Python and Salesforce

simple-salesforce

https://github.com/simple-salesforce/simple-salesforce

oktana

About Me

K.B. Carte | https://kbcarte.com

Role: Developer
Hire Date: February 2020 *~6 months ago*
Experience:

- Web Development (PHP, WordPress, CSS, JS/jQuery, Python)
- Game Development (Blender Modeling and BGE, Godot, Adobe Flash)
- Hacking and Security (hackthebox.eu, hackaday.com)
- 3D Printing
- Robotics and Embedded Systems
  - Parallax BASIC Stamp
  - Arduino
  - AVR
  - RaspberryPi
  - BeagleBoard
  - ESP8266

Get to know Python

https://python.org

- Interpreted, Interactive, Object-Oriented, Scripting Language built on C
- Includes **1,731** libraries (Standard Library) Batteries Included!
- Package Management is really easy: **PIP** and **PIPEnv**
- Easy to learn and simple syntax
- MANY different ways to use it:
  - Web apps or API's (Django, Flask)
  - Salesforce (simple-salesforce)
  - Data Science (NumPy, SciPy, Pandas)
  - Visuals and Reporting (Matplotlib, Seaborn, Bokeh, Plotly)
  - Automation (Selenium, Robot Framework, Tavern)
  - Machine Learning / AI (TensorFlow, Keras, PyTorch, Scikit-learn)
  - Robotics and IoT (MicroPython, PySerial, PyVISA, PyFirmata)

Syntax Example

```
print("Hello World!")

x = 10
y = x+20

z = ["a", "b", "c"]
for i in z:
    print(i)

def my_function():
    my_var = "Hello"
    my_var2 = " World!"
    return my_var + my_var2

class my_class:
    def __init__(self):
        self.my_var = "Oktana"
    def my_method(self, my_arg):
        return my_arg + self.my_var + " class method!"
```

Note the use of indentation. Instead of curly braces “{}” like other languages, Python code blocks and scope is determined by the indentation levels.

You can’t mix tabs and spaces. One or the other, not both. Pep8 recommends 4 **spaces** per indentation.

Uses Dynamic Type Variables. No need to say what type the variable is, Python automatically converts the var to what ever is being assigned to it.

__init__(self) is how class constructors are created. The double underscore methods are special. Also called “magic methods” or “dunder methods”

simple-salesforce

https://github.com/simple-salesforce/simple-salesforce

"*Simple Salesforce is a basic Salesforce.com REST API client built for Python 3.3, 3.4, 3.5, and 3.6. The goal is to provide a very low-level interface to the REST Resource and APEX API, returning a dictionary of the API JSON response.*"

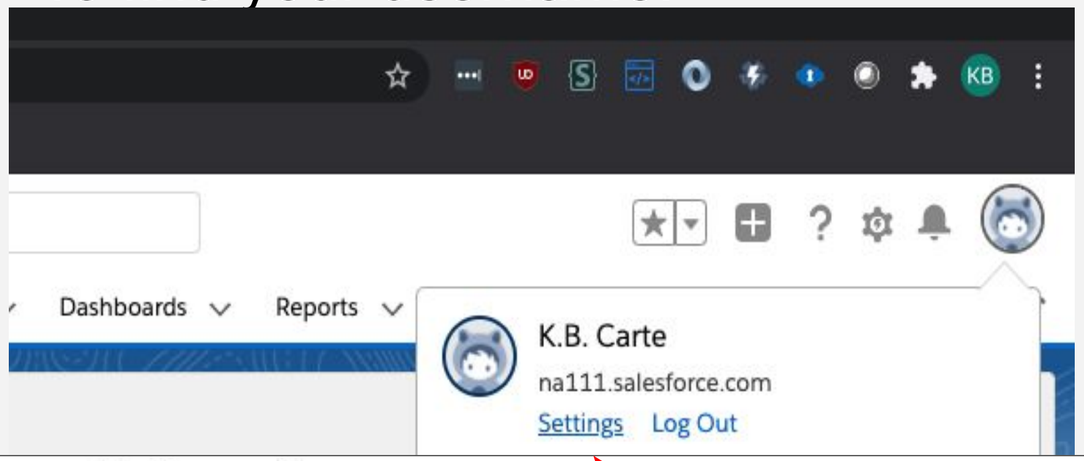
The lib makes it **very** easy to interact with our salesforce org. We only need three things to get started. Our username (email), password, and security token.

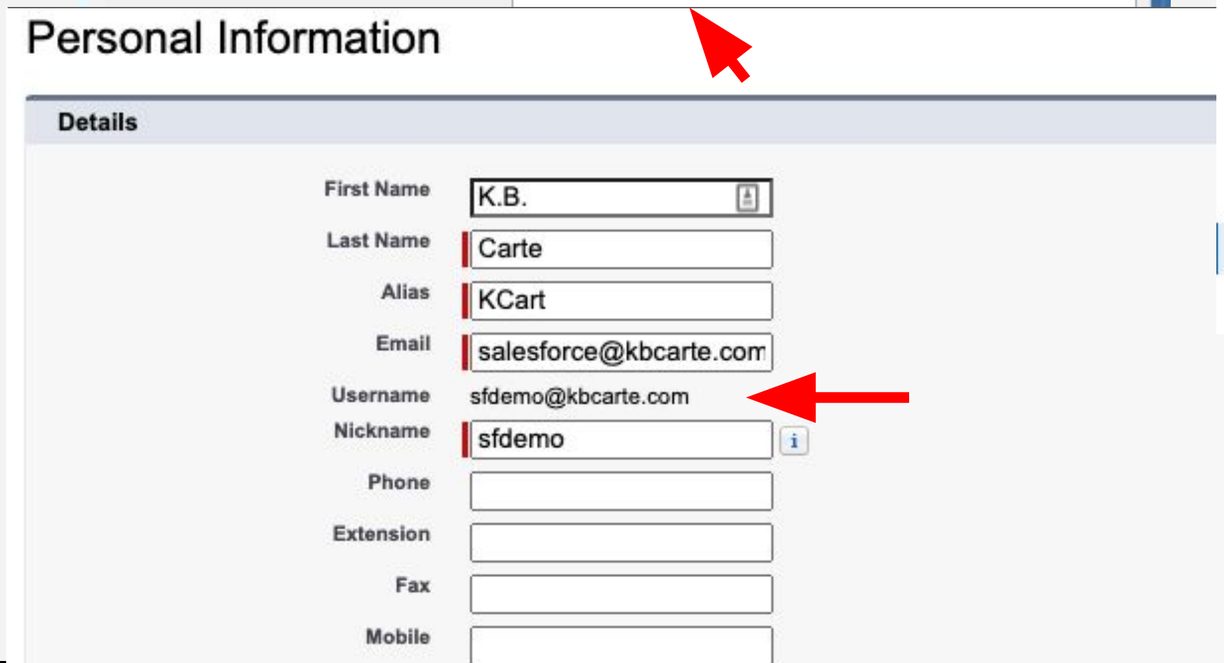
```
from simple_salesforce import Salesforce
sf = Salesforce(username='myemail@example.com',
                password='password',
                security_token='token')
```

Now we have a new object called "sf" that we can use to interact with our org. The object has a lot of methods to help with everything from SOQL queries, DML like insert and delete, even bulk API is included!

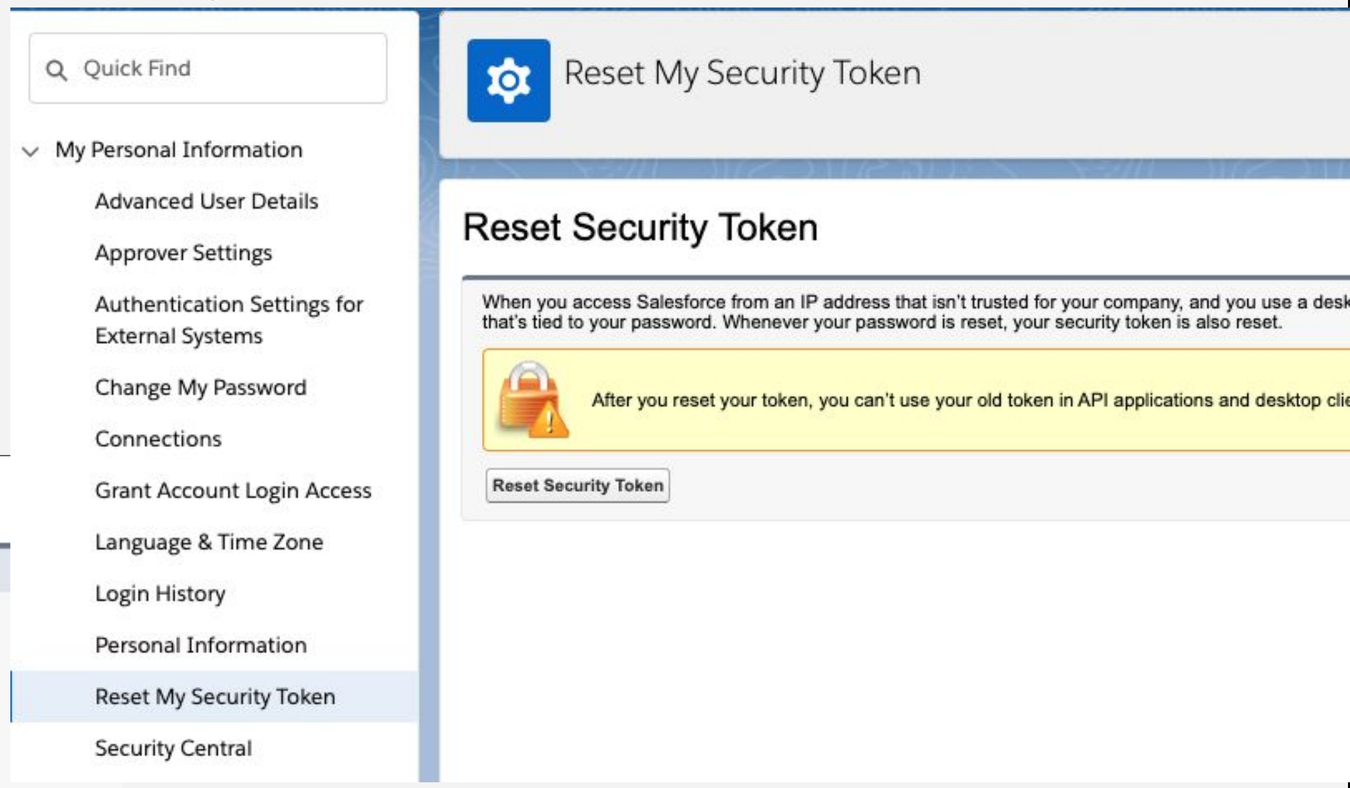
Login Information

To find your username:





Security Token:



Quick word on PIP

Pip is just a package manager that Python uses. It is just like any other manager such as npm, apt, pacman, ruby gems, php composer, etc…

The server/host pip uses is called **PyPi**. When we run the command `pip install <package_name>` or `pipenv install <package_name>` it is installed from the PyPi servers. This is more than okay for the most part and the vast majority of the time it's fine to have the lib pulled from PyPi. **BUT** there are times, like now, where the version on PyPi isn't the latest version. So there is a way to install libs and packages from Github directly.

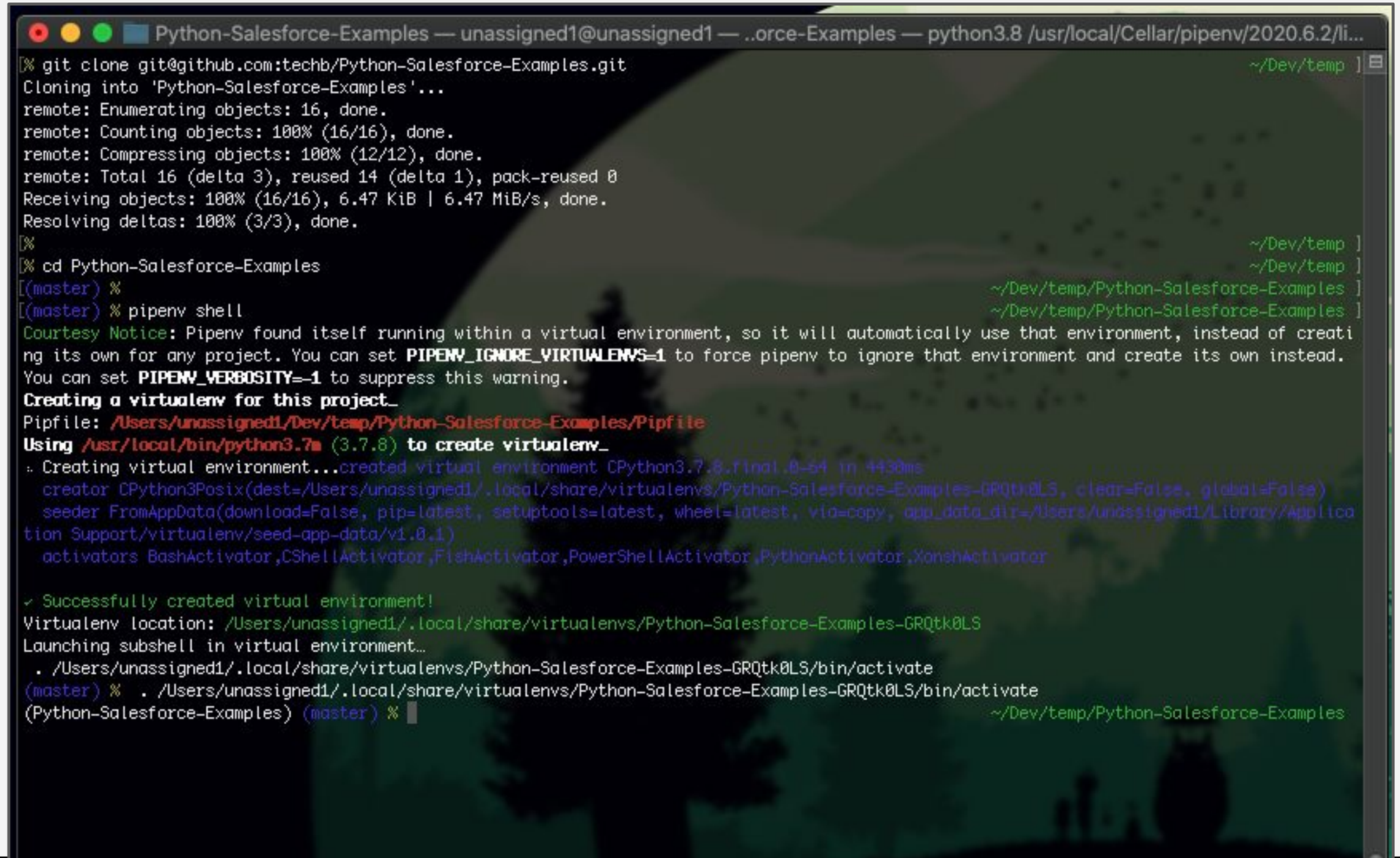
While working on this example, there was an update to simple-salesforce lib. Github has the current version and we need the format_soql method from it. **BUT** pip (PyPi) hasn't been updated with the new version as of yet, July 2 2020.

So, we'll need to install it via it's repo on Github.

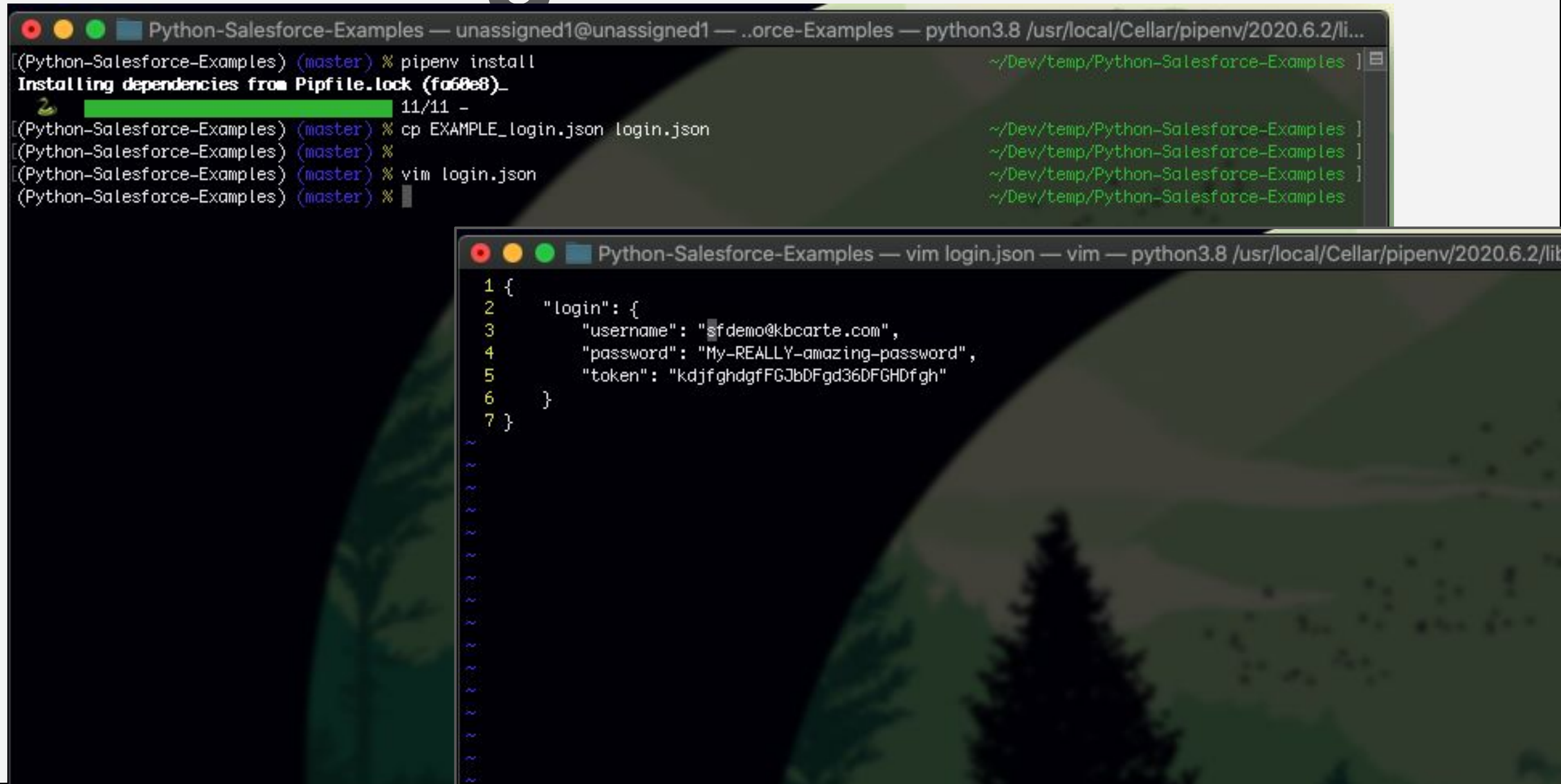
```
pipenv install -e
git+https://github.com/simple-salesforce/simple-salesforce.git#egg=simple-salesforce
```

If installing this demo from the repo. The `Pipfile` already contains the git repo as the lib source. Thanks to pipenv, it took care of automatically adding the required dependency.

Installing the Demo



Installing the Demo Cont...



Live Demo Time!

Q&A

Thank you!

oktana