Cisco ACI API (part2)

Cobra, Arya & ACItoolKit

2023. 7

YAME! ACI API series

100) Introduction – API inspector

110) Introduction – ACItoolkit, cobra

120) CPU Util

130) Endpoint Tracker

140) APIC Port Binding

150) APIC vPC Binding

160) MSO Port Binding

170) Log

110) Introduction – ACI Tool Kit, cobra

- Postman
- Cobra SDK

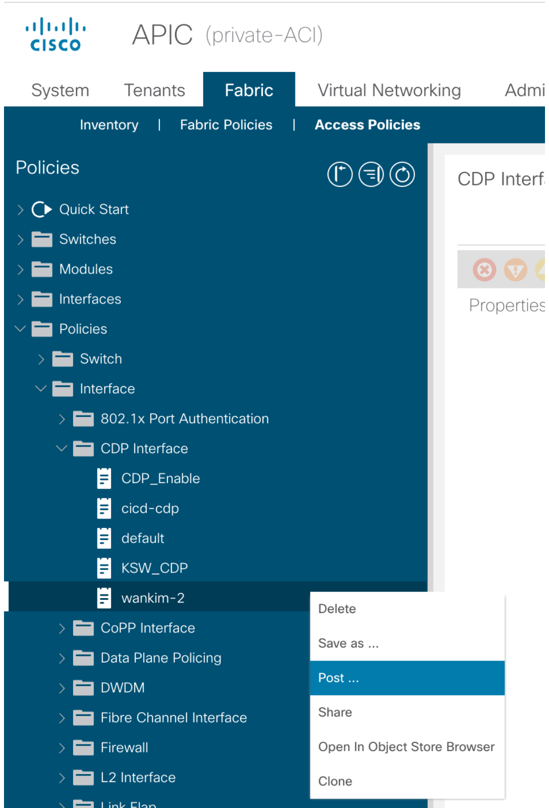
- ARYA
- Web ARYA

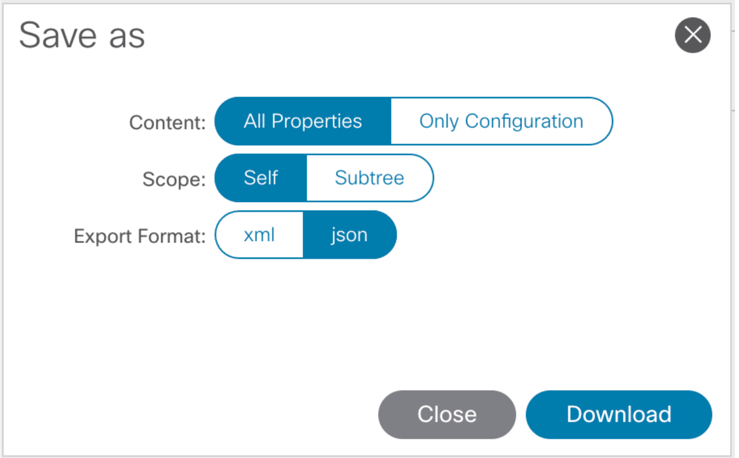
- ACI Tool Kit

Postman

Creating ACI Scripts the Easy Way

DOWNLOAD JSON IN THE GUI

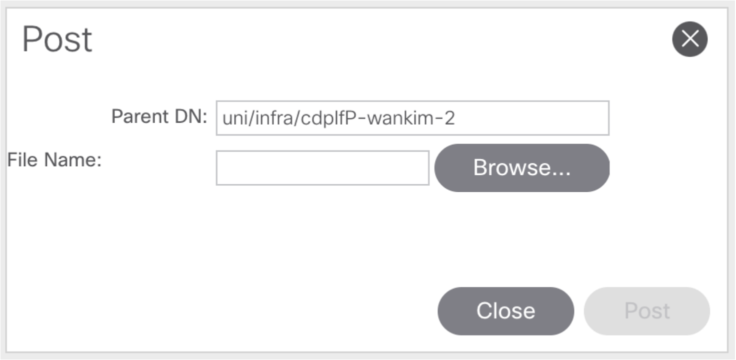




Content: [✓ All Properties] Only Configuration

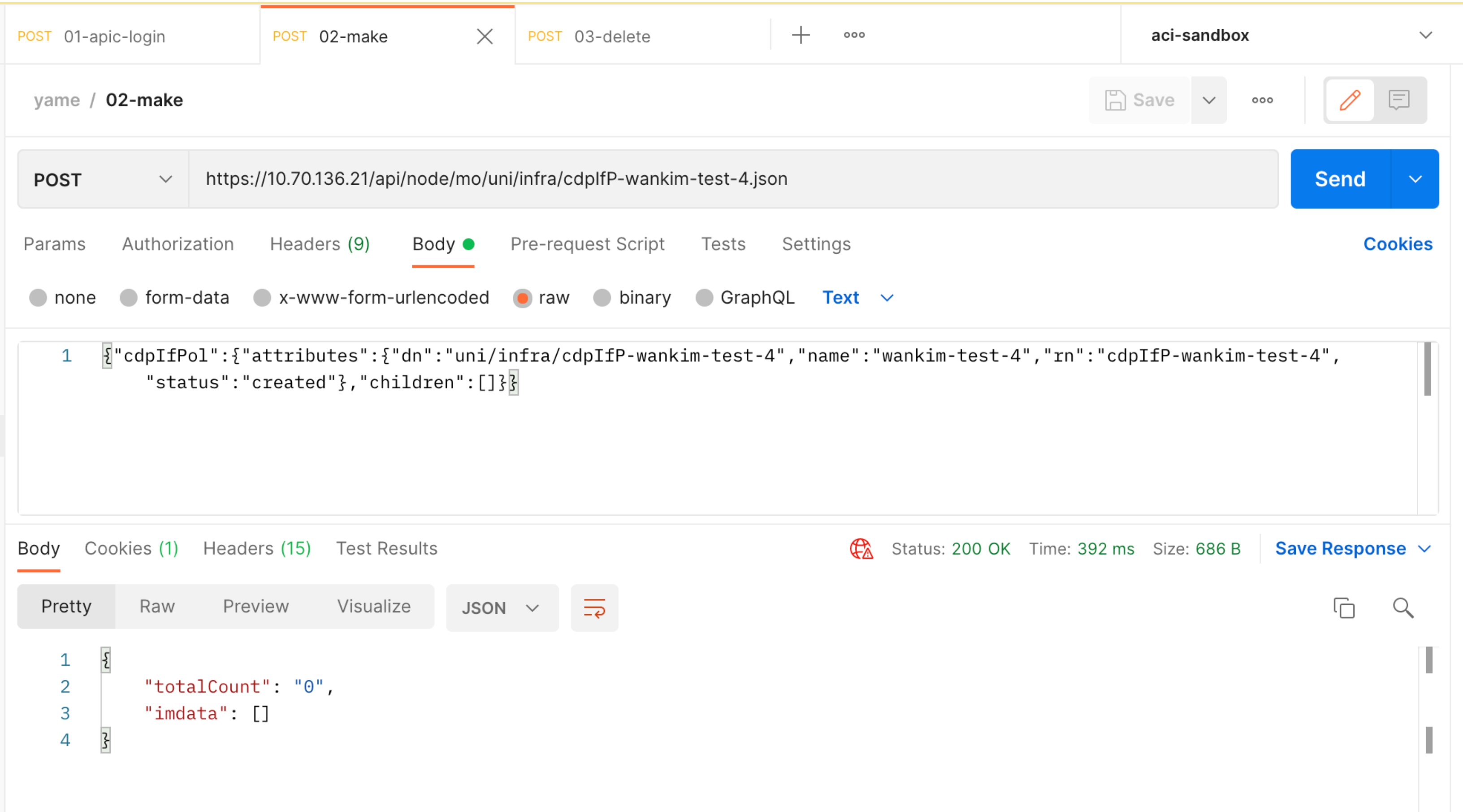
Scope: [✓ Self] Subtree

Export Format: xml [✓ json]



Parent DN: uni/infra/cdpIfP-wankim-2

File Name:



POST https://10.70.136.21/api/node/mo/uni/infra/cdpIfP-wankim-test-4.json

```
{"cdpIfPol":{"attributes":{"dn":"uni/infra/cdpIfP-wankim-test-4","name":"wankim-test-4","rn":"cdpIfP-wankim-test-4",
    "status":"created"},"children":[]}}
```

Status: 200 OK Time: 392 ms Size: 686 B

```
{
    "totalCount": "0",
    "imdata": []
}
```

Cobra SDK

Cobra SDK?

- It’s a native Python language binding for APIC REST API
- Cobra is ACI’s SDK.
- Supports lookups, creations, modifications, deletions
- Objects in Cobra are a **1:1 representation of objects in the MIT**
  - As a result, policy created via GUI/JSON/XML can be used as a programming template, for more rapid development
  - All data has the client side consistency checks performed

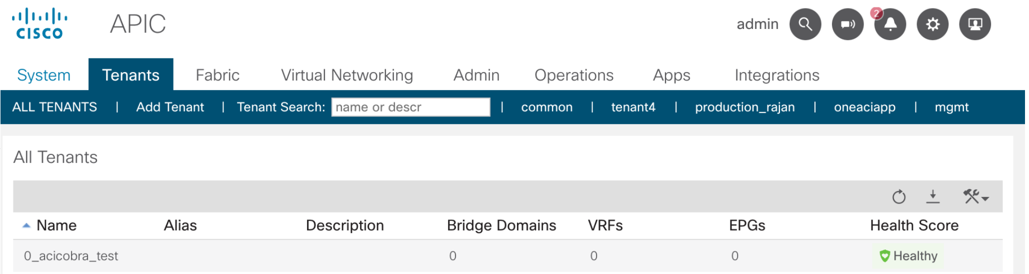
Install Cobra

Install and Docs : https://cobra.readthedocs.io/en/latest/index.html

Examples : https://github.com/datacenter/cobra

- **Install**
  - Can’t download the files SDK and model.
  - So, download from the git and install manually using PIP command.
    - move to the directory : /110-ACI-API/cobra
    - `pip3 install acicobra-5.2.3.0.7-py2.py3-none-any.whl`
    - `pip3 install acimodel-5.2.3.0.7-py2.py3-none-any.whl`

```
(venv) wankim@WANKIM-M-P1E1 100-ACI-API % python3
Python 3.9.13 (main, May 24 2022, 21:28:31)
[Clang 13.1.6 (clang-1316.0.21.2)] on darwin
Type "help", "copyright", "credits" or "license" for mor
>>> from credentials import *
>>> import cobra.mit.access
>>> import cobra.mit.request
>>> import cobra.mit.session
>>> import cobra.model.fv
>>> import cobra.model.pol
>>>
>>> auth = cobra.mit.session.LoginSession(URL, LOGIN, PASSWORD)
>>> session = cobra.mit.access.MoDirectory(auth)
>>> session.login()
/Users/wankim/Documents/Py/yamebook2_youtube_not_git/100-ACI-API/venv/lib/python3.9/site-
packages/urllib3/connectionpool.py:1045: InsecureRequestWarning: Unverified HTTPS request is being made to host
'sandboxapicdc.cisco.com'. Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
  warnings.warn(
>>>
>>> tenant_name = "0_acicobra_test"
>>> root = cobra.model.pol.Uni('')
>>> new_tenant = cobra.model.fv.Tenant(root, tenant_name)
>>>
>>> config_request = cobra.mit.request.ConfigRequest()
>>> config_request.addMo(new_tenant)
>>> session.commit(config_request)
/Users/wankim/Documents/Py/yamebook2_youtube_not_git/100-ACI-API/venv/lib/python3.9/site-
packages/urllib3/connectionpool.py:1045: InsecureRequestWarning: Unverified HTTPS request is being made to host
'sandboxapicdc.cisco.com'. Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
  warnings.warn(
<Response [200]>
>>>
```


cisco
APIC
admin
System
Tenants
Fabric
Virtual Networking
Admin
Operations
Apps
Integrations
ALL TENANTS | Add Tenant | Tenant Search: name or descr | common | tenant4 | production_rajan | oneaciapp | mgmt

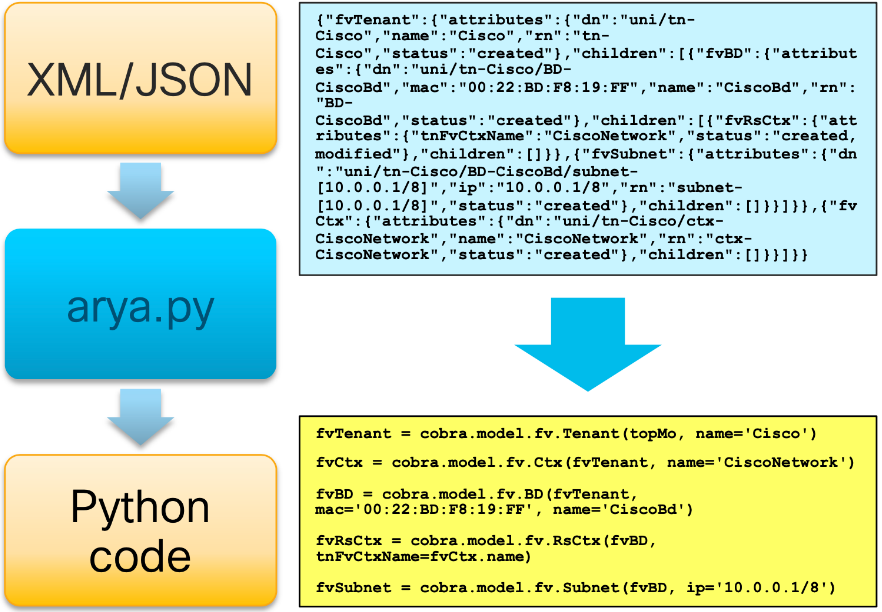
All Tenants

Name	Alias	Description	Bridge Domains	VRFs	EPGs	Health Score
0_acicobra_test			0	0	0	Healthy

ARYA
(APIC Rest pYthon Adapter)

APIC REST to Python Adapter: arya.py

- GUI creates REST
- API Inspector shows REST
- arya.py creates code from REST
- Auto-generate code to automate tasks, without heavy lifting
- Available at http://github.com/datacenter/arya



```
{"fvTenant":{"attributes":{"dn":"uni/tn-
Cisco","name":"Cisco","rn":"tn-
Cisco","status":"created"},"children":[{"fvBD":{"attribut
es":{"dn":"uni/tn-Cisco/BD-
CiscoBd","mac":"00:22:BD:F8:19:FF","name":"CiscoBd","rn":
"BD-
CiscoBd","status":"created"},"children":[{"fvRsCtx":{"att
ributes":{"tnFvCtxName":"CiscoNetwork","status":"created,
modified"},"children":[]}},{"fvSubnet":{"attributes":{"dn
":"uni/tn-Cisco/BD-CiscoBd/subnet-
[10.0.0.1/8]","ip":"10.0.0.1/8","rn":"subnet-
[10.0.0.1/8]","status":"created"},"children":[]}}]}},{"fv
Ctx":{"attributes":{"dn":"uni/tn-Cisco/ctx-
CiscoNetwork","name":"CiscoNetwork","rn":"ctx-
CiscoNetwork","status":"created"},"children":[]}}]}}
```

```
fvTenant = cobra.model.fv.Tenant(topMo, name='Cisco')

fvCtx = cobra.model.fv.Ctx(fvTenant, name='CiscoNetwork')

fvBD = cobra.model.fv.BD(fvTenant,
mac='00:22:BD:F8:19:FF', name='CiscoBd')

fvRsCtx = cobra.model.fv.RsCtx(fvBD,
tnFvCtxName=fvCtx.name)

fvSubnet = cobra.model.fv.Subnet(fvBD, ip='10.0.0.1/8')
```

Install

- git clone https://github.com/datacenter/arya.git
- cd arya
- python3 setup.py install

Convert JSON(from inspector) to Python Code

```
(venv) wankim@WANKIM-M-P1E1 arya % arya.py -f cdpIfP-wankim-2.json -i xx.xx.xx.xx -u admin -p xxxxx
#!/usr/bin/env python
'''
Autogenerated code using arya.py
Original Object Document Input:
{"totalCount":"1","imdata":[{"cdpIfPol":{"attributes":{"adminSt":"disabled","annotation":"","descr":"","dn":"uni/infra/cdpIfP-wankim-2","name":"wankim-
2","nameAlias":"","ownerKey":"","ownerTag":"","userdom":":all:"}}}]}
'''
raise RuntimeError('Please review the auto generated code before ' +
                    'executing the output. Some placeholders will ' +
                    'need to be changed')

# list of packages that should be imported for this code to work
import cobra.mit.access
import cobra.mit.request
import cobra.mit.session
import cobra.model.cdp
import cobra.model.infra
import cobra.model.pol
from cobra.internal.codec.xmlcodec import toXMLStr

# log into an APIC and create a directory object
ls = cobra.mit.session.LoginSession('https://xx.xx.xx.xx', 'admin', ‘xxxxx')
md = cobra.mit.access.MoDirectory(ls)
md.login()

# the top level object on which operations will be made
polUni = cobra.model.pol.Uni('')
infraInfra = cobra.model.infra.Infra(polUni)

# build the request using cobra syntax
cdpIfPol = cobra.model.cdp.IfPol(infraInfra, adminSt='disabled', annotation='', descr='', name='wankim-2', nameAlias='', ownerKey='', ownerTag='',
userdom=':all:')


# commit the generated code to APIC
print(toXMLStr(infraInfra))
c = cobra.mit.request.ConfigRequest()
c.addMo(infraInfra)
md.commit(c)
```

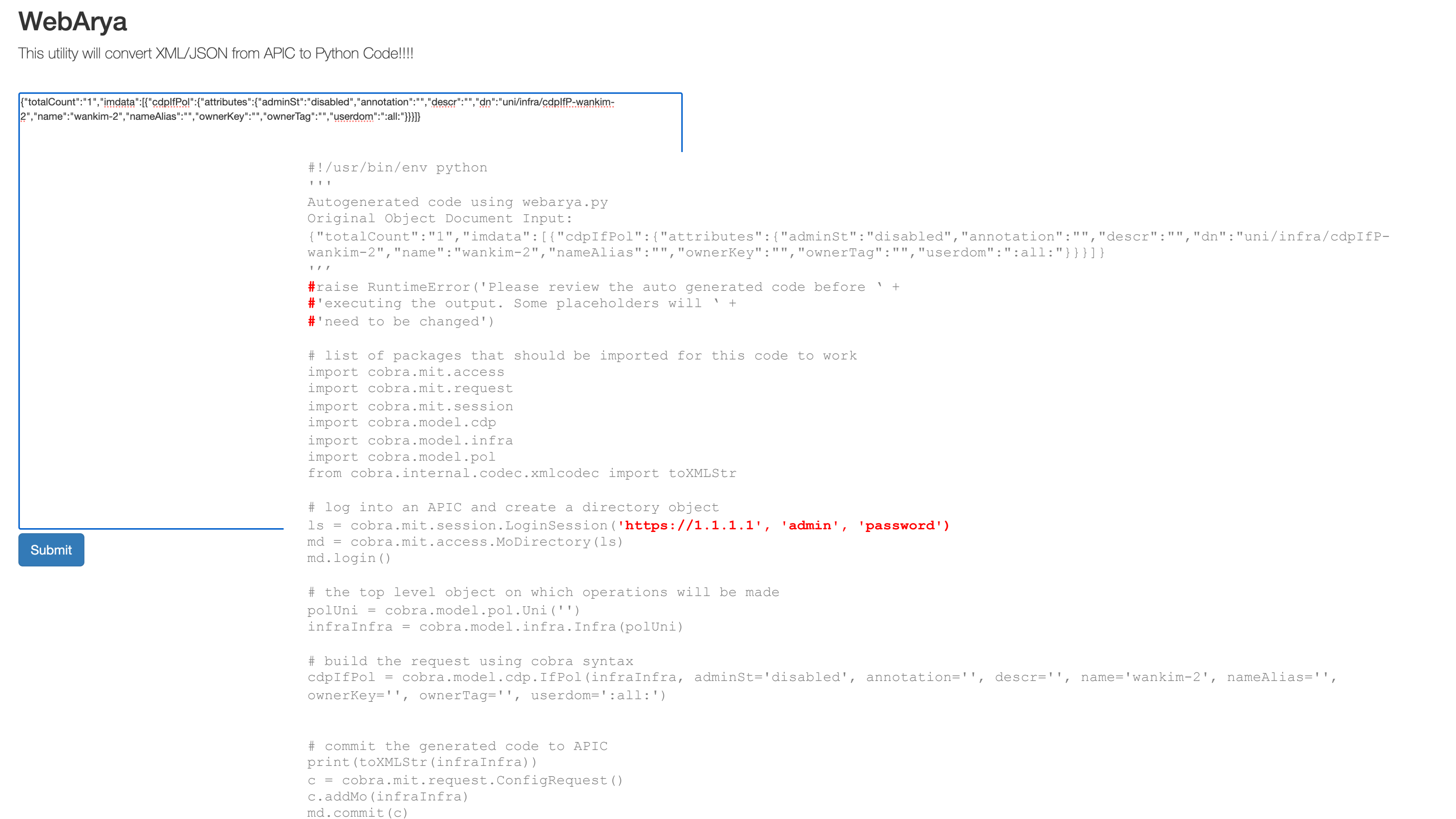
WEB ARYA

https://github.com/datacenter/webarya

web

no need to install. Just Run!

```
(venv) wankim@WANKIM-M-P1E1 webarya % python3 webarya.py
 * Serving Flask app 'webarya'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI
server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:8888
 * Running on http://10.70.153.76:8888
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 124-846-690
```



WebArya

This utility will convert XML/JSON from APIC to Python Code!!!!

{"totalCount":"1","imdata":[{"cdpIfPol":{"attributes":{"adminSt":"disabled","annotation":"","descr":"","dn":"uni/infra/cdpIfP-wankim-2","name":"wankim-2","nameAlias":"","ownerKey":"","ownerTag":"","userdom":":all:"}}}]}

```
#!/usr/bin/env python
'''
Autogenerated code using webarya.py
Original Object Document Input:
{"totalCount":"1","imdata":[{"cdpIfPol":{"attributes":{"adminSt":"disabled","annotation":"","descr":"","dn":"uni/infra/cdpIfP-
wankim-2","name":"wankim-2","nameAlias":"","ownerKey":"","ownerTag":"","userdom":":all:"}}}]}
'''
#raise RuntimeError('Please review the auto generated code before ' +
#'executing the output. Some placeholders will ' +
#'need to be changed')

# list of packages that should be imported for this code to work
import cobra.mit.access
import cobra.mit.request
import cobra.mit.session
import cobra.model.cdp
import cobra.model.infra
import cobra.model.pol
from cobra.internal.codec.xmlcodec import toXMLStr

# log into an APIC and create a directory object
ls = cobra.mit.session.LoginSession('https://1.1.1.1', 'admin', 'password')
md = cobra.mit.access.MoDirectory(ls)
md.login()

# the top level object on which operations will be made
polUni = cobra.model.pol.Uni('')
infraInfra = cobra.model.infra.Infra(polUni)

# build the request using cobra syntax
cdpIfPol = cobra.model.cdp.IfPol(infraInfra, adminSt='disabled', annotation='', descr='', name='wankim-2', nameAlias='',
ownerKey='', ownerTag='', userdom=':all:')


# commit the generated code to APIC
print(toXMLStr(infraInfra))
c = cobra.mit.request.ConfigRequest()
c.addMo(infraInfra)
md.commit(c)
```

ACI Tool Kit

ACI Toolkit

Document : https://acitoolkit.readthedocs.io/en/latest/index.html
Examples : https://github.com/datacenter/acitoolkit (over 40 sample scripts)

- from Cisco Live DEVWKS-2001

- Simple toolkit built on top of APIC's REST API
- Set of **simple Python classes**
  - Python Library
  - Used to generate REST API calls
  - Runs locally
  - Small number of classes
  - "Intuitive" names
- Preserves the ACI basic concepts
  - Tenants, EPGs, Contracts, etc.

- ACI Toolkit was built to only perform the most common operations on a fabric.
- Cobra SDK Python library is a fully functional, 1:1 mapping of the MIT.

- the ACI Toolkit can be used to create scripts that build a new application with multiple EPGs, assign the EPGs to the proper Bridge Domain, have them provide and consume contracts, and connect them to physical and virtual infrastructure.

- It also has some ready-made applications that provide visualizations and store persistent data, such as endpoint locations and fabric configurations.

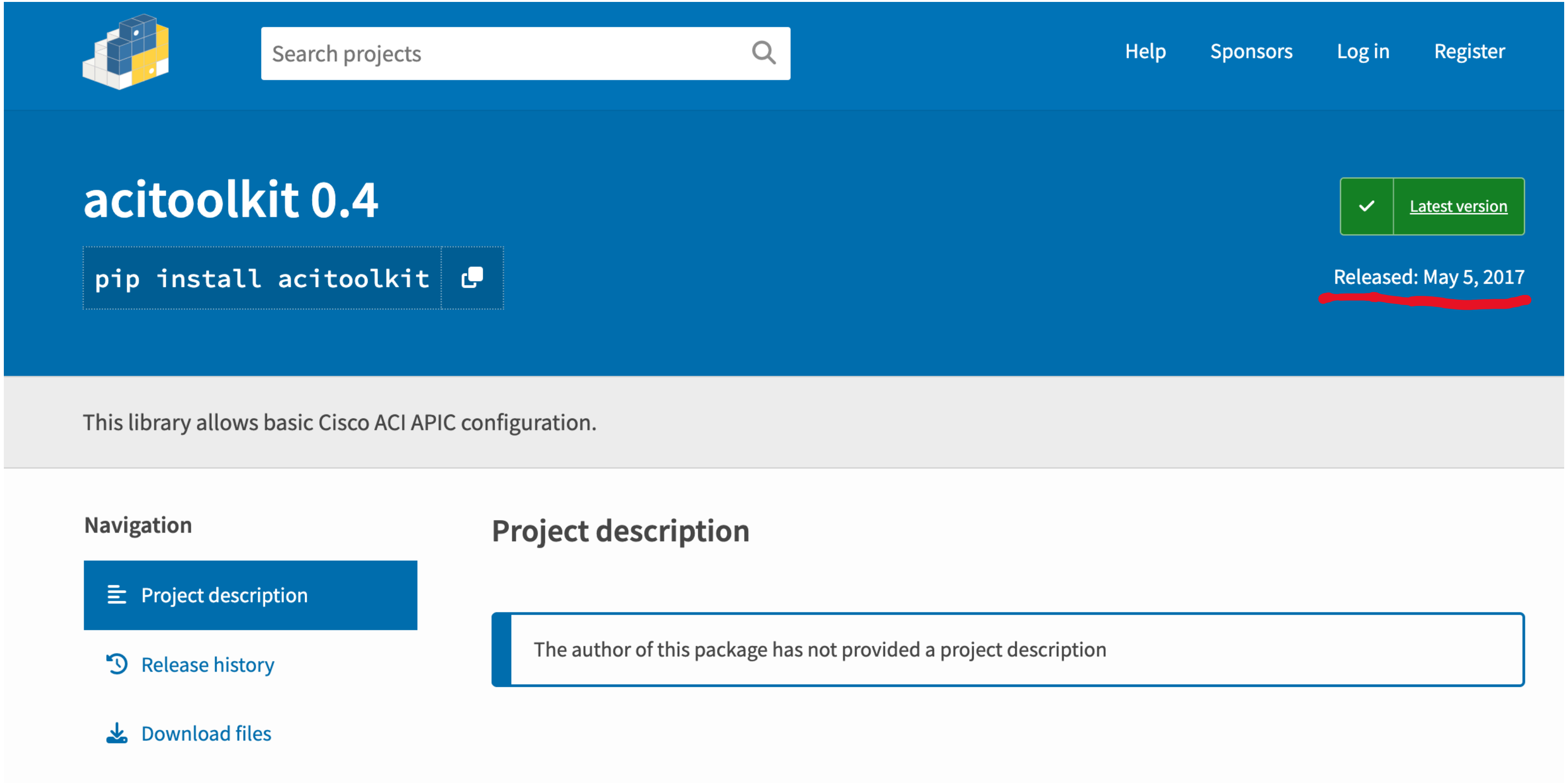
ACI Toolkit install and create the tenant

```
(venv) wankim@WANKIM-M-P1E1 100-ACI-API % pip install acitoolkit

(venv) wankim@WANKIM-M-P1E1 100-ACI-API % python3
Python 3.9.13 (main, May 24 2022, 21:28:31)
[Clang 13.1.6 (clang-1316.0.21.2)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from acitoolkit import acitoolkit
>>> session = acitoolkit.Session('https://sandboxapicdc.cisco.com','admin','!v3G@!4@Y')
>>> session.login()
<Response [200]>
>>>
>>> tenant_name = "0aci-api-test2"
>>> new_tenant = acitoolkit.Tenant(tenant_name)
>>> session.push_to_apic(new_tenant.get_url(), new_tenant.get_json())
<Response [200]>
```

Not Updated for 6 Years since May 5, 2017

https://acitoolkit.readthedocs.io/en/latest/



acitoolkit 0.4

```
pip install acitoolkit
```

Released: May 5, 2017

This library allows basic Cisco ACI APIC configuration.

Project description

The author of this package has not provided a project description