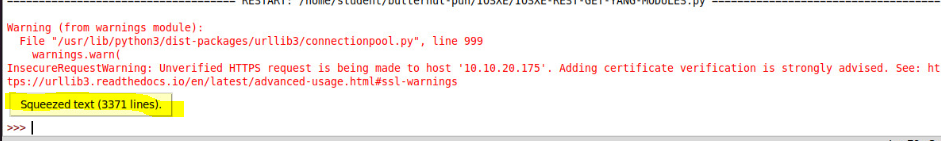
**IOS XE RESTCONF**

Directions: Complete each of the scripts below and submit them by the date specified in the assignment sheet and Blackboard. Submit **copies of your code** and **screenshots of the code running with each task**. Also, be sure to use the document, **Script Requirements as a guide** to writing good code. **Full credit will not be earned if you do not meet these script requirements.** **10 points**

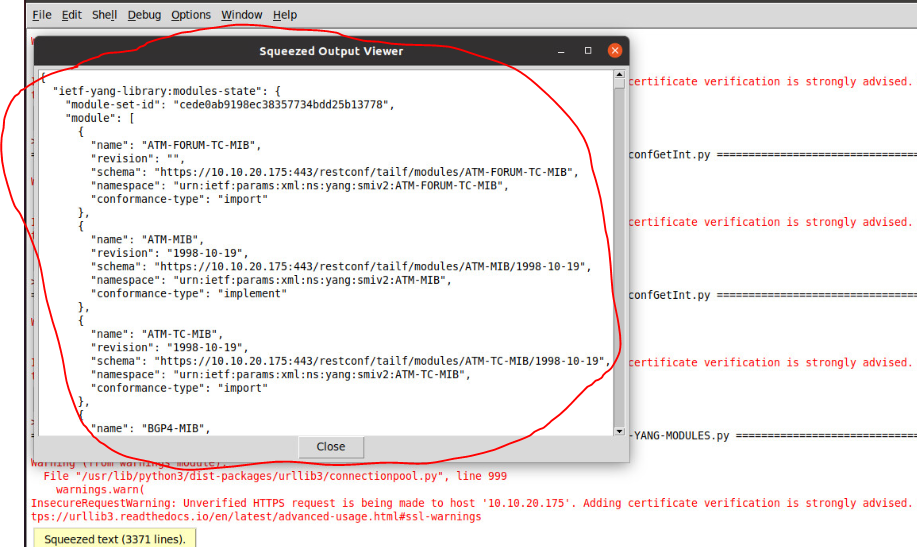
**ISO XE YANG Data Model**

YANG is a data model that we can use with RESTCONF and NETCONF to access our IOS XE devices. In this lab, we will be exploring RESTCONF for configuring and monitoring our devices. We are going to configure a simple setting on an interface. To do so, we will explore one of our YANG model’s schema.

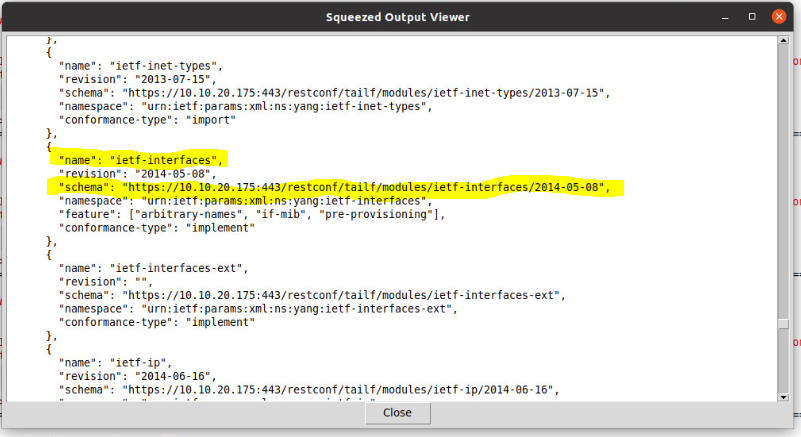
1. Set up IOS XE devices for RESTCONF by typing the following in global configuration on your IOS XE routers:
   1. *ip http server*
   2. *ip http secure-server*
   3. *restconf*
2. Update the Git, <https://github.com/dwright314/turnipTheBeet.git>
   1. In the git repository directory, type *git pull*
3. Open the script, IOSXE-REST-GET-YANG-MODULES.py
4. In this lab, we are going to run our first RESTCONF call on our IOSXE device and ask the device which YANG modules it supports.
5. Run the script. Note the script details and note that we are making a GET HTTP request to the RESTCONF API. Make sure the IP address in the URL is one of your IOSXE devices. The result will look something like the following:



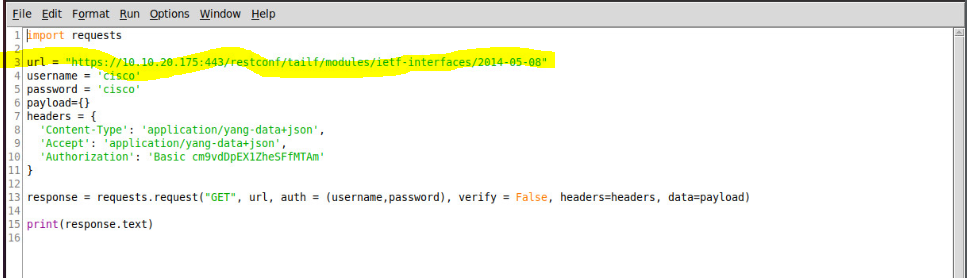
1. You can expand the squeezed output to browse your available YANG Modules. This won’t make much sense yet, but gradually we will acquire an introductory knowledge of this framework. When you expand the squeezed dictionary, it will look something like the following:



1. Scroll down to ietf-interfaces. Note that lowercase words will come after all the uppercase words alphabetically.



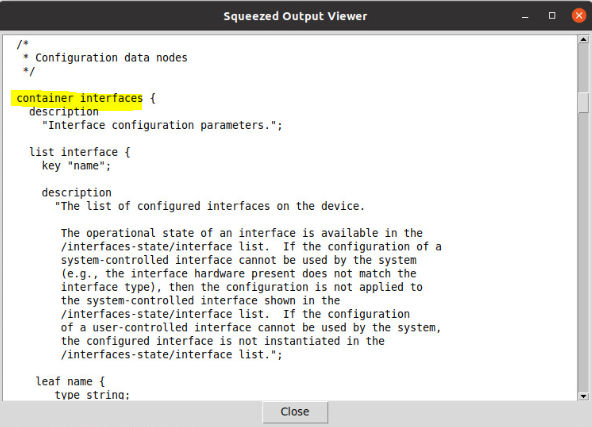
1. We are now able to identify the location of the schema. Copy the schema URL and paste it into your prior code and be sure to save the code as a new script before running it.



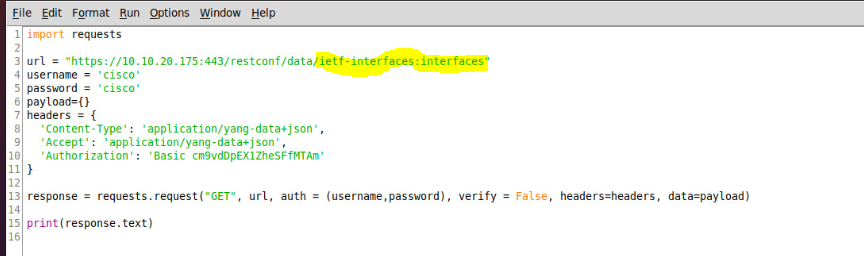
A screenshot of a computer

Description automatically generated with medium confidence

1. Note that everything is the same as our script, IOSXE-REST-GET-YANG-MODULES.py, except the URL. View your squeezed data. Note that as we scroll through the YANG schema, we can see some containers like interfaces and interfaces-state. See if you can find them:



1. Go back to your IOSXE-REST-GET-YANG-MODULES.py and save it under a different filename in your folder. Make the following modifications to your URL:



1. Note that ietf-interfaces was returned in Step 7 (/data/ietf-interfaces) and that :interfaces was our container returned in step 9 above. When you run the script with the above URL, you should have a nested dictionary that contains all of your interface names and their ip addresses.

A screenshot of a computer program

Description automatically generated with medium confidence

1. **Write a script that uses the above URL to get a list of IP addresses from an IOSXE device and returns a list of interfaces with address as shown below. The script must have a function that accepts an IP address for an IOSXE device and returns a dictionary to the main script of only interfaces with IP addresses. Your main script should then pass the dictionary to a printInt() function that prints the interfaces in the format shown below. Note that the loopback interface will cause an error in the printout of the dictionary if you try to call the address key. It doesn’t have an address. You can use a selection statement (if) or a try block to address this. Note that the work will be done in your functions. Your main script should only look the following:**

import requests

import json

deviceIP = ’10.10.20.175’

intDict = getInts(deviceIP) #This gets the interfaces model

printInt(intDict) #This iterates the dictionary that is returned

Your output should look like:

