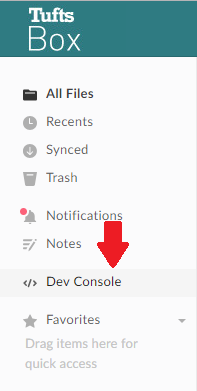
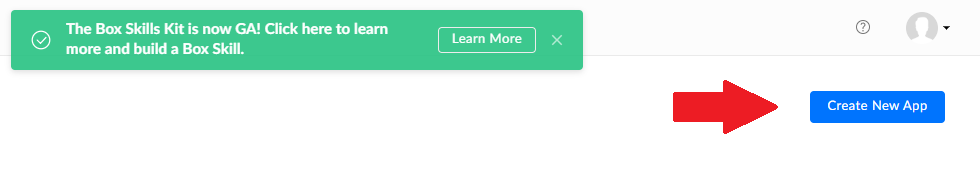
**Connecting to Box via the API**

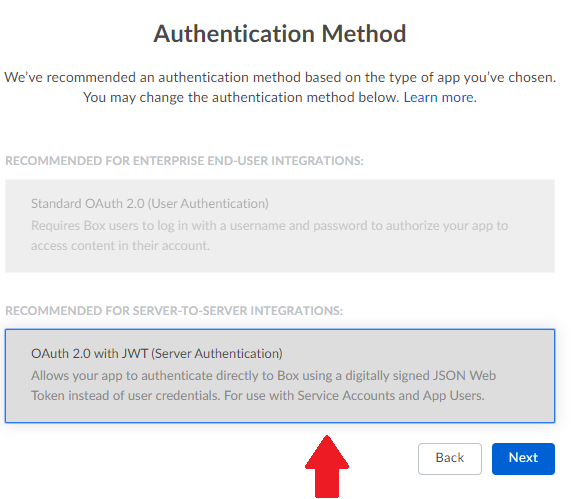
**Purpose:**

Create a connection to the Box API so you can read, write, and manage files in Box folders programmatically.

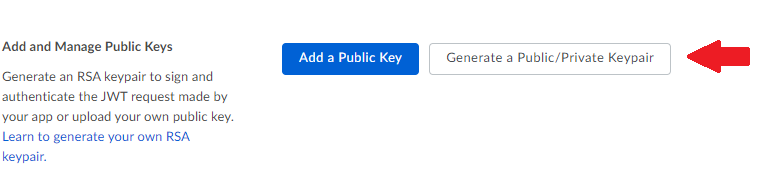
**Method:**

**Create app in the Box dev console:**

* + In your own institutional Box account, go to the Dev Console
    - ****
  + Create a new a new app in the Box developer’s console (upper right)
    - ****
  + Choose “Enterprise” and click Next
  + Choose the **OAuth with JWT** (JSON Web Tokens) authentication method. This will allow the app to access Box via the API, programmatically.

****

* + - Save the Bearer authentication token and set it aside if you want, although most likely you won’t need this
  + Set up the authentication settings in the app’s configuration screen
    - See the screen capture in the Appendix for the settings you’ll need
    - The goal with these settings is to allow access that will let the app write and read as another user, and use public and private keys to authenticate
  + Generate public/private key pair
    - Using the button in the “Add and Manage Public Keys” section

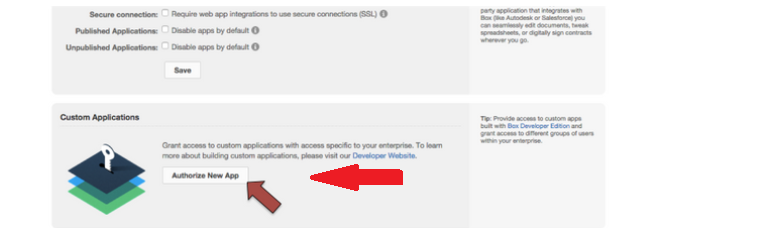
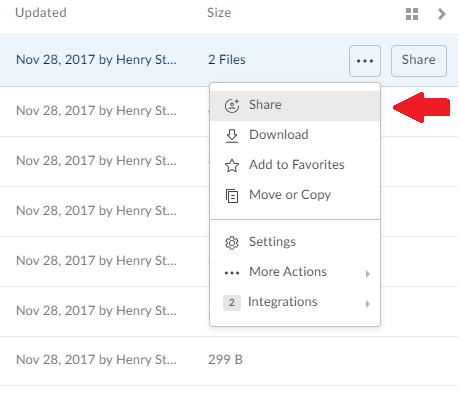
****

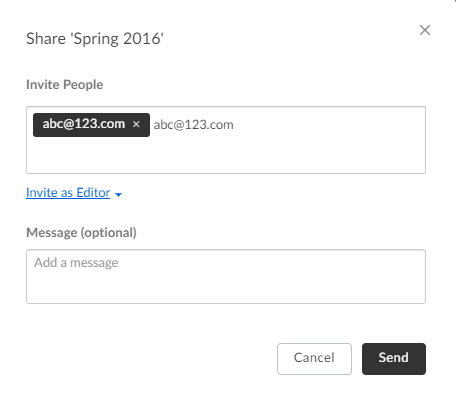
* + - **Download the JSON for this keypair and save it in some place safe and secure!**

This is the only way you’ll be able to authenticate via this app, and if you lose it you’ll have to generate a new pair

* + - The JSON that contains this keypair is how you’re going to authenticate.
    - You’ll be providing a path to this file as an argument to the command when you run the scripts below, so putting it in the same directory as the scripts is a good idea.

**Have your institutional Box admin approve this app**

* + Ask whoever is your Box admin in your institution to approve the app
  + They will see a screen something like this
    - ****
* **Create a user with Python on behalf of the app**
  + **See example Python script at <link>**
  + modify this script for your institution, particularly you’ll need to put the config file in the same directory as the script on your machine (preferably) so you can specify the path as an argument
  + You also need to provide a name for this user in the place noted in the script
  + run the following command from within the directory you download from github. This will install the necessary packages (pandas and numpy)
    - pip install –r requirements.txt
  + create a new non-human user on behalf of the developers app to read and write to Box with the connectToBox-GetLogin.py script. The one argument to this script is the path to the config json file you generated in the Box dev console in your app.
    - python connectToBox-GetLogin.py <path to config file>
  + This is a non-human user, but you’ll grant it editor permissions on the folder you want to write to just like you would a human user, using the user’s email that you’ll receive when you create the user
  + Once you have the new app user’s email that you generated in the script above, add it as an editor in the Box app
    - 

****

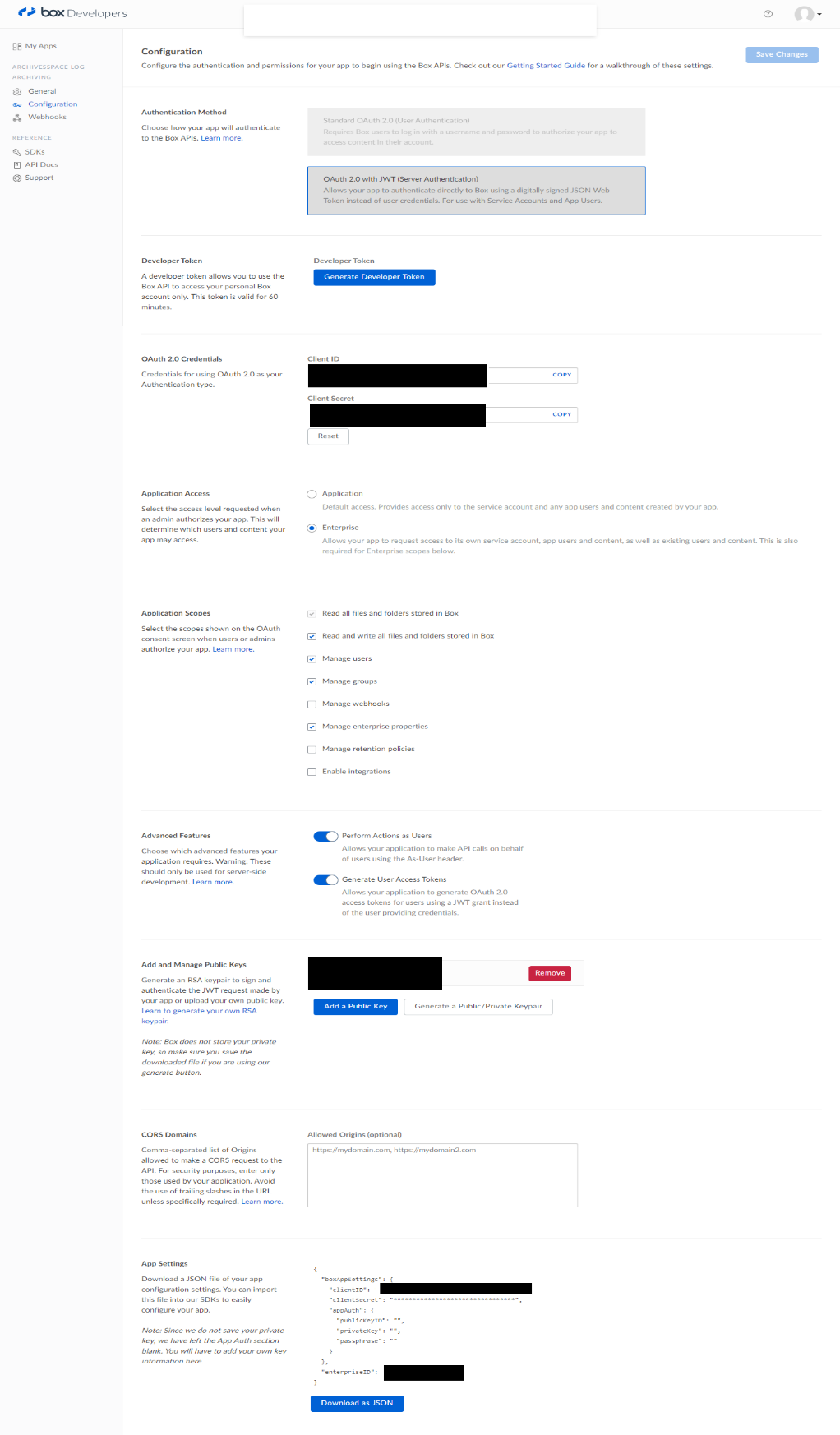
**Authenticate as the new app user, and read and write to Box**

* + The second Python script authenticates with the JSON public private key for your Box app, and becomes the new app user
  + **See example Python script at <link>**
  + Dependencies are already installed, so you can just run this script, after you edit it for your institution:
    - Enter the user ID from the process above in the place noted in the script
    - enter the folder ID of the folder you want to write to in the place noted in the script
  + This script takes 2 arguments:
    - 1: The filename of the input file you want to write to Box
    - The path to the config file you generated in the Box dev console within your app
  + Read and write to the specified folder, or perform other Box API functions as described at <https://developer.box.com/reference>

**References:**

* <https://developer.box.com/reference>
* **Github page**

**Appendix:**

****