

---

# **twilio-python Documentation**

*Release 5.7.0*

**Twilio Inc.**

**Nov 17, 2017**



---

## Contents

---

|          |                                       |           |
|----------|---------------------------------------|-----------|
| <b>1</b> | <b>Installation</b>                   | <b>1</b>  |
| <b>2</b> | <b>Getting Started</b>                | <b>3</b>  |
| <b>3</b> | <b>User Guide</b>                     | <b>5</b>  |
| <b>4</b> | <b>Upgrade Plan</b>                   | <b>45</b> |
| <b>5</b> | <b>API Reference</b>                  | <b>47</b> |
| <b>6</b> | <b>Deploying to Google App Engine</b> | <b>87</b> |
| <b>7</b> | <b>Frequently Asked Questions</b>     | <b>89</b> |
| <b>8</b> | <b>Changelog</b>                      | <b>91</b> |
| <b>9</b> | <b>Support and Development</b>        | <b>93</b> |
|          | <b>Python Module Index</b>            | <b>95</b> |



# CHAPTER 1

---

## Installation

---

Install from PyPi using `pip`, a package manager for Python.

```
pip install twilio
```

Don't have `pip` installed? Try installing it, by running this from the command line:

```
curl https://raw.githubusercontent.com/pypa/pip/master/contrib/get-pip.py | python
```

Or, install the library by downloading [the source](#), installing `setuptools`, navigating in the Terminal to the folder containing the **twilio-python** library, and then running:

```
python setup.py install
```



## CHAPTER 2

---

### Getting Started

---

The `/getting-started` will get you up and running in a few quick minutes. This guide assumes you understand the core concepts of Twilio. If you've never used Twilio before, don't fret! Just read [about how Twilio works](#) and then jump in!



Functionality is split over three different sub-packages within **twilio-python**. Below are in-depth guides to specific portions of the library.

## 3.1 REST API

Query the Twilio REST API to create phone calls, send messages and more!

### 3.1.1 Accessing REST Resources

To access Twilio REST resources, you'll first need to instantiate a *TwilioRestClient*.

#### Authentication

The *TwilioRestClient* needs your Twilio credentials. While these can be passed in directly to the constructor, we suggest storing your credentials as environment variables. Why? You'll never have to worry about committing your credentials and accidentally posting them somewhere public.

The *TwilioRestClient* looks for `TWILIO_ACCOUNT_SID` and `TWILIO_AUTH_TOKEN` inside the current environment.

With those two values set, create a new *TwilioRestClient*.

```
from twilio.rest import TwilioRestClient

conn = TwilioRestClient()
```

If you'd rather not use environment variables, pass your account credentials directly to the the constructor.

```
from twilio.rest import TwilioRestClient

ACCOUNT_SID = "XXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"
client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
```

## Proxies

*TwilioRestClient* supports HTTP and SOCKS4/5 proxies. You can change the proxy configuration at any time with the *Connection* class:

```
from twilio.rest.resources import Connection
from twilio.rest.resources.connection import PROXY_TYPE_SOCKS5

Connection.set_proxy_info(
    'example.com',
    5000,
    proxy_type=PROXY_TYPE_SOCKS5,
    proxy_user='username',
    proxy_pass='password',
)
```

The *TwilioRestClient* will retrieve and use the current proxy information for each request.

## Listing Resources

The *TwilioRestClient* gives you access to various list resources. *ListResource.list*, by default, returns the most recent 50 instance resources.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
resources = client.calls.list()
```

*resource.ListResource.list()* accepts paging arguments. The following will return page 3 with page size of 25.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
resources = client.calls.list(page=3, page_size=25)
```

## Listing All Resources

Sometimes you'd like to retrieve all records from a list resource. Instead of manually paging over the resource, the `resources.ListResource.iter` method returns a generator. After exhausting the current page, the generator will request the next page of results.

**Warning:** Accessing all your records can be slow. We suggest only doing so when you absolutely need all the records.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
for number in client.phone_numbers.iter():
    print number.friendly_name
```

## Get an Individual Resource

To get an individual instance resource, use `resources.ListResource.get()`. Provide the sid of the resource you'd like to get.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)

call = client.calls.get("CA123")
print call.to
```

### 3.1.2 Messages

The Messages resource manages all interaction with Twilio SMS and MMS messages. For more information, see the [Message REST Resource](#) documentation.

#### Sending a Text Message

The Message resource allows you to send an SMS message in a few lines of code.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
```

```
message = client.messages.create(
    body="Hello Monkey!", # Message body, if any
    to="+12125551234",
    from_="+15105551234",
)
print message.sid
```

If you want to send a message from a [short code](#) on Twilio, just set `from_` to your short code's number.

## Sending a Picture Message

To send a picture message (MMS), set `media_url` to the url of the picture you wish to send.

Don't forget to check the [availability of MMS in your area](#) before using this functionality.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)

message = client.messages.create(
    body="Hello Monkey!", # Message body, if any
    to="+12125551234",
    from_="+15105551234",
    media_url="http://example.com/image1.jpg"
)
```

You can send multiple pictures in the same message by setting `media_url` to a list of urls.

```
message = client.messages.create(
    body="Hello Monkey!", # Message body, if any
    to="+12125551234",
    from_="+15105551234",
    media_url=[ # List of media URLs, if any
        "http://example.com/image1.jpg",
        "http://example.com/image2.jpg",
    ],
)
```

## Retrieving Sent Messages

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)

for message in client.messages.list():
    print message.body
```

## Filtering Your Messages

The `list()` methods supports filtering on `to`, `from_`, and `date_sent`. The following will only show messages to “+5466758723” on January 1st, 2011.

```
from datetime import date
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)

messages = client.messages.list(
    to="+5466758723",
    date_sent=date(2011,1,1),
)

for message in messages:
    print message.body
```

## Redacting or Deleting Message Records

To protect your users’ privacy and/or comply with legal requirements, Twilio allows you to redact your Message bodies or delete the records outright.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
message_sid = "MM123"

client.messages.redact(message_sid)
message = client.messages.get(message_sid)
print message.body # Will be an empty string

client.messages.delete(message_sid) # Deletes record entirely, subsequent requests_
↳ will return 404
```

### 3.1.3 Phone Calls

The `Calls` resource manages all interaction with Twilio phone calls, including the creation and termination of phone calls. For more information, see the [Calls REST Resource](#) documentation.

#### Making a Phone Call

The `Calls` resource allows you to make outgoing calls. Before a call can be successfully started, you’ll need a to set up a url endpoint which outputs valid `TwiML`. This can be done with the `:class: twiml.Response` class, [get started here](#).

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
call = client.calls.create(to="9991231234", from_="9991231234",
                          url="http://foo.com/call.xml")

print call.length
print call.sid
```

## Retrieve a Call Record

If you already have a *Call* sid, you can use the client to retrieve that record.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
sid = "CA12341234"
call = client.calls.get(sid)
```

## Delete a Call Record

You can delete your *Call* resources from Twilio's storage to protect your users' privacy and/or comply with legal requirements.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
sid = "CA12341234"
client.calls.delete(sid)
```

## Accessing Specific Call Resources

Each *Call* resource also has access to its notifications, recordings, and transcriptions. These attributes are *ListResources*, just like the *Calls* resource itself.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
```

```

sid = "CA12341234"
call = client.calls.get(sid)

print call.notifications.list()
print call.recordings.list()
print call.transcriptions.list()

```

However, what if you only have a `call_sid`, and not the actual *Resource*? No worries, as `Calls.list()` can be filtered based on a given `call_sid`.

```

from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
sid = "CA24234"
print client.notifications.list(call=sid)
print client.recordings.list(call=sid)
print client.transcriptions.list(call=sid)

```

## Modifying Live Calls

The *Call* resource makes it easy to find current live calls and redirect them as necessary

```

from twilio.rest import TwilioRestClient
from twilio.rest.resources import Call

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
calls = client.calls.list(status=Call.IN_PROGRESS)
for c in calls:
    c.route(
        "http://twimlets.com/holdmusic?Bucket=com.twilio.music.ambient",
        method="POST"
    )

```

Ending all live calls is also possible

```

from twilio.rest import TwilioRestClient
from twilio.rest.resources import Call

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
calls = client.calls.list(status=Call.IN_PROGRESS)
for c in calls:
    c.hangup()

```

Note that `hangup()` will also cancel calls currently queued.

If you already have a *Call* sid, you can use the *Calls* resource to update the record without having to use *get()* first.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
sid = "CA12341234"
client.calls.update(sid, method="POST",
    url="http://twimlets.com/holdmusic?Bucket=com.twilio.music.ambient")
```

Hanging up the call also works.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
sid = "CA12341234"
client.calls.hangup(sid)
```

### 3.1.4 Phone Numbers

With Twilio you can search and buy real phone numbers using the API.

For more information, see the [IncomingPhoneNumbers REST Resource](#) documentation.

#### Searching and Buying a Number

Finding numbers to buy couldn't be easier. The example below searches for a number with US area code '530'. Once we find one, we'll purchase it for our account.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
numbers = client.phone_numbers.search(area_code=530)

if numbers:
    numbers[0].purchase()
else:
    print "No numbers in 530 available"
```

#### Local, Toll Free, and Mobile Numbers

By default, *PhoneNumbers.search()* looks for local phone numbers. To search for numbers for a given type, include the desired type in the *type* paramter.

Available types are: - *local* - *tollfree* - *mobile*

```
# Local
numbers = client.phone_numbers.search(type="local")

# Toll Free
numbers = client.phone_numbers.search(type="tollfree")

# Mobile
numbers = client.phone_numbers.search(type="mobile")
```

Similarly, you can purchase numbers of a given type. You must still include the `phone_number` or `area_code` parameters.

```
# Local
numbers = client.phone_numbers.purchase(type='local', phone_number '(919) 123-4567')

# Toll Free
numbers = client.phone_numbers.purchase(type='tollfree', phone_number '(919) 123-4567')

# Mobile
numbers = client.phone_numbers.purchase(type='mobile', phone_number '(919) 123-4567')
```

## Numbers Containing Words

Phone number search also supports looking for words inside phone numbers. The following example will find any phone number with “FOO” in it.

```
numbers = client.phone_numbers.search(contains="FOO")
```

You can use the “\*” wildcard to match any character. The following example finds any phone number that matches the pattern “D\*D”.

```
numbers = client.phone_numbers.search(contains="D*D")
```

## International Numbers

By default, the client library will look for US numbers. Set the `country` keyword to a country code of your choice to search for international numbers.

```
numbers = client.phone_numbers.search(country="FR")
```

`PhoneNumbers.search()` method has plenty of other options to augment your search :

- `region`: When searching the US, show numbers in this state
- `postal_code`: Only show numbers in this area code
- `rate_center`: US only.
- `near_lat_long`: Find numbers near this latitude and longitude.
- `distance`: Search radius for a Near- query in miles.

The [AvailablePhoneNumbers REST Resource](#) documentation has more information on the various search options.

## Buying a Number

If you've found a phone number you want, you can purchase the number.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
number = client.phone_numbers.purchase(phone_number="+15305431234")
```

However, it's easier to purchase numbers after finding them using search (as shown in the first example).

## Updating Properties on a Number

To update the properties on a phone number, such as the Voice URL, call `update()` on the phone number object, with any of the parameters listed in the [IncomingPhoneNumbers Resource documentation](#)

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
for number in client.phone_numbers.list(api_version="2010-04-01"):
    number.update(voice_url="http://twimlets.com/holdmusic?" +
                  "Bucket=com.twilio.music.ambient",
                  status_callback="http://example.com/callback")
```

## Changing Applications

An *Application* encapsulates all necessary URLs for use with phone numbers. Update an application on a phone number using `update()`.

```
from twilio.rest import TwilioRestClient

phone_sid = "PNXXXXXXXXXXXXXXXXXXXXXXXXXXXX"

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
number = client.phone_numbers.update(phone_sid, sms_application_sid="AP123")
```

See *Applications* for instructions on updating and maintaining Applications.

## Validate a Phone Number

See validation instructions here: *Caller Ids*:

### 3.1.5 Accounts

Managing Twilio accounts is straightforward. Update your own account information or create and manage multiple subaccounts.

For more information, see the [Account REST Resource](#) documentation.

#### Updating Account Information

Use the `Account.update()` to modify one of your accounts. Right now the only valid attribute is *FriendlyName*.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
account = client.accounts.get(ACCOUNT_SID)
account.update(friendly_name="My Awesome Account")
```

#### Creating Subaccounts

Subaccounts are easy to make.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
subaccount = client.accounts.create(name="My Awesome SubAccount")
```

#### Managing Subaccounts

Say you have a subaccount for Client X with an account sid *AC123*

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)

# Client X's subaccount
subaccount = client.accounts.get('AC123')
```

Client X hasn't paid you recently, so let's suspend their account.

```
subaccount.suspend()
```

If it was just a misunderstanding, reenable their account.

```
subaccount.activate()
```

Otherwise, close their account permanently.

```
subaccount.close()
```

**Warning:** This action can't be undone.

### 3.1.6 Conferences and Participants

The *Conference* resource manages all interaction with Twilio conferences, such as listing and managing participants. For more information, see the [Conference REST Resource](#) and [Participant REST Resource](#) documentation.

#### Listing Conferences

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
conferences = client.conferences.list()

for conference in conferences:
    print conference.sid
```

#### Filtering Conferences

The *Conferences.list()* method supports filtering on `status`, `date_updated`, `date_created` and `friendly_name`. The following code will return a list of all in-progress conferences and print their friendly name.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
conferences = client.conferences.list(status="in-progress")

for conference in conferences:
    print conference.friendly_name
```

#### Listing Participants

Each *Conference* has a `participants` instance which represents all current users in the conference

```

from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
conference = client.conferences.get("CF123")

for participant in conference.participants.list():
    print participant.sid

```

*Conferences* and *Participants* are subclasses of *ListResource*. Therefore, their instances have the inherited methods such as `count()`.

## Managing Participants

Each *Conference* has a `participants` function that returns a *Participants* instance. This behavior differs from other list resources because *Participants* needs a participant sid AND a conference sid to access the participants resource.

Participants can be either muted or kicked out of the conference. The following code kicks out the first participant and mutes the rest.

```

from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
participants = client.participants("CF123").list()

if len(participants) == 0:
    return

# Kick the first person out
participants.pop().kick()

# And mute the rest
for participant in participants:
    participant.mute()

```

## 3.1.7 Applications

An application inside of Twilio is just a set of URLs and other configuration data that tells Twilio how to behave when one of your Twilio numbers receives a call or message.

For more information, see the [Application REST Resource](#) documentation.

### Listing Your Applications

The following code will print out the `friendly_name` for each *Application*.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
for app in client.applications.list():
    print app.friendly_name
```

## Filtering Applications

You can filter applications by FriendlyName

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
for app in client.applications.list(friendly_name="FOO"):
    print app.sid
```

## Creating an Application

When creating an application, no fields are required. We create an application with only a `friendly_name`. The `Applications.create()` method accepts many other arguments for url configuration.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
application = client.applications.create(friendly_name="My New App")
```

## Updating an Application

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
url = "http://twimlets.com/holdmusic?Bucket=com.twilio.music.ambient"
app_sid = 'AP123' # the app you'd like to update
application = client.applications.update(app_sid, voice_url=url)
```

## Deleting an Application

```

from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

app_sid = 'AP123' # the app you'd like to delete
client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
client.applications.delete(app_sid)

```

### 3.1.8 Notifications

A *Notification* resource represents a single log entry made by Twilio in the course of handling your calls or your use of the REST API. For more information, see the [Notifications REST Resource](#) documentation.

#### Listing Your Notifications

The following code will print out additional information about each of your current *Notification* resources.

```

from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
for notification in client.notifications.list():
    print notification.more_info

```

You can filter transcriptions by log and message\_date. The log value is 0 for *ERROR* and 1 for *WARNING*.

```

from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)

ERROR = 0

for notification in client.notifications.list(log=ERROR):
    print notification.error_code

```

**Note:** Due to the potentially voluminous amount of data in a notification, the full HTTP request and response data is only returned in the *Notification* instance resource representation.

## Deleting Notifications

Your account can sometimes generate an inordinate amount of *Notification* resources. The *Notifications* resource allows you to delete unnecessary notifications.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
client.notifications.delete("NO123")
```

## 3.1.9 Recordings

A *Recording* resource represents an individually recorded call. Recordings are generated when you use the *Record* TwiML verb. For more information, see the [Recordings REST Resource](#) documentation.

### Audio Formats

Each *Recording* has a *formats* dictionary which lists the audio formats available for each recording. Below is an example *formats* dictionary.

```
{
  "mp3": "https://api.twilio.com/cowbell.mp3",
  "wav": "http://www.dailywav.com/0112/noFateButWhatWeMake.wav",
}
```

### Listing Your Recordings

The following code will print out the duration for each *Recording*.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
for recording in client.recordings.list():
    print recording.duration
```

You can filter recordings by *CallSid* by passing the *Sid* as *call*. Filter recordings using *before* and *after* dates.

The following will only show recordings made before January 1, 2011.

```
from datetime import date
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"
```

```
client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
for recording in client.recordings.list(before=date(2011,1,1)):
    print recording.duration
```

## Deleting Recordings

The *Recordings* resource allows you to delete unnecessary recordings.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
client.recordings.delete("RC123")
```

## Accessing Related Transcriptions

The *Recordings* allows you to retrieve associated transcriptions. The following prints out the text for each of the transcriptions associated with this recording.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
recording = client.recordings.get("RC123")

for transcription in recording.transcriptions.list():
    print transcription.transcription_text
```

### 3.1.10 Transcriptions

A *Transcription* resource represents a transcription of a recording. The transcription text itself is the result of converting an audio recording to readable text. Transcriptions are generated from recordings via the *Twiml <Record>* verb. Using the API, you can only read your transcription records.

For more information, see the [Transcriptions REST Resource](#) documentation.

#### Listing Your Transcriptions

The following code will print out the transcription text of each *Transcription*.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"
```

```
client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
for transcription in client.transcriptions.list():
    print transcription.transcriptiontext
```

### 3.1.11 Usage API

#### Usage Records

You can query your UsageRecords to see the activity of your Twilio account. Here we'll make a query for the activity over the last day.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)

todays_usage = client.usage.records.today
for category in todays_usage:
    print '{0}: {1}'.format(category.description, category.price)
```

This will print out the amount spent for each usage category over the last day. To see all of the possible usage categories, as well as all of the possible time ranges to query over, check out the [REST API UsageRecord docs](<https://www.twilio.com/docs/api/rest/usage-records#usage-all-categories>).

#### Usage Triggers

You can also set up a UsageTrigger. UsageTriggers notify you once your usage reaches a certain level. Here we'll set up a UsageTrigger to tell us when we've sent 1000 total messages.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)

trigger = client.usage.triggers.create(
    friendly_name="1000 messages",
    usage_category="sms",
    trigger_value="1000",
    callback_url="http://example.com/thousand-sms",
    trigger_by="count"
)
```

Once this trigger is created, Twilio will make a POST to “<http://example.com/thousand-sms>” once your account has sent 1000 messages.

## Relative Triggers

The previous example is helpful, but it is an example of an *absolute* trigger value. What if your account has already sent 1000 messages, and you want to be notified once it sends the next 1000, regardless of how many you've previously sent? To do this, you must add a '+' in front of the `trigger_value`, so Twilio knows this is a relative count. Here's an example that shows how to set a relative trigger.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)

trigger = client.usage.triggers.create(
    friendly_name="1000 messages",
    usage_category="sms",
    trigger_value="+1000",
    callback_url="http://example.com/thousand-sms",
    trigger_by="count"
)
```

Once this trigger is created, Twilio will make a POST to “<http://example.com/thousand-sms>” once your account has sent 1000 more messages.

### 3.1.12 Caller Ids

#### Validate a Phone Number

Validating a phone number can be done with the Python helper library.

The code block below provides a simple example.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
response = client.caller_ids.validate("+449876543212")
print response.validation_code
```

Twilio will call the provided number and wait for the validation code to be entered.

#### Delete a Phone Number

Deleting a validated phone number is just as easy as validating.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
account = "ACXXXXXXXXXXXXXXXXXXXX"
token = "YYYYYYYYYYYYYYYYYYYY"
client = TwilioRestClient(account, token)
```

```
response = client.caller_ids.list(phone_number="+15555555555")
callerid = response[0]
callerid.delete()
```

### 3.1.13 Queues and Members

The *Queue* resource allows you to query and manage the state of individual call queues. For more information, see the [Queue REST Resource](#).

The *Members* resource is a subresource of a Queue resource. It represents the set of members currently in a queue. See the [Member REST Resource](#) documentation for more information.

#### Listing Queues

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
queues = client.queues.list()

for queue in queues:
    print queue.sid
```

#### Listing Queue Members

Each *Queue* has a *queue\_members* instance which represents all current calls in the queue.

```
from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
queue = client.queues.get("QU123")

for member in queue.queue_members.list():
    print member.call_sid
```

#### Getting a specific Queue Member

To retrieve information about a specific member in the queue, each *Members* has a *Members.get()* method. *Members.get()* accepts one argument. The argument can either be a *call\_sid* that's in the queue, in which case *get()* will return a *Member* instance representing that call, or the argument can be `Front`, in which case *Get()* will return a *Member* instance representing the first call in the queue.

```

from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"
QUEUE_SID = "QUaaaaaaaaaaaaaa"
CALL_SID = "CAxxxxxxxxxxxxxxxx"
client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
members = client.queues.get(QUEUE_SID).queue_members

# Get the first call in the queue
print members.get('Front').date_enqueued

# Get the call with the given call sid in the queue
print members.get(CALL_SID).current_position

```

## Dequeuing Queue Members

To dequeue a specific member from the queue, each *Members* has a *dequeue()* method. *dequeue()* accepts an argument and two optional keyword arguments. The first argument is the url of the twiml document to be executed when the member is dequeued. The other two are *call\_sid* and *method*, their default values are 'Front' and 'GET'

```

from twilio.rest import TwilioRestClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"
QUEUE_SID = "QUaaaaaaaaaaaaaa"

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)
members = client.queues.get(QUEUE_SID).queue_members

# Dequeue the first call in the queue
print members.dequeue('http://www.twilio.com/welcome/call')

```

## 3.1.14 Sip In

### Getting started with Sip

Connect your SIP endpoints to Twilio and start building voice apps with Twilio's APIs and application stack. If you're unfamiliar with SIP, please see the [SIP API Documentation](#) on our website.

### Creating a Sip Domain

The Domains resource allows you to create a new domain. To create a new domain, you'll need to choose a unique domain that lives under sip.twilio.com. For example, dunder-mifflin-scranton.sip.twilio.com. For more information, see the [Domains resource documentation](#) on our website.

```

from twilio.rest import TwilioRestClient

# Your Account Sid and Auth Token from twilio.com/user/account
account_sid = "ACXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
auth_token = "YYYYYYYYYYYYYYYYYYYY"

```

```
client = TwilioRestClient(account_sid, auth_token)

domain = client.sip.domains.create(
    friendly_name="The Office Domain",
    voice_url="http://example.com/voice",
    domain_name="dunder-mifflin-scranton.sip.twilio.com",
)
print domain.sid
```

## Creating a new IpAccessControlList

To control access to your new domain, you'll need to explicitly grant access to individual ip addresses. To do this, you'll first need to create an IpAccessControlList to hold the ip addresses you wish to allow.

```
from twilio.rest import TwilioRestClient

# Your Account Sid and Auth Token from twilio.com/user/account
account_sid = "ACXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
auth_token = "YYYYYYYYYYYYYYYYYYYYYYYYYYY"
client = TwilioRestClient(account_sid, auth_token)

ip_acl = client.sip.ip_access_control_lists.create(
    friendly_name="The Office IpAccessControlList",
)
print ip_acl.sid
```

## Adding a new IpAddress

Now it's time to add an IpAddress to your new IpAccessControlList.

```
from twilio.rest import TwilioRestClient

# Your Account Sid and Auth Token from twilio.com/user/account
account_sid = "ACXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
auth_token = "YYYYYYYYYYYYYYYYYYYYYYYYYYY"
client = TwilioRestClient(account_sid, auth_token)

ip_address = client.sip.ip_addresses(
    "AL456", # IpAccessControlList sid
).create(
    friendly_name="Dwight's Computer",
    ip_address="192.168.1.42",
)
print ip_address.sid
```

## Adding an IpAccessControlList to a Domain

Once you've created a Domain and an IpAccessControlList you need to associate them. To do this, create an IpAccessControlListMapping.

```
from twilio.rest import TwilioRestClient

# Your Account Sid and Auth Token from twilio.com/user/account
```

```

account_sid = "ACXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
auth_token = "YYYYYYYYYYYYYYYYYYYYYYY"
client = TwilioRestClient(account_sid, auth_token)

ip_access_control_list_mapping = client.sip.ip_access_control_list_mappings(
    "SD456", # SIP Domain sid
).create(ip_access_control_list_sid="AL789")
print ip_access_control_list_mapping.sid

```

## 3.2 TaskRouter

Query the Twilio TaskRouter API to set up workspaces and task routing, and create capability tokens to authorize your client-side code to safely update state.

### 3.2.1 TaskRouter

Twilio TaskRouter is a system for distributing tasks such as phone calls, leads, support tickets, and other work items to the people and processes that can best handle them.

For more information, see the *TaskRouter documentation* <<https://www.twilio.com/docs/taskrouter>>.

#### Creating a Workspace

A Workspace is a container for your Tasks, Workers, TaskQueues, Workflows and Activities. Each of these items exists within a single Workspace and will not be shared across Workspaces.

The following code will create a new *Workspace* resource and print its unique ID.

```

from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYYYYY"

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)

workspace = client.workspaces.create(
    friendly_name="Customer Support",
    template="FIFO", # Sets up default activities and a FIFO TaskQueue
)
print workspace.sid

```

The following code will get an instance of an existing workspace resource

```

from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYYYYY"

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)
workspace = client.workspaces.get(WORKSPACE_SID)
print workspace.friendly_name

```

The following code will get the list of all existing workspace resources

```
from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)
for workspace in client.workspaces.list():
    print workspace.friendly_name
```

The following code will create a update an existing *Workspace* resource

```
from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)
workspace = client.workspaces.update(
    WORKSPACE_SID,
    friendly_name='Test Workspace',
    event_callback_uri="http://www.example.com",
    template='FIFO')
```

The following code will delete an existing workspace resource

```
from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)
client.workspaces.delete(WORKSPACE_SID)
```

## Workflows

Workflows control how tasks will be prioritized and routed into TaskQueues, and how Tasks should escalate in priority or move across queues over time. Workflows are described in a simple JSON format and can be modified through the REST API or through the account portal.

The following code will create a new *Workflow* resource and print its unique ID:

```
from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZ"

# Some JSON to configure the Workflow. See the documentation at
# http://www.twilio.com/docs/taskrouter for more details.
```

```

CONFIG = """
{
  "task_routing":{
    "filters":[
      {
        "friendly_name":"Gold Tickets",
        "expression":"customer_value == 'Gold' AND type == 'ticket'",
        "targets":[
          {
            "task_queue_sid":"WQ0123456789abcdef0123456789abcdef",
            "priority":"2"
          }
        ]
      }
    ],
    "default_filter":{
      "task_queue_sid":"WQabcdef01234567890123456789abcdef"
    }
  }
}
"""

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)

workflow = client.workflows(WORKSPACE_SID).create(
    friendly_name="Incoming Call Flow",
    assignment_callback_url="https://example.com/callback",
    fallback_assignment_callback_url="https://example.com/callback2",
    configuration=CONFIG
)
print workflow.sid

```

The following code will get a instance of an existing workflow resource

```

from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZ"

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)

workflow = client.workflows(WORKSPACE_SID).get(WORKFLOW_SID)
print workflow.friendly_name

```

The following code will get a list of all existing workflow resources

```

from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZ"

```

```

client = TwilioTaskRouterClient (ACCOUNT_SID, AUTH_TOKEN)

for workflow in client.workflows (WORKSPACE_SID).list ()
    print workflow.friendly_name

```

The following code will update an existing workflow resource

```

from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZZZ"

# Some JSON to configure the Workflow. See the documentation at
# http://www.twilio.com/docs/taskrouter for more details.
CONFIG = """
{
  "task_routing":{
    "filters":[
      {
        "friendly_name":"Gold Tickets",
        "expression":"customer_value == 'Gold' AND type == 'ticket'",
        "targets":[
          {
            "task_queue_sid":"WQ0123456789abcdef0123456789abcdef",
            "priority":"2"
          }
        ]
      },
      {
        "targets": [
          {
            "queue": "WQ2acd4c1a41ffadce5d1bac9e1ce2fa9f",
            "priority": "1"
          }
        ],
        "friendly_name": "Marketing",
        "expression": "type == 'marketing'"
      }
    ],
    "default_filter":{
      "task_queue_sid":"WQabcdef01234567890123456789abcdef"
    }
  }
}
"""

client = TwilioTaskRouterClient (ACCOUNT_SID, AUTH_TOKEN)

workflow = client.workflows (WORKSPACE_SID).update (
    WORKFLOW_SID,
    friendly_name="Incoming Call Flow",
    assignment_callback_url="https://example.com/callback",
    fallback_assignment_callback_url="https://example.com/callback2",

```

```

    configuration=CONFIG
)
print workflow.sid

```

The following code will delete an existing *Workflow*

```

from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZZZ"

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)

client.workflows(WORKSPACE_SID).delete(
    WORKFLOW_SID
)

```

## Activities

Activities describe the current status of your Workers, which determines whether they are eligible to receive task assignments. Workers are always set to a single Activity.

To create a new Activity:

```

from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZZZ"

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)
activity = client.activities(WORKSPACE_SID).create(
    friendly_name="Coffee Break",
    available=False, # Whether workers are available to handle tasks during this_
    ↪activity
)

```

To get an existing activity resource

```

from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZZZ"

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)

```

```
activity = client.activities(WORKSPACE_SID).get(ACTIVITY_SID)
print activity.friendly_name
```

To get a list of existing activity resources

```
from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZZZ"

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)
for activity in client.activities(WORKSPACE_SID).list():
    print activity.friendly_name
```

To update an existing Activity

```
from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZZZ"

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)
activity = client.activities(WORKSPACE_SID).update(
    ACTIVITY_SID,
    friendly_name="Coffee Break",
    available=True,
)
```

To delete an existing Activity

```
from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZZZ"

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)
activity = client.activities(WORKSPACE_SID).delete(
    ACTIVITY_SID
)
```

## Workers

Workers represent an entity that is able to perform tasks, such as an agent working in a call center, or a salesperson handling leads.

To create a new *Worker*:

```
from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZZZ"

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)
worker = client.workers(WORKSPACE_SID).create(
    friendly_name="Jamie",
    attributes="""{
        "phone": "+14155551234",
        "languages": ["EN", "ES"]
    }
    """
)
print worker.sid
```

To get an existing worker instance

```
from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZZZ"

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)
worker = client.workers(WORKSPACE_SID).get(WORKER_SID)
print worker.friendly_name;
```

To get an existing worker list

```
from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZZZ"

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)
for worker in client.workers(WORKSPACE_SID).list():
    print worker.friendly_name;
```

To update an existing *Worker*

```
from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"
```

```
# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZZZZZZZ"

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)
worker = client.workers(WORKSPACE_SID).update(
    WORKER_SID,
    friendly_name="Jamie Howe",
    attributes="""{
        "phone": "+14155551234",
        "languages": ["EN", "ES", "DE"]
    }
    """
)
print worker.sid
```

To delete an existing *Worker*

```
from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZZZZZZZ"

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)
client.workers(WORKSPACE_SID).delete(
    WORKER_SID
)
```

## TaskQueues

TaskQueues are the resource you use to categorize Tasks and describe which Workers are eligible to handle those Tasks. As your Workflows process Tasks, those Tasks will pass through one or more TaskQueues until the Task is assigned and accepted by an eligible Worker.

To create a new *TaskQueue*:

```
from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZZZZZZZ"

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)

queue = client.task_queues(WORKSPACE_SID).create(
    friendly_name="Sales",
    # The Activity to assign workers when a task is reserved for them
    reservation_activity_sid="WA111111111111",
    # The Activity to assign workers when a task is assigned to them
    assignment_activity_sid="WA222222222222",
```

```
)
print queue.sid
```

To get an existing `:class`TaskQueue`` instance

```
from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZZZ"

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)

queue = client.task_queues(WORKSPACE_SID).get(TASKQUEUE_SID)
print queue.sid
```

To get an existing `:class`TaskQueue`` list

```
from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZZZ"

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)

for queue in client.task_queues(WORKSPACE_SID).list():
    print queue.sid
```

To update an existing `TaskQueue`

```
from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZZZ"

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)

queue = client.task_queues(WORKSPACE_SID).update(
    TASKQUEUE_SID,
    friendly_name="Sales+Pre-Sales",
    # The Activity to assign workers when a task is reserved for them
    reservation_activity_sid="WA111111111111",
    # The Activity to assign workers when a task is assigned to them
    assignment_activity_sid="WA222222222222",
)
print queue.sid
```

To delete an existing *TaskQueue*

```
from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZZZ"

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)

queue = client.task_queues(WORKSPACE_SID).delete(
    TASKQUEUE_SID
)
print queue.sid
```

## Tasks

A Task instance resource represents a single item of work waiting to be processed.

To create a new *Task* via the REST API:

```
from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZZZ"
WORKFLOW_SID = "WWXXXXXXXXXXXXXXXXXX"
# Some JSON containing attributes for this task. User-defined.
TASK_ATTRIBUTES = """{
    "type": "call",
    "contact": "+15558675309",
    "customer-value": "gold",
    "task-reason": "support",
    "callSid": "CA42ed11..."
}"""

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)
task = client.tasks(WORKSPACE_SID).create(
    attributes=TASK_ATTRIBUTES,
    assignment_status='pending',
    workflow_sid=WORKFLOW_SID
)
print task.sid
```

To get an existing *Task* instance

```
from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
```

```

AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZZZZZ"
WORKFLOW_SID = "WWXXXXXXXXXXXXXXXXXX"
# Some JSON containing attributes for this task. User-defined.
TASK_ATTRIBUTES = """{
    "type": "call",
    "contact": "+2014068777",
    "customer-value": "gold",
    "task-reason": "support",
    "callSid": "CA42ed11..."
}"""

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)
task = client.tasks(WORKSPACE_SID).delete(TASK_SID)
print task.attributes

```

To get an existing *Task* list

```

from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZZZZZ"
WORKFLOW_SID = "WWXXXXXXXXXXXXXXXXXX"
# Some JSON containing attributes for this task. User-defined.
TASK_ATTRIBUTES = """{
    "type": "call",
    "contact": "+2014068777",
    "customer-value": "gold",
    "task-reason": "support",
    "callSid": "CA42ed11..."
}"""

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)
for task in client.tasks(WORKSPACE_SID).list():
    print task.attributes

```

To update an existing *Task*

```

from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZZZZZ"
WORKFLOW_SID = "WWXXXXXXXXXXXXXXXXXX"
# Some JSON containing attributes for this task. User-defined.
TASK_ATTRIBUTES = """{
    "type": "call",

```

```

        "contact": "+2014068777",
        "customer-value": "gold",
        "task-reason": "support",
        "callSid": "CA42ed11..."
    }"""

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)
task = client.tasks(WORKSPACE_SID).update(
    TASK_SID,
    attributes=TASK_ATTRIBUTES,
    assignment_status='pending',
    workflow_sid=WORKFLOW_SID
)
print task.sid

```

To delete an existing *Task*

```

from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZZZ"
WORKFLOW_SID = "WWXXXXXXXXXXXXXXXXXX"
# Some JSON containing attributes for this task. User-defined.
TASK_ATTRIBUTES = """{
    "type": "call",
    "contact": "+2014068777",
    "customer-value": "gold",
    "task-reason": "support",
    "callSid": "CA42ed11..."
}"""

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)
client.tasks(WORKSPACE_SID).delete(
    TASK_SID
)

```

Using Workflow builder helper classes to create a *Workflow* resource.

```

from twilio.rest import TwilioTaskRouterClient

# To find these visit https://www.twilio.com/user/account
ACCOUNT_SID = "ACXXXXXXXXXXXXXXXXXXXX"
AUTH_TOKEN = "YYYYYYYYYYYYYYYYYYYY"

# See previous examples to create a Workspace
WORKSPACE_SID = "WSZZZZZZZZZZZZZZZZ"

rules = [
    WorkflowRule("1==1", [WorkflowRuleTarget("WQeae4fc2f4db7f377c5d3758fb08b79b7",
↪ "1==1", 1, 20)], "SomeQ"),
    WorkflowRule("1==1", [WorkflowRuleTarget("WQ19ebe92fb33522f018b5a31d805d94da",
↪ "1==1", 1, 210)], "SomeOtherQ")
]

```

```

]
default_target = WorkflowRuleTarget("WQ9963154bf3122d0a0558f3763951d916", "1==1",
↪None, None)
config = WorkflowConfig(rules, default_target)
print config.to_json()

client = TwilioTaskRouterClient(ACCOUNT_SID, AUTH_TOKEN)

workflow = client.workflows(WORKSPACE_SID).create(
    friendly_name= "Incoming Call Flow",
    assignment_callback_url= "https://example.com/callback",
    fallback_assignment_callback_url= "https://example.com/callback2",
    configuration= config.to_json()
)

print workflow.sid

```

### 3.2.2 TaskRouter Capability Tokens

TaskRouter’s Worker.js library lets you add `TaskRouter` Worker Activity controls and event notifications to your web applications. Worker.js uses a Websocket connection to TaskRouter to receive realtime notifications of Worker Reservations and Task details, and provides a simple API for modifying a Worker’s current Activity.

TaskRouter uses Twilio capability tokens to delegate scoped access to TaskRouter resources to your JavaScript application. Twilio capability tokens conform to the JSON Web Token (commonly referred to as a JWT and pronounced “jot”) standard, which allow for limited-time use of credentials by a third party. Your web server needs to generate a Twilio capability token and provide it to your JavaScript application in order to register a TaskRouter worker.

`TaskRouterCapability` is responsible for the creation of these capability tokens. You’ll need your Twilio AccountSid and AuthToken, the Sid of the Workspace you want to authorize access to, and the Sid of the Worker you’re granting authorization for.

```

from twilio.task_router import TaskRouterCapability

# Get these values from https://twilio.com/user/account
account_sid = "AC123"
auth_token = "secret"

# Create a Workspace and Worker in the TaskRouter account portal
# or through the TaskRouter API
workspace_sid = "WS456"
worker_sid = "WK789"

capability = TaskRouterCapability(account_sid, auth_token,
                                workspace_sid, worker_sid)

```

By default, the Capability object will allow the Worker.js process to read from and write to the websockets used to communicate events, and also to fetch the list of available activities in the workspace.

There are three additional permissions you can grant using the Capability token, and in most cases you’ll want to allow all of them for your application:

#### Attribute Fetch

This authorizes requests to retrieve the registered Worker’s attributes from the TaskRouter API.

```
capability.allow_worker_fetch_attributes()
```

### Worker Activity Update

This authorizes updates to the registered Worker's current Activity.

```
capability.allow_worker_activity_updates()
```

### Task Reservation Update

This authorizes updates to a Task's reservation status.

```
capability.allow_task_reservation_updates()
```

### Generate a Token

```
token = capability.generate_token()
```

By default, this token will expire in one hour. If you'd like to change the token expiration, `generate_token()` takes an optional `ttl` argument.

```
token = capability.generate_token(ttl=600)
```

This token will now expire in 10 minutes. If you haven't guessed already, `ttl` is expressed in seconds.

## 3.3 TwiML

Generates valid TwiML for controlling and manipulating phone calls.

### 3.3.1 TwiML Creation

TwiML creation begins with the *Response* verb. Each successive verb is created by calling various methods on the response, such as `say()` or `play()`. These methods return the verbs they create to ease creation of nested TwiML. To finish, call the `toxml()` method on the *Response*, which returns raw TwiML.

To get started with TwiML, read the [TwiML documentation](#) on our website.

The following code block creates a simple TwiML Response.

```
from twilio import twiml

r = twiml.Response()
r.say("Hello")
print str(r)
```

The output is perfectly formatted XML:

```
<?xml version="1.0" encoding="utf-8"?>
<Response><Say>Hello</Say></Response>
```

The verb methods (outlined in the *complete reference*) take the body (only text) of the verb as the first argument. All attributes are keyword arguments.

```
from twilio import twiml

r = twiml.Response()
r.play("https://api.twilio.com/cowbell.mp3", loop=5)
print str(r)
```

```
<?xml version="1.0" encoding="utf-8"?>
<Response>
  <Play loop="3">https://api.twilio.com/cowbell.mp3</Play>
</Response>
```

The Message and Sms verbs are slightly special: because from is a Python keyword, use the sender parameter to specify the number the message should come from:

```
from twilio import twiml

r = twiml.Response()
m = r.message("Hello MMS Monkey!", sender="+14155551234")
print str(r)
```

```
<?xml version="1.0" encoding="utf-8"?>
<Response>
  <Message from="+14155551234">Hello MMS Monkey!</Message>
</Response>
```

Python 2.6+ added the with statement for context management. Using with, the module can *almost* emulate Ruby blocks.

```
from twilio import twiml

r = twiml.Response()
r.say("hello")
with r.gather(finishOnKey=4) as g:
    g.say("world")
print str(r)
```

which returns the following

```
<?xml version="1.0" encoding="utf-8"?>
<Response>
  <Say>Hello</Say>
  <Gather finishOnKey="4"><Say>World</Say></Gather>
</Response>
```

If you don't want the XML declaration in your output, use the toxml () method

```
from twilio import twiml

r = twiml.Response()
r.say("hello")
with r.gather(finishOnKey=4) as g:
    g.say("world")
print r.toxml(xml_declaration=False)
```

```
<Response>
  <Say>Hello</Say>
  <Gather finishOnKey="4"><Say>World</Say></Gather>
</Response>
```

## 3.4 Utilities

Small functions useful for validating requests are coming from Twilio

### 3.4.1 Validate Incoming Requests

Twilio requires that your TwiML-serving web server be open to the public. This is necessary so that Twilio can retrieve TwiML from urls and POST data back to your server.

However, there may be people out there trying to spoof the Twilio service. Luckily, there's an easy way to validate that incoming requests are from Twilio and Twilio alone.

An in-depth guide to our security features can be [found in our online documentation](#).

Before you can validate requests, you'll need four pieces of information:

- your Twilio Auth Token (found in your [Dashboard](#))
- the POST data for the request
- the requested URL
- the X-Twilio-Signature header value

Obtaining the last three pieces of information depends on the framework you are using to process requests. The below example assumes that you have the POST data as a dictionary and the url and X-Twilio-Signature as strings.

The below example will print out a confirmation message if the request is actually from Twilio.

```
from twilio.util import RequestValidator

AUTH_TOKEN = 'YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY'

validator = RequestValidator(AUTH_TOKEN)

# the callback URL you provided to Twilio
url = "http://www.example.com/my/callback/url.xml"

# the POST variables attached to the request (eg "From", "To")
post_vars = {}

# X-Twilio-Signature header value
signature = "HpS7PBa1Agvt4OtO+wZp75IuQa0=" # will look something like that

if validator.validate(url, post_vars, signature):
    print "Confirmed to have come from Twilio."
else:
    print "NOT VALID. It might have been spoofed!"
```

## Trailing Slashes

If your URL uses an “index” page, such as `index.php` or `index.html` to handle the request, such as: `https://mycompany.com/twilio` where the real page is served from `https://mycompany.com/twilio/index.php`, then Apache or PHP may rewrite that URL a little bit so it’s got a trailing slash, such as `https://mycompany.com/twilio/` for example.

Using the code above, or similar code in another language, you could end up with an incorrect hash because Twilio built the hash using `https://mycompany.com/twilio` and you may have built the hash using `https://mycompany.com/twilio/`. More information can be found in our documentation on validating requests.

## 3.4.2 Generate Capability Tokens

`Twilio Client` allows you to make and receive connections in the browser. You can place a call to a phone on the PSTN network, all without leaving your browser. See the [Twilio Client Quickstart](#) to get up and running with Twilio Client.

Capability tokens are used by `Twilio Client` to provide connection security and authorization. The [Capability Token documentation](#) explains in depth the purpose and features of these tokens.

`TwilioCapability` is responsible for the creation of these capability tokens. You’ll need your Twilio AccountSid and AuthToken.

```
from twilio.util import TwilioCapability

# Find these values at twilio.com/user/account
account_sid = "AC123123"
auth_token = "secret"

capability = TwilioCapability(account_sid, auth_token)
```

### Allow Incoming Connections

Before a device running `Twilio Client` can receive incoming connections, the instance must first register a name (such as “Alice” or “Bob”). The `allow_client_incoming()` method adds the client name to the capability token.

```
capability.allow_client_incoming("Alice")
```

### Allow Outgoing Connections

To make an outgoing connection from a `Twilio Client` device, you’ll need to choose a `Twilio Application` to handle TwiML URLs. A Twilio Application is a collection of URLs responsible for outputting valid TwiML to control phone calls and SMS.

```
# Twilio Application Sid
application_sid = "APabe7650f654fc34655fc81ae71caa3ff"
capability.allow_client_outgoing(application_sid)
```

### Generate a Token

```
token = capability.generate()
```

By default, this token will expire in one hour. If you’d like to change the token expiration, `generate()` takes an optional `expires` argument.

```
token = capability.generate(expires=600)
```

This token will now expire in 10 minutes. If you haven't guessed already, `expires` is expressed in seconds.

## CHAPTER 4

---

### Upgrade Plan

---

*twilio-python* 3.0 introduced backwards-incompatible changes to the API. See the `/upgrade-guide` for step-by-step instructions for migrating to 3.0. In many cases, the same methods are still offered, just in different locations.



A complete guide to all public APIs found in *twilio-python*. Auto-generated, so only use when you really need to dive deep into the library.

## 5.1 API Documentation

A complete API reference to the *twilio* module.

### 5.1.1 *twilio*

**class** `twilio.TwilioRestException` (*status, uri, msg='', code=None, method='GET'*)

A generic 400 or 500 level exception from the Twilio API

#### Parameters

- **status** (*int*) – the HTTP status that was returned for the exception
- **uri** (*str*) – The URI that caused the exception
- **msg** (*str*) – A human-readable message for the error
- **method** (*str*) – The HTTP method used to make the request
- **code** (*int | None*) – A Twilio-specific error code for the error. This is not available for all errors.

### 5.1.2 *twilio.rest*

**class** `twilio.rest.TwilioRestClient` (*account=None, token=None, base='https://api.twilio.com', version='2010-04-01', timeout=<Unset Timeout Value>, request\_account=None*)

A client for accessing the Twilio REST API

**Parameters**

- **account** (*str*) – Your Account SID from [your dashboard](#)
- **token** (*str*) – Your Auth Token from [your dashboard](#)
- **timeout** (*float*) – The socket and read timeout for requests to Twilio

**dependent\_phone\_numbers** (*address\_sid*)Return a *DependentPhoneNumbers* instance for the *Address* with the given *address\_sid***feedback** (*call\_sid*)Return a *CallFeedback* instance for the *Call* with the given *call\_sid***members** (*queue\_sid*)Return a *Members* instance for the *Queue* with the given *queue\_sid***participants** (*conference\_sid*)Return a *Participants* instance for the *Conference* with given *conference\_sid***request** (*path, method=None, vars=None*)

sends a request and gets a response from the Twilio REST API

Deprecated since version 3.0.

**Parameters**

- **path** – the URL (relative to the endpoint URL, after the /v1
- **url** – the HTTP method to use, defaults to POST
- **vars** – for POST or PUT, a dict of data to send

**Returns** Twilio response in XML or raises an exception on error**Raises** a *ValueError* if the path is invalid**Raises** a *NotImplementedError* if the method is unknown

This method is only included for backwards compatability reasons. It will be removed in a future version

```
class twilio.rest.TwilioTaskRouterClient (account=None, token=None,
                                         base='https://taskrouter.twilio.com', version='v1',
                                         timeout=<Unset Timeout Value>,
                                         request_account=None)
```

A client for accessing the Twilio TaskRouter API

**Parameters**

- **account** (*str*) – Your Account SID from [your dashboard](#)
- **token** (*str*) – Your Auth Token from [your dashboard](#)
- **timeout** (*float*) – The socket and read timeout for requests to Twilio

**activities** (*workspace\_sid*)Return a *Activities* instance for the *Activity* with the given *workspace\_sid***events** (*workspace\_sid*)Return a *Events* instance for the *Event* with the given *workspace\_sid***request** (*path, method=None, vars=None*)

sends a request and gets a response from the Twilio REST API

Deprecated since version 3.0.

**Parameters**

- **path** – the URL (relative to the endpoint URL, after the /v1
- **url** – the HTTP method to use, defaults to POST
- **vars** – for POST or PUT, a dict of data to send

**Returns** Twilio response in XML or raises an exception on error

**Raises** a `ValueError` if the path is invalid

**Raises** a `NotImplementedError` if the method is unknown

This method is only included for backwards compatability reasons. It will be removed in a future version

**reservations** (*workspace\_sid, task\_sid*)

Return a `Reservations` instance for the `Reservation` with the given `workspace_sid` and `task_sid`

**task\_queues** (*workspace\_sid*)

Return a `TaskQueues` instance for the `TaskQueue` with the given `workspace_sid`

**tasks** (*workspace\_sid*)

Return a `Tasks` instance for the `Task` with the given `workspace_sid`

**worker\_reservations** (*workspace\_sid, worker\_sid*)

Return a `Reservations` instance for the `Reservation` with the given `workspace_sid` and `worker_sid`

**workers** (*workspace\_sid*)

Return a `Workers` instance for the `Worker` with the given `workspace_sid`

**workflows** (*workspace\_sid*)

Return a `Workflows` instance for the `Workflow` with the given `workspace_sid`

### 5.1.3 twilio.rest.resources

`class twilio.rest.resources.ListResource(*args, **kwargs)`

**get** (*sid*)

Get an instance resource by its sid

Usage:

```
message = client.messages.get("SM1234")
print message.body
```

**Return type** `InstanceResource`

**Raises** a `TwilioRestException` if a resource with that sid does not exist, or the request fails

**iter** (*\*\*kwargs*)

Return all instance resources using an iterator

This will fetch a page of resources from the API and yield them in turn. When the page is exhausted, this will make a request to the API to retrieve the next page. Hence you may notice a pattern - the library will loop through 50 objects very quickly, but there will be a delay retrieving the 51st as the library must make another request to the API for resources.

Example usage:

```
for message in client.messages:
    print message.sid
```

```
class twilio.rest.resources.InstanceResource (parent, sid)
    The object representation of an instance response from the Twilio API
```

**Parameters**

- **parent** (*ListResource*) – The parent list class for this instance resource. For example, the parent for a *Call* would be a *Calls* object.
- **sid** (*str*) – The 34-character unique identifier for this instance

**Accounts**

```
class twilio.rest.resources.Accounts (*args, **kwargs)
    A list of Account resources
```

**activate** (*sid*)

Reactivate an account, Alias to update

**close** (*sid*)

Permenently deactivate an account, Alias to update

**create** (*\*\*kwargs*)

Returns a newly created sub account resource.

**Parameters** **friendly\_name** – Update the description of this account.**list** (*\*\*kwargs*)Returns a page of *Account* resources as a list. For paging informtion see *ListResource***Parameters**

- **friendly\_name** (*date*) – Only list accounts with this friendly name
- **status** (*date*) – Only list accounts with this status

**suspend** (*sid*)

Temporarily suspend an account, Alias to update

**update** (*sid, \*\*kwargs*)**Parameters**

- **sid** – Account identifier
- **friendly\_name** – Update the description of this account.
- **status** – Alter the status of this account

Use CLOSED to irreversibly close this account, SUSPENDED to temporarily suspend it, or ACTIVE to reactivate it.

```
class twilio.rest.resources.Account (parent, sid)
    An Account resource
```

**sid**

A 34 character string that uniquely identifies this account.

**date\_created**

The date that this account was created, in GMT in RFC 2822 format

**date\_updated**

The date that this account was last updated, in GMT in RFC 2822 format.

**friendly\_name**

A human readable description of this account, up to 64 characters long. By default the FriendlyName is your email address.

**status**

The status of this account. Usually active, but can be suspended if you've been bad, or closed if you've been horrible.

**auth\_token**

The authorization token for this account. This token should be kept a secret, so no sharing.

**ACTIVE = 'active'**

A constant representing an active account

**CLOSED = 'closed'**

A constant representing a closed account

**SUSPENDED = 'suspended'**

A constant representing a suspended account

**activate ()**

Reactivate this account

**close ()**

Permanently deactivate this account

**suspend ()**

Temporarily suspend this account

**update (\*\*kwargs)**

Update the status of an account.

**Parameters**

- **friendly\_name** – Update the description of this account.
- **status** – Alter the status of this account

Use *CLOSED* to irreversibly close this account, *SUSPENDED* to temporarily suspend it, or *ACTIVE* to reactivate it.

## Addresses

```
class twilio.rest.resources.Addresses (*args, **kwargs)
```

**create** (*customer\_name*, *street*, *city*, *region*, *postal\_code*, *iso\_country*, *friendly\_name=None*)

Create an *Address*.

**Parameters**

- **customer\_name** (*str*) – Your customer's name
- **street** (*str*) – The number and street of your address
- **city** (*str*) – The city of you or your customer's address
- **region** (*str*) – The region or state
- **postal\_code** (*str*) – The postal code of your address
- **iso\_country** (*str*) – The ISO 3166-1 alpha-2 (two-character) country code, e.g. 'US' or 'AU'

- **friendly\_name** (*str*) – A user-defined name for this address (optional; up to 64 characters)

**delete** (*sid*)

Delete an *Address*.

**Parameters** *sid* (*str*) – The sid of the Address to delete.

**update** (*sid*, *\*\*kwargs*)

Update an *Address* with the given parameters.

Parameters are described above in *create()*, with the exception that *iso\_country* cannot be updated on an existing Address (create a new one instead).

**class** `twilio.rest.resources.Address` (*parent*, *sid*)

An Address resource. See <https://www.twilio.com/docs/api/rest/address>

**friendly\_name**

A human-readable description of this address. Maximum 64 characters.

**customer\_name**

Your or your customer's name or business name.

**street**

The number and street address where you or your customer are located.

**city**

The city in which you or your customer are located.

**region**

The state or region in which you or your customer are located.

**postal\_code**

The postal code in which you or your customer are located.

**iso\_country**

The ISO country code of your or your customer's address.

**update** (*\*\*kwargs*)

Update this phone number instance.

Parameters are as described in *Addresses.create()*, with the exception that *iso\_country* cannot be updated on an existing Address (create a new one instead).

## DependentPhoneNumbers

**class** `twilio.rest.resources.DependentPhoneNumbers` (*\*args*, *\*\*kwargs*)

A list of purchased phone numbers that depend on a particular *Address*.

Included numbers are those that require an address on file and have no other candidate addresses of the appropriate type (local, foreign) associated with the owning account.

If this list has entries for a given Address, that address cannot be deleted until the numbers are released from your account or alternate addresses are provided to satisfy the requirements.

This resource is read-only and cannot be updated or deleted, but will reflect the current state of the owning account's addresses (i.e. if you add another address that satisfies a number's requirements, it will not appear in subsequent requests to this list resource).

**class** `twilio.rest.resources.DependentPhoneNumber` (*parent*, *sid*)

A purchased phone number that depends on a particular *Address*.

Attributes are the same as *PhoneNumber*.

DependentPhoneNumbers are a read-only resource and cannot be updated or deleted.

## Applications

```
class twilio.rest.resources.Applications(*args, **kwargs)
```

```
create(**kwargs)
```

Create an *Application* with any of these optional parameters.

### Parameters

- **friendly\_name** – A human readable description of the application, with maximum length 64 characters.
- **api\_version** – Requests to this application’s URLs will start a new TwiML session with this API version. Either 2010-04-01 or 2008-08-01.
- **voice\_url** – The URL that Twilio should request when somebody dials a phone number assigned to this application.
- **voice\_method** – The HTTP method that should be used to request the VoiceUrl. Either GET or POST.
- **voice\_fallback\_url** – A URL that Twilio will request if an error occurs requesting or executing the TwiML defined by VoiceUrl.
- **voice\_fallback\_method** – The HTTP method that should be used to request the VoiceFallbackUrl. Either GET or POST.
- **status\_callback** – The URL that Twilio will request to pass status parameters (such as call ended) to your application.
- **status\_callback\_method** – The HTTP method Twilio will use to make requests to the StatusCallback URL. Either GET or POST.
- **voice\_caller\_id\_lookup** – Do a lookup of a caller’s name from the CNAM database and post it to your app. Either true or false.
- **sms\_url** – The URL that Twilio should request when somebody sends an SMS to a phone number assigned to this application.
- **sms\_method** – The HTTP method that should be used to request the SmsUrl. Either GET or POST.
- **sms\_fallback\_url** – A URL that Twilio will request if an error occurs requesting or executing the TwiML defined by SmsUrl.
- **sms\_fallback\_method** – The HTTP method that should be used to request the SmsFallbackUrl. Either GET or POST.
- **sms\_status\_callback** – Twilio will make a POST request to this URL to pass status parameters (such as sent or failed) to your application if you specify this application’s Sid as the ApplicationSid on an outgoing SMS request.

```
delete(sid)
```

Delete an *Application*

```
list(**kwargs)
```

Returns a page of *Application* resources as a list. For paging information see *ListResource*

**Parameters** **friendly\_name** (*date*) – List applications with this friendly name

**update** (*sid*, *\*\*kwargs*)

Update an *Application* with the given parameters.

All the parameters are describe above in *create()*

**class** `twilio.rest.resources.Application` (*parent*, *sid*)

An application resource

**sid**

A 34 character string that uniquely identifies this application.

**date\_created**

The date that this application was created, in GMT in RFC 2822 format

**date\_updated**

The date that this application was last updated, in GMT in RFC 2822 format.

**friendly\_name**

A human readable description of this application, up to 64 characters long. By default the FriendlyName is your email address.

**status**

The status of this account. Usually active, but can be suspended if you've been bad, or closed if you've been horrible.

**api\_version**

Requests to this application will start a new TwiML session with this API version.

**voice\_url**

URL Twilio will request when a phone number assigned to this application receives a call.

**voice\_method**

The HTTP method Twilio will use when requesting the above Url. Either GET or POST.

**voice\_fallback\_url**

The URL that Twilio will request if an error occurs retrieving or executing the TwiML requested by Url.

**voice\_fallback\_method**

The HTTP method Twilio will use when requesting the VoiceFallbackUrl. Either GET or POST.

**status\_callback**

The URL that Twilio will request to pass status parameters (such as call ended) to your application.

**status\_callback\_method**

The HTTP method Twilio will use to make requests to the StatusCallback URL. Either GET or POST.

**voice\_caller\_id\_lookup**

Look up the caller's caller-ID name from the CNAM database (additional charges apply). Either true or false.

**sms\_url**

The URL Twilio will request when a phone number assigned to this application receives an incoming SMS message.

**sms\_method**

The HTTP method Twilio will use when making requests to the SmsUrl. Either GET or POST.

**sms\_fallback\_url**

The URL that Twilio will request if an error occurs retrieving or executing the TwiML from SmsUrl.

**sms\_fallback\_method**

The HTTP method Twilio will use when requesting the above URL. Either GET or POST.

**sms\_status\_callback**

Twilio will make a POST request to this URL to pass status parameters (such as sent or failed) to your application if you specify this application's Sid as the ApplicationSid on an outgoing SMS request.

**uri**

The URI for this resource, relative to <https://api.twilio.com>

**delete()**

Delete this application

**update(\*\*kwargs)**

Update this application

## Calls

**class** `twilio.rest.resources.Calls(*args, **kwargs)`

A list of Call resources

**cancel(sid)**

If this call is queued or ringing, cancel the call. Will not affect in-progress calls.

**Parameters** `sid` – A Call Sid for a specific call

**Returns** Updated `Call` resource

**create** (`to`, `from_`, `url`, `status_method=None`, `status_events=None`, `**kwargs`)

Make a phone call to a number.

**Parameters**

- **to** (`str`) – The phone number to call
- **from\_** (`str`) – The caller ID (must be a verified Twilio number)
- **url** (`str`) – The URL to read TwiML from when the call connects
- **method** (`None` (defaults to 'POST'), 'GET', or 'POST') – The HTTP method Twilio should use to request the url
- **fallback\_url** (`str`) – A URL that Twilio will request if an error occurs requesting or executing the TwiML at url
- **fallback\_method** (`None` (will make 'POST' request), 'GET', or 'POST') – The HTTP method that Twilio should use to request the fallback\_url
- **status\_callback** (`str`) – A URL that Twilio will request when the call ends to notify your app.
- **status\_method** (`str`) – The HTTP method Twilio should use when requesting the above URL.
- **status\_events** (`list`) – A list of call progress events Twilio should send status callback requests on. One or more of: 'initiated', 'ringing', 'answered', 'completed'. Defaults to ['completed'] if not provided. 'completed' events are sent free of charge; see [twilio.com](https://www.twilio.com) for current pricing on others.
- **if\_machine** (`None`, 'Continue', or 'Hangup') – Tell Twilio to try and determine if a machine (like voicemail) or a human has answered the call. See more in our [answering machine documentation](#).
- **send\_digits** (`None` or any combination of (0-9), '#', '\*' or 'w' (to insert a half second pause)) – A string of keys to dial after connecting to the number.

- **timeout** (*int*) – The integer number of seconds that Twilio should allow the phone to ring before assuming there is no answer.
- **application\_sid** (*str*) – The 34 character sid of the application Twilio should use to handle this phone call. Should not be used in conjunction with the url parameter.

**Returns** A *Call* object

**create\_instance** (*body*)

Create an InstanceResource via a POST to the List Resource

**Parameters** **body** (*dict*) – Dictionary of POST data

**delete** (*sid*)

Delete the given Call record from Twilio.

**delete\_instance** (*sid*)

Delete an InstanceResource via DELETE

body: string – HTTP Body for the quest

**feedback** (*sid, quality\_score, issue=None*)

Create feedback for the given call.

**Parameters**

- **sid** – A Call Sid for a specific call
- **quality\_score** – The quality of the call
- **issue** – A list of issues experienced during the call

**Returns** A *CallFeedback* object

**get** (*sid*)

Get an instance resource by its sid

Usage:

```
message = client.messages.get("SM1234")
print message.body
```

**Return type** *InstanceResource*

**Raises** a *TwilioRestException* if a resource with that sid does not exist, or the request fails

**get\_instance** (*sid*)

Request the specified instance resource

**get\_instances** (*params*)

Query the list resource for a list of InstanceResources.

Raises a *TwilioRestException* if requesting a page of results that does not exist.

**Parameters**

- **params** (*dict*) – List of URL parameters to be included in request
- **page** (*int*) – The page of results to retrieve (most recent at 0)
- **page\_size** (*int*) – The number of results to be returned.

**Returns** – the list of resources

**hangup** (*sid*)

If this call is currently active, hang up the call. If this call is scheduled to be made, remove the call from the queue.

**Parameters** *sid* – A Call Sid for a specific call

**Returns** Updated *Call* resource

**iter** (*\*args, \*\*kwargs*)

Returns an iterator of *Call* resources.

**Parameters**

- **after** (*date*) – Only list calls started after this datetime
- **before** (*date*) – Only list calls started before this datetime

**list** (*\*args, \*\*kwargs*)

Returns a page of *Call* resources as a list. For paging information see *ListResource*

**Parameters**

- **after** (*date*) – Only list calls started after this datetime
- **before** (*date*) – Only list calls started before this datetime

**request** (*method, uri, \*\*kwargs*)

Send an HTTP request to the resource.

**Raises** a *TwilioRestException*

**route** (*sid, url, method='POST'*)

Route the specified *Call* to another url.

**Parameters**

- **sid** – A Call Sid for a specific call
- **url** – A valid URL that returns TwiML.
- **method** – The HTTP method Twilio uses when requesting the URL.

**Returns** Updated *Call* resource

**update\_instance** (*sid, body*)

Update an InstanceResource via a POST

*sid*: string – String identifier for the list resource *body*: dictionary – Dict of items to POST

**class** `twilio.rest.resources.Call` (*parent, sid*)

A call resource

**sid**

A 34 character string that uniquely identifies this resource.

**parent\_call\_sid**

A 34 character string that uniquely identifies the call that created this leg.

**date\_created**

The date that this resource was created, given as GMT in RFC 2822 format.

**date\_updated**

The date that this resource was last updated, given as GMT in RFC 2822 format.

**account\_sid**

The unique id of the Account responsible for creating this call.

**to**

The phone number that received this call. e.g., +16175551212 (E.164 format)

**from\_**

The phone number that made this call. e.g., +16175551212 (E.164 format)

**phone\_number\_sid**

If the call was inbound, this is the Sid of the IncomingPhoneNumber that received the call. If the call was outbound, it is the Sid of the OutgoingCallerId from which the call was placed.

**status**

A string representing the status of the call. May be QUEUED, RINGING, IN-PROGRESS, COMPLETED, FAILED, BUSY or NO\_ANSWER.

**start\_time**

The start time of the call, given as GMT in RFC 2822 format. Empty if the call has not yet been dialed.

**end\_time**

The end time of the call, given as GMT in RFC 2822 format. Empty if the call did not complete successfully.

**duration**

The length of the call in seconds. This value is empty for busy, failed, unanswered or ongoing calls.

**price**

The charge for this call in USD. Populated after the call is completed. May not be immediately available.

**direction**

A string describing the direction of the call. inbound for inbound calls, outbound-api for calls initiated via the REST API or outbound-dial for calls initiated by a <Dial> verb.

**answered\_by**

If this call was initiated with answering machine detection, either human or machine. Empty otherwise.

**forwarded\_from**

If this call was an incoming call forwarded from another number, the forwarding phone number (depends on carrier supporting forwarding). Empty otherwise.

**caller\_name**

If this call was an incoming call from a phone number with Caller ID Lookup enabled, the caller's name. Empty otherwise.

**cancel ()**

If the called is queued or ringing, cancel the calls. Will not affect in progress calls

**delete ()**

Delete the specified *Call* record from Twilio.

**hangup ()**

If this call is currently active, hang up the call. If this call is scheduled to be made, remove the call from the queue

**route (\*\*kwargs)**

Route the specified *Call* to another url.

**Parameters**

- **url** – A valid URL that returns TwiML.
- **method** – HTTP method Twilio uses when requesting the above URL.

## Caller Ids

**class** `twilio.rest.resources.CallerIds` (*\*args*, *\*\*kwargs*)

A list of *CallerId* resources

**delete** (*sid*)

Deletes a specific *CallerId* from the account.

**list** (*\*\*kwargs*)

### Parameters

- **phone\_number** – Show caller ids with this phone number.
- **friendly\_name** – Show caller ids with this friendly name.

**update** (*sid*, *\*\*kwargs*)

Update a specific *CallerId*

**validate** (*phone\_number*, *\*\*kwargs*)

Begin the validation process for the given number.

Returns a dictionary with the following keys

**account\_sid**: The unique id of the Account to which the Validation Request belongs.

**phone\_number**: The incoming phone number being validated, formatted with a '+' and country code e.g., +16175551212

**friendly\_name**: The friendly name you provided, if any.

**validation\_code**: The 6 digit validation code that must be entered via the phone to validate this phone number for Caller ID.

### Parameters

- **phone\_number** – The phone number to call and validate
- **friendly\_name** – A description for the new caller ID
- **call\_delay** – Number of seconds to delay the validation call.
- **extension** – Digits to dial after connecting the validation call.

**Returns** A response dictionary

**class** `twilio.rest.resources.CallerId` (*parent*, *sid*)

**sid**

A 34 character string that uniquely identifies this resource.

**date\_created**

The date that this resource was created, given in RFC 2822 format.

**date\_updated**

The date that this resource was last updated, given in RFC 2822 format.

**friendly\_name**

A human readable descriptive text for this resource, up to 64 characters long. By default, the FriendlyName is a nicely formatted version of the phone number.

**account\_sid**

The unique id of the Account responsible for this Caller Id.

**phone\_number**

The incoming phone number. Formatted with a '+' and country code e.g., +16175551212 (E.164 format).

**uri**

The URI for this resource, relative to <https://api.twilio.com>.

**delete()**

Deletes this caller ID from the account.

**update(\*\*kwargs)**

Update the CallerId

## Conferences

```
class twilio.rest.resources.Conferences(*args, **kwargs)
```

**list(\*args, \*\*kwargs)**

Return a list of *Conference* resources

**Parameters**

- **status** – Show conferences with this status
- **friendly\_name** – Show conferences with this exact friendly\_name
- **updated\_after** (*date*) – List conferences updated after this date
- **updated\_before** (*date*) – List conferences updated before this date
- **created\_after** (*date*) – List conferences created after this date
- **created\_before** (*date*) – List conferences created before this date

```
class twilio.rest.resources.Conference(parent, sid)
```

**sid**

A 34 character string that uniquely identifies this conference.

**friendly\_name**

A user provided string that identifies this conference room.

**status**

A string representing the status of the conference. May be init, in-progress, or completed.

**date\_created**

The date that this conference was created, given as GMT in RFC 2822 format.

**date\_updated**

The date that this conference was last updated, given as GMT in RFC 2822 format.

**account\_sid**

The unique id of the Account responsible for creating this conference.

**uri**

The URI for this resource, relative to <https://api.twilio.com>.

**participants**

The *Participants* resource, listing people currently in this conference

## Connect Apps

**class** `twilio.rest.resources.ConnectApps` (\*args, \*\*kwargs)  
 A list of Connect App resources

**list** (\*\*kwargs)  
 Returns a page of *ConnectApp* resources as a list. For paging information see *ListResource*

**class** `twilio.rest.resources.ConnectApp` (parent, sid)  
 An authorized connect app

**sid**  
 A 34 character string that uniquely identifies this resource.

**date\_created**  
 The date that this resource was created, given as GMT in RFC 2822 format.

**date\_updated**  
 The date that this resource was last updated, given as GMT in RFC 2822 format.

**account\_sid**  
 The unique id of the Account responsible for creating this call.

**permissions**  
 The list of permissions that your ConnectApp requests.

**friendly\_name**  
 A human readable name for this resource.

**description**  
 A more detailed human readable description of this resource.

**company\_name**  
 The company name set for this Connect App.

**homepage\_url**  
 The public URL where users can obtain more information about this Connect App.

**authorize\_redirect\_url**  
 The URL the user's browser will redirect to after Twilio authenticates the user and obtains authorization for this Connect App.

**deauthorize\_callback\_url**  
 The URL to which Twilio will send a request when a user de-authorizes this Connect App.

**deauthorize\_callback\_method**  
 The HTTP method to be used when making a request to the deauthorize callback url.

**uri**  
 The URI for this resource, relative to <https://api.twilio.com>.

## Notifications

**class** `twilio.rest.resources.Notifications` (\*args, \*\*kwargs)

**delete** (sid)  
 Delete a given Notification

**list** (\*args, \*\*kwargs)  
 Returns a page of *Notification* resources as a list. For paging information see *ListResource*.

**NOTE:** Due to the potentially voluminous amount of data in a notification, the full HTTP request and response data is only returned in the Notification instance resource representation.

#### Parameters

- **after** (*date*) – Only list notifications logged after this datetime
- **before** (*date*) – Only list notifications logged before this datetime
- **log\_level** – If 1, only shows errors. If 0, only show warnings

```
class twilio.rest.resources.Notification(parent, sid)
```

#### **sid**

A 34 character string that uniquely identifies this resource.

#### **date\_created**

The date that this resource was created, given in RFC 2822 format.

#### **date\_updated**

The date that this resource was last updated, given in RFC 2822 format.

#### **account\_sid**

The unique id of the Account responsible for this notification.

#### **call\_sid**

CallSid is the unique id of the call during which the notification was generated. Empty if the notification was generated by the REST API without regard to a specific phone call.

#### **api\_version**

The version of the Twilio in use when this notification was generated.

#### **log**

An integer log level corresponding to the type of notification: 0 is ERROR, 1 is WARNING.

#### **error\_code**

A unique error code for the error condition. You can lookup errors, with possible causes and solutions, in our Error Dictionary.

#### **more\_info**

A URL for more information about the error condition. The URL is a page in our Error Dictionary.

#### **message\_text**

The text of the notification.

#### **message\_date**

The date the notification was actually generated, given in RFC 2822 format. Due to buffering, this may be slightly different than the DateCreated date.

#### **request\_url**

The URL of the resource that generated the notification. If the notification was generated during a phone call: This is the URL of the resource on YOUR SERVER that caused the notification. If the notification was generated by your use of the REST API: This is the URL of the REST resource you were attempting to request on Twilio's servers.

#### **request\_method**

The HTTP method in use for the request that generated the notification. If the notification was generated during a phone call: The HTTP Method use to request the resource on your server. If the notification was generated by your use of the REST API: This is the HTTP method used in your request to the REST resource on Twilio's servers.

**request\_variables**

The Twilio-generated HTTP GET or POST variables sent to your server. Alternatively, if the notification was generated by the REST API, this field will include any HTTP POST or PUT variables you sent to the REST API.

**response\_headers**

The HTTP headers returned by your server.

**response\_body**

The HTTP body returned by your server.

**uri**

The URI for this resource, relative to <https://api.twilio.com>

**delete()**

Delete this notification

**Participants**

```
class twilio.rest.resources.Participants(*args, **kwargs)
```

**delete**(*call\_sid*)

Remove the participant from the given conference

**kick**(*call\_sid*)

Remove the participant from the given conference

**list**(\*\**kwargs*)

Returns a list of *Participant* resources in the given conference

**Parameters**

- **conference\_sid** – Conference this participant is part of
- **muted** (*boolean*) – If True, only show participants who are muted

**mute**(*call\_sid*)

Mute the given participant

**unmute**(*call\_sid*)

Unmute the given participant

**update**(*sid*, \*\**kwargs*)**Parameters**

- **sid** – Participant identifier
- **muted** (*boolean*) – If true, mute this participant

```
class twilio.rest.resources.Participant(parent, sid)
```

**call\_sid**

A 34 character string that uniquely identifies the call that is connected to this conference

**conference\_sid**

A 34 character string that identifies the conference this participant is in

**date\_created**

The date that this resource was created, given in RFC 2822 format.

**date\_updated**

The date that this resource was last updated, given in RFC 2822 format.

**account\_sid**

The unique id of the Account that created this conference

**muted**

true if this participant is currently muted. false otherwise.

**start\_conference\_on\_enter**

Was the startConferenceOnEnter attribute set on this participant (true or false)?

**end\_conference\_on\_exit**

Was the endConferenceOnExit attribute set on this participant (true or false)?

**uri**

The URI for this resource, relative to <https://api.twilio.com>.

**kick()**

Remove the participant from the given conference

**mute()**

Mute the participant

**unmute()**

Unmute the participant

## Phone Numbers

```
class twilio.rest.resources.PhoneNumbers (base_uri, auth, timeout=<Unset Timeout Value>)
```

**delete** (*sid*)

Release this phone number from your account. Twilio will no longer answer calls to this number, and you will stop being billed the monthly phone number fees. The phone number will eventually be recycled and potentially given to another customer, so use with care. If you make a mistake, contact us... we may be able to give you the number back.

```
list (type=None, **kwargs)
```

**Parameters**

- **phone\_number** – Show phone numbers that match this pattern.
- **friendly\_name** – Show phone numbers with this friendly name
- **type** – Filter numbers by type. Available types are ‘local’, ‘mobile’, or ‘tollfree’

You can specify partial numbers and use ‘\*’ as a wildcard.

```
purchase (status_callback_url=None, **kwargs)
```

Attempt to purchase the specified number. The only required parameters are **either** phone\_number or area\_code

**Returns** Returns a *PhoneNumber* instance on success, *False* on failure

**Raises** A *TypeError* if neither phone\_number or area\_code

is specified.

```
search (**kwargs)
```

**Parameters**

- **type** – The type of phone number to search for.
- **country** (*str*) – Only show numbers for this country (iso2)
- **region** (*str*) – When searching the US, show numbers in this state
- **postal\_code** (*str*) – Only show numbers in this area code
- **rate\_center** (*str*) – US only.
- **near\_lat\_long** (*str*) – Find close numbers within Distance miles. Should be string of format “{lat},{long}”
- **distance** (*integer*) – Search radius for a Near- query in miles.
- **beta** (*boolean*) – Whether to include numbers new to the Twilio platform.

**transfer** (*sid, account\_sid*)

Transfer the phone number with *sid* from the current account to another identified by *account\_sid*

**update** (*sid, \*\*kwargs*)

Update this phone number instance

**class** `twilio.rest.resources.PhoneNumber` (*parent, sid*)

An IncomingPhoneNumber object

**sid**

A 34 character string that uniquely identifies this resource.

**date\_created**

The date that this resource was created, given as GMT RFC 2822 format.

**date\_updated**

The date that this resource was last updated, in GMT RFC 2822 format.

**friendly\_name**

A human readable descriptive text for this resource, up to 64 characters long. By default, the FriendlyName is a nicely formatted version of the phone number.

**account\_sid**

The unique id of the Account responsible for this phone number.

**phone\_number**

The incoming phone number. e.g., +16175551212 (E.164 format)

**api\_version**

Calls to this phone number will start a new TwiML session with this API version.

**voice\_caller\_id\_lookup**

Look up the caller’s caller-ID name from the CNAM database (additional charges apply). Either true or false.

**voice\_url**

The URL Twilio will request when this phone number receives a call.

**voice\_method**

The HTTP method Twilio will use when requesting the above Url. Either GET or POST.

**voice\_fallback\_url**

The URL that Twilio will request if an error occurs retrieving or executing the TwiML requested by Url.

**voice\_fallback\_method**

The HTTP method Twilio will use when requesting the VoiceFallbackUrl. Either GET or POST.

**status\_callback**

The URL that Twilio will request to pass status parameters (such as call ended) to your application.

**status\_callback\_method**

The HTTP method Twilio will use to make requests to the StatusCallback URL. Either GET or POST.

**sms\_url**

The URL Twilio will request when receiving an incoming SMS message to this number.

**sms\_method**

The HTTP method Twilio will use when making requests to the SmsUrl. Either GET or POST.

**sms\_fallback\_url**

The URL that Twilio will request if an error occurs retrieving or executing the TwiML from SmsUrl.

**sms\_fallback\_method**

The HTTP method Twilio will use when requesting the above URL. Either GET or POST.

**uri**

The URI for this resource, relative to <https://api.twilio.com>.

**beta**

(boolean) Phone numbers new to the Twilio platform are marked as beta.

**delete ()**

Release this phone number from your account. Twilio will no longer answer calls to this number, and you will stop being billed the monthly phone number fees. The phone number will eventually be recycled and potentially given to another customer, so use with care. If you make a mistake, contact us... we may be able to give you the number back.

**load (entries)**

Set the proper Account owner of this phone number

**transfer (account\_sid)**

Transfer the phone number with sid from the current account to another identified by account\_sid

**update (\*\*kwargs)**

Update this phone number instance.

**class** twilio.rest.resources.**AvailablePhoneNumber** (*parent*)

An available phone number resource

**friendly\_name**

A nicely-formatted version of the phone number.

**phone\_number**

The phone number, in E.164 (i.e. "+1") format.

**lata**

The LATA of this phone number.

**rate\_center**

The rate center of this phone number.

**latitude**

The latitude coordinate of this phone number.

**longitude**

The longitude coordinate of this phone number.

**region**

The two-letter state or province abbreviation of this phone number.

**postal\_code**

The postal (zip) code of this phone number.

**iso\_country**

The country for this number

**address\_requirements**

Whether the phone number requires you or your customer to have an Address registered with Twilio. Possible values are 'none', 'any', 'local', or 'foreign'.

**beta**

(boolean) Phone numbers new to the Twilio platform are marked as beta.

## Queues

```
class twilio.rest.resources.Queues(*args, **kwargs)
```

**create** (*name*, *\*\*kwargs*)

Create an *Queue* with any of these optional parameters.

**Parameters**

- **name** – A human readable description of the application, with maximum length 64 characters.
- **max\_size** – The limit on calls allowed into the queue (optional)

**delete** (*sid*)

Delete a *Queue*. Can only be run on empty queues.

**Parameters** *sid* – String identifier for a Queue resource

**list** (*\*\*kwargs*)

Returns a page of *Queue* resources as a list sorted by DateUpdated. For paging information see *ListResource*

**update** (*sid*, *\*\*kwargs*)

Update a *Queue*

**Parameters**

- **sid** – String identifier for a Queue resource
- **friendly\_name** – A new friendly name for this queue
- **max\_size** – A new max size. Changing a max size to less than the current size results in the queue rejecting incoming requests until it shrinks below the new max size

```
class twilio.rest.resources.Queue(parent, sid)
```

An instance of a Queue

**sid**

A 34 character string that uniquely identifies this queue.

**friendly\_name**

A user-provided string that identifies this queue.

**current\_size**

The count of calls currently in the queue.

**max\_size**

The upper limit of calls allowed to be in the queue. *unlimited* is an option. The default is 100.

**average\_wait\_time**

The average wait time of the members of this queue in seconds. This is calculated at the time of the request.

**uri**

The URI for this resource, relative to <https://api.twilio.com>.

**queue\_members**

A *Members* object holding the *Member* objects in this queue.

**delete()**

Delete this queue. Can only be run on empty queues.

**update(\*\*kwargs)**

Update this queue

**Parameters**

- **friendly\_name** – A new friendly name for this queue
- **max\_size** – A new max size. Changing a max size to less than the current size results in the queue rejecting incoming requests until it shrinks below the new max size

## Queue Members

**class** `twilio.rest.resources.Members(*args, **kwargs)`

A list of *Member* objects

**dequeue** (*url*, *call\_sid*='Front', \*\*kwargs)

Dequeues a member from the queue and have the member's call begin executing the TwiML document at the url.

**Parameters**

- **call\_sid** – Call sid specifying the member, if not given, the member at the front of the queue will be used
- **url** – url of the TwiML document to be executed.

**list** (\*\*kwargs)

Returns a list of *Member* resources in the given queue

**Parameters** **queue\_sid** – *Queue* this participant is part of

**class** `twilio.rest.resources.Member(parent, sid)`

A Member of a queue

**call\_sid**

A 34 character string that uniquely identifies the call that is enqueued.

**date\_enqueued**

The date that the member was enqueued, given in RFC 2822 format.

**wait\_time**

The number of seconds the member has been in the queue.

**position**

This member's current position in the queue.

**uri**

The URI for this resource, relative to <https://api.twilio.com>.

## Recordings

`class twilio.rest.resources.Recordings (*args, **kwargs)`

**delete** (*sid*)

Delete the given recording

**iter** (\*args, \*\*kwargs)

Returns an iterator of *Recording* resources.

### Parameters

- **after** (*date*) – Only list recordings logged after this datetime
- **before** (*date*) – Only list recordings logged before this datetime

**list** (\*args, \*\*kwargs)

Returns a page of *Recording* resources as a list. For paging information see *ListResource*.

### Parameters

- **after** (*date*) – Only list recordings logged after this datetime
- **before** (*date*) – Only list recordings logged before this datetime
- **call\_sid** – Only list recordings from this *Call*

`class twilio.rest.resources.Recording (*args, **kwargs)`

**sid**

A 34 character string that uniquely identifies this resource.

**date\_created**

The date that this resource was created, given in RFC 2822 format.

**date\_updated**

The date that this resource was last updated, given in RFC 2822 format.

**account\_sid**

The unique id of the Account responsible for this recording.

**call\_sid**

The call during which the recording was made.

**duration**

The length of the recording, in seconds.

**api\_version**

The version of the API in use during the recording.

**uri**

The URI for this resource, relative to <https://api.twilio.com>

**subresource\_uris**

The list of subresources under this account

**formats**

A dictionary of the audio formats available for this recording

```
{
  "wav": "http://www.dailywav.com/0112/noFateButWhatWeMake.wav",
  "mp3": "https://api.twilio.com/cowbell.mp3",
}
```

```
delete ()  
    Delete this recording
```

## Sandbox

```
class twilio.rest.resources.Sandboxes (*args, **kwargs)
```

```
get ()  
    Request the specified instance resource
```

```
update (**kwargs)  
    Update your Twilio Sandbox
```

```
class twilio.rest.resources.Sandbox (parent, sid)
```

```
pin  
    An 8 digit number that gives access to this sandbox.
```

```
account_sid  
    The unique id of the Account connected to this sandbox.
```

```
phone_number  
    The phone number of the sandbox. Formatted with a '+' and country code e.g., +16175551212 (E.164 format).
```

```
voice_url  
    The URL Twilio will request when the sandbox number is called.
```

```
voice_method  
    The HTTP method to use when requesting the above URL. Either GET or POST.
```

```
sms_url  
    The URL Twilio will request when receiving an incoming SMS message to the sandbox number.
```

```
sms_method  
    The HTTP method to use when requesting the sms URL. Either GET or POST.
```

```
date_created  
    The date that this resource was created, given in RFC 2822 format.
```

```
date_updated  
    The date that this resource was last updated, given in RFC 2822 format.
```

```
uri  
    The URI for this resource, relative to https://api.twilio.com
```

```
update (**kwargs)  
    Update your Twilio Sandbox
```

## Short Codes

```
class twilio.rest.resources.ShortCodes (*args, **kwargs)
```

```
list (**kwargs)  
    Returns a page of ShortCode resources as a list. For paging information see ListResource.
```

**Parameters**

- **short\_code** – Only show the ShortCode resources that match this pattern. You can specify partial numbers and use ‘\*’ as a wildcard for any digit.
- **friendly\_name** – Only show the ShortCode resources with friendly names that exactly match this name.

**update** (*sid*, *url=None*, *method=None*, *fallback\_url=None*, *fallback\_method=None*, *\*\*kwargs*)

Update a specific *ShortCode*, by specifying the *sid*.

**Parameters**

- **friendly\_name** – Description of the short code, with maximum length 64 characters.
- **api\_version** – SMSs to this short code will start a new TwiML session with this API version.
- **url** – The URL that Twilio should request when somebody sends an SMS to the short code.
- **method** – The HTTP method that should be used to request the url.
- **fallback\_url** – A URL that Twilio will request if an error occurs requesting or executing the TwiML at the url.
- **fallback\_method** – The HTTP method that should be used to request the *fallback\_url*.

**class** `twilio.rest.resources.ShortCode` (*parent*, *sid*)

**sid**

A 34 character string that uniquely identifies this resource.

**date\_created**

The date that this resource was created, given in RFC 2822 format.

**date\_updated**

The date that this resource was last updated, given in RFC 2822 format.

**friendly\_name**

A human readable descriptive text for this resource, up to 64 characters long. By default, the *friendly\_name* is just the short code.

**account\_sid**

The unique id of the Account that owns this short code.

**short\_code**

The short code. e.g., 894546.

**api\_version**

SMSs to this short code will start a new TwiML session with this API version.

**sms\_url**

The URL Twilio will request when receiving an incoming SMS message to this short code.

**sms\_method**

The HTTP method Twilio will use when making requests to the *sms\_url*. Either GET or POST.

**sms\_fallback\_url**

The URL that Twilio will request if an error occurs retrieving or executing the TwiML from *sms\_url*.

**sms\_fallback\_method**

The HTTP method Twilio will use when requesting the above URL. Either GET or POST.

**uri**

The URI for this resource, relative to <https://api.twilio.com>.

## SMS Messages

```
class twilio.rest.resources.SmsMessages(*args, **kwargs)
```

```
create(from_=None, **kwargs)
```

Create and send a SMS Message.

### Parameters

- **to** (*str*) – The destination phone number.
- **from\_** (*str*) – The phone number sending this message (must be a verified Twilio number)
- **body** (*str*) – The message you want to send, limited to 160 characters.
- **status\_callback** – A URL that Twilio will POST to when your message is processed.
- **application\_sid** (*str*) – The 34 character sid of the application Twilio should use to handle this phone call.

Usage:

**instance**

alias of *SmsMessage*

```
list(*args, **kwargs)
```

Returns a page of *SmsMessage* resources as a list. For paging information see *ListResource*.

### Parameters

- **to** – Only show SMS messages to this phone number.
- **from** – Only show SMS messages from this phone number.
- **after** (*date*) – Only list SMS messages sent after this date.
- **before** (*date*) – Only list SMS message sent before this date.
- **date\_sent** (*date*) – Only list SMS message sent on this date.
- **from\_** – Only show SMS messages from this phone number.
- **after** – Only list SMS messages logged after this datetime
- **before** – Only list SMS messages logged before this datetime

```
class twilio.rest.resources.SmsMessage(parent, sid)
```

An instance of an SMS Message

**sid**

A 34 character string that uniquely identifies this resource.

**date\_created**

The date that this resource was created, given in RFC 2822 format.

**date\_updated**

The date that this resource was last updated, given in RFC 2822 format.

**date\_sent**

The date that the SMS was sent, given in RFC 2822 format.

**account\_sid**

The unique id of the Account that sent this SMS message.

**from**

The phone number that initiated the message in E.164 format. For incoming messages, this will be the remote phone. For outgoing messages, this will be one of your Twilio phone numbers.

**to**

The phone number that received the message in E.164 format. For incoming messages, this will be one of your Twilio phone numbers. For outgoing messages, this will be the remote phone.

**body**

The text body of the SMS message.

**status**

The status of this SMS message. Either queued, sending, sent, or failed.

**direction**

The direction of this SMS message. `incoming` for incoming messages, `outbound-api` for messages initiated via the REST API, `outbound-call` for messages initiated during a call or `outbound-reply` for messages initiated in response to an incoming SMS.

**price**

The amount billed for the message.

**api\_version**

The version of the Twilio API used to process the SMS message.

**uri**

The URI for this resource, relative to <https://api.twilio.com>

## Transcriptions

```
class twilio.rest.resources.Transcriptions(*args, **kwargs)
```

**delete** (*sid*)

Delete the given transcription

**list** (*\*\*kwargs*)

Return a list of *Transcription* resources

```
class twilio.rest.resources.Transcription(parent, sid)
```

**sid**

A 34 character string that uniquely identifies this resource.

**date\_created**

The date that this resource was created, given in RFC 2822 format.

**date\_updated**

The date that this resource was last updated, given in RFC 2822 format.

**account\_sid**

The unique id of the Account responsible for this transcription.

**status**

A string representing the status of the transcription: in-progress, completed or failed.

**recording\_sid**

The unique id of the Recording this Transcription was made of.

**duration**

The duration of the transcribed audio, in seconds.

**transcription\_text**

The text content of the transcription.

**price**

The charge for this transcript in USD. Populated after the transcript is completed. Note, this value may not be immediately available.

**uri**

The URI for this resource, relative to <https://api.twilio.com>

**delete ()**

Delete this transcription

## 5.1.4 twilio.rest.resources.task\_router

### Workspaces

**class** `twilio.rest.resources.task_router.Workspaces (*args, **kwargs)`

A list of Workspace resources

**create** (*friendly\_name*, \*\*kwargs)

Create a Workspace.

**Parameters**

- **friendly\_name** – Human readable description of this workspace (for example “Customer Support” or “2014 Election Campaign”).
- **event\_callback\_url** – If provided, the Workspace will publish events to this URL. You can use this to gather data for reporting. See [Workspace Events](#) for more information.
- **template** – One of the available template names. Will pre-configure this Workspace with the Workflow and Activities specified in the template. Currently “FIFO” is the only available template, which will configure Work Distribution Service with a set of default activities and a single queue for first-in, first-out distribution.

**delete** (*sid*)

Delete the given workspace

**update** (*sid*, \*\*kwargs)

Update a *Workspace* with the given parameters.

All the parameters are describe above in `create ()`

**class** `twilio.rest.resources.task_router.Workspace (*args, **kwargs)`

A Workspace resource. See the *TaskRouter API reference* <<https://www.twilio.com/docs/taskrouter/workspaces>>\_ for more information.

**sid**

The unique ID of the Workspace

**account\_sid**

The ID of the account that owns this Workspace

**friendly\_name**

Human readable description of this workspace (for example “Sales Call Center” or “Customer Support Team”)

**default\_activity\_sid**

The ID of the default `Activity` that will be used when new `Workers` are created in this `Workspace`.

**default\_activity\_name**

The human readable name of the default activity. Read only.

**timeout\_activity\_sid**

The ID of the `Activity` that will be assigned to a `Worker` when a `Task` reservation times out without a response.

**timeout\_activity\_name**

The human readable name of the timeout activity. Read only.

**event\_callback\_url**

An optional URL where the `Workspace` will publish events. You can use this to gather data for reporting.

**date\_created**

The time the `Workspace` was created, given as UTC in ISO 8601 format.

**date\_updated**

The time the `Workspace` was last updated, given as UTC in ISO 8601 format.

## Workflows

**class** `twilio.rest.resources.task_router.Workflows` (\*args, \*\*kwargs)

A list of `Workflow` resources

**create** (*friendly\_name*, *configuration*, *assignment\_callback\_url*, \*\*kwargs)

Create a `Workflow`.

### Parameters

- **friendly\_name** – A string representing a human readable name for this `Workflow`. Examples include ‘Inbound Call Workflow’ or ‘2014 Outbound Campaign’.
- **configuration** – JSON document configuring the rules for this `Workflow`.
- **assignment\_callback\_url** – A valid URL for the application that will process task assignment events.
- **fallback\_assignment\_callback\_url** – If the request to the `assignment_callback_url` fails, the assignment callback will be made to this URL.
- **task\_reservation\_timeout** – An integer value controlling how long in seconds `Work Distribution Service` will wait for a confirmation response from your application after assigning a `Task` to a worker. Defaults to 120 seconds.

**delete** (*sid*)

Delete the given workflow

**update** (*sid*, \*\*kwargs)

Update a `Workflow` with the given parameters.

All the parameters are describe above in `create()`

**class** `twilio.rest.resources.task_router.Workflow` (\*args, \*\*kwargs)

A `Workflow` resource.

See the *TaskRouter API reference* <<https://www.twilio.com/docs/taskrouter/workflows>>\_ for more information.

**sid**

The unique ID of the `Workflow`.

**account\_sid**

The ID of the account that owns this Workflow.

**workspace\_sid**

The ID of the *Workspace* that contains this Workflow.

**friendly\_name**

A human-readable description of this Workflow.

**assignment\_callback\_url**

The URL that will be called whenever a *Task* managed by this Workflow is assigned to a *Worker*.

**fallback\_assignment\_callback\_url**

If the request to the *assignment\_callback\_url* fails, the assignment callback will be made to this URL.

**configuration**

A JSON document configuring the rules for this workflow.

**task\_reservation\_timeout**

Determines how long TaskRouter will wait for a confirmation response from your application after assigning a Task to a worker. Defaults to 120 seconds.

**date\_created**

The time this workflow was created, as UTC in ISO 8601 format.

**date\_updated**

The time this workflow was last updated, as UTC in ISO 8601 format.

## Activities

**class** twilio.rest.resources.task\_router.**Activities** (\*args, \*\*kwargs)

A list of Activity resources

**create** (*friendly\_name*, *available*)

Create an Activity.

### Parameters

- **friendly\_name** – A human-readable name for the activity, such as ‘On Call’, ‘Break’, ‘Email’, etc. Must be unique in this Workspace. These names will be used to calculate and expose statistics about workers, and give you visibility into the state of each of your workers.
- **available** – Boolean value indicating whether the worker should be eligible to receive a Task when they occupy this Activity. For example, a call center might have an activity named ‘On Call’ with an availability set to ‘false’.

**delete** (*sid*)

Delete the given activity

**update** (*sid*, \*\*kwargs)

Update an Activity with the given parameters.

All the parameters are describe above in *create()*

## Workers

**class** twilio.rest.resources.task\_router.**Workers** (*base\_uri*, *auth*, *timeout*, \*\*kwargs)

A list of Worker resources

**create** (*friendly\_name*, *\*\*kwargs*)

Create a Workflow.

#### Parameters

- **friendly\_name** – String representing user-friendly name for the Worker.
- **activity\_sid** – A valid Activity describing the worker’s initial state.
- **attributes** – JSON object describing this worker. For example: { ‘email’: ‘Bob@foo.com’, ‘phone’: ‘8675309’ }. This data will be passed to the Assignment Callback URL whenever Work Distribution Service assigns a Task to this worker.

**delete** (*sid*)

Delete the given worker

**update** (*sid*, *\*\*kwargs*)

Update a *Worker* with the given parameters.

All the parameters are describe above in *create()*

**class** `twilio.rest.resources.task_router.Worker` (*\*args*, *\*\*kwargs*)

A Worker resource.

See the *TaskRouter API reference* <<https://www.twilio.com/docs/taskrouter/workers>>\_ for more information.

**sid**

The unique ID of this Worker.

**account\_sid**

The unique ID of the account that owns this worker.

**workspace\_sid**

The unique ID of the *Workspace* that contains this worker.

**friendly\_name**

A human-readable name for this worker.

**attributes**

A JSON object describing this worker. For example, for a Worker that handles English language phone calls:

```
{“language”:”english”,”task-type”:”phone”}
```

These attributes determine which *TaskQueue* this worker will subscribe to. These attributes will also be passed to the Assignment Callback URL whenever TaskRouter assigns a *Task* to this worker, so you can also use this as a place to store information that you’ll need when routing a Task to the Worker (for example, the Worker’s phone number or Twilio Client name).

**available**

A Boolean value indicating whether the worker can be assigned another task. When true, the worker can be assigned a new task; when false, the worker will not be assigned any tasks.

**activity\_sid**

The unique ID of the *Activity* this Worker is currently performing.

**activity\_name**

A string describing the worker’s current activity. Workers may only perform Activities that exist in this Workspace.

**date\_created**

The time this worker was created, given in UTC ISO 8601 format.

**date\_updated**

The time this worker was last updated, given in UTC ISO 8601 format.

**date\_status\_changed**

The time of the last change to this worker's activity. Used to calculate :class: *Workflow* statistics.

## TaskQueues

**class** `twilio.rest.resources.task_router.TaskQueues` (*base\_uri, auth, timeout, \*\*kwargs*)

A list of TaskQueue resources

**create** (*friendly\_name, assignment\_activity\_sid, reservation\_activity\_sid, \*\*kwargs*)

Create a TaskQueue.

### Parameters

- **friendly\_name** – Human readable description of this TaskQueue (for example “Support - Tier 1”, “Sales” or “Escalation”)
- **assignment\_activity\_sid** – ActivitySID to assign workers once a task is assigned for them.
- **reservation\_activity\_sid** – ActivitySID to assign workers once a task is reserved for them.

**delete** (*sid*)

Delete the given task queue

**update** (*sid, \*\*kwargs*)

Update a *TaskQueue* with the given parameters.

All the parameters are describe above in *create()*

**class** `twilio.rest.resources.task_router.TaskQueue` (*\*args, \*\*kwargs*)

A TaskQueue resource.

See the *TaskRouter API reference* <<https://www.twilio.com/docs/taskrouter/taskqueues>>\_ for more information.

**sid**

The unique ID of this TaskQueue.

**account\_sid**

The unique ID of the account that owns this TaskQueue.

**workspace\_sid**

The unique ID of the *Workspace* that contains this TaskQueue.

**friendly\_name**

Human-readable description of this TaskQueue (e.g. “Customer Support” or “Sales”).

**target\_workers**

The worker selection expressions associated with this TaskQueue.

**reservation\_activity\_sid**

The *Activity* to assign a *Worker* when they are reserved for a *Task* from this TaskQueue. Defaults to ‘Reserved for Task’.

**assignment\_activity\_sid**

The *Activity* to assign a *Worker* when they accept a *Task* from this Taskqueue. Defaults to ‘Unavailable for Assignment’.

## Tasks

**class** `twilio.rest.resources.task_router.Tasks(*args, **kwargs)`

A list of Task resources

**create** (*attributes*, *workflow\_sid*, *\*\*kwargs*)

Create a Task.

### Parameters

- **attributes** – Url-encoded JSON string describing the attributes of this task. This data will be passed back to the Workflow’s AssignmentCallbackURL when the Task is assigned to a Worker. An example task: { ‘task\_type’: ‘call’, ‘twilio\_call\_sid’: ‘...’, ‘customer\_ticket\_number’: ‘12345’ }.
- **workflow\_sid** – The workflow\_sid for the Workflow that you would like to handle routing for this Task.
- **timeout** – If provided, time-to-live for the task in seconds, before it is automatically canceled

**delete** (*sid*)

Delete the given task

**update** (*sid*, *\*\*kwargs*)

Update a *Task* with the given parameters.

All the parameters are describe above in *create()*

**class** `twilio.rest.resources.task_router.Task(*args, **kwargs)`

A Task resource.

See the *TaskRouter API reference* <<https://www.twilio.com/docs/taskrouter/tasks>>\_ for more information.

**sid**

The unique ID for this Task.

**account\_sid**

The ID of the account that owns this Task.

**workspace\_sid**

The ID of the *Workspace* that contains this Task.

**workflow\_sid**

The ID of the *Workflow* that is responsible for routing this Task.

**attributes**

The user-defined JSON string describing the custom attributes of this work.

**age**

The number of seconds since this Task was created.

**priority**

The current priority score of the task, as assigned by the Workflow. Tasks with higher values will be assigned before tasks with lower values.

**task\_queue\_sid**

The ID of the current TaskQueue this task occupies, controlled by the Workflow.

**assignment\_status**

A string representing the assignment state of the task. May be ‘pending’, ‘reserved’, ‘assigned’, or ‘canceled’.

**reason**

The reason the task was canceled (if applicable).

**date\_created**

The date this task was created, as UTC in ISO 8601 format.

**date\_updated**

The date this task was last updated, as UTC in ISO 8601 format.

## Events

```
class twilio.rest.resources.task_router.Events(*args, **kwargs)
```

**get\_instances** (*params*)

Query the list resource for a list of InstanceResources. Raises a *TwilioRestException* if requesting a page of results that does not exist.

**Parameters**

- **params** (*dict*) – List of URL parameters to be included in request
- **page\_token** (*str*) – Token of the page of results to retrieve
- **page\_size** (*int*) – The number of results to be returned.

**Returns** – the list of resources

**list** (*\*\*kwargs*)

**Returns a page of *Event* resources as a list. For paging information see *NextGenListResource***

**Parameters**

- **minutes** – (Optional, Default=15) Definition of the interval in minutes prior to now.
- **start\_date** – (Optional, Default=15 minutes prior) Filter events by a start date.
- **end\_date** – (Optional, Default=Now) Filter events by an end date.
- **resource\_sid** – (Optional) Sid of the event resource.
- **event\_type** – (Optional) The type of event to filter by.

```
class twilio.rest.resources.task_router.Event(*args, **kwargs)
```

An Event resource representing a state change in a TaskRouter Workspace.

See the *TaskRouter API reference* <<https://www.twilio.com/docs/taskrouter/events>>\_ for more information.

**event\_type**

An identifier for this event.

**account\_sid**

The unique ID of the Account that owns this Event.

**description**

A description of the event.

**resource\_type**

The type of object this event is most relevant to (*Task*, *Reservation*, *Worker*).

**resource\_sid**

The unique ID of the object this event is most relevant to.

**event\_date**

The time this event was sent, in UTC ISO 8601 format.

**event\_data**

Data about this specific Event.

## 5.1.5 twilio.twiml

Make sure to check out the TwiML overview and tutorial at <https://www.twilio.com/docs/api/twiml>

**class** `twilio.twiml.Response` (*\*\*kwargs*)

Twilio response object.

**dial** (*number=None, \*\*kwargs*)

Return a newly created *Dial* verb, nested inside this *Response*

**enqueue** (*name, \*\*kwargs*)

Return a newly created *Enqueue* verb, nested inside this *Response*

**gather** (*\*\*kwargs*)

Return a newly created *Gather* verb, nested inside this *Response*

**hangup** (*\*\*kwargs*)

Return a newly created *Hangup* verb, nested inside this *Response*

**leave** (*\*\*kwargs*)

Return a newly created *Leave* verb, nested inside this *Response*

**message** (*msg=None, \*\*kwargs*)

Return a newly created *Message* verb, nested inside this *Response*

**pause** (*\*\*kwargs*)

Return a newly created *Pause* verb, nested inside this *Response*

**play** (*url=None, digits=None, \*\*kwargs*)

Return a newly created *Play* verb, nested inside this *Response*

**record** (*\*\*kwargs*)

Return a newly created *Record* verb, nested inside this *Response*

**redirect** (*url=None, \*\*kwargs*)

Return a newly created *Redirect* verb, nested inside this *Response*

**reject** (*reason=None, \*\*kwargs*)

Return a newly created *Hangup* verb, nested inside this *Response*

**say** (*text, \*\*kwargs*)

Return a newly created *Say* verb, nested inside this *Response*

**sms** (*msg, \*\*kwargs*)

Return a newly created *Sms* verb, nested inside this *Response*

### Primary Verbs

**class** `twilio.twiml.Say` (*text, \*\*kwargs*)

The *Say* verb converts text to speech that is read back to the caller.

#### Parameters

- **voice** – allows you to choose a male or female voice to read text back.

- **language** – allows you pick a voice with a specific language’s accent and pronunciations. Twilio currently supports languages ‘en’ (English), ‘es’ (Spanish), ‘fr’ (French), and ‘de’ (German), ‘en-gb’ (English Great Britain”).
- **loop** – specifies how many times you’d like the text repeated. Specifying ‘0’ will cause the the *Say* verb to loop until the call is hung up. Defaults to 1.

**class** `twilio.twiml.Play` (*url=None, digits=None, \*\*kwargs*)  
Play DTMF digits or audio from a URL.

#### Parameters

- **url** (*str*) – point to an audio file. The MIME type on the file must be set correctly. At least one of *url* and *digits* must be specified. If both are given, the digits will play prior to the audio from the URL.
- **digits** (*str*) – a string of digits to play. To pause before playing digits, use leading ‘w’ characters. Each ‘w’ will cause Twilio to wait 0.5 seconds instead of playing a digit. At least one of *url* and *digits* must be specified. If both are given, the digits will play first.
- **loop** (*int*) – specifies how many times you’d like the text repeated. Specifying ‘0’ will cause the the *Play* verb to loop until the call is hung up. Defaults to 1.

**class** `twilio.twiml.Dial` (*number=None, \*\*kwargs*)  
Dial another phone number and connect it to this call

#### Parameters

- **action** – submit the result of the dial to this URL
- **method** – submit to ‘action’ url using GET or POST
- **timeout** (*int*) – The number of seconds to waits for the called party to answer the call
- **hangupOnStar** (*bool*) – Allow the calling party to hang up on the called party by pressing the ‘\*’ key
- **timeLimit** (*int*) – The maximum duration of the Call in seconds
- **callerId** – The caller ID that will appear to the called party
- **record** (*bool*) – Record both legs of a call within this <Dial>

**class** `twilio.twiml.Gather` (*\*\*kwargs*)  
Gather digits from the caller’s keypad

#### Parameters

- **action** – URL to which the digits entered will be sent
- **method** – submit to ‘action’ url using GET or POST
- **numDigits** – how many digits to gather before returning
- **timeout** – wait for this many seconds before returning
- **finishOnKey** – key that triggers the end of caller input

**class** `twilio.twiml.Record` (*\*\*kwargs*)  
Record audio from caller

#### Parameters

- **action** – submit the result of the dial to this URL
- **method** – submit to ‘action’ url using GET or POST

- **maxLength** – maximum number of seconds to record
- **timeout** – seconds of silence before considering the recording done

## Secondary Verbs

**class** `twilio.twiml.Hangup(**kwargs)`  
Hangup the call

**class** `twilio.twiml.Redirect(url='', **kwargs)`  
Redirect call flow to another URL

### Parameters

- **url** – specifies the url which Twilio should query to retrieve new TwiML. The default is the current url
- **method** – specifies the HTTP method to use when retrieving the url

**class** `twilio.twiml.Reject(**kwargs)`  
Hangup the call

**Parameters** **reason** – not sure

**class** `twilio.twiml.Pause(**kwargs)`  
Pause the call

**Parameters** **length** – specifies how many seconds Twilio will wait silently before continuing on.

**class** `twilio.twiml.Message(msg=None, **kwargs)`  
Send an MMS Message to a phone number.

### Parameters

- **to** – whom to send message to
- **sender** – whom to send message from.
- **action** – url to request after the message is queued
- **method** – submit to ‘action’ url using GET or POST
- **statusCallback** – url to hit when the message is actually sent

**class** `twilio.twiml.Enqueue(name, **kwargs)`  
Enqueue the call into a specific queue.

### Parameters

- **name** – friendly name for the queue
- **action** – url to a twiml document that executes when the call leaves the queue. When dequeued via a <Dial> verb, this url is executed after the bridged parties disconnect
- **method** – HTTP method for action GET/POST
- **waitUrl** – url to a twiml document that executes while the call is on the queue
- **waitUrlMethod** – HTTP method for waitUrl GET/POST

**class** `twilio.twiml.Leave(**kwargs)`  
Signals the call to leave its queue

## Nouns

**class** `twilio.twiml.Conference` (*name*, *\*\*kwargs*)  
Specify conference in a nested Dial element.

### Parameters

- **name** – friendly name of conference
- **muted** (*bool*) – keep this participant muted
- **beep** (*bool*) – play a beep when this participant enters/leaves
- **startConferenceOnEnter** (*bool*) – start conf when this participants joins
- **endConferenceOnExit** (*bool*) – end conf when this participants leaves
- **waitUrl** – TwiML url that executes before conference starts
- **waitMethod** – HTTP method for waitUrl GET/POST

**class** `twilio.twiml.Number` (*number*, *\*\*kwargs*)  
Specify phone number in a nested Dial element.

### Parameters

- **number** – phone number to dial
- **sendDigits** – key to press after connecting to the number

**class** `twilio.twiml.Client` (*name*, *\*\*kwargs*)  
Specify a client name to call in a nested Dial element.

**Parameters** **name** – Client name to connect to

**class** `twilio.twiml.Queue` (*name*, *\*\*kwargs*)  
Specify queue in a nested Dial element.

### Parameters

- **name** – friendly name for the queue
- **url** – url to a twiml document that executes after a call is dequeued and before the call is connected
- **method** – HTTP method for url GET/POST

## 5.1.6 twilio.util

**class** `twilio.util.RequestValidator` (*token*)

**compute\_signature** (*uri*, *params*, *utf=False*)  
Compute the signature for a given request

### Parameters

- **uri** – full URI that Twilio requested on your server
- **params** – post vars that Twilio sent with the request
- **utf** – whether return should be bytestring or unicode (python3)

**Returns** The computed signature

**validate** (*uri, params, signature*)

Validate a request from Twilio

**Parameters**

- **uri** – full URI that Twilio requested on your server
- **params** – post vars that Twilio sent with the request
- **signature** – expected signature in HTTP X-Twilio-Signature header

**Returns** True if the request passes validation, False if not

**class** `twilio.util.TwilioCapability` (*account\_sid, auth\_token*)

A token to control permissions with Twilio Client

**Parameters**

- **account\_sid** (*str*) – the account sid to which this token is granted access
- **auth\_token** (*str*) – the secret key used to sign the token. Note, this auth token is not visible to the user of the token.

**Returns** A new TwilioCapability with zero permissions

**allow\_client\_incoming** (*client\_name*)

If the user of this token should be allowed to accept incoming connections then configure the TwilioCapability through this method and specify the client name.

**Parameters** **client\_name** (*str*) – Client name to accept calls from

**allow\_client\_outgoing** (*application\_sid, \*\*kwargs*)

Allow the user of this token to make outgoing connections.

Keyword arguments are passed to the application.

**Parameters** **application\_sid** (*str*) – Application to contact

**generate** (*expires=3600*)

Generate a valid JWT token with an expiration date.

**Parameters** **expires** (*int*) – The token lifetime, in seconds. Defaults to 1 hour (3600)



---

## Deploying to Google App Engine

---

Want to run your app on Google App Engine? We've got a full guide to getting started here:

### 6.1 Deploying to Google App Engine

Adding the *twilio-python* library to a Google App Engine project is a little tricky. Let's walk you through one way to do it.

#### 6.1.1 Laying Out Your Project

The key is to lay out your project in a way that makes sense.

1. Create a project folder called `twilio-demo`. You can name the folder anything you like, but for the rest of this tutorial, we'll assume it's named `twilio-demo`. At the command line, add a `virtualenv` inside of that folder, by running:

```
mkdir twilio-demo      # Creates a new twilio-demo folder
cd twilio-demo         # Move into that folder
virtualenv .           # Create a new virtual environment
```

You should now have a directory structure that looks like this:

```
twilio-demo
- bin
- include
- lib
```

We'll get to the Google App Engine part in a few steps.

2. Now let's install the `twilio-python` library in our `Virtualenv`. If your current working directory is `twilio-demo`, we want to source the `activate` file in the `bin` folder, then install the library with `pip`.

```
source bin/activate      # Activate our virtual environment
pip install twilio      # Install the twilio-python library
```

- Now let's add a new folder called `src`. This is the folder that contains your `app.yaml` and your other Google App Engine files. You can add this at the command line. If your current directory is `twilio-demo`:

```
mkdir src
```

- Create a basic `app.yaml` file in your `src` directory, per the instructions Google App Engine provides. Your folder structure should now look something like this:

```
twilio-demo
- bin
| - activate
| - ... about 20 files
- include
| - python2.7 -> /path/to/system/python-2.7
- lib
| - python2.7      # This folder contains a bunch of files
- src
  - app.yaml
  - helloworld.py
```

- Link the `twilio-python` library into your `src` directory. We are going to use a symbolic link. Google will pick this up and import the library into the correct place. In the terminal, run these three commands from the `src` directory inside `twilio-demo`:

```
ln -s ../lib/python2.7/site-packages/twilio .
ln -s ../lib/python2.7/site-packages/httplib2 .
ln -s ../lib/python2.7/site-packages/six.py .
```

This should create a symbolic link inside the `src` directory to the `twilio-python` module. You can test the link as follows. Inside the `twilio-demo/src` folder, create a file called `helloworld.py` and put this inside it:

```
import webapp2
import twilio

class MainPage(webapp2.RequestHandler):
    def get(self):
        self.response.headers['Content-Type'] = 'text/plain'
        self.response.write("The twilio version is " + twilio.__version__)

app = webapp2.WSGIApplication([('/', MainPage)],
                             debug=True)
```

- Finally, configure your app in Google App Engine and deploy it. Here are the settings you want in Google App Engine - Note the folder path ends with `twilio-demo/src`.

Once App Engine is running locally, in your browser, you should be able to navigate to `http://localhost` + the provided Port and view the `twilio` library version number, as well as deploy your app to Google. Once you have this set up, adding more complicated actions using the `twilio` library should be a snap.

Hope that helps! If you have questions, we're always listening at [help@twilio.com](mailto:help@twilio.com).

---

## Frequently Asked Questions

---

What to do if you get an `ImportError`, and some advice about how to format phone numbers.

### 7.1 Frequently Asked Questions

Hopefully you can find an answer here to one of your questions. If not, please contact [help@twilio.com](mailto:help@twilio.com).

#### 7.1.1 ImportError messages

If you get an error that looks like this:

```
Traceback (most recent call last):
  File "twilio.py", line 1, in <module>
    from twilio.rest import TwilioRestClient
  File "../twilio-python/docs/twilio.py", line 1, in <module>
    from twilio.rest import TwilioRestClient
ImportError: No module named rest
```

Check to make sure that you don't have a file named `twilio.py`; Python will try to load the Twilio library from your `twilio.py` file instead of from the Twilio library.

If you get an error that looks like this:

```
Traceback (most recent call last):
  File "test.py", line 1, in <module>
    import twilio.rest
ImportError: No module named twilio.rest
```

Your Python installation cannot find the library.

Check which versions of `pip` and Python you are running with this command in the Terminal:

```
which -a python
which -a pip
```

`pip` needs to install the Twilio library to a path that your Python executable can read from. Sometimes there will be more than one version of `pip`, like `pip-2.5`, `pip-2.7` etc. You can find all of them by running `compgen -c | grep pip` (works with Bash on \*nix machines). There can also be more than one version of Python, especially if you have Macports or homebrew.

You also may be using an outdated version of the `twilio-python` library, which did not use a `twilio.rest.TwilioRestClient` object. Check which version of the `twilio` library you have installed by running this command:

```
$ pip freeze | grep twilio           # Or pip-2.7 freeze etc.
```

The latest version (as of January 2012) is 3.3. If you are running an outdated version, you can upgrade with this command:

```
$ pip install --upgrade twilio
```

Note that if you have code that uses the older version of the library, it may break when you upgrade your site.

## 7.1.2 Formatting phone numbers

Twilio always returns phone numbers that are formatted in the [E.164 format](#), like this: `+12125551234`. However your users may enter them like this: `(212) 555-1234`. This can lead to problems when, for example, Twilio makes a POST request to your server with the `From` phone number as `+12125551234`, but you stored the phone number in your database as `(212) 555-1234`, causing a database lookup to fail.

We suggest that you convert the number to E.164 format before you store it in the database. The [phonenumbers](#) library is excellent for this purpose. Install it like this:

```
$ pip install phonenumbers
```

Then you can convert user input to phone numbers like this:

```
import phonenumbers

def convert_to_e164(raw_phone):
    if not raw_phone:
        return

    if raw_phone[0] == '+':
        # Phone number may already be in E.164 format.
        parse_type = None
    else:
        # If no country code information present, assume it's a US number
        parse_type = "US"

    phone_representation = phonenumbers.parse(raw_phone, parse_type)
    return phonenumbers.format_number(phone_representation,
        phonenumbers.PhoneNumberFormat.E164)

print convert_to_e164('(212) 555 1234') # prints +12125551234
```

## CHAPTER 8

---

### Changelog

---

See the `/changelog` for version history.



## CHAPTER 9

---

### Support and Development

---

All development occurs over on [Github](#). To checkout the source,

```
git clone git@github.com:twilio/twilio-python.git
```

Report bugs using the Github [issue tracker](#).

If you have questions that aren't answered by this documentation, ask the [#twilio IRC channel](#)



**t**

twilio, 47  
twilio.rest, 5  
twilio.rest.resources, 21  
twilio.rest.resources.sms\_messages, 7  
twilio.rest.resources.task\_router, 27  
twilio.task\_router, 39  
twilio.twiml, 40  
twilio.util, 42



## A

- Account (class in twilio.rest.resources), 50
- account\_sid (twilio.rest.resources.Call attribute), 57
- account\_sid (twilio.rest.resources CallerId attribute), 59
- account\_sid (twilio.rest.resources.Conference attribute), 60
- account\_sid (twilio.rest.resources.ConnectApp attribute), 61
- account\_sid (twilio.rest.resources.Notification attribute), 62
- account\_sid (twilio.rest.resources.Participant attribute), 64
- account\_sid (twilio.rest.resources.PhoneNumber attribute), 65
- account\_sid (twilio.rest.resources.Recording attribute), 69
- account\_sid (twilio.rest.resources.Sandbox attribute), 70
- account\_sid (twilio.rest.resources.ShortCode attribute), 71
- account\_sid (twilio.rest.resources.SmsMessage attribute), 72
- account\_sid (twilio.rest.resources.task\_router.Event attribute), 80
- account\_sid (twilio.rest.resources.task\_router.Task attribute), 79
- account\_sid (twilio.rest.resources.task\_router.TaskQueue attribute), 78
- account\_sid (twilio.rest.resources.task\_router.Worker attribute), 77
- account\_sid (twilio.rest.resources.task\_router.Workflow attribute), 76
- account\_sid (twilio.rest.resources.task\_router.Workspace attribute), 74
- account\_sid (twilio.rest.resources.Transcription attribute), 73
- Accounts (class in twilio.rest.resources), 50
- activate() (twilio.rest.resources.Account method), 51
- activate() (twilio.rest.resources.Accounts method), 50
- ACTIVE (twilio.rest.resources.Account attribute), 51
- Activities (class in twilio.rest.resources.task\_router), 76
- activities() (twilio.rest.TwilioTaskRouterClient method), 48
- activity\_name (twilio.rest.resources.task\_router.Worker attribute), 77
- activity\_sid (twilio.rest.resources.task\_router.Worker attribute), 77
- Address (class in twilio.rest.resources), 52
- address\_requirements (twilio.rest.resources.AvailablePhoneNumber attribute), 67
- Addresses (class in twilio.rest.resources), 51
- age (twilio.rest.resources.task\_router.Task attribute), 79
- allow\_client\_incoming() (twilio.util.TwilioCapability method), 85
- allow\_client\_outgoing() (twilio.util.TwilioCapability method), 85
- answered\_by (twilio.rest.resources.Call attribute), 58
- api\_version (twilio.rest.resources.Application attribute), 54
- api\_version (twilio.rest.resources.Notification attribute), 62
- api\_version (twilio.rest.resources.PhoneNumber attribute), 65
- api\_version (twilio.rest.resources.Recording attribute), 69
- api\_version (twilio.rest.resources.ShortCode attribute), 71
- api\_version (twilio.rest.resources.SmsMessage attribute), 73
- Application (class in twilio.rest.resources), 54
- Applications (class in twilio.rest.resources), 53
- assignment\_activity\_sid (twilio.rest.resources.task\_router.TaskQueue attribute), 78
- assignment\_callback\_url (twilio.rest.resources.task\_router.Workflow attribute), 76
- assignment\_status (twilio.rest.resources.task\_router.Task attribute), 79
- attributes (twilio.rest.resources.task\_router.Task attribute), 79
- attributes (twilio.rest.resources.task\_router.Worker attribute), 77

auth\_token (twilio.rest.resources.Account attribute), 51  
 authorize\_redirect\_url (twilio.rest.resources.ConnectApp attribute), 61  
 available (twilio.rest.resources.task\_router.Worker attribute), 77  
 AvailablePhoneNumber (class in twilio.rest.resources), 66  
 average\_wait\_time (twilio.rest.resources.Queue attribute), 67

## B

beta (twilio.rest.resources.AvailablePhoneNumber attribute), 67  
 beta (twilio.rest.resources.PhoneNumber attribute), 66  
 body (twilio.rest.resources.SmsMessage attribute), 73

## C

Call (class in twilio.rest.resources), 57  
 call\_sid (twilio.rest.resources.Member attribute), 68  
 call\_sid (twilio.rest.resources.Notification attribute), 62  
 call\_sid (twilio.rest.resources.Participant attribute), 63  
 call\_sid (twilio.rest.resources.Recording attribute), 69  
 caller\_name (twilio.rest.resources.Call attribute), 58  
 CallerId (class in twilio.rest.resources), 59  
 CallerIds (class in twilio.rest.resources), 59  
 Calls (class in twilio.rest.resources), 55  
 cancel() (twilio.rest.resources.Call method), 58  
 cancel() (twilio.rest.resources.Calls method), 55  
 city (twilio.rest.resources.Address attribute), 52  
 Client (class in twilio.twiml), 84  
 close() (twilio.rest.resources.Account method), 51  
 close() (twilio.rest.resources.Accounts method), 50  
 CLOSED (twilio.rest.resources.Account attribute), 51  
 company\_name (twilio.rest.resources.ConnectApp attribute), 61  
 compute\_signature() (twilio.util.RequestValidator method), 84  
 Conference (class in twilio.rest.resources), 60  
 Conference (class in twilio.twiml), 84  
 conference\_sid (twilio.rest.resources.Participant attribute), 63  
 Conferences (class in twilio.rest.resources), 60  
 configuration (twilio.rest.resources.task\_router.Workflow attribute), 76  
 ConnectApp (class in twilio.rest.resources), 61  
 ConnectApps (class in twilio.rest.resources), 61  
 create() (twilio.rest.resources.Accounts method), 50  
 create() (twilio.rest.resources.Addresses method), 51  
 create() (twilio.rest.resources.Applications method), 53  
 create() (twilio.rest.resources.Calls method), 55  
 create() (twilio.rest.resources.Queues method), 67  
 create() (twilio.rest.resources.SmsMessages method), 72  
 create() (twilio.rest.resources.task\_router.Activities method), 76

create() (twilio.rest.resources.task\_router.TaskQueues method), 78  
 create() (twilio.rest.resources.task\_router.Tasks method), 79  
 create() (twilio.rest.resources.task\_router.Workers method), 76  
 create() (twilio.rest.resources.task\_router.Workflows method), 75  
 create() (twilio.rest.resources.task\_router.Workspaces method), 74  
 create\_instance() (twilio.rest.resources.Calls method), 56  
 current\_size (twilio.rest.resources.Queue attribute), 67  
 customer\_name (twilio.rest.resources.Address attribute), 52

## D

date\_created (twilio.rest.resources.Account attribute), 50  
 date\_created (twilio.rest.resources.Application attribute), 54  
 date\_created (twilio.rest.resources.Call attribute), 57  
 date\_created (twilio.rest.resources.CallerId attribute), 59  
 date\_created (twilio.rest.resources.Conference attribute), 60  
 date\_created (twilio.rest.resources.ConnectApp attribute), 61  
 date\_created (twilio.rest.resources.Notification attribute), 62  
 date\_created (twilio.rest.resources.Participant attribute), 63  
 date\_created (twilio.rest.resources.PhoneNumber attribute), 65  
 date\_created (twilio.rest.resources.Recording attribute), 69  
 date\_created (twilio.rest.resources.Sandbox attribute), 70  
 date\_created (twilio.rest.resources.ShortCode attribute), 71  
 date\_created (twilio.rest.resources.SmsMessage attribute), 72  
 date\_created (twilio.rest.resources.task\_router.Task attribute), 80  
 date\_created (twilio.rest.resources.task\_router.Worker attribute), 77  
 date\_created (twilio.rest.resources.task\_router.Workflow attribute), 76  
 date\_created (twilio.rest.resources.task\_router.Workspace attribute), 75  
 date\_created (twilio.rest.resources.Transcription attribute), 73  
 date\_enqueued (twilio.rest.resources.Member attribute), 68  
 date\_sent (twilio.rest.resources.SmsMessage attribute), 72  
 date\_status\_changed (twilio.rest.resources.task\_router.Worker attribute), 78

- date\_updated (twilio.rest.resources.Account attribute), 50
  - date\_updated (twilio.rest.resources.Application attribute), 54
  - date\_updated (twilio.rest.resources.Call attribute), 57
  - date\_updated (twilio.rest.resources CallerId attribute), 59
  - date\_updated (twilio.rest.resources.Conference attribute), 60
  - date\_updated (twilio.rest.resources.ConnectApp attribute), 61
  - date\_updated (twilio.rest.resources.Notification attribute), 62
  - date\_updated (twilio.rest.resources.Participant attribute), 63
  - date\_updated (twilio.rest.resources.PhoneNumber attribute), 65
  - date\_updated (twilio.rest.resources.Recording attribute), 69
  - date\_updated (twilio.rest.resources.Sandbox attribute), 70
  - date\_updated (twilio.rest.resources.ShortCode attribute), 71
  - date\_updated (twilio.rest.resources.SmsMessage attribute), 72
  - date\_updated (twilio.rest.resources.task\_router.Task attribute), 80
  - date\_updated (twilio.rest.resources.task\_router.Worker attribute), 77
  - date\_updated (twilio.rest.resources.task\_router.Workflow attribute), 76
  - date\_updated (twilio.rest.resources.task\_router.Workspace attribute), 75
  - date\_updated (twilio.rest.resources.Transcription attribute), 73
  - deauthorize\_callback\_method (twilio.rest.resources.ConnectApp attribute), 61
  - deauthorize\_callback\_url (twilio.rest.resources.ConnectApp attribute), 61
  - default\_activity\_name (twilio.rest.resources.task\_router.Workspace attribute), 75
  - default\_activity\_sid (twilio.rest.resources.task\_router.Workspace attribute), 74
  - delete() (twilio.rest.resources.Addresses method), 52
  - delete() (twilio.rest.resources.Application method), 55
  - delete() (twilio.rest.resources.Applications method), 53
  - delete() (twilio.rest.resources.Call method), 58
  - delete() (twilio.rest.resources CallerId method), 60
  - delete() (twilio.rest.resources CallerIds method), 59
  - delete() (twilio.rest.resources.Calls method), 56
  - delete() (twilio.rest.resources.Notification method), 63
  - delete() (twilio.rest.resources.Notifications method), 61
  - delete() (twilio.rest.resources.Participants method), 63
  - delete() (twilio.rest.resources.PhoneNumber method), 66
  - delete() (twilio.rest.resources.PhoneNumbers method), 64
  - delete() (twilio.rest.resources.Queue method), 68
  - delete() (twilio.rest.resources.Queues method), 67
  - delete() (twilio.rest.resources.Recording method), 70
  - delete() (twilio.rest.resources.Recordings method), 69
  - delete() (twilio.rest.resources.task\_router.Activities method), 76
  - delete() (twilio.rest.resources.task\_router.TaskQueues method), 78
  - delete() (twilio.rest.resources.task\_router.Tasks method), 79
  - delete() (twilio.rest.resources.task\_router.Workers method), 77
  - delete() (twilio.rest.resources.task\_router.Workflows method), 75
  - delete() (twilio.rest.resources.task\_router.Workspaces method), 74
  - delete() (twilio.rest.resources.Transcription method), 74
  - delete() (twilio.rest.resources.Transcriptions method), 73
  - delete\_instance() (twilio.rest.resources.Calls method), 56
  - dependent\_phone\_numbers() (twilio.rest.TwilioRestClient method), 48
  - DependentPhoneNumber (class in twilio.rest.resources), 52
  - DependentPhoneNumbers (class in twilio.rest.resources), 52
  - dequeue() (twilio.rest.resources.Members method), 68
  - description (twilio.rest.resources.ConnectApp attribute), 61
  - description (twilio.rest.resources.task\_router.Event attribute), 80
  - Dial (class in twilio.twiml), 82
  - dial() (twilio.twiml.Response method), 81
  - direction (twilio.rest.resources.Call attribute), 58
  - direction (twilio.rest.resources.SmsMessage attribute), 73
  - duration (twilio.rest.resources.Call attribute), 58
  - duration (twilio.rest.resources.Recording attribute), 69
  - duration (twilio.rest.resources.Transcription attribute), 74
- ## E
- enable\_conference\_on\_exit (twilio.rest.resources.Participant attribute), 64
  - end\_time (twilio.rest.resources.Call attribute), 58
  - Enqueue (class in twilio.twiml), 83
  - enqueue() (twilio.twiml.Response method), 81
  - error\_code (twilio.rest.resources.Notification attribute), 62
  - Event (class in twilio.rest.resources.task\_router), 80
  - event\_callback\_url (twilio.rest.resources.task\_router.Workspace attribute), 75
  - event\_data (twilio.rest.resources.task\_router.Event attribute), 81
  - event\_date (twilio.rest.resources.task\_router.Event attribute), 80

event\_type (twilio.rest.resources.task\_router.Event attribute), 80

Events (class in twilio.rest.resources.task\_router), 80

events() (twilio.rest.TwilioTaskRouterClient method), 48

## F

fallback\_assignment\_callback\_url (twilio.rest.resources.task\_router.Workflow attribute), 76

feedback() (twilio.rest.resources.Calls method), 56

feedback() (twilio.rest.TwilioRestClient method), 48

formats (twilio.rest.resources.Recording attribute), 69

forwarded\_from (twilio.rest.resources.Call attribute), 58

friendly\_name (twilio.rest.resources.Account attribute), 50

friendly\_name (twilio.rest.resources.Address attribute), 52

friendly\_name (twilio.rest.resources.Application attribute), 54

friendly\_name (twilio.rest.resources.AvailablePhoneNumber attribute), 66

friendly\_name (twilio.rest.resources.CallerId attribute), 59

friendly\_name (twilio.rest.resources.Conference attribute), 60

friendly\_name (twilio.rest.resources.ConnectApp attribute), 61

friendly\_name (twilio.rest.resources.PhoneNumber attribute), 65

friendly\_name (twilio.rest.resources.Queue attribute), 67

friendly\_name (twilio.rest.resources.ShortCode attribute), 71

friendly\_name (twilio.rest.resources.task\_router.TaskQueue attribute), 78

friendly\_name (twilio.rest.resources.task\_router.Worker attribute), 77

friendly\_name (twilio.rest.resources.task\_router.Workflow attribute), 76

friendly\_name (twilio.rest.resources.task\_router.Workspace attribute), 74

from (twilio.rest.resources.SmsMessage attribute), 73

from\_ (twilio.rest.resources.Call attribute), 58

## G

Gather (class in twilio.twiml), 82

gather() (twilio.twiml.Response method), 81

generate() (twilio.util.TwilioCapability method), 85

get() (twilio.rest.resources.Calls method), 56

get() (twilio.rest.resources.ListResource method), 49

get() (twilio.rest.resources.Sandboxes method), 70

get\_instance() (twilio.rest.resources.Calls method), 56

get\_instances() (twilio.rest.resources.Calls method), 56

get\_instances() (twilio.rest.resources.task\_router.Events method), 80

## H

Hangup (class in twilio.twiml), 83

hangup() (twilio.rest.resources.Call method), 58

hangup() (twilio.rest.resources.Calls method), 56

hangup() (twilio.twiml.Response method), 81

homepage\_url (twilio.rest.resources.ConnectApp attribute), 61

## I

instance (twilio.rest.resources.SmsMessages attribute), 72

InstanceResource (class in twilio.rest.resources), 50

iso\_country (twilio.rest.resources.Address attribute), 52

iso\_country (twilio.rest.resources.AvailablePhoneNumber attribute), 67

iter() (twilio.rest.resources.Calls method), 57

iter() (twilio.rest.resources.ListResource method), 49

iter() (twilio.rest.resources.Recordings method), 69

## K

kick() (twilio.rest.resources.Participant method), 64

kick() (twilio.rest.resources.Participants method), 63

## L

lata (twilio.rest.resources.AvailablePhoneNumber attribute), 66

latitude (twilio.rest.resources.AvailablePhoneNumber attribute), 66

Leave (class in twilio.twiml), 83

leave() (twilio.twiml.Response method), 81

list() (twilio.rest.resources.Accounts method), 50

list() (twilio.rest.resources.Applications method), 53

list() (twilio.rest.resources.CallerIds method), 59

list() (twilio.rest.resources.Calls method), 57

list() (twilio.rest.resources.Conferences method), 60

list() (twilio.rest.resources.ConnectApps method), 61

list() (twilio.rest.resources.Members method), 68

list() (twilio.rest.resources.Notifications method), 61

list() (twilio.rest.resources.Participants method), 63

list() (twilio.rest.resources.PhoneNumbers method), 64

list() (twilio.rest.resources.Queues method), 67

list() (twilio.rest.resources.Recordings method), 69

list() (twilio.rest.resources.ShortCodes method), 70

list() (twilio.rest.resources.SmsMessages method), 72

list() (twilio.rest.resources.task\_router.Events method), 80

list() (twilio.rest.resources.Transcriptions method), 73

ListResource (class in twilio.rest.resources), 49

load() (twilio.rest.resources.PhoneNumber method), 66

log (twilio.rest.resources.Notification attribute), 62

longitude (twilio.rest.resources.AvailablePhoneNumber attribute), 66

## M

max\_size (twilio.rest.resources.Queue attribute), 67  
 Member (class in twilio.rest.resources), 68  
 Members (class in twilio.rest.resources), 68  
 members() (twilio.rest.TwilioRestClient method), 48  
 Message (class in twilio.twiml), 83  
 message() (twilio.twiml.Response method), 81  
 message\_date (twilio.rest.resources.Notification attribute), 62  
 message\_text (twilio.rest.resources.Notification attribute), 62  
 more\_info (twilio.rest.resources.Notification attribute), 62  
 mute() (twilio.rest.resources.Participant method), 64  
 mute() (twilio.rest.resources.Participants method), 63  
 muted (twilio.rest.resources.Participant attribute), 64

## N

Notification (class in twilio.rest.resources), 62  
 Notifications (class in twilio.rest.resources), 61  
 Number (class in twilio.twiml), 84

## P

parent\_call\_sid (twilio.rest.resources.Call attribute), 57  
 Participant (class in twilio.rest.resources), 63  
 Participants (class in twilio.rest.resources), 63  
 participants (twilio.rest.resources.Conference attribute), 60  
 participants() (twilio.rest.TwilioRestClient method), 48  
 Pause (class in twilio.twiml), 83  
 pause() (twilio.twiml.Response method), 81  
 permissions (twilio.rest.resources.ConnectApp attribute), 61  
 phone\_number (twilio.rest.resources.AvailablePhoneNumber attribute), 66  
 phone\_number (twilio.rest.resources.CallerId attribute), 59  
 phone\_number (twilio.rest.resources.PhoneNumber attribute), 65  
 phone\_number (twilio.rest.resources.Sandbox attribute), 70  
 phone\_number\_sid (twilio.rest.resources.Call attribute), 58  
 PhoneNumber (class in twilio.rest.resources), 65  
 PhoneNumbers (class in twilio.rest.resources), 64  
 pin (twilio.rest.resources.Sandbox attribute), 70  
 Play (class in twilio.twiml), 82  
 play() (twilio.twiml.Response method), 81  
 position (twilio.rest.resources.Member attribute), 68  
 postal\_code (twilio.rest.resources.Address attribute), 52  
 postal\_code (twilio.rest.resources.AvailablePhoneNumber attribute), 66  
 price (twilio.rest.resources.Call attribute), 58  
 price (twilio.rest.resources.SmsMessage attribute), 73  
 price (twilio.rest.resources.Transcription attribute), 74  
 priority (twilio.rest.resources.task\_router.Task attribute), 79  
 purchase() (twilio.rest.resources.PhoneNumbers method), 64

## Q

Queue (class in twilio.rest.resources), 67  
 Queue (class in twilio.twiml), 84  
 queue\_members (twilio.rest.resources.Queue attribute), 68  
 Queues (class in twilio.rest.resources), 67

## R

rate\_center (twilio.rest.resources.AvailablePhoneNumber attribute), 66  
 reason (twilio.rest.resources.task\_router.Task attribute), 79  
 Record (class in twilio.twiml), 82  
 record() (twilio.twiml.Response method), 81  
 Recording (class in twilio.rest.resources), 69  
 recording\_sid (twilio.rest.resources.Transcription attribute), 73  
 Recordings (class in twilio.rest.resources), 69  
 Redirect (class in twilio.twiml), 83  
 redirect() (twilio.twiml.Response method), 81  
 region (twilio.rest.resources.Address attribute), 52  
 region (twilio.rest.resources.AvailablePhoneNumber attribute), 66  
 Reject (class in twilio.twiml), 83  
 reject() (twilio.twiml.Response method), 81  
 request() (twilio.rest.resources.Calls method), 57  
 request() (twilio.rest.TwilioRestClient method), 48  
 request() (twilio.rest.TwilioTaskRouterClient method), 48  
 request\_method (twilio.rest.resources.Notification attribute), 62  
 request\_url (twilio.rest.resources.Notification attribute), 62  
 request\_variables (twilio.rest.resources.Notification attribute), 62  
 RequestValidator (class in twilio.util), 84  
 reservation\_activity\_sid (twilio.rest.resources.task\_router.TaskQueue attribute), 78  
 reservations() (twilio.rest.TwilioTaskRouterClient method), 49  
 resource\_sid (twilio.rest.resources.task\_router.Event attribute), 80  
 resource\_type (twilio.rest.resources.task\_router.Event attribute), 80  
 Response (class in twilio.twiml), 81  
 response\_body (twilio.rest.resources.Notification attribute), 63

response\_headers (twilio.rest.resources.Notification attribute), 63  
 route() (twilio.rest.resources.Call method), 58  
 route() (twilio.rest.resources.Calls method), 57

## S

Sandbox (class in twilio.rest.resources), 70  
 Sandboxes (class in twilio.rest.resources), 70  
 Say (class in twilio.twiml), 81  
 say() (twilio.twiml.Response method), 81  
 search() (twilio.rest.resources.PhoneNumbers method), 64  
 short\_code (twilio.rest.resources.ShortCode attribute), 71  
 ShortCode (class in twilio.rest.resources), 71  
 ShortCodes (class in twilio.rest.resources), 70  
 sid (twilio.rest.resources.Account attribute), 50  
 sid (twilio.rest.resources.Application attribute), 54  
 sid (twilio.rest.resources.Call attribute), 57  
 sid (twilio.rest.resources.CallerId attribute), 59  
 sid (twilio.rest.resources.Conference attribute), 60  
 sid (twilio.rest.resources.ConnectApp attribute), 61  
 sid (twilio.rest.resources.Notification attribute), 62  
 sid (twilio.rest.resources.PhoneNumber attribute), 65  
 sid (twilio.rest.resources.Queue attribute), 67  
 sid (twilio.rest.resources.Recording attribute), 69  
 sid (twilio.rest.resources.ShortCode attribute), 71  
 sid (twilio.rest.resources.SmsMessage attribute), 72  
 sid (twilio.rest.resources.task\_router.Task attribute), 79  
 sid (twilio.rest.resources.task\_router.TaskQueue attribute), 78  
 sid (twilio.rest.resources.task\_router.Worker attribute), 77  
 sid (twilio.rest.resources.task\_router.Workflow attribute), 75  
 sid (twilio.rest.resources.task\_router.Workspace attribute), 74  
 sid (twilio.rest.resources.Transcription attribute), 73  
 sms() (twilio.twiml.Response method), 81  
 sms\_fallback\_method (twilio.rest.resources.Application attribute), 54  
 sms\_fallback\_method (twilio.rest.resources.PhoneNumber attribute), 66  
 sms\_fallback\_method (twilio.rest.resources.ShortCode attribute), 71  
 sms\_fallback\_url (twilio.rest.resources.Application attribute), 54  
 sms\_fallback\_url (twilio.rest.resources.PhoneNumber attribute), 66  
 sms\_fallback\_url (twilio.rest.resources.ShortCode attribute), 71  
 sms\_method (twilio.rest.resources.Application attribute), 54  
 sms\_method (twilio.rest.resources.PhoneNumber attribute), 66  
 sms\_method (twilio.rest.resources.Sandbox attribute), 70

sms\_method (twilio.rest.resources.ShortCode attribute), 71  
 sms\_status\_callback (twilio.rest.resources.Application attribute), 54  
 sms\_url (twilio.rest.resources.Application attribute), 54  
 sms\_url (twilio.rest.resources.PhoneNumber attribute), 66  
 sms\_url (twilio.rest.resources.Sandbox attribute), 70  
 sms\_url (twilio.rest.resources.ShortCode attribute), 71  
 SmsMessage (class in twilio.rest.resources), 72  
 SmsMessages (class in twilio.rest.resources), 72  
 start\_conference\_on\_enter (twilio.rest.resources.Participant attribute), 64  
 start\_time (twilio.rest.resources.Call attribute), 58  
 status (twilio.rest.resources.Account attribute), 51  
 status (twilio.rest.resources.Application attribute), 54  
 status (twilio.rest.resources.Call attribute), 58  
 status (twilio.rest.resources.Conference attribute), 60  
 status (twilio.rest.resources.SmsMessage attribute), 73  
 status (twilio.rest.resources.Transcription attribute), 73  
 status\_callback (twilio.rest.resources.Application attribute), 54  
 status\_callback (twilio.rest.resources.PhoneNumber attribute), 65  
 status\_callback\_method (twilio.rest.resources.Application attribute), 54  
 status\_callback\_method (twilio.rest.resources.PhoneNumber attribute), 66  
 street (twilio.rest.resources.Address attribute), 52  
 subresource\_uris (twilio.rest.resources.Recording attribute), 69  
 suspend() (twilio.rest.resources.Account method), 51  
 suspend() (twilio.rest.resources.Accounts method), 50  
 SUSPENDED (twilio.rest.resources.Account attribute), 51

## T

target\_workers (twilio.rest.resources.task\_router.TaskQueue attribute), 78  
 Task (class in twilio.rest.resources.task\_router), 79  
 task\_queue\_sid (twilio.rest.resources.task\_router.Task attribute), 79  
 task\_queues() (twilio.rest.TwilioTaskRouterClient method), 49  
 task\_reservation\_timeout (twilio.rest.resources.task\_router.Workflow attribute), 76  
 TaskQueue (class in twilio.rest.resources.task\_router), 78  
 TaskQueues (class in twilio.rest.resources.task\_router), 78  
 Tasks (class in twilio.rest.resources.task\_router), 79  
 tasks() (twilio.rest.TwilioTaskRouterClient method), 49

- timeout\_activity\_name (twilio.rest.resources.task\_router.Workspace attribute), 75  
 timeout\_activity\_sid (twilio.rest.resources.task\_router.Workspace attribute), 75  
 to (twilio.rest.resources.Call attribute), 57  
 to (twilio.rest.resources.SmsMessage attribute), 73  
 Transcription (class in twilio.rest.resources), 73  
 transcription\_text (twilio.rest.resources.Transcription attribute), 74  
 Transcriptions (class in twilio.rest.resources), 73  
 transfer() (twilio.rest.resources.PhoneNumber method), 66  
 transfer() (twilio.rest.resources.PhoneNumbers method), 65  
 twilio (module), 47  
 twilio.rest (module), 5, 14, 47  
 twilio.rest.resources (module), 9, 12, 16, 17, 19–21, 23, 24, 49  
 twilio.rest.resources.sms\_messages (module), 7  
 twilio.rest.resources.task\_router (module), 27, 74  
 twilio.task\_router (module), 39  
 twilio.twiml (module), 40, 81  
 twilio.util (module), 42, 43, 84  
 TwilioCapability (class in twilio.util), 85  
 TwilioRestClient (class in twilio.rest), 47  
 TwilioRestException (class in twilio), 47  
 TwilioTaskRouterClient (class in twilio.rest), 48
- ## U
- unmute() (twilio.rest.resources.Participant method), 64  
 unmute() (twilio.rest.resources.Participants method), 63  
 update() (twilio.rest.resources.Account method), 51  
 update() (twilio.rest.resources.Accounts method), 50  
 update() (twilio.rest.resources.Address method), 52  
 update() (twilio.rest.resources.Addresses method), 52  
 update() (twilio.rest.resources.Application method), 55  
 update() (twilio.rest.resources.Applications method), 53  
 update() (twilio.rest.resources CallerId method), 60  
 update() (twilio.rest.resources CallerIds method), 59  
 update() (twilio.rest.resources.Participants method), 63  
 update() (twilio.rest.resources.PhoneNumber method), 66  
 update() (twilio.rest.resources.PhoneNumbers method), 65  
 update() (twilio.rest.resources.Queue method), 68  
 update() (twilio.rest.resources.Queues method), 67  
 update() (twilio.rest.resources.Sandbox method), 70  
 update() (twilio.rest.resources.Sandboxes method), 70  
 update() (twilio.rest.resources.ShortCodes method), 71  
 update() (twilio.rest.resources.task\_router.Activities method), 76  
 update() (twilio.rest.resources.task\_router.TaskQueues method), 78  
 update() (twilio.rest.resources.task\_router.Tasks method), 79
- update() (twilio.rest.resources.task\_router.Workspace method), 77  
 update() (twilio.rest.resources.task\_router.Workflows method), 75  
 update() (twilio.rest.resources.task\_router.Workspaces method), 74  
 update\_instance() (twilio.rest.resources.Calls method), 57  
 uri (twilio.rest.resources.Application attribute), 55  
 uri (twilio.rest.resources CallerId attribute), 60  
 uri (twilio.rest.resources.Conference attribute), 60  
 uri (twilio.rest.resources.ConnectApp attribute), 61  
 uri (twilio.rest.resources.Member attribute), 68  
 uri (twilio.rest.resources.Notification attribute), 63  
 uri (twilio.rest.resources.Participant attribute), 64  
 uri (twilio.rest.resources.PhoneNumber attribute), 66  
 uri (twilio.rest.resources.Queue attribute), 68  
 uri (twilio.rest.resources.Recording attribute), 69  
 uri (twilio.rest.resources.Sandbox attribute), 70  
 uri (twilio.rest.resources.ShortCode attribute), 71  
 uri (twilio.rest.resources.SmsMessage attribute), 73  
 uri (twilio.rest.resources.Transcription attribute), 74
- ## V
- validate() (twilio.rest.resources CallerIds method), 59  
 validate() (twilio.util.RequestValidator method), 84  
 voice\_caller\_id\_lookup (twilio.rest.resources.Application attribute), 54  
 voice\_caller\_id\_lookup (twilio.rest.resources.PhoneNumber attribute), 65  
 voice\_fallback\_method (twilio.rest.resources.Application attribute), 54  
 voice\_fallback\_method (twilio.rest.resources.PhoneNumber attribute), 65  
 voice\_fallback\_url (twilio.rest.resources.Application attribute), 54  
 voice\_fallback\_url (twilio.rest.resources.PhoneNumber attribute), 65  
 voice\_method (twilio.rest.resources.Application attribute), 54  
 voice\_method (twilio.rest.resources.PhoneNumber attribute), 65  
 voice\_method (twilio.rest.resources.Sandbox attribute), 70  
 voice\_url (twilio.rest.resources.Application attribute), 54  
 voice\_url (twilio.rest.resources.PhoneNumber attribute), 65  
 voice\_url (twilio.rest.resources.Sandbox attribute), 70
- ## W
- wait\_time (twilio.rest.resources.Member attribute), 68  
 Worker (class in twilio.rest.resources.task\_router), 77  
 worker\_reservations() (twilio.rest.TwilioTaskRouterClient method), 49  
 Workers (class in twilio.rest.resources.task\_router), 76

workers() (twilio.rest.TwilioTaskRouterClient method),  
49

Workflow (class in twilio.rest.resources.task\_router), 75

workflow\_sid (twilio.rest.resources.task\_router.Task attribute), 79

Workflows (class in twilio.rest.resources.task\_router), 75

workflows() (twilio.rest.TwilioTaskRouterClient method), 49

Workspace (class in twilio.rest.resources.task\_router), 74

workspace\_sid (twilio.rest.resources.task\_router.Task attribute), 79

workspace\_sid (twilio.rest.resources.task\_router.TaskQueue attribute), 78

workspace\_sid (twilio.rest.resources.task\_router.Worker attribute), 77

workspace\_sid (twilio.rest.resources.task\_router.Workflow attribute), 76

Workspaces (class in twilio.rest.resources.task\_router),  
74