Introduction to Earth Engine + Code Editor (JavaScript)

1. (Recommended) Sign up with gmail or personal email account (non-USGS) at <https://signup.earthengine.google.com/#!/> in order to access the Developer’s User Group (<https://groups.google.com/forum/#!forum/google-earth-engine-developers>)
   1. Other benefits: 1. Log-in on your home computer 2. Extra 250GB free storage!
2. Become familiar with the GEE API reference documentation: <https://developers.google.com/earth-engine/>
3. Check out the breakout sessions and training materials from [GEEUS 2016](http://earthenginesummit2016.earthoutreach.org/) and [GEEUS 2017](https://events.withgoogle.com/google-earth-engine-user-summit-2017/).
4. Training sessions YouTube playlist from GEEUS 2016: <https://www.youtube.com/playlist?list=PLWw80tqUZ5J9_3E_9C_bK8zt0mGHfvOrj>

And from GEEUS 2017: <https://www.youtube.com/playlist?list=PLWw80tqUZ5J_cclRAvZJXn4XMgj5x5QBI>

1. More useful videos and tutorials found here: <https://developers.google.com/earth-engine/tutorials>
   1. Recommend starting here: <https://developers.google.com/earth-engine/tutorial_js_01>
   2. Then here: <https://developers.google.com/earth-engine/tutorial_api_01>
2. Classroom intro labs and resources: <https://developers.google.com/earth-engine/edu>
3. Solid beginner and intermediate step-by-step resources: <https://developers.google.com/earth-engine/ttt>

Getting started with the GEE Python API

1. See minimal installation: <https://developers.google.com/earth-engine/python_install>
2. Or follow the install instructions below which seem to work at USGS EROS.

Installing the Earth Engine Python API using the conda packaging manager [WINDOWS 7 GUIDE ONLY]

Table of Contents:

Installing the conda Package Manager

Windows

Creating a conda Environment for the Earth Engine API

Authenticating to Earth Engine

Adding IPython Notebooks and SciPy to the conda Environment

# **Installing the conda Package Manager**

[**conda**](http://conda.pydata.org/docs/intro.html) is a cross-platform application for installing software packages. Although it is not specific to installing Python packages, it is being adopted by the scientific Python community for it's ability to handle Python packages that have dependencies on shared libraries. conda is open source, and is available as part of the [Anaconda](http://docs.continuum.io/anaconda/index.html) Python distribution, managed by Continuum Analytics.

The easiest way to install conda, is to download and install the Anaconda Python Distribution available. Binary installers are available for various operating systems, and specific instructions/notes for each operating system are listed below.

If you don't want to install the full Anaconda distribution (which is about 350MB), there are alternative methods of installing conda: you can install the minimal conda installation (a.k.a. [miniconda](http://conda.pydata.org/miniconda.html)), install the [conda package from PyPI](https://pypi.python.org/pypi/conda), or to download and install from the [source code](https://github.com/conda/conda).

## **Windows 7**

1. Go to<http://continuum.io/downloads> and download the Python 2.7 Graphical Installer for your platform (32 bit or 64 bit).

2. Execute the Graphical Installer package, and follow the instructions.

3. When asked where it should be installed, choose “Just me”. By default, Anaconda will be installed in C:\Users\<username>\AppData\Local\Continuum\Anaconda.

4. Go to your Start Menu > All Programs > Anaconda folder, and start the “Anaconda Command Prompt”.

5. List the system path

echo %PATH%

If the system path does not start with a path that includes the Anaconda installation location, see the troubleshooting section below.

https://github.com/ContinuumIO/anaconda-issues/issues/41

## **Updating conda**

1. Once you have installed conda, run the following to command to update conda to the most recent version.

conda update conda

To check which version of conda you are running, execute the following command:

conda –version

# **Creating a conda Environment for the Earth Engine API**

This section illustrates how to build a conda environment that allows you to run Python scripts against the Earth Engine API. Note that these instructions were written using conda version 3.5.5.

1. Create a new conda environment that includes Python 2.7.

ENV\_NAME=ee-python

conda create -n $ENV\_NAME python=2.7

2. A plan for downloading and linking packages will be presented, which you can review before proceeding. Once you proceed, a directory in your anaconda directory will be created.

3. Next activate this environment:

[Windows] activate $ENV\_NAME

Don't do this now, but once you are done using a conda environment, you can deactivate the environment by executing the following code:

[Windows] deactivate

4. Install cryptography (which includes cffi)

conda install -y cryptography

5. Install SimpleJSON

conda install -y simplejson

6. Install PIP into the conda environment, so that it can be used to install Python packages that are not available as conda packages, but are available in the Python Package Index ([PyPI](https://pypi.python.org/)).

conda install -y pip

7. Next install the Earth Engine Python API package

pip install earthengine-api

8. Test that you can successfully import the Earth Engine API package, by running:

python -c "import ee; print ee.\_\_version\_\_"

**Note for Windows users:** *Make sure that you execute this command in a directory that does not contains python.exe (such as the default Anaconda Command Prompt) , so that the python command correctly executes the python executable in the conda environment and has access to the Python packages that you have installed.*

## **Authenticating to Earth Engine**

<https://developers.google.com/earth-engine/command_line>

To setup your connection to the Earth Engine backend servers, set up your authentication credentials using the command line. (prompts user to authorize access to GEE via OAuth2.

earthengine authenticate

**Note:** Be sure to have a browser that enables cookies when performing the authentication steps here. (Chrome should work, IE admin settings might not?)

**Run a test Python Command for Earth Engine (@ USGS EROS)**

1. After setting up your authentication credentials, run the following command in your conda environment to see if your user account is configured correctly.

earthengine ls

SSL Workaround instructions...

At present (OCT-2017), the EROS firewall blocks the earthengine.google.com ('https://earthengine.googleapis.com/api') endpoint during SSL validation (EE stack uses httplib2 to handle authentication) so for now one workaround is to override the ssl check:

In ```data.py ~ line 713``` Override the SSL check (the quick and dirty):

**http = httplib2.Http(timeout=(\_deadline\_ms / 1000.0) or None,**

**disable\_ssl\_certificate\_validation=True)**

But ultimately we should fix (overwrite) the CA\_CERTS\_PATH, → i.e. be able to explicitly direct httplib2 when instantiating the http connection object with something like:

HTTPLIB\_CA\_CERTS\_PATH = os.environ.get('HTTPLIB\_CA\_CERTS\_PATH')

http = httplib2.Http(timeout=(\_deadline\_ms / 1000.0) or None,

ca\_certs=HTTPLIB\_CA\_CERTS\_PATH)

But this 2nd option does not seem to solve the problem at this time??

## **You are now ready to run the Google Earth Engine Python API**

**You can stop here!**

Otherwise…

The following contains additional information for installing IPython and SciPy, if interested.

## **Adding IPython Notebooks and SciPy to the conda Environment**

In many cases, additional tools may make working with the Earth Engine Python API easier. This section will walk through the installation of several useful tools:

●  [IPython Notebooks](http://ipython.org/notebook.html) - a browser-based interactive computing environment that can display various types of output

●  [SciPy](http://www.scipy.org/scipylib/index.html) library - a collection of efficient numerical routines for scientific analysis, and

●  [matplotlib](http://matplotlib.org/) - a 2D plotting library for Python

Instructions for installing in a conda environment:

1. Install the IPython Notebook, SciPy and matplotlib packages, using the conda package manager:

conda install ipython-notebook scipy matplotlib

2. Create a directory where the IPython Notebooks will live.

mkdir -p ~/temp/ipynb

cd ~/temp/ipynb

[Windows] mkdir %HOMEPATH%\temp\ipynb

[Windows] cd %HOMEPATH%\temp\ipynb

3. Download a basic notebook that accesses the Earth Engine Python API using the following command. (Note that despite the line wrapping, this is a single command.)

curl -O<https://raw.githubusercontent.com/tylere/IPython-Notebooks/master/ee-demos/basic_tests/Earth_Engine_API_basic_test.ipynb>

Curl is not available by default in Windows, so navigate in a browser to: https://github.com/tylere/IPython-Notebooks/blob/master/ee-demos/basic\_tests/Earth\_Engine\_API\_basic\_test.ipynb

and then save the "Raw" file link into the temporary directory you previously created..

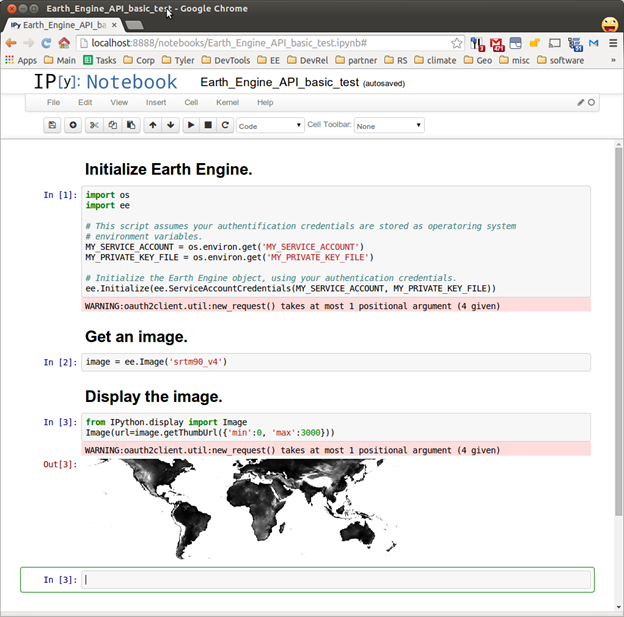
4. Start the IPython Notebook Server, which will open in a browser and display files and directories below the directory where you start the notebook:

ipython notebook

5. Click on the file Earth\_Engine\_API\_Basic\_Test.ipynb, which will open the notebook in a new tab.

6. Look through the commands, and click the play button (near the top-middle of the page) to run each of the notebook cells. Note that IPython Notebooks allow you to run arbitrary code, so do not run untrusted code without first reviewing it!!!

If everything is working correctly, you browser window should look similar to the following screenshot.



7. To stop the IPython Notebook server, hit Ctrl-c twice in the terminal where the server was started.

8. If you want to experiment with additional IPython Notebooks, here are a few others to try:

# IPython In Depth tutorial notebooks

curl -L https://github.com/ipython/ipython-in-depth/zipball/master -o ipython-in-depth.zip

unzip ipython-in-depth.zip

# Notebooks presented at SciPy 2014

curl -L<https://github.com/tylere/SciPy-2014/zipball/master> -o scipy-ee.zip

unzip scipy-ee.zip

# Exploratory Computing with Python

# http://mbakker7.github.io/exploratory\_computing\_with\_python/

curl -L<https://github.com/mbakker7/exploratory_computing_with_python/zipball/master> -o exploratory\_computing\_with\_python.zip

unzip exploratory\_computing\_with\_python.zip