

---

# **google-api-python-client-helpers**

## **Documentation**

*Release 1.2.6*

**Ross Vandegrift**

**Aug 17, 2021**



---

## Contents

---

<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Examples</b>	<b>5</b>
<b>3</b>	<b>Reference</b>	<b>7</b>
<b>4</b>	<b>Indicies and tables</b>	<b>11</b>
	<b>Python Module Index</b>	<b>13</b>
	<b>Index</b>	<b>15</b>



This library collects helpers that make google-api-python-client a bit nicer to use.

Contents:



# CHAPTER 1

---

## Installation

---

To install `google-api-python-client-helpers` from PyPI:

```
$ pip install google-api-python-client-helpers
```

To install from source, clone the repo from [Github](#). Then:

```
$ python setup.py install
```





## 2.1 build\_subresource

Some of the Google APIs have multiple levels of subresources. For example, to manage a GCP service account's customer managed keys, you can use `googleapiclienthelpers.discovery.build_subresource()` like this:

```
from googleapiclienthelpers.discovery import build_subresource

client = build_subresource('iam.projects.serviceAccounts.keys', 'v1')
```

## 2.2 IAM bindings

Many GCP resources provide `getIamPolicy` and `setIamPolicy` methods to modify permissions. To update a permission, you must fetch the policy, modify it, and set the new policy. A simpler, uniform interface is provided by `googleapiclienthelpers.iam.add_binding()` and `googleapiclienthelpers.iam.remove_binding()`. They make simple policy changes straightforward:

```
from googleapiclienthelpers.discovery import build_subresource
import googleapiclienthelpers.iam

client = build_subresource('cloudresourcemanager.projects', 'v1beta1')
googleapiclienthelpers.iam.add_binding(
    client,
    'roles/viewer',
    'user:my.user@example.com',
    resource='my-cool-project',
)
```

```
from googleapiclienthelpers.discovery import build_subresource
import googleapiclienthelpers.iam
```

(continues on next page)

(continued from previous page)

```
client = build_subresource('pubsub.projects.topics', 'v1')

googleapiclienthelpers.iam.remove_binding(
    client,
    'roles/viewer',
    'serviceAccount:my_svc_acct@my-cool-project.iam.gserviceaccount.com',
    topic='projects/my-cool-project/topics/interesting-stuff',
)
```

The bindings you request must meet any underlying constraints on `setIamPolicy` calls. For example, the API cannot be used to add a user as a project owner.

## 2.3 Waiter

Sometimes you need to block for an operation to complete. The *Waiter* makes this easy to do in a generic way. For instance, suppose you want to launch a Deployment Manager deployment and wait until it is complete:

```
from googleapiclienthelpers.discovery import build_subresource
from googleapiclienthelpers.waiter import Waiter

deployments = build_subresource('deploymentmanager.deployments', 'v2')
r = deployments.insert(project='example', body={
    'name': 'example',
    'target': {...},
}).execute()

waiter = Waiter(deployments.get, project='example', deployment='example')
waiter.wait('status', 'DONE')
```

To create a *Waiter*, you must supply:

1. The resource method you want to wait on.
2. All positional and keyword arguments required to invoke it.

Then, call the `wait()` method to start the process. The first argument is the response key that the waiter will look for. The second argument is the status it should wait for.

By default, this will poll the resource once every two seconds for a max of 60 retries. These values can be overridden.

Some helpers that make google-api-python-client a bit nicer to use.

### 3.1 discovery - helpers for API discovery

`googleapiclienthelpers.discovery.build_subresource` (*servicePath*, *version*, *\*\*kargs*)

Construct a Resource for interacting with an API's subresources.

Construct a Resource object for interacting with an API. The *servicePath* is a dot-delimited string. The first element is the *serviceName* from the Discovery service. Subsequent elements are the names of subresources accessed via the resulting client. The *version* is the version of the *serviceName* from the Discovery service.

**Some example servicePaths:** `iam.roles`      `iam.projects.serviceAccounts.keys`      `cloudresourcemanager.cloudresourcemanager.projects`

#### Parameters

- **servicePath** – string, see above description.
- **version** – string, the version of the service.
- **kargs** – all other arguments are passed on to build

**Returns** A Resource object with methods for interacting with the service.

### 3.2 iam - helpers for IAM bindings

`googleapiclienthelpers.iam.add_binding` (*client*, *role*, *member*, *\*\*kargs*)

Generic function to add an IAM binding to any GCP resource.

This function provides a simple, generic way to add a member to a role for any resource. The caller must supply a client that provides `getIamPolicy` and `setIamPolicy` methods. It preserves the policy's etag to prevent concurrent updates.

**Parameters**

- **client** – Resource, any googleapiclient resource or subresource.
- **role** – string, name of the role to modify (e.g., roles/viewer).
- **member** – string, the member to add (e.g., user:email@domain.com, serviceAccount:wooo@yea.iam.gserviceaccount.com).
- **\*\*kargs** – passed onto client's getIamPolicy and setIamPolicy. This must include parameter that identifies the resource whose policy will be modified. The parameter varies; check the docs on client.getIamPolicy for details.

**Returns** the result of client.setIamPolicy

`googleapiclienthelpers.iam.get_policy(client, **kargs)`

Retrieve a policy with GET or POST, as required

Some getIamPolicy methods are called with GET, and must have an empty body. Other getIamPolicy methods are called with POST, and must have a non-empty body equal to {}. This function provides a common interface to both.

**Parameters**

- **client** – Resource, any googleapiclient resource or subresource.
- **\*\*kargs** – passed onto client.getIamPolicy

**Returns** The result of client.getIamPolicy(\*\*kargs).execute()

`googleapiclienthelpers.iam.get_role_bindings(policy, role)`

Find the given role's binding from the given IAM policy.

Given an IAM policy and a desired role, return the binding associated with that role. If the role is not present in the policy, return None.

**Parameters**

- **policy** – list, the result of any getIamPolicy method.
- **role** – string, the role to pull from the policy.

**Returns** A list of dictionaries, or None if the role is not present.

`googleapiclienthelpers.iam.remove_binding(client, role, member, **kargs)`

Generic function to remove an IAM binding from any GCP resource.

This function provides a simple, generic way to remove a member from a role for any resource. The caller must supply a client that provides getIamPolicy and setIamPolicy methods. It preserves the policy's etag to prevent concurrent updates.

**Parameters**

- **client** – Resource, any googleapiclient resource or subresource.
- **role** – string, name of the role to modify (e.g., roles/viewer).
- **member** – string, the member to add (e.g., user:email@domain.com, serviceAccount:wooo@yea.iam.gserviceaccount.com).
- **\*\*kargs** – passed onto client's getIamPolicy and setIamPolicy. This must include parameter that identifies the resource whose policy will be modified. The parameter varies; check the docs on client.getIamPolicy for details.

**Returns** the result of client.setIamPolicy

### 3.3 waiter - generic way to wait for state

**class** googleapiclienthelpers.waiter.Waiter (*func, \*args, \*\*kwargs*)

Wait for a resource to reach a desired state.

This class implements a generic way to wait for an operation to reach a desired state.

To use: create a new instance by providing a callable and the arguments that it requires. Then, call wait() and the Waiter will poll the callable until the desired status is reached.

The callable you pass must have a .execute() method, like the Resource objects returned by googleapiclient.

For example, you might start an instance and wait until it reaches the RUNNING status: >>> instances = build\_subresource('compute.instances', 'v1') >>> instances.insert(...) >>> waiter = Waiter(instances.get, project=..., zone=..., instance=...) >>> waiter.wait('status', 'RUNNING')

**wait** (*field, value, retries=60, interval=2, terminal\_values=[]*)

Wait until an object reached the desired status

When called, the Waiter will invoke the callable provided at construction and capture its return. The Waiter extracts field, and checks if it's equal to value. If it is, the full response from the callable is returned.

**wait()** handles three kinds of field parameter:

- dict-like return where field is a string that specified a key.
- object-like return where field is a string naming an attribute.
- any kind of return where field is a one-place callable that extracts the needed data to compare.

#### Parameters

- **field** – string or callable, the data to extract.
- **value** – varies, the value indicating the desired state.
- **retries** – int, maximum number of times to poll.
- **interval** – float, time to sleep between polls.
- **terminal\_values** – list<string> of values that indicate an unrecoverable state. If encountered an exception is raised.

**exception** googleapiclienthelpers.waiter.WaiterException (*\*args, response=None*)



## CHAPTER 4

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`





### g

`googleapiclienthelpers`, [7](#)  
`googleapiclienthelpers.discovery`, [7](#)  
`googleapiclienthelpers.iam`, [7](#)  
`googleapiclienthelpers.waiter`, [9](#)



## A

`add_binding()` (*in module googleapiclienthelpers.iam*), 7

## B

`build_subresource()` (*in module googleapiclienthelpers.discovery*), 7

## G

`get_policy()` (*in module googleapiclienthelpers.iam*), 8

`get_role_bindings()` (*in module googleapiclienthelpers.iam*), 8

`googleapiclienthelpers` (*module*), 7

`googleapiclienthelpers.discovery` (*module*), 7

`googleapiclienthelpers.iam` (*module*), 7

`googleapiclienthelpers.waiter` (*module*), 9

## R

`remove_binding()` (*in module googleapiclienthelpers.iam*), 8

## W

`wait()` (*googleapiclienthelpers.waiter.Waiter method*), 9

`Waiter` (*class in googleapiclienthelpers.waiter*), 9

`WaiterException`, 9