

# Twitter Assignment

July 2, 2018

You will now develop a way of controlling Cozmo by sending tweets. For this you will use the framework you have downloaded from github ([https://github.com/HWURoboticsLab/Leaps\\_material](https://github.com/HWURoboticsLab/Leaps_material)). What I want to do is to send messages to the robot, which will drive to robot towards a cube, lift the cube and bring the cube towards you.

First you will have to register your twitter app. To do this you will need a twitter account. If you do not have one please create it by going to [twitter.com](https://twitter.com). After you have created your twitter account please go to <https://apps.twitter.com/app/new> and click sign in (located in the top right corner).

Follow the instructions on the screen and click "create your twitter application" (see figure 1).

After this you will see a success screen with an API key on it. Select "Keys and Access Tokens" tab (see figure 2) and click "Generate an Access Token and Secret" (see figure 3). Copy your consumer key + secret, and access token + secret key (see figure 4) to `cozmo_twitter_keys` (located in `code/SDK_examples/lib/cozmo_twitter_keys.py`) and replace the XXXX with your keys.

Look at the advanced tutorial (<https://github.com/anki/cozmo-python-sdk/blob/master/docs/source/tutorial-advanced.rst>) for information on what you can send to the cozmo. Look at the `tweet_at_cozmo` app ([https://github.com/anki/cozmo-python-sdk/blob/master/examples/apps/tweet\\_at\\_cozmo.py](https://github.com/anki/cozmo-python-sdk/blob/master/examples/apps/tweet_at_cozmo.py)) to see how you can implement these things.

Recap:

- Create a Twitter account if you have not got one already
- Connect the robot to twitter by requesting an API key
- Move the robot around by sending messages to twitter by sending `do_drive` and `do_turn` tweets and implementing the code

The image shows a web browser window with the URL `https://apps.twitter.com/app/new`. The page title is "Create an application | Twitter". The browser's address bar shows the URL and a "Secure" icon. The page content includes a header with the Twitter logo and "Application Management". The main section is titled "Create an application". Below this title, there are two main sections: "Application Details" and "Developer Agreement".

**Application Details**

**Name \***

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

**Description \***

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

**Website \***

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens. (If you don't have a URL yet, just put a placeholder here but remember to change it later.)

**Callback URLs**

Where should we return after successfully authenticating? OAuth 1.0a applications must explicitly specify their `oauth_callback` URL(s) here, as well as include the one of the URLs below in the request token step. To restrict your application from using callbacks, leave this field blank.

**Developer Agreement**

☐ Yes, I have read and agree to the [Twitter Developer Agreement](#).

**Buttons:**

- Add a Callback URL** (link)
- Create your Twitter application** (button)

**Annotations:**

- A red arrow points from the text "Go to this site" to the browser's address bar.
- A red arrow points from the text "Fill in everything with a \*" to the "Name" input field.
- A red arrow points from the text "Check" to the "Developer Agreement" checkbox.
- A red arrow points from the text "Click" to the "Create your Twitter application" button.

Figure 1: Fill in the boxes

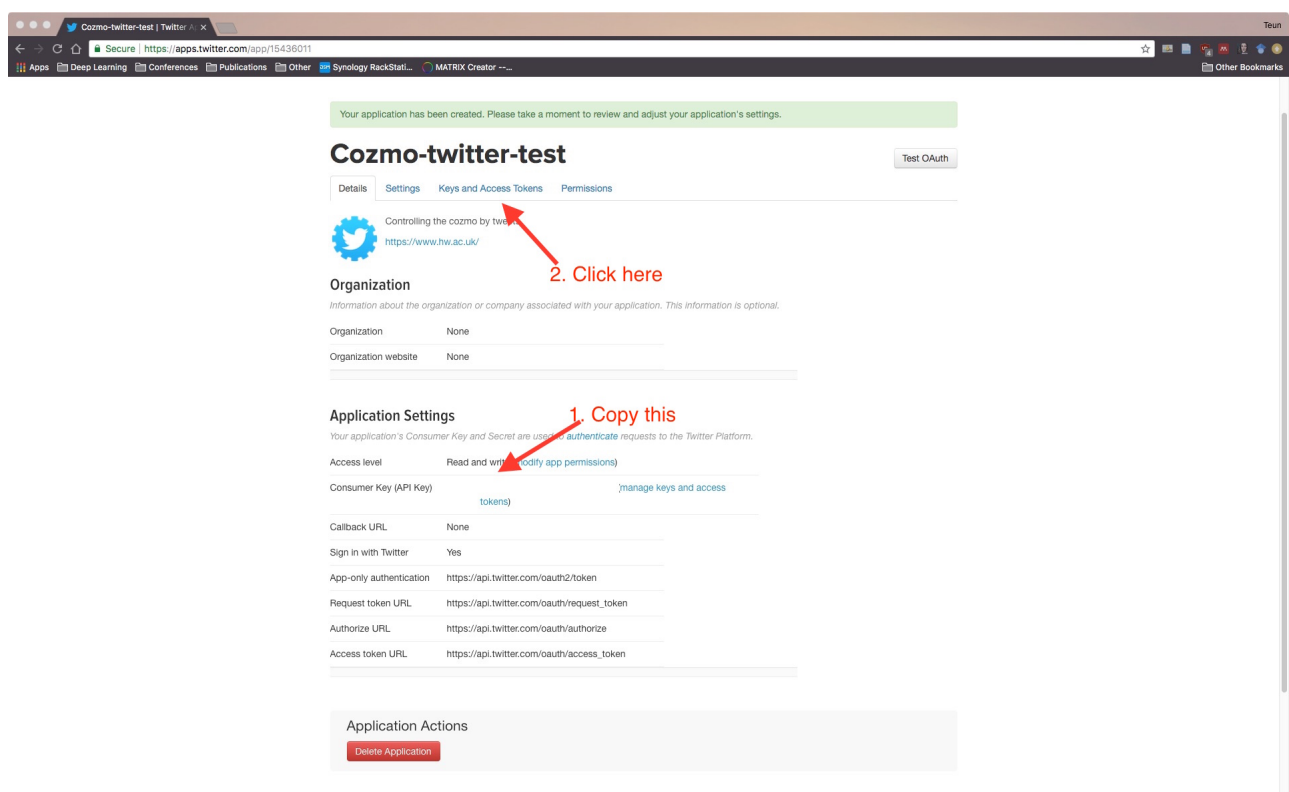


Figure 2: After a success you will see this screen

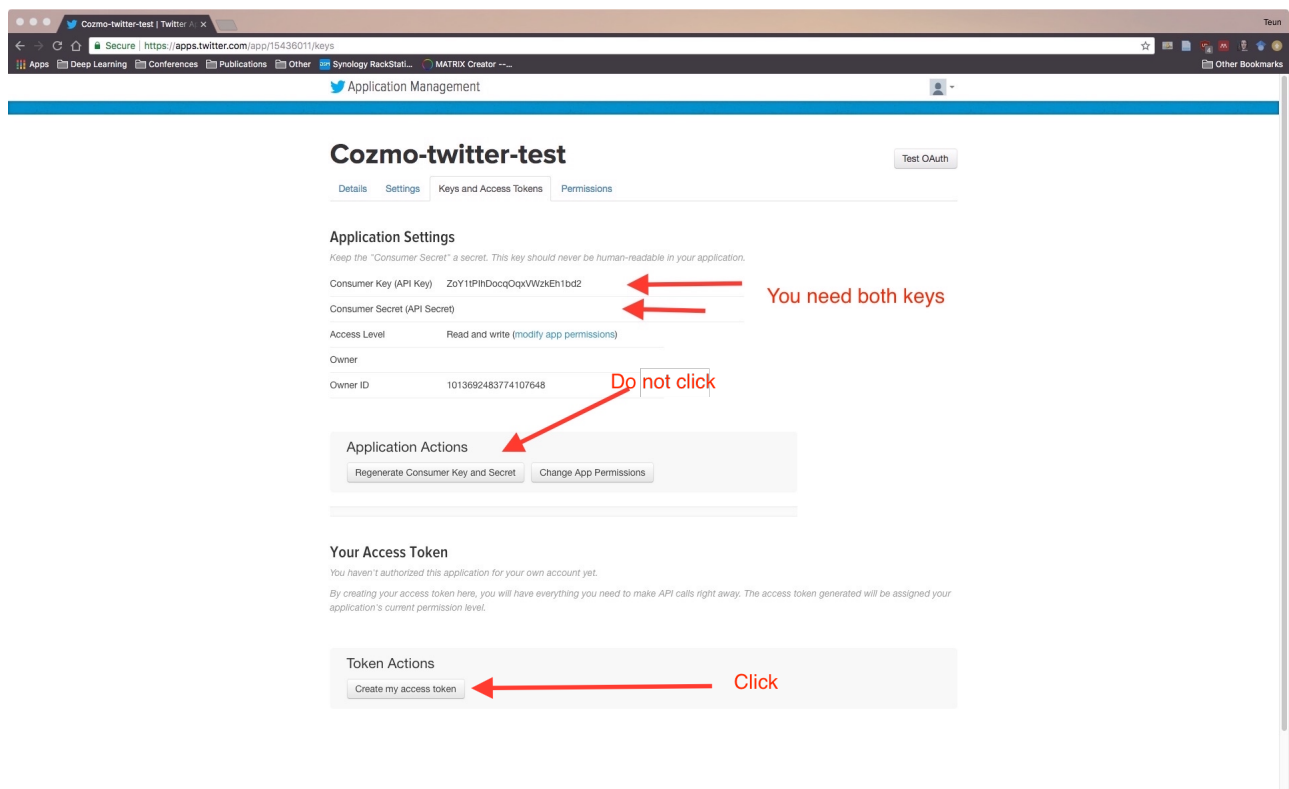


Figure 3: First click the bottom button

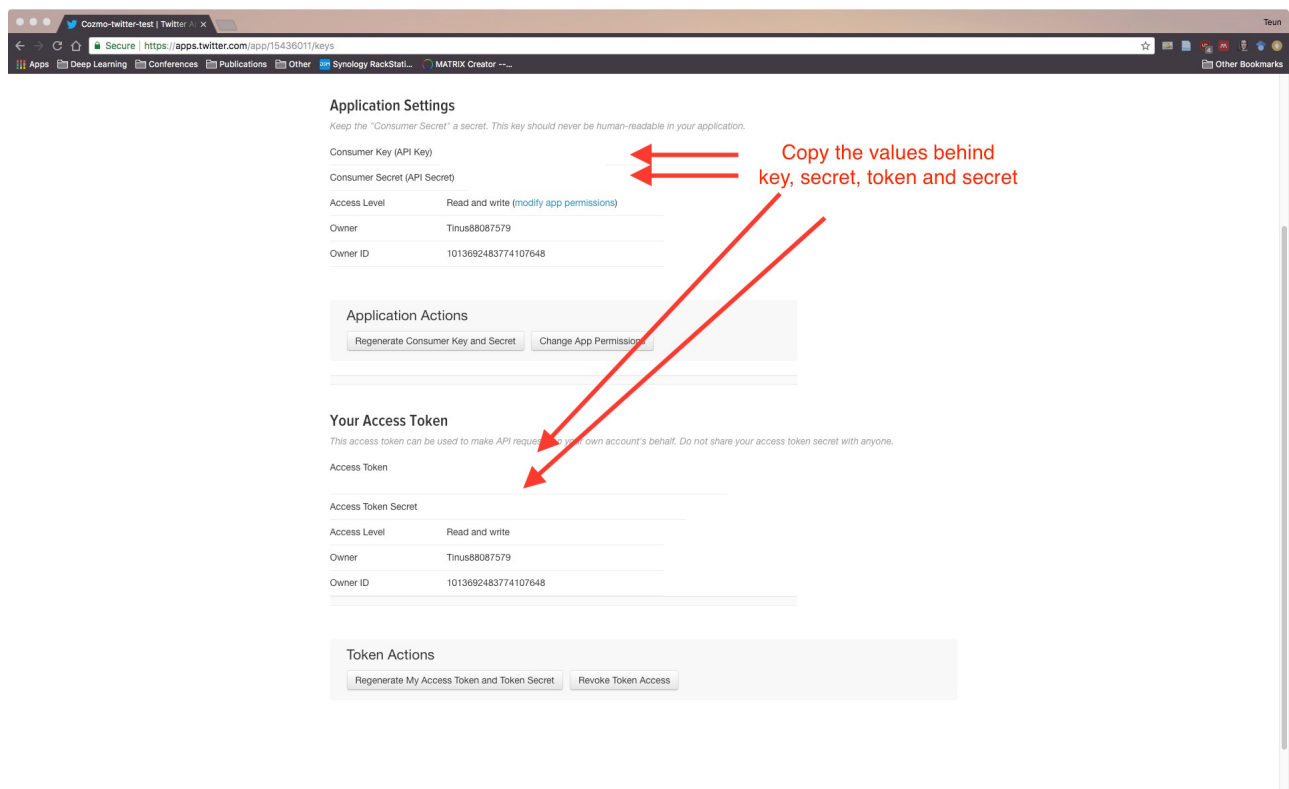


Figure 4: Copy the api and consumer keys

- Pickup a cube using twitter by sending do\_lift tweet and implementing the code
- Make the robot move towards the cube, pick it up and bring it to you.
- Look in code-SDK\_examples-docs-source-tutorial-advanced.rst for ways to use the twitter API
- Use (import) the classes in code-SDK\_examples-lib to connect to twitter and to receive the messages

An example of your class is given below:

```
import cozmo
from cozmo.util import degrees
import twitter_helpers
import cozmo_twitter_keys as twitter_keys
class ReactToTweetsStreamListener \
(twitter_helpers.CozmoTweetStreamListener):
    '''React to Tweets sent to our Cozmo, live, as they happen...'''

    def __init__(self, coz, twitter_api):
        super().__init__(coz, twitter_api)

    # Useful during development - an easy way to delete all of Cozmo's tweets
    # def do_deleteall(self, cmd_args, kw_args):
    #     cozmo.logger.info('Deleting all of your tweets')
    #     twitter_helpers.delete_all_tweets(self.twitter_api)
    #     return None

    def do_drive(self, cmd_args, kw_args):
        """drive X"""
        usage = "'drive X' where X is number of seconds to drive for"
        error_message = ""

        drive_duration = extract_float(cmd_args)

        if drive_duration is not None:
            drive_speed = 50
            drive_dir = "forwards"
            if drive_duration < 0:
                drive_speed = -drive_speed
                drive_duration = -drive_duration
                drive_dir = "backwards"

            self.cozmo.drive_wheels \
            (drive_speed, drive_speed, duration=drive_duration)
            return "I drove " + drive_dir + " for " +
            str(drive_duration) + " seconds!"
```

```
return "Error: _usage_=" + usage + error_message
```