PyCirrus

Everything you need
to know about PyCirrus

By Sebastian Guerrero

Table Of Contents

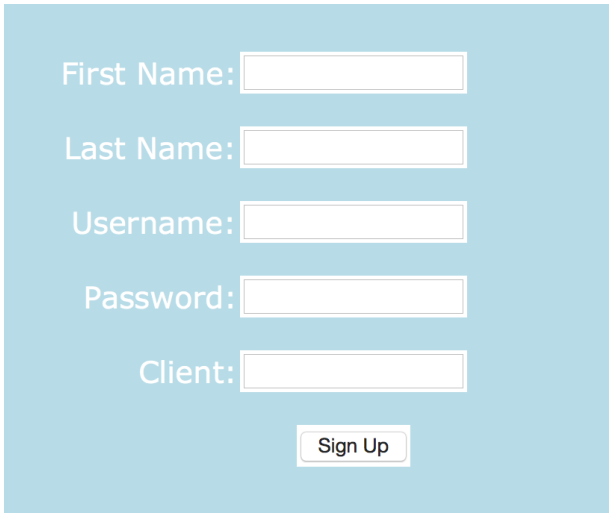
Intro

Hello, and welcome to PyCirrus! The way this application works is via your own Dropbox application it helps store Database Information. PyCirrus is made of two parts. PyCirrus the interface that lets you see and manipulate your data, and Pycirrus2 the module that allows you to build awesome applications by letting you create and manage your data via cloud. All of this with your data safely stored in your own Dropbox.

The way the application works is it uses both python 2.7 and Dropbox python API to manage files you create in a CSV format so that all your applications can access them. Before getting into how the module or the API works we need to learn about the interface and how to login.

New User

Before beginning to use the application you must first make a login which consists of multiple requirements.



First Name= Your first name

Last name = Your last name

Username = a unque identifier that can consist of any string of letters or digits just no ";"

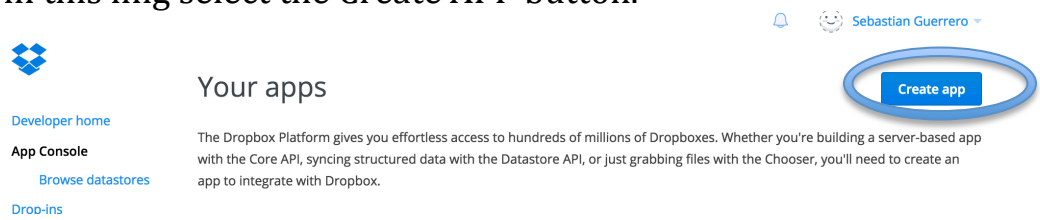
Password: Your password is important to login, and we store it as a hash so it is safely stored

Client= A string of letters that allows us and yourself to access your cloud

Getting Client

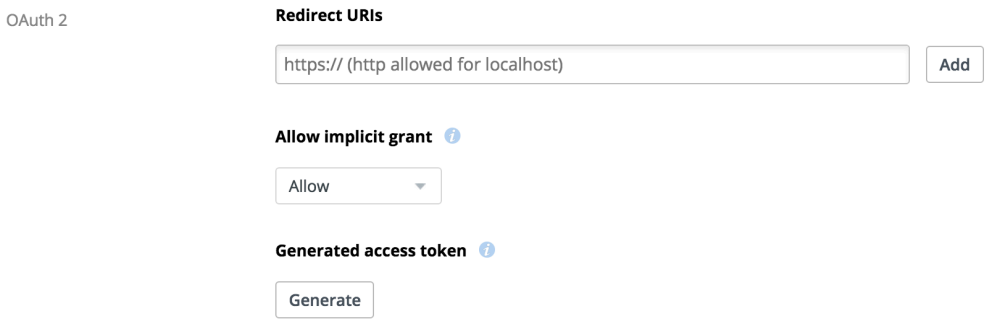
Getting a client string is really simple, especially if you are familiar with dropbox. Here we will list the steps on how to get this client id

1. If you do not have a client go to Dropbox website and make a login. It will ask if you want to download Dropbox in to your computer, it is not needed for the application to work. https://www.dropbox.com/login
2. Once you have account you have to create an application to create a Dropbox application https://www.dropbox.com/developers/apps
3. One in this ling select the Create APP button.



4. You will be taking to a screen with multiple option. Make sure you select the correct ones. First you have to choose **Dropbox API APP** this means that you will use the api in a program. For safety reasons in the next step and for increase in speed make the next option **Yes, My app will only access files it creates.** This guarantees that any api including the one you make cannot access things that already existed. Next name the application whatever you want, the name will not be used but for organization you should name it 'PyCirrusData'. Then click Confirm at you should be done
5. It will take you to a window with information about your application. There is a section called Oauth2, in there is an option to generate an access token. This is very important get the access

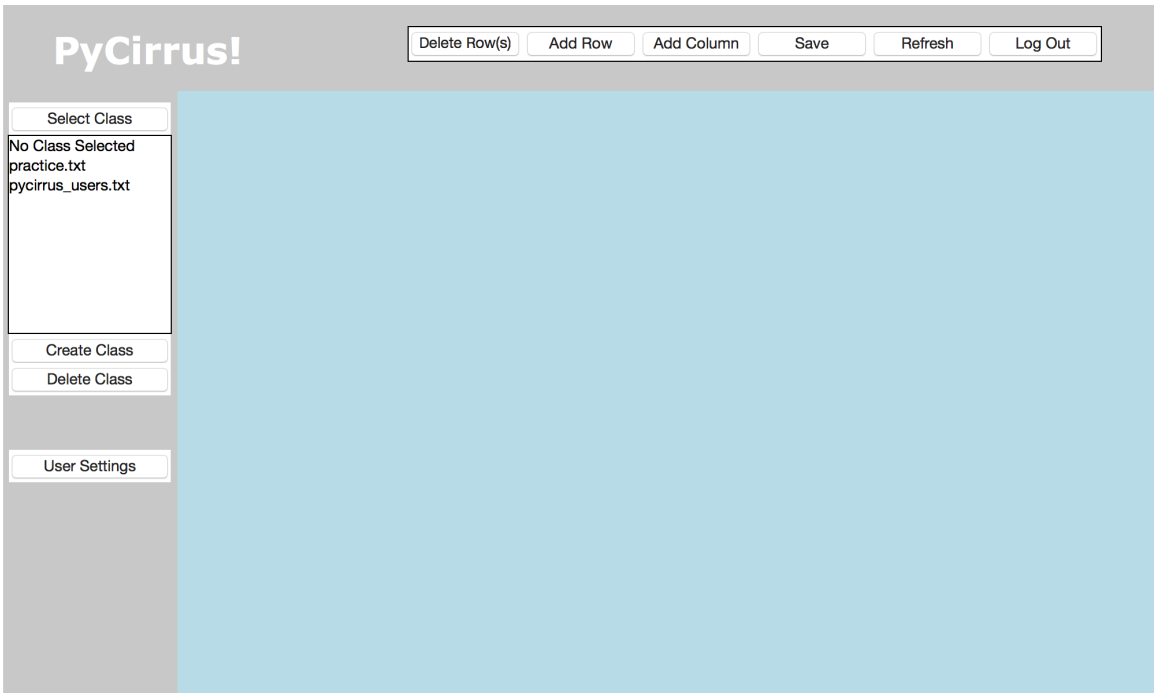
token and this is your token.



6. Now all you need to do is login using all the data nd your specific token. The token allows us to manage your data in the application and allows yourself with the use of Pycirrus2 to manage your data.

Buttons

The interface has several features and buttons that allows for you to manage your data.



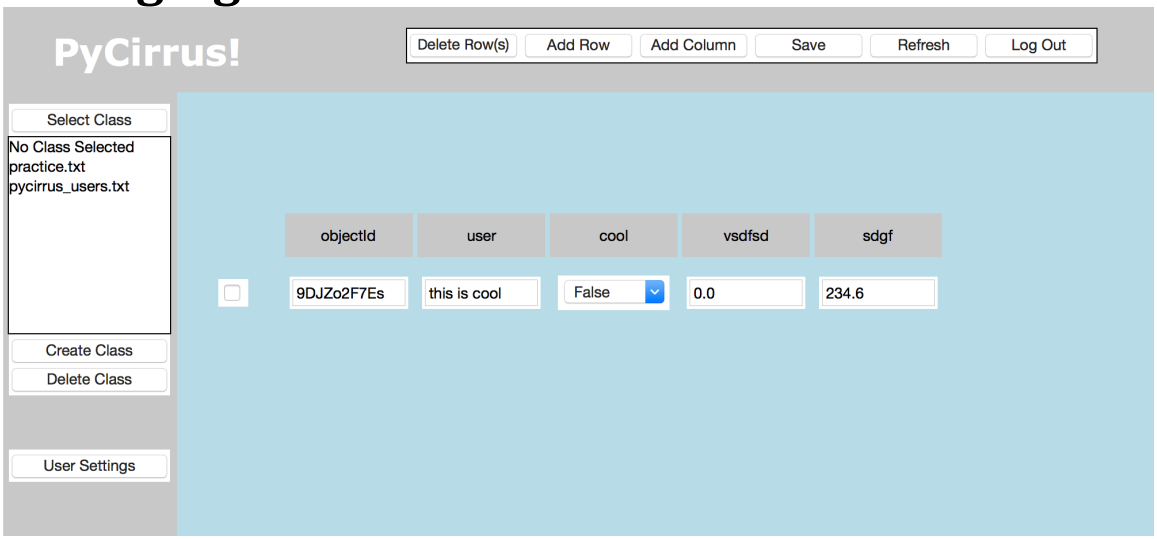
Create Class = This allows you to create a new data structure with a specific name. part of the requirements is that it must include '.txt' at the end and that the name does not include an under score '_'.

Delete Class= If no class is selected this will do nothing and ask that you do select one. Above the delete class we have a section full of classes that you as a user have created. Once a class is selected you can choose to delete it completely, from all existence. A dialog will come up letting you know if you are sure of your decision. If so the program will terminate but in the progress it will delete the class

Select Class= This allows you to select a class that is highlighted in order for you to be able to see it and manage it.

User Settings =This lets you see your info that you pu when making the account. Incase you want to change them you can do it via this application.

Managing Data



This is how data in your classes is displayed. There are multiple buttons to know so that you can manage it

Delete Row(s) = Next to each row of data there is a toggle button. If you select more than one and then press the deleted row(s) button these rows will be deleted from existence.

Add Row= This brings up a dialog box with inputs to add for each of the rows, at the end it will create that information. Something to notice is that you must put the correct type of data, weather it is int, float, Boolean, str, or list.

Add Column= It creates new column in the data and allows for the type of data you want.

Save= if you change any of the data in any of the fields, clicking safe will make sure those changes are saved

Refresh= refreshes the information and displays it again incase you want to undo changes or see if any changes were made

Log Out= logs the user out and takes the to the login screen.

PyCirrus2 Module

This part of the documentation is to demonstrate how to use the module.

To begin using the module you must first download the pycirrus2 file and import it to your project.

```
import pycirrus2
```

Before you begin using the functions you have to initiate the connection between your application and the dropboc api. This is called flow. Then you have to connect it with your dropbox account. This is where the client ID comes in. you can have multiple dropbox Clients.

```
import pycirrus2

flow = pycirrs2.flowS()
c = pycirrus2.clientS('Client ID')
```

Once you established your client, you can use this to extrapolate data.But, incase you want to create data straight from your code you need to use the create class function that has specific requirements. The input of columns is a list made of tuples including the name of the column and the type it should be. Remember no underscores or semi-colons

```
pycirrus2.createClass(c,"nameOfProject.txt",[('name',str)('newsletter',bool)('numbe
r',int)]
```

Once you have a class the way the module works is it creates text files that it downloads and upload to dropbox. Every class consists of two files one named access another without that. This is built for security so that no two users access the same data at the same time. Once you know this there are simple uses of your application. We will now look at on how to manage data with the code. To do this you must create access to a specific class.

```
Class1 = Class(client, className)
```

.getCSVLIST()= returns a list of all the data in the specified class

.saveUpdates(csv) = You must input the same data type that the class has and it will update it

.addRow(*inputs) = Add the things you want to add to each row, keeping in mind that they math the original type and it will safely save them on the cloud

.deleteRow(num) = deletes a specific row

.addCol(Column Names) = adds a column with the format of how you add a class in a tuple

.deleteME= Deletes the specific class you specfied