* calls *qualified_py_ident()*, so the above also applies to *PyIdent* arguments passed to *P()*.

name
Name of the generated file.

Type str

P(**args*)
Add a new line to the output buffer.

Add a new line to the output buffer containing a stringified version of the passed arguments. For arguments that are of class *PyIdent* *qualified_py_ident()* is called. This will add the import path to the generated files import list and write the fully qualified name of the Python identifier, if necessary.

Parameters ***args** – Items that make up the content of the new line. All args are printed on the same line. There is no whitespace added between the individual args.

__init__(*name: str*, *py_import_path:* protogen.PyImportPath)

print_import()
Set the mark to print the imports in the output buffer.

The current location in the output buffer will be used to print the imports collected by *qualified_py_ident()*. Only one location can be set. Consecutive calls will overwrite previous calls.

Example

```
>>> g.P("# My python file")
>>> g.P()
>>> g.print_imports()
>>> g.P()
>>> g.P("# more content following after the imports..")
```

qualified_py_ident(*ident:* protogen.PyIdent) → str
Obtain the qualified Python identifier name with respect to the generated file.

If `ident.py_import_path` and the `import_path` of the generated file refer to different Python modules, the `ident.py_import_path` will be added to the list of imports of the generated file and the fully qualified name of *ident* will be returned. If `ident.py_import_path` and the `import_path` of the generated file refer to the same Python module, the `ident.py_name` will be returned and nothing will be added to the list of imports of the generated file.

Parameters **ident** (PyIdent) – The identifier to obtain the qualified name for.

Returns The qualified identifier name.

Return type str

set_indent(*level: int*) → int
Set the indentation level.

Set the indentation level such that consecutive calls to *P()* are indented automatically to that level.

Parameters **level** (*int*) – The new indentation level.

Returns The old indentation level.

Return type int

Raises **ValueError** – If level is less than zero.

Example

```
>>> g.P("class MyClass:")
>>> reset = g.set_indent(4)
>>> g.P("def __init__():")
>>> g.P("    pass")
>>> g.set_indent(reset)
```

exception protogen.**InvalidDescriptorError**(*full_name: str*, *msg: str*)
Error raised when a descriptor is invalid.

This error is raied if a descriptor is considered invalid. A descriptor might be considered invalid for various reasons. For example: * a FieldDescriptor may be of TYPE_ENUM but not declare a type_name * a FieldDescriptor may be of TYPE_MESSAGE but not declare a type_name

__init__(*full_name: str*, *msg: str*)

class protogen.**Kind**(*value*)

Kind is an enumeration of the different value types of a field.

class protogen.**Location**(*source_file: str*, *path: List[int]*, *leading_detached_comments: List[str]*, *leading_comments: str*, *trailing_comments: str*)

A proto location.

A Location identifies a piece of source code in a .proto file which corresponds to a particular definition. This information is particular useful as it contains the comments that are associated with a certain part (e.g. a message or field) of the `.proto` file.

source_file

Name of the file the location is from.

Type str

path

Identifies which part of the FileDescriptor was defined at the location.

Type List[int]

leading_comments

Comments directly attached (leading) to the location. Not separated with a blank.

Type str

trailing_comments

Comments directly attached (trailing) to the location. Not separated with a blank.

Type str

leading_detached_comments

Comments that are leading to the current location and detached from it by at least one blank line.

Type List[str]

Examples

The following example explains the different kind of comments.

```
optional int32 foo = 1;  // Comment attached to foo.
// Comment attached to bar.
optional int32 bar = 2;

optional string baz = 3;
// Comment attached to baz.
// Another line attached to baz.

// Comment attached to qux.
//
// Another line attached to qux.
optional double qux = 4;

// Detached comment for corge. This is not leading or trailing comments
```

(continues on next page)

(continued from previous page)

```
// to qux or corge because there are blank lines separating it from
// both.

// Detached comment for corge paragraph 2.

optional string corge = 5;
/* Block comment attached
* to corge.  Leading asterisks
* will be removed. */
/* Block comment attached to
* grault. */
optional int32 grault = 6;

// ignored detached comments.
```

__init__(*source_file: str*, *path: List[int]*, *leading_detached_comments: List[str]*, *leading_comments: str*, *trailing_comments: str*)

class protogen.**Message**(*proto: google.protobuf.descriptor_pb2.DescriptorProto*, *parent_file:* protogen.File, *parent: Optional[*protogen.Message*]*, *path: List[int]*)

A proto message.

This is the `protogen` equivalent to a protobuf Descriptor. The messages attributes are obtained from the Descriptor it is derived from and references to other `protogen` classes that have been resolved in the resolution process. It represents a Protobuf message defined within an *.proto* file.

proto
The raw Descriptor of the message.

Type google.protobuf.descriptor_pb2.DescriptorProto

py_ident
Python identifier for the Python class of the message.

Type *PyIdent*

full_name
Full proto name of the message.

Type str

parent_file
The file the message is defined in.

Type *File*

parent
The parent message in case this is a nested message. `None`, for top-level messages.

Type *Message* or None

fields
Message field declarations. This includes fields defined within oneofs.

Type List[*Field*]

oneofs
Oneof declarations.

Type List[*OneOf*]

enums
Nested enum declarations.

Type List[*Enum*]

messages List[Message]
Nested message declarations.

extensions
Nested extension declations.

Type List[Extension]

location
Comments associated with the message.

Type *Location*

__init__(*proto: google.protobuf.descriptor_pb2.DescriptorProto*, *parent_file:* protogen.File, *parent: Optional[*protogen.Message*]*, *path: List[int]*)

class protogen.**Method**(*proto: google.protobuf.descriptor_pb2.MethodDescriptorProto*, *parent:* protogen.Service, *path: List[int]*)
A proto service method.

This is the `protogen` equivalent to a protobuf MethodDescriptor. The methods attributes are obtained from the MethodDescriptor it is derived from and references to other `protogen` classes that have been resolved in the resolution process. It represents a Protobuf method declared within a Protobuf service definition.

proto
The raw MethodDescriptor of the method.

Type google.protobuf.descriptor_pb2.MethodDescriptorProto

py_name
Python name of the method. A snake cased version of the proto name.

Type str

full_name
Full proto name of the method.

Type str

grpc_path
The grpc path of the method. Derived from the service and method name: `"/{service name}/{method name}"`

Type str

parent
The service the method is declared in.

Type *Service*

input
The input message of the method.

Type *Message*

output
The output message of the method.

Type *Message*

location
Comments associated with the method.

Type *Location*

__init__(*proto: google.protobuf.descriptor_pb2.MethodDescriptorProto*, *parent:* protogen.Service, *path: List[int]*)

class protogen.**OneOf**(*proto: google.protobuf.descriptor_pb2.OneofDescriptorProto*, *parent:* protogen.Message, *path: List[int]*)

A proto Oneof.

This is the `protogen` equivalent to a protobuf OneofDescriptor. The oneofs attributes are obtained from the OneofDescriptor it is derived from and references to other `protogen` classes that have been resolved in the resolution process. It represents a Protobuf oneof declared within a Protobuf message definition.

proto
The raw OneofDescritor of the oneof.

Type google.protobuf.descriptor_pb2.OneofDescriptorProto

full_name
Full proto name of the oneof.

Type str

parent
The message the oneof is declared in.

Type *Message*

fields
Fields that are part of the oneof.

Type List[*Field*]

location
Comments associated with the oneof.

Type *Location*

__init__(*proto: google.protobuf.descriptor_pb2.OneofDescriptorProto*, *parent:* protogen.Message, *path: List[int]*)

class protogen.**Options**(**, py_import_func: Callable[[str, str], protogen.PyImportPath] = <function default_py_import_func>, input: Optional[BinaryIO] = None, output: Optional[BinaryIO] = None, supported_features: List[google.protobuf.compiler.plugin_pb2.CodeGeneratorResponse.Feature] = []*)

Options for resolving a raw CodeGeneratorRequest to `protogen` classes.

In the resolution process, the raw FileDescriptors, Descriptors, ServiceDescriptors etc. that are contained in the CodeGeneratorRequest provided by protoc are turned into their corresponding `protogen` classes (*File*, *Message*, *Service*).

Use *run()* to run a code generation function.

__init__(**, py_import_func: Callable[[str, str], protogen.PyImportPath] = <function default_py_import_func>, input: Optional[BinaryIO] = None, output: Optional[BinaryIO] = None, supported_features: List[google.protobuf.compiler.plugin_pb2.CodeGeneratorResponse.Feature] = []*)

Create options for the resolution process.

Parameters

- **py_import_func** (*Callable[[str, str], PyImportPath], optional*) – Defines how to derive *PyImportPath* for the *File* objects in the resolution process. This also influences the *PyIdent* attributes that are part of *Message*, *Enum*, and *Service* classes as their import paths are inherited from the *File* they are defined in. Defaults to use *default_py_import_func()*.
- **input** (*BinaryIO, optional*) – The input stream to read the CodeGeneratorRequest from. Defaults to `sys.stdin.buffer` if set as None.
- **output** (*BinaryIO, optional*) – The output stream to write the CodeGeneratorResponse to. Defaults to `sys.stdout.buffer` if set as None.
- **supported_features** (*List[str]*) – List of features that are supported by the plugin. This list will be delegated to protoc via the CodeGeneratorresponse.supported_features field. For example, to indicate that the plugin supports optionals, provide *google.protobuf.compiler.plugin_pb2.CodeGeneratorResponse.Feature.FEATURE_PROTO3_OPTIONAL.* in the list.

run(*f: Callable[[*protogen.Plugin*], None]*)

Start resolution process and run `f` with the *Plugin* containing the resolved classes.

run waits for protoc to write the CodeGeneratorRequest to `input`, resolves the raw FileDescriptors, Descriptors, ServiceDescriptors etc. contained in it to their corresponding `protogen` classes and creates a new *Plugin* with the resolved classes. `f` is then called with the *Plugin* as argument. Once `f` returns, *Options* will collect the CodeGeneratorResponse from the *Plugin* that contains information of all *GeneratedFile*s that have been created on the plugin. The response is written to `output` for protoc to pick it up. protoc writes the generated files to disk.

Parameters **f** (*Callable[[Plugin], None]*) – Function to run with the Plugin containing the resolved classes.

class protogen.**Plugin**(*parameter: Dict[str, str], files_to_generate: List[*protogen.File*], registry:* protogen.Registry)

An invocation of a protoc plugin.

Provides access to the resolved `protogen` classes as parsed from the CodeGeneratorRequest read from protoc and is used to create a CodeGeneratorResponse that is returned back to protoc. To add a new generated file to the response, use *new_generated_file()*.

parameter

Parameter passed to the plugin using `{plugin name}_opt=<key>=<value>` or ``<plugin>_out=<key>=<value>` command line flags.

Type Dict[str, str]

files_to_generate

Set of files to code generation is request for. These are the files explictly passed to protoc as command line arguments.

Type List[*File*]

registry

The registry that was used in the resolution process for this plugin.

Type *Registry*

__init__(*parameter: Dict[str, str], files_to_generate: List[*protogen.File*], registry:* protogen.Registry)

error(*msg: str*)

Record an error.

The error will be reported back to protoc. No output will be produced in case of an error. produce any output. Will act as a no-op for consecutive calls; only the first error is reported back.

Parameters **msg** (*str*) – Error message to report back to protoc. This will appear on the command line when the error is displayed.

new_generated_file(*name: str*, *py_import_path:* protogen.PyImportPath) → *protogen.GeneratedFile*
Create a new generated file.

The generated file will be added to the output of the plugin.

Parameters

- **name** (*str*) – Filename of the generated file.
- **py_import_path** (PyImportPath) – Python import path of the new generated file. This is used to decide whether to print the fully qualified name or the simple name for a Python identifier when using *GeneratedFile.P*. See *GeneratedFile*.

Returns The new generated file.

Return type *GeneratedFile*

class protogen.**PyIdent**(*py_import_path:* protogen.PyImportPath, *py_name: str*)
An identifier for a Python class, function or variable.

A Python class, function or variable is uniquely identified by its import path (e.g. `google.protobuf.timestamp_pb2`), that references the module its defined in, and name (eg *Timestamp*).

py_import_path
The Python import path of the identifier.

Type *PyImportPath*

py_name
Name of the class, function or variable.

Type str

__init__(*py_import_path:* protogen.PyImportPath, *py_name: str*)
Create a new Python identifier.

The recommended way to initialize a new *PyIdent* is using *PyImportPath.indent()* instead.

```
>>> grpc_pkg = protogen.PyImportPath("grpc")
>>> grpc_pkg.ident("unary_unary")
```

class protogen.**PyImportPath**(*path: str*)
A Python import path.

Represents a Python import path as used in a Python import statement. In Python, the import path is used to identify the module to import. An import path "google.protobuf.timestamp_pb2" refers to the "google/protobuf/timestamp_pb2.py" module and might be imported as follows:

```
>>> import google.protobuf.timestamp_pb2
```

or

```
>>> from google.protobuf.timestamp_pb2 import Timestamp
```

This is just a simple wrapper class around the import string. It is used in the *GeneratedFile* to keep track of which import statements need to be included in the output of the generated file as well as how a *PyIdent* needs to be referred to in the output the generated file.

Example

Use the *PyImportPath* class to take advantage of the import resolution mechanism provided by the *GeneratedFile*:

```
>>> import protogen
>>> grpc_pkg = protogen.PyImportPath("grpc")
>>> # g is of type protogen.GeneratedFile
>>> g.P("def my_method(request):")
>>> g.P("  ", grpc_pkg.ident("unary_unary"), "(request)")
```

That way *grpc_pkg* will be added automatically to the import list of *g*.

__init__(*path: str*)
Create a new Python import path wrapping *path*.

ident(*name: str*) → *protogen.PyIdent*
Create a *PyIdent* with *self* as import path and name as *py_name*.

Parameters **name** (*str*) – Python name of the identifier.

Returns The python identifier.

Return type *PyIdent*

class protogen.**Registry**
A registry for protogen types.

A registry holds referneces to *File*, *Service*, *Enum* and *Message* objects that have been resolved within a resolution process (see *Options.run()*).

__init__()
Create a new, empty registry.

all_enums() → List[*protogen.Enum*]
Get all registered enums.

all_files() → List[*protogen.File*]
Get all registered files.

all_messages() → List[*protogen.Message*]
Get all registered messages.

all_services() → List[*protogen.Service*]
Get all registered services.

enum_by_name(*name: str*) → Optional[*protogen.Enum*]
Get an enum by its full name.

Parameters **name** (*str*) – The full (proto) name of the enum to retrieve.

Returns **enum** – The enum or *None* if no enum with that name has been registered.

Return type *Enum* or None

enums_by_package(*package: str*, *top_level_only: bool = False*) → List[*protogen.Enum*]
Get enums by proto package.

Parameters

- **package** (*str*) – The proto package to get enums for.
- **top_level_only** (*bool, optional, default=False*) – If True, only top level enums are returned. Otherwise nested enums are included.

Returns The enums.

Return type List[*Enum*]

file_by_name(*name: str*) → Optional[*protogen.File*]
Get a file by its full name.

Parameters **name** (*str*) – The full (proto) name of the file to retrieve.

Returns **file** – The file or *None* if no file with that name has been registered.

Return type *File* or None

files_by_package(*package: str*) → List[*protogen.File*]
Get files by proto package.

Parameters **package** (*str*) – The proto package to get files for.

Returns The files.

Return type List[*File*]

message_by_name(*name: str*) → Optional[*protogen.Message*]
Get a message by its full name.

Parameters **name** (*str*) – The full (proto) name of the message to retrieve.

Returns **message** – The message or *None* if no message with that name has been registered.

Return type *Message* or None

messages_by_package(*package: str*, *top_level_only: bool = False*) → List[*protogen.Message*]
Get messages by proto package.

Parameters

- **package** (*str*) – The proto package to get messages for.
- **top_level_only** (*bool, optional, default=False*) – If True, only top level message are returned. Otherwise nested messages are included.

Returns The messages.

Return type List[*Message*]

resolve_enum_type(*reference_scope: str*, *proto_name: str*) → Optional[*protogen.Enum*]
Resolve an enum name to an enum.

Searches for an enum within the registry by its proto name. If the *proto_name* has a leading dot the name is treated as fully qualified, otherwise the enum is resolved relative to the reference scope using C++ scoping rules.

E.g. given a *reference_scope* of *"mycom.cloud.datastore.v1.Hello*" and a *proto_name' of `"World"* the registry would be search for (in that order):

- mycom.cloud.datastore.v1.Hello.World
- mycom.cloud.datastore.v1.World
- mycom.cloud.datastore.World
- mycom.cloud.World
- mycom.World
- World

and the first existing enum type would be returned.

Parameters

- **reference_scope** (*str*) – The current scope that acts as starting points in the enum type resolution process.
- **proto_name** (*str*) – The proto (enum type) name to resolve.

Returns **response** – The resolved protogen enum type, or *None* if no enum with that name could be found under the reference scope.

Return type *protogen.Enum* | None

resolve_message_type(*reference_scope: str*, *proto_name: str*) → Optional[*protogen.Message*]

Resolve a message name to a message.

Searches for a message within the registry by its proto name. If the *proto_name* has a leading dot the name is treated as fully qualified, otherwise the message is resolved relative to the reference scope using C++ scoping rules.

E.g. given a *reference_scope* of *"mycom.cloud.datastore.v1.Hello*" and a *proto_name' of `"World"* the registry would be search for (in that order):

- mycom.cloud.datastore.v1.Hello.World
- mycom.cloud.datastore.v1.World
- mycom.cloud.datastore.World
- mycom.cloud.World
- mycom.World
- World

and the first existing message type would be returned.

Parameters

- **reference_scope** (*str*) – The current scope that acts as starting points in the message type resolution process.
- **proto_name** (*str*) – The proto (message type) name to resolve.

Returns **response** – The resolved protogen message type, or *None* if no message with that name could be found under the reference scope.

Return type *protogen.Message* | None

service_by_name(*name: str*) → Optional[*protogen.Service*]

Get a service by its full name.

Parameters **name** (*str*) – The full (proto) name of the service to retrieve.

Returns **service** – The service or *None* if no service with that name has been registered.

Return type *Service* or None

services_by_package(*package: str*) → List[*protogen.Service*]

Get services by proto package.

Parameters **package** (*str*) – The proto package to get services for.

Returns The services.

Return type List[*Service*]

exception protogen.**ResolutionError**(*file: str*, *desc: str*, *ref: str*)

Error raised when type or enum name can not be resolved.

This error is raised if a reference to a message or enum could not be resolved. References to messages and enum might be declared in MethodDescriptors or FieldDescriptors.

file
The proto file that contains the descriptor that referes to a type that could not be resolved.

Type str

desc
The full name of the descriptor that holds the reference

Type str

ref
The type or enum reference that can not be resolved.

Type str

__init__(*file: str*, *desc: str*, *ref: str*)

class protogen.**Service**(*proto: google.protobuf.descriptor_pb2.ServiceDescriptorProto*, *parent:* protogen.File, *path: List[int]*)
A proto service.

This is the `protogen` equivalent to a protobuf ServiceDescriptor. The services attributes are obtained from the ServiceDescriptor it is derived from and references to other `protogen` classes that have been resolved in the resolution process. It represents a Protobuf service defined within an *.proto* file.

proto
The raw ServiceDescriptor of the service.

Type google.protobuf.descriptor_pb2.ServiceDescriptorProto

py_ident
Python identifier for the Python class of the service.

Type *PyIdent*

full_name
Full proto name of the service.

Type str

parent_file
The file the Service is defined in.

Type *File*

methods
Service method declarations.

Type List[*Method*]

location
Comments associated with the service.

Type *Location*

__init__(*proto: google.protobuf.descriptor_pb2.ServiceDescriptorProto*, *parent:* protogen.File, *path: List[int]*)

protogen.**default_py_import_func**(*filename: str*, *package: str*) → *protogen.PyImportPath*
Return the Python import path for a file.

Return the Python import path for a file following the behaviour of the offical Python protoc plugin that generates for each input file *path/to/file.proto* a corresponding *path/to/file_pb2.py* file. This function is used as the default `py_import_func` parameter in :func:`Options.__init__`.

Parameters

- **filename** (*str*) – Filename of the proto file to request the import path for.
- **package** (*str*) – Proto package name of the file to request the import path for.

Returns The Python import path for the file.

Return type *PyImportPath*

Example

```
>>> default_py_import_func("google/protobuf/field_mask.proto", "google.protobuf")
"google.protobuf.field_mask_pb2"
```

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

p
protogen, 3

INDEX

Symbols

__init__() (*protogen.CodeGeneratorResponse method*), 4
__init__() (*protogen.Enum method*), 5
__init__() (*protogen.EnumValue method*), 5
__init__() (*protogen.Field method*), 7
__init__() (*protogen.File method*), 8
__init__() (*protogen.GeneratedFile method*), 9
__init__() (*protogen.InvalidDescriptorError method*), 10
__init__() (*protogen.Location method*), 11
__init__() (*protogen.Message method*), 12
__init__() (*protogen.Method method*), 13
__init__() (*protogen.OneOf method*), 13
__init__() (*protogen.Options method*), 13
__init__() (*protogen.Plugin method*), 14
__init__() (*protogen.PyIdent method*), 15
__init__() (*protogen.PyImportPath method*), 16
__init__() (*protogen.Registry method*), 16
__init__() (*protogen.ResolutionError method*), 19
__init__() (*protogen.Service method*), 19

A

all_enums() (*protogen.Registry method*), 16
all_files() (*protogen.Registry method*), 16
all_messages() (*protogen.Registry method*), 16
all_services() (*protogen.Registry method*), 16

C

Cardinality (*class in protogen*), 3
cardinality (*protogen.Field attribute*), 6
CodeGeneratorResponse (*class in protogen*), 4

D

default_py_import_func() (*in module protogen*), 19
dependencies (*protogen.File attribute*), 8
desc (*protogen.ResolutionError attribute*), 19

E

Enum (*class in protogen*), 4
enum (*protogen.Field attribute*), 6
enum_by_name() (*protogen.Registry method*), 16
enums (*protogen.File attribute*), 8
enums (*protogen.Message attribute*), 11
enums_by_package() (*protogen.Registry method*), 16
EnumValue (*class in protogen*), 5
error() (*protogen.Plugin method*), 14
extendee (*protogen.Field attribute*), 6
Extension (*in module protogen*), 5
extensions (*protogen.Message attribute*), 12

F

Field (*class in protogen*), 5
fields (*protogen.Message attribute*), 11
fields (*protogen.OneOf attribute*), 13
File (*class in protogen*), 7
file (*protogen.ResolutionError attribute*), 19
file_by_name() (*protogen.Registry method*), 17
file_content() (*protogen.CodeGeneratorResponse method*), 4
files_by_package() (*protogen.Registry method*), 17
files_to_generate (*protogen.Plugin attribute*), 14
full_name (*protogen.Enum attribute*), 4
full_name (*protogen.EnumValue attribute*), 5
full_name (*protogen.Field attribute*), 6
full_name (*protogen.Message attribute*), 11
full_name (*protogen.Method attribute*), 12
full_name (*protogen.OneOf attribute*), 13
full_name (*protogen.Service attribute*), 19

G

generate (*protogen.File attribute*), 8
generated_filename_prefix (*protogen.File attribute*), 7
GeneratedFile (*class in protogen*), 8
grpc_path (*protogen.Method attribute*), 12

H

has_file() (*protogen.CodeGeneratorResponse method*), 4

I

ident() (*protogen.PyImportPath method*), 16

input (*protogen.Method attribute*), 12
InvalidDescriptorError, 9
is_list() (*protogen.Field method*), 7
is_map() (*protogen.Field method*), 7

K

Kind (*class in protogen*), 10
kind (*protogen.Field attribute*), 6

L

leading_comments (*protogen.Location attribute*), 10
leading_detached_comments (*protogen.Location attribute*), 10
Location (*class in protogen*), 10
location (*protogen.Enum attribute*), 5
location (*protogen.EnumValue attribute*), 5
location (*protogen.Field attribute*), 6
location (*protogen.Message attribute*), 12
location (*protogen.Method attribute*), 12
location (*protogen.OneOf attribute*), 13
location (*protogen.Service attribute*), 19

M

map_key() (*protogen.Field method*), 7
map_value() (*protogen.Field method*), 7
Message (*class in protogen*), 11
message (*protogen.Field attribute*), 6
message_by_name() (*protogen.Registry method*), 17
messages (*protogen.File attribute*), 8
messages_by_package() (*protogen.Registry method*), 17
Method (*class in protogen*), 12
methods (*protogen.Service attribute*), 19
module
    protogen, 3

N

name (*protogen.GeneratedFile attribute*), 8
new_generated_file() (*protogen.Plugin method*), 15
number (*protogen.EnumValue attribute*), 5

O

OneOf (*class in protogen*), 13
oneof (*protogen.Field attribute*), 6
oneofs (*protogen.Message attribute*), 11
Options (*class in protogen*), 13
output (*protogen.Method attribute*), 12

P

P() (*protogen.GeneratedFile method*), 8
parameter (*protogen.Plugin attribute*), 14
parent (*protogen.Enum attribute*), 4
parent (*protogen.EnumValue attribute*), 5
parent (*protogen.Field attribute*), 6
parent (*protogen.Message attribute*), 11
parent (*protogen.Method attribute*), 12
parent (*protogen.OneOf attribute*), 13
parent_file (*protogen.Enum attribute*), 4
parent_file (*protogen.Field attribute*), 6
parent_file (*protogen.Message attribute*), 11
parent_file (*protogen.Service attribute*), 19
path (*protogen.Location attribute*), 10
Plugin (*class in protogen*), 14
print_import() (*protogen.GeneratedFile method*), 9
proto (*protogen.CodeGeneratorResponse attribute*), 4
proto (*protogen.Enum attribute*), 4
proto (*protogen.EnumValue attribute*), 5
proto (*protogen.Field attribute*), 6
proto (*protogen.File attribute*), 7
proto (*protogen.Message attribute*), 11
proto (*protogen.Method attribute*), 12
proto (*protogen.OneOf attribute*), 13
proto (*protogen.Service attribute*), 19
protogen
    module, 3
py_ident (*protogen.Enum attribute*), 4
py_ident (*protogen.EnumValue attribute*), 5
py_ident (*protogen.Message attribute*), 11
py_ident (*protogen.Service attribute*), 19
py_import_path (*protogen.File attribute*), 7
py_import_path (*protogen.PyIdent attribute*), 15
py_name (*protogen.Field attribute*), 6
py_name (*protogen.Method attribute*), 12
py_name (*protogen.PyIdent attribute*), 15
py_package_name (*protogen.File attribute*), 7
PyIdent (*class in protogen*), 15
PyImportPath (*class in protogen*), 15

Q

qualified_py_ident() (*protogen.GeneratedFile method*), 9

R

ref (*protogen.ResolutionError attribute*), 19
Registry (*class in protogen*), 16
registry (*protogen.Plugin attribute*), 14
ResolutionError, 18
resolve_enum_type() (*protogen.Registry method*), 17
resolve_message_type() (*protogen.Registry method*), 18
run() (*protogen.Options method*), 14

S

Service (*class in protogen*), 19
service_by_name() (*protogen.Registry method*), 18
services (*protogen.File attribute*), 8

services_by_package() (*protogen.Registry method*), 18
set_indent() (*protogen.GeneratedFile method*), 9
source_file (*protogen.Location attribute*), 10

T

trailing_comments (*protogen.Location attribute*), 10

V

values (*protogen.Enum attribute*), 5