|  |  |  |
| --- | --- | --- |
| **RM\_DSWE\_EE\_API.docx** | | |
| [Readme log file providing a template for other log file format and content. This file is placed in the root directory for the project (see data storage directories). For illustrative purposes, the sections were populated using actual notes from a previous project].  This file documents research into APIs and JSON, and Python scripts to download DSWE data using an API.  Grey Text indicates mistakes retained for future reference but not directly included in the current product. | | |
| **Overall Goal/This activity’s objective(s)** | | |
| [This is where the overall goal of the greater project, as well as the activity documented in this file are briefly described. Directly below I provide an example from a previously conducted project/activity. Replace this with your own].  The overall goal of the project is to derive DSWE products through temporal processing [more details to come later]. The objective for the activity documented here is to create scripts for downloading DSWE data using an API. | | |
| **Data storage directories** | | |
| [The directory structure for the project is carefully documented here. Replace this with your own directory structure – I’ll provide the top level directories to you].  **/DSWE\_EaD/API** This is the base directory (including this log file)  **/DSWE\_EaD/API/practice** This folder contains practice code and notes for me to learn working with APIs  **/DSWE\_EaD/API/code** This folder contains real scripts for getting data  **/DSWE\_EaD/API/notes** Contains text documents with notes that I’ve taken to help with understanding  **/DSWE\_EaD/API/other\_data** Contains scripts for accessing data from other sources | | |
| **File naming conventions** | | |
| [Provide a ‘key’ to file naming conventions here. This example is very detailed (Annie was wonderfully detail oriented). The key is that the reader can decipher your thinking when looking at lists of files.]  **tutorial\_[...].py** file contains notes and code written while following a tutorial | | |
| **Contributors** | | |
| [List everyone who has made edits to this file/participated in conversation within it].  Text colors in this log indicate the identity of the contributor.  Madeline Hunt  John Jones | | |
| **Processing [Here’s where the daily log of processing begins. Try to make this easily readable. Use screen shots to convey results. Add commentary/interpretations. I’ll insert questions and comments here as well. Put ‘dead ends’ in grey text, but don’t delete them. Put show stopping problems in yellow text. Put significant results in red text].** | | |
| Working with APIs: learning from tutorials | | |
| 13 July 2020 | | |
| Goals: I’m doing research into using APIs with python (starting out here <https://www.dataquest.io/blog/python-api-tutorial/>). I’ll work through this tutorial, and then look at using other APIs (for example, spotify or reddit). I’ll also start skimming through the EarthExplorer documentation to see similarities between it and the other APIs I play around with to understand better what I need to do. Then I will look up a short tutorial on working with JSON data. Finally, I’ll look at getting DSWE data using the EarthExplorer API. | | |
| *practice/ tutorial\_ISS.py* | Followed a short tutorial on getting data from APIs (<https://www.dataquest.io/blog/python-api-tutorial/>); learned how to use the *get* function and the basics behind what APIs are. Also learned basics of JSON (so might not need to do a JSON tutorial -- it seems pretty straightforward) | |
| *practice/ tutorial\_spotify.py* | Generally following a youtube video (<https://www.youtube.com/watch?v=xdq6Gz33khQ>) and tutorial (<https://medium.com/@maxtingle/getting-started-with-spotifys-api-spotipy-197c3dc6353b>) on using the Spotify API.  This is where I stopped for today (about ⅓ through the video tutorial, after figuring out how to get a token and get authenticated).  Note to self: the spotify API appears to be much more complicated than the EarthExplorer API, so just having learned the basics of authentication might be enough to start work with EarthExplorer tomorrow. | |
| Questions:   * Do I need to request access for EarthExplorer through EROS (<https://ers.cr.usgs.gov/profile/access>)? Do I need to fill this out by hand, or will my API request fill this out for me (is having an account sufficient, or do I need to fill out the form for every time I need to request data)? (I hope this question makes sense) * Will this request access give me a one-time token, or should filling this out programmatically be something I include in my Python scripts?   You need to create an account that will allow you additional capabilities for downloading data from EROS (and actually NASA land process related distributed archives). You can’t access the documentation on the EarthExplorer API without an account. This account won’t give you machine-to-machine access. We’ll have to investigate whether that is possible. But I already have it. So in the worst case scenario, I’ll use the code you write to access the data we need. Does that make sense? | | |
| 14 July 2020 | | |
| Goals:   * Read through all EarthExplorer documentation this morning and take notes on anything useful * Then, finish working through spotify API tutorial (<https://www.youtube.com/watch?v=xdq6Gz33khQ>) -- on second thought after yesterday, I think I will go ahead and finish the tutorial, because it looks like I’ll have to write my own scripts in the same level of detail as done in the video * Look through a few JSON tutorials to see if they give any new information (don’t spend more than an hour on this unless things get really confusing for some reason) * Finally, use code written for spotify tutorial and the examples given in Perl and PHP in the documentation to start work on the EE API (see next section)   (Note to self: look more into the USGS python package -- it might already have scripts in place for working with EE’s API, so you could use this package or modify some of the scripts) | | |
| *notes/ earthExplorer.txt* | Contains notes while reading the EarthExplorer documentation | |
| *practice/ tutorial\_spotify.py* | Finished the spotify tutorial and built a nice class that I can use as general code for working with the EE API. Also double checked code with code on his github. This was a really good tutorial for building an API client! It looks like there are 2 other videos about APIs in this series   * There’s code for a google maps client on the github that looks similar to what I went through and wrote for spotify -- maybe I can use this to help me with EE as well (<https://github.com/codingforentrepreneurs/30-Days-of-Python/blob/master/tutorial-reference/Day%2020/google_maps_client.ipynb>) | |
| *notes/ json.txt* | General notes on JSON and using the json package in python | |
| Making a client for the EarthExplorer (EE) API | | |
| *code/ client\_perlToPython.py* | Attempting to take the example client given in the EE documentation, see how it works, and convert it to python (<https://earthexplorer.usgs.gov/inventory/example/json-InventoryClient-pm>). So far, most of the smaller subroutines seem to be checking that inputs are in the correct format; the bulk of the code is in the \_callService subroutine, which looks like the requests.get() function in python plus checking status codes. This appears to be the most important piece of the perl example so far, so I am sticking with converting it to python first. This perl example seemed like a good place to start in making my API code, and I’ll be cross-referencing with the USGS package in python which looks like it does part of what I need as well (<http://kapadia.github.io/usgs/_modules/usgs/api.html>). | |
| *code/ earthExplorerAPI.py* | I found an example of the Earth Explorer API in python -- in the landsatxplore package (<https://github.com/yannforget/landsatxplore>). I am actually going to base my code off of what is in their github -- needless to say, it’s much easier to interpret than perl. The code looks like it was already translated straight from the perl example, so there’s not much new here and it looks similar to what I started to do above. I will abandon the above file for now and just work here. | |
| Summary and work for tomorrow: Aiming to finish bare-bones API code in earthExplorerAPI.py using the landsatxplore package for reference, and then I’ll spend tomorrow afternoon (hopefully) neatening the code up and running it on the test page in the documentation ([https://earthexplorer.usgs.gov/inventory/documentation/test](https://earthexplorer.usgs.gov/inventory/documentation/sample)).  Questions:   * None right now, other than just making sure that I am approaching this task in a productive/meaningful way | | |
| 15 July 2020 | | |
| Goals:   * Finish going through landsat API code (<https://github.com/yannforget/landsatxplore>) and build my own API script * Neaten up the code and test on the EE test page ([https://earthexplorer.usgs.gov/inventory/documentation/test](https://earthexplorer.usgs.gov/inventory/documentation/sample)) * Work on getting authentication code working -- see if I can request access to test on real data | | |
| *code/ earthExplorerAPI.py* | So far, just copying the landsatxplore code with minor changes to see if it works -- after I basically copy it, I will work on modifying it for this project (<https://github.com/yannforget/landsatxplore>)  This morning: got an error when running my code -- going to investigate it by installing the landsatxplore package to see if it will run with my credentials | |
| *code/ dataModels.py* | Writing code for the data models given in the documentation and the landsat package -- I’ll have to see if I need to add more based on what kinds of data we need (for now, it just has coordinates, spatial filter, and temporal filter). | |
| *practice/ tutorial\_landsat.py* | Downloaded the landsatxplore python package to test if it worked with my credentials, so I would have a baseline for how my API code would work. Typed in the simple example provided but got an error:  EarthExplorerError: EE: Application Offline Application Unavailable  I also am unable to access the documentation -- I think this explains the error I was having with my code earlier, but I can’t look at the documentation code or test my code in any way because the site is down. | |
| Looking at other sources of data | | |
| While the EarthExplorer website is down, I will look at how to get data from other sites. | | |
| *other\_data/ request\_prism\_data.py* | Python functions to download climate data from PRISM (<https://prism.oregonstate.edu/>); followed outline on <https://prism.oregonstate.edu/documents/PRISM_downloads_web_service.pdf> for instructions.  Can download zip file of BIL data (daily, monthly, annually, or normals) -- finished this script today | |
| *other\_data/ request\_drought\_data.py* | Python functions to download JSON data from the US Drought Monitor (<https://droughtmonitor.unl.edu/>) -- finished this script today | |
| Summary and work for tomorrow: wrote scripts to easily get data from the two websites Dr. Jones sent; finished “first draft” of API code, but couldn’t test as intended because the EarthExplorer website was down. Tomorrow, I will test my code on both the test site and try using real credentials, and finish up the EE API code.  Questions:   * None today! | | |
| 16 July 2020 | | |
| Goals: | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  |  |  |