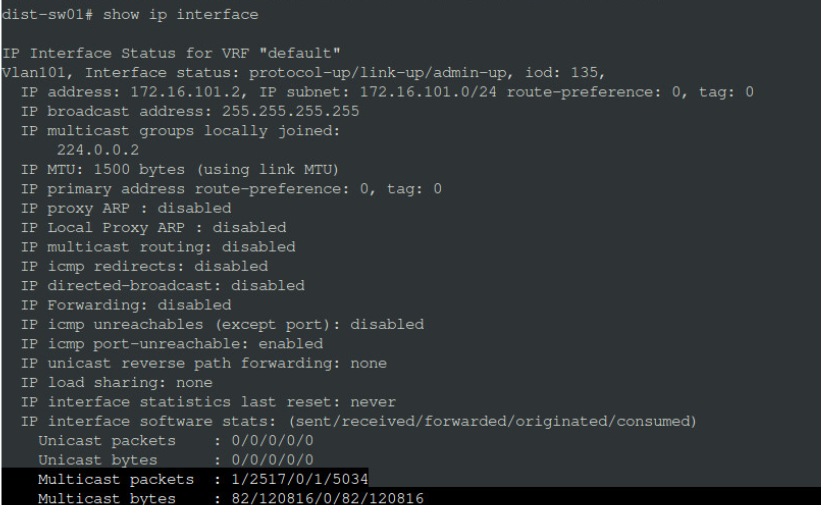
**Week Five Assessment**

Directions: Complete the scripts below and submit the code and screenshots of it running by the due date. 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.** **40 points**

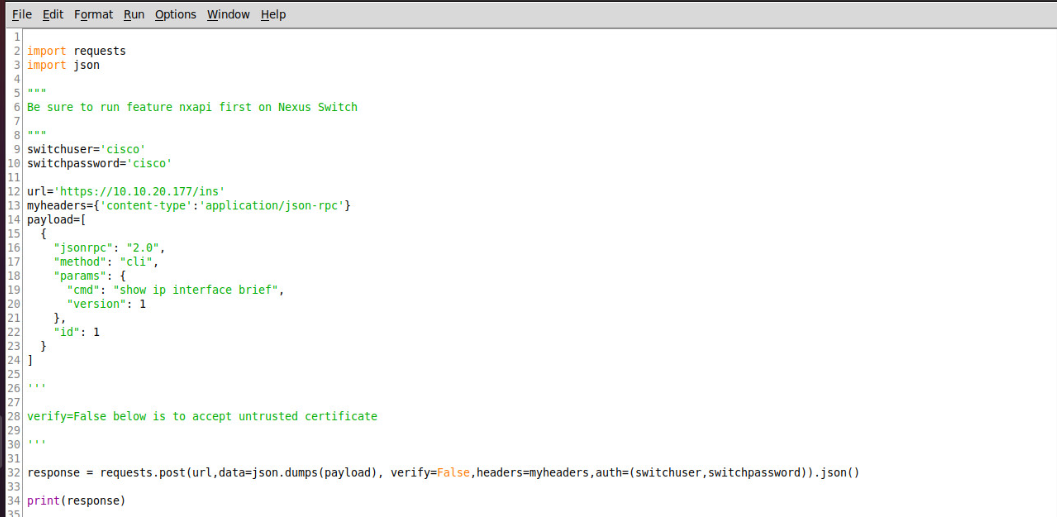
1. Get a Cisco Modeling Lab Reservation. Once in, save a screenshot of a “show ip interface” command on one of your device consoles.



1. Click on the console for dist-sw01, one of your NXOS Switches
2. Type feature nxapi on the switch in configuration mode.



1. The following code has been generated for you by the NXOS API Sandbox for the Nexus API interface, much like Postman did for your cards. A copy of the code is in a .py file called WeekFiveAssessmentAPI.pycode which you can copy to your Linux Terminal to use from Blackboard. You will want to rename it with a .py extension:

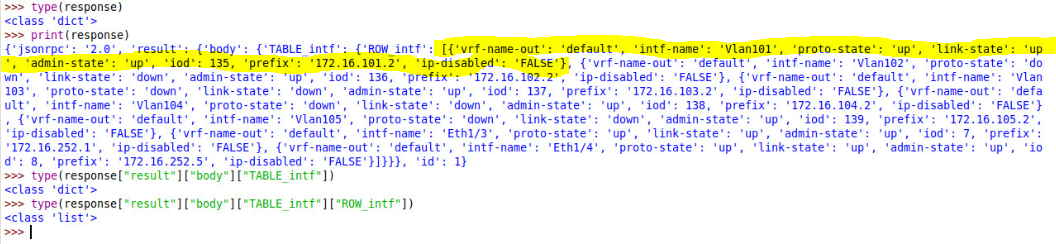


Note that 10.10.20.177 is the management address of dist-sw01 switch, a NEXOS 9000 series switch.

1. You won’t need to change anything in this code, but you will add one feature which will involve a few lines of code. You will print out a list of interfaces and IP addresses as shown below when it runs:



1. The good news is that you have done something like this before, and this won’t be too difficult if you think about it. However, I will provide some hints as follows:
   1. In the code in number 4 above, url, myheaders, and payload are all variable objects that have values and were auto-generated by the NXOS software toolkit that we will use later this semester. Notice that the payload is passing a “show ip interface brief” command via the API interface on line 19. This line can be modified for other commands like a show version.
   2. switchuser and switchpassword are strings defined to be passed in the requests.post() method. When you run this, there will be a warning about trusting certificates. That is acceptable in this situation. We are communicating with https to the NXOS switch, and we are receiving an untrusted certificate. The variable, response, is the value returned from the switch. It is a dictionary object. See below:



You have seen this kind of nested object before. Note the square bracket after “ROW\_intf” :. It indicates that a LIST follows, and in fact, it is a list of Dictionaries. See the type() outputs above for nested object types. I have highlighted one interface of the seven returned by the switch. This is very much like the card program from Week Four.

The object, response[“result”][“body”][“TABLE\_intf”][“ROW\_intf”] is a list just like your cardsDrawn[“cards”] List in Week Four. That means you can iterate through response[“result”][“body”][“TABLE\_intf”][“ROW\_intf”] with a For loop and access each dictionary key/pair as needed. You will need to access the values of dict[“intf\_name”], dict[“proto-state”], etc. where dict is the iterated dictionary variable in the list (i.e., for dict in iteratedList :). Within the loop you will be able to access dict[“key”] to generate the printout as shown in number 5 above. Remember that you can use “\t” in a string to print a tab, which should help you print the output in a readable form.

Graphical user interface, text, application

Description automatically generated