

CISCO *Live!*

GO BEYOND

#CiscoLiveAPJC



Simplify Your NetDevOps

Unleashing the Possibilities of Cisco Catalyst Routing
& SDWAN SDK

Adrian ILIESIU, Senior Developer Advocate
@aidevnet
DEVNET-2137



#CiscoLiveAPJC

Cisco Webex App

Questions?

Use Cisco Webex App to chat with the speaker after the session

How

- 1 Find this session in the Cisco Live Mobile App
- 2 Click “Join the Discussion”
- 3 Install the Webex App or go directly to the Webex space
- 4 Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until November 15, 2024.



<https://ciscolive.ciscoevents.com/ciscolivebot/#DEVNET-2137>

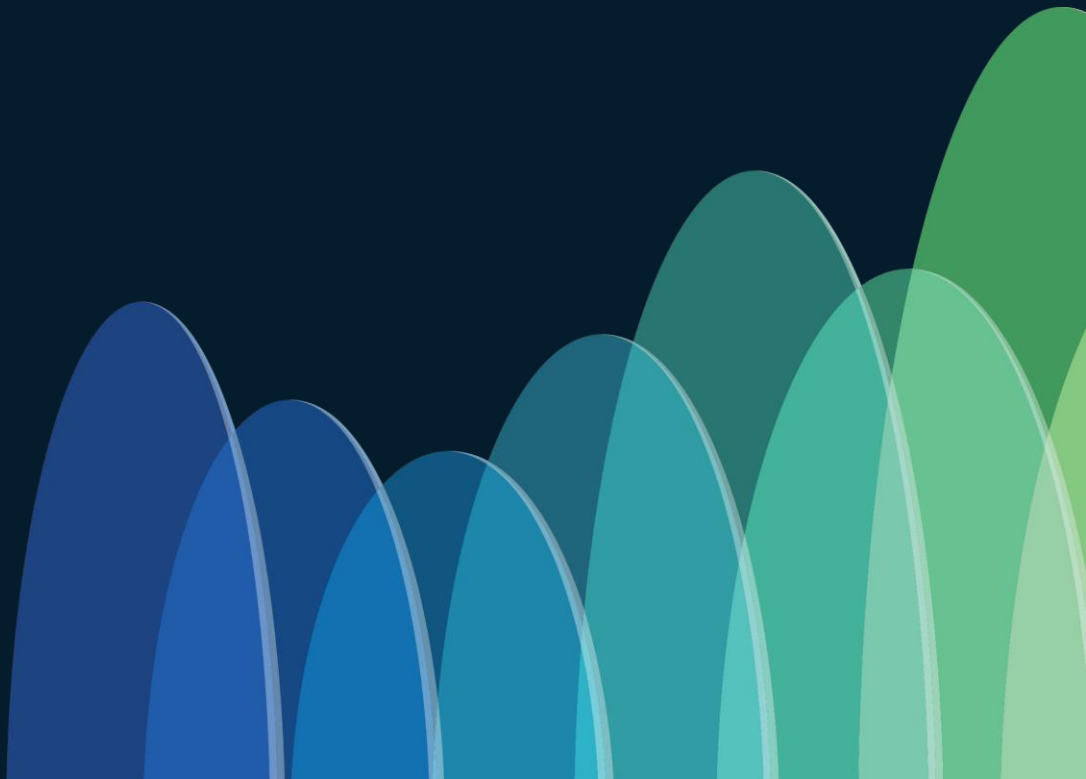




Agenda

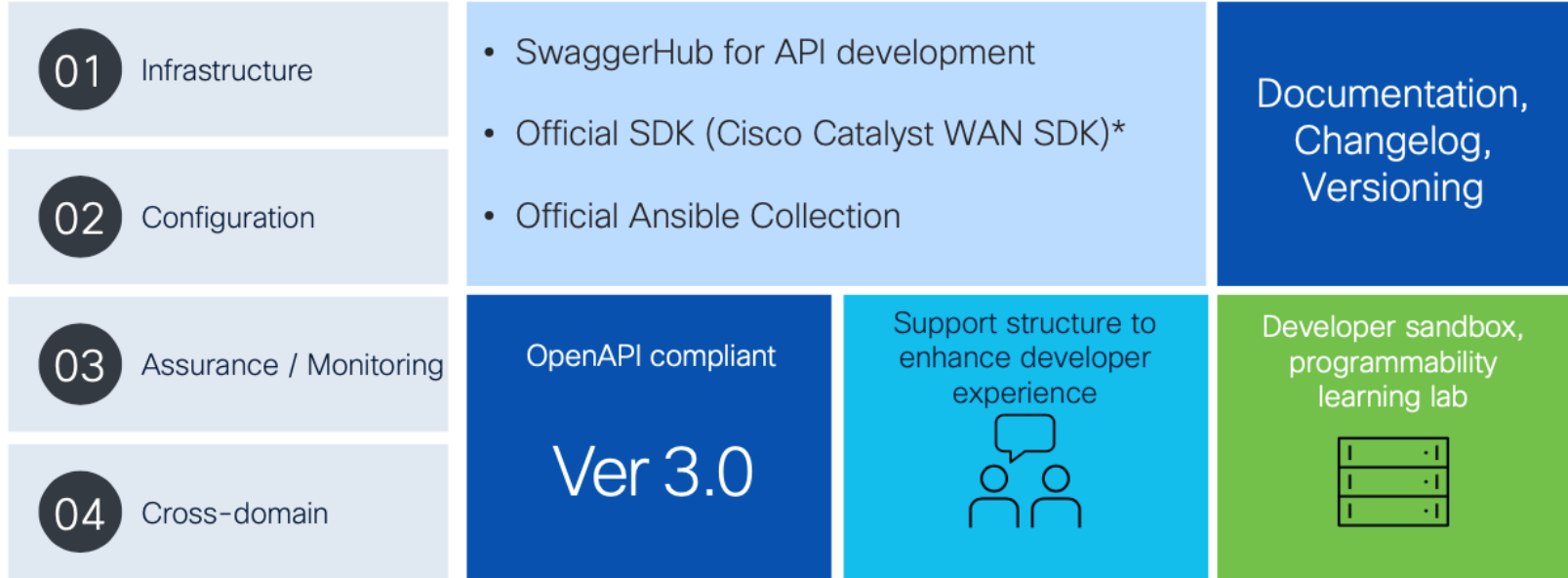
- Introduction
- What is an SDK
- What is Catalyst WAN SDK
- Architecture
- SDK Repo and Sample Workflow
- Demo
- Resources

Introduction



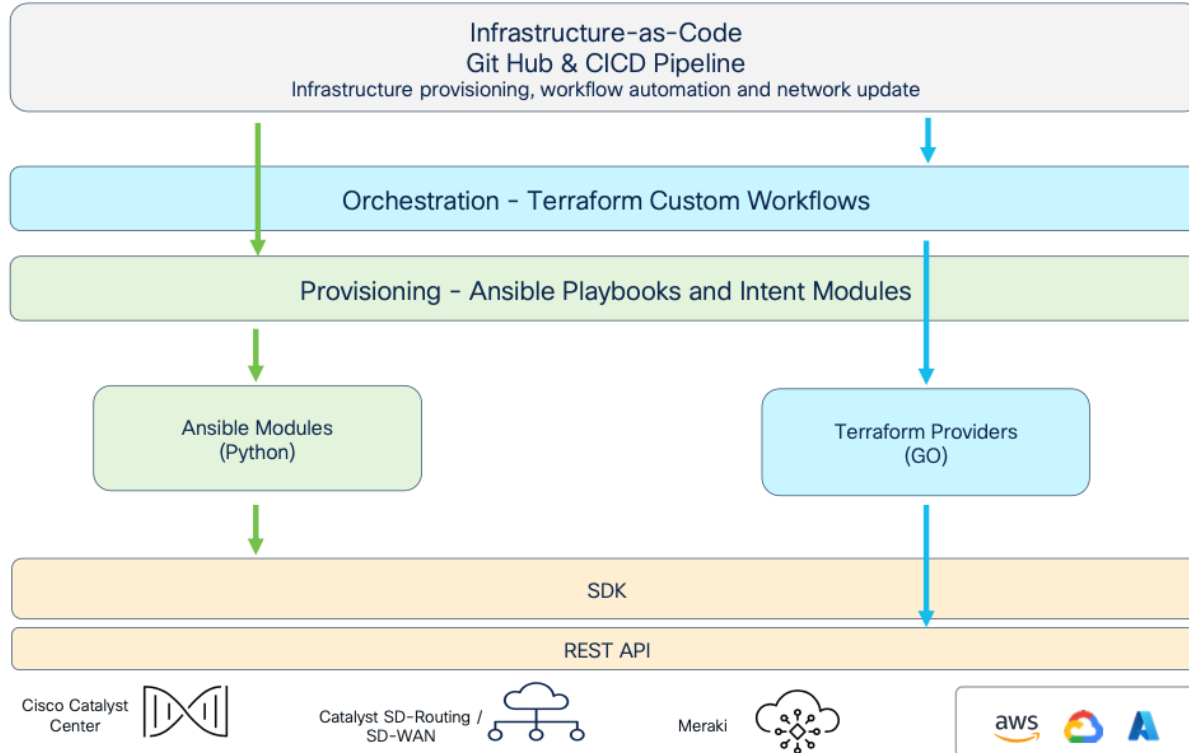
Catalyst SD-Routing/SD-WAN Automation

API/SDK/Workflow-based Use Cases



Automation with Catalyst SD-WAN

Architecture



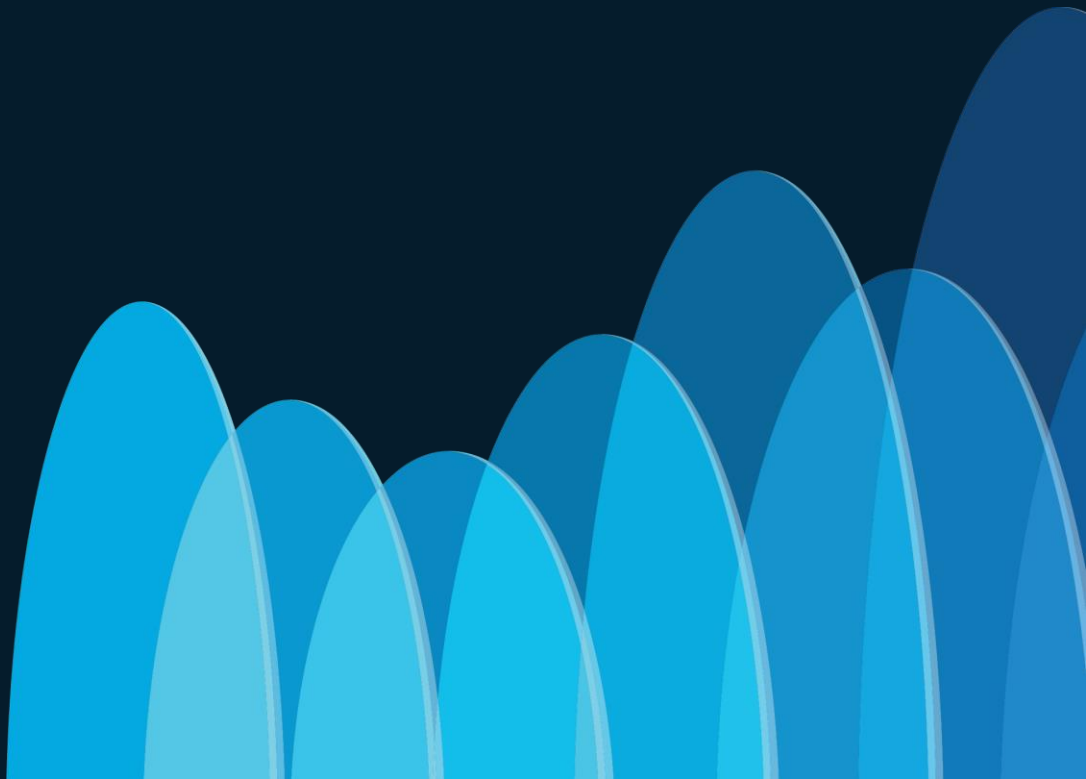
What is an SDK



What is an SDK

- is a collection of programming language specific libraries, tools and utilities that make it easy for users to interact with your application
- designed to streamline the development process, making it easier and faster for developers to create applications for specific software, operating systems, hardware platforms, etc.

What is Catalyst WAN SDK



Cisco Catalyst WAN SDK

It's more than just an SDK

- Official Cisco supported SDK
- Universal Catalyst SD-WAN Manager API Python library
- Designed for multi-endpoints workflows
- Open Source
- Easy to use
- Catalyst SD-WAN Manager version supported
 - 20.14, 20.13, 20.12, 20.11, 20.10, 20.9, 20.6, 20.3



Cisco Catalyst WAN SDK

Python 3.8 | 3.9 | 3.10 | 3.11 | 3.12

Cisco Catalyst WAN SDK – Built-In Workflows

Completed:

- ST to MT migration
- MT to ST migration
- MT: individual tenant backup & restore

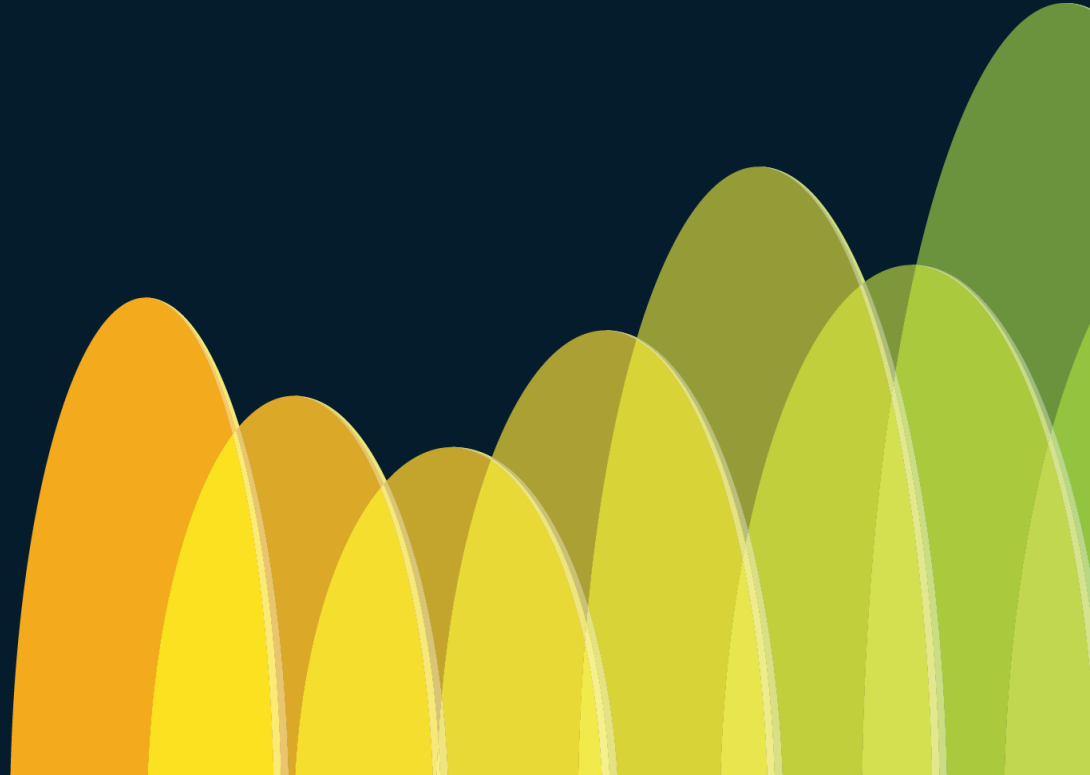
Work in progress:

- Software upgrade
- Template migration (UX 1.0 -> UX 2.0)

Roadmap:

- vEdge configuration to cEdge configuration migration
- MT to MT migration

Architecture



Catalyst WAN SDK Architecture

- Layer 3: API

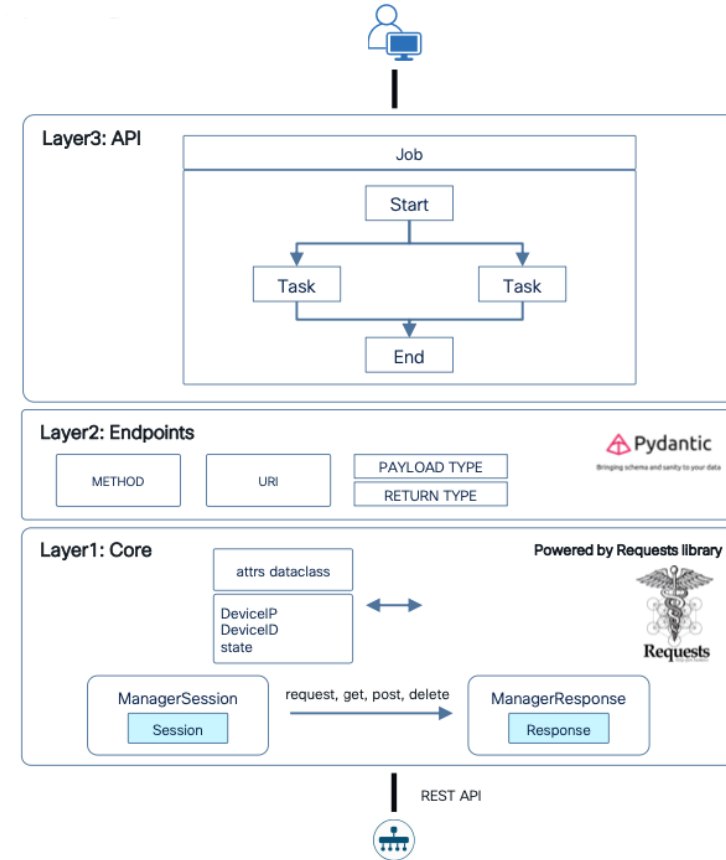
- Runs one level higher and can handle a series of API calls
- Stateful, logic, can make decisions between endpoint calls
- Workflows are built here
- Available via: **ManagerSession.api**

- Layer 2: Endpoints

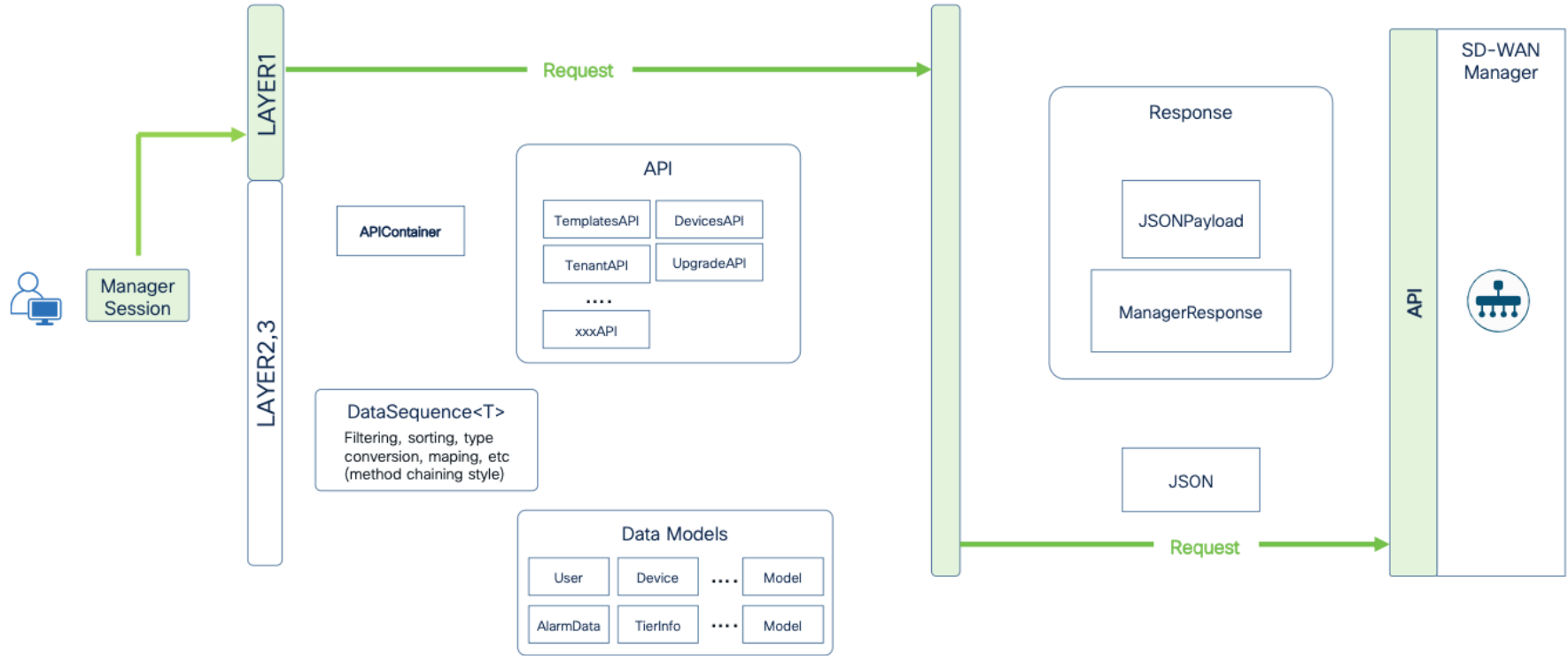
- Creates data-models based on pydantic library
 - Work with object.attributes instead of raw JSON payload
- Version handling
- State-less and Logic-less: only method + URI + data
- Available via: **ManagerSession.endpoints**

- Layer 1: Core

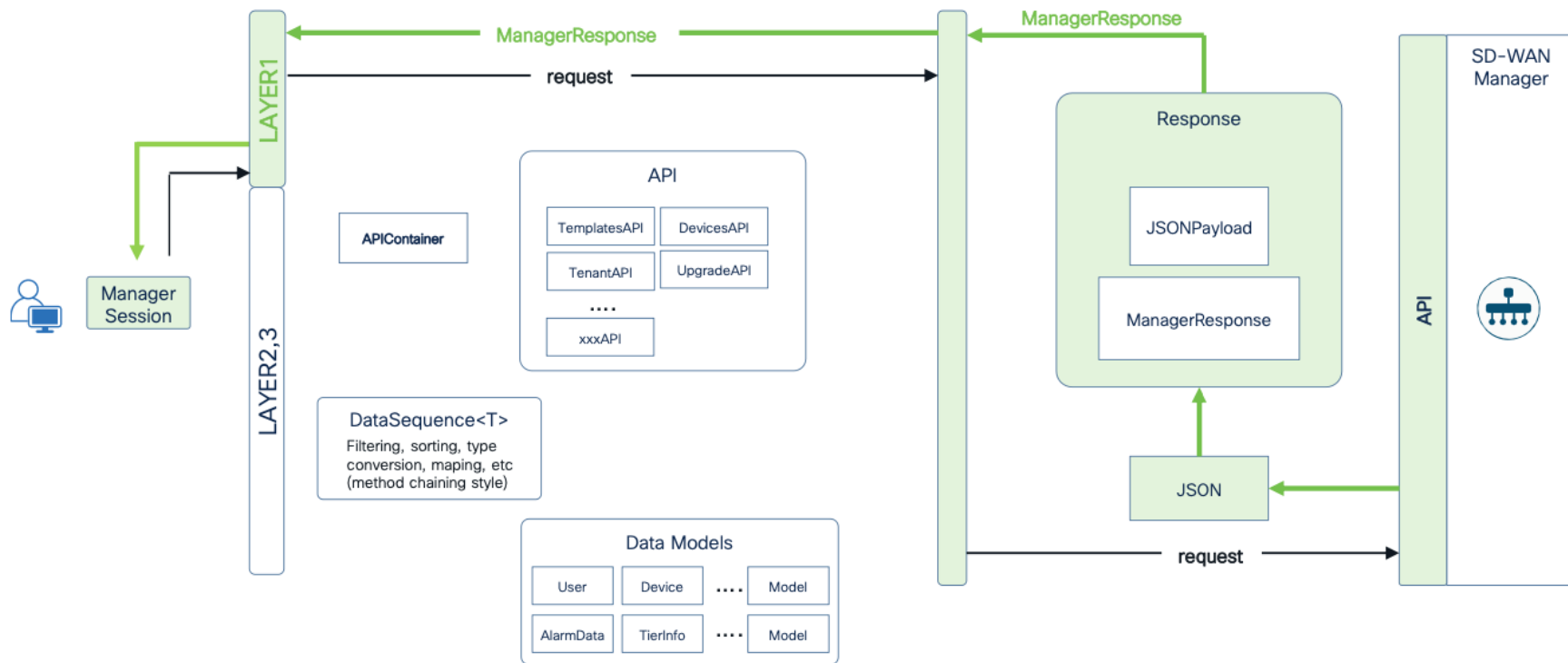
- Manages user session including authentication and session mgmt.
- Extends Requests library to ManageSession & ManageResponse
- Provides framework for data-model creation based on attrs library



(1) Using RAW APIs - Request



(1) Using RAW APIs - Response



Output

Config Group



```
{
  "id": "bdaec87d-831e-4852-b7fb-25b14c6791ed",
  "name": "edge_basic",
  "description": "Do not modify!",
  "source": null,
  "solution": "sdwan",
  "lastUpdatedBy": "jmb",
  "lastUpdatedOn": 1710421112116,
  "createdBy": "jmb",
  "createdOn": 1710406377623,
  "profiles": [
    {
      "id": "4a53c92c-17b2-4804-9563-b9144a8fac22",
      "name": "edge_basic_system",
      "solution": "sdwan",
      "type": "system",
      "description": null,
      "lastUpdatedBy": "jmb",
      "lastUpdatedOn": 1710422367658,
      "createdBy": "jmb",
      "createdOn": 1710406371889
    },
    {
      "id": "57a3e3f9-ec2c-4a59-8ad8-79e647a6fda9",
      "name": "edge_basic_wan",
      "solution": "sdwan",
      "type": "transport",
      "description": null,
      "lastUpdatedBy": "jmb",
      "lastUpdatedOn": 1710406367740,
      "createdBy": "jmb",
      "createdOn": 1710406364543
    },
    {
      "id": "416090a3-37c3-49a2-a88d-90c8cde9a89f",
      "name": "edge_basic_lan",
      "solution": "sdwan",
      "type": "service",
      "description": null,
      "lastUpdatedBy": "jmb",
      "lastUpdatedOn": 1710406371688,
      "createdBy": "jmb",
      "createdOn": 1710406368185
    }
  ]
}
```

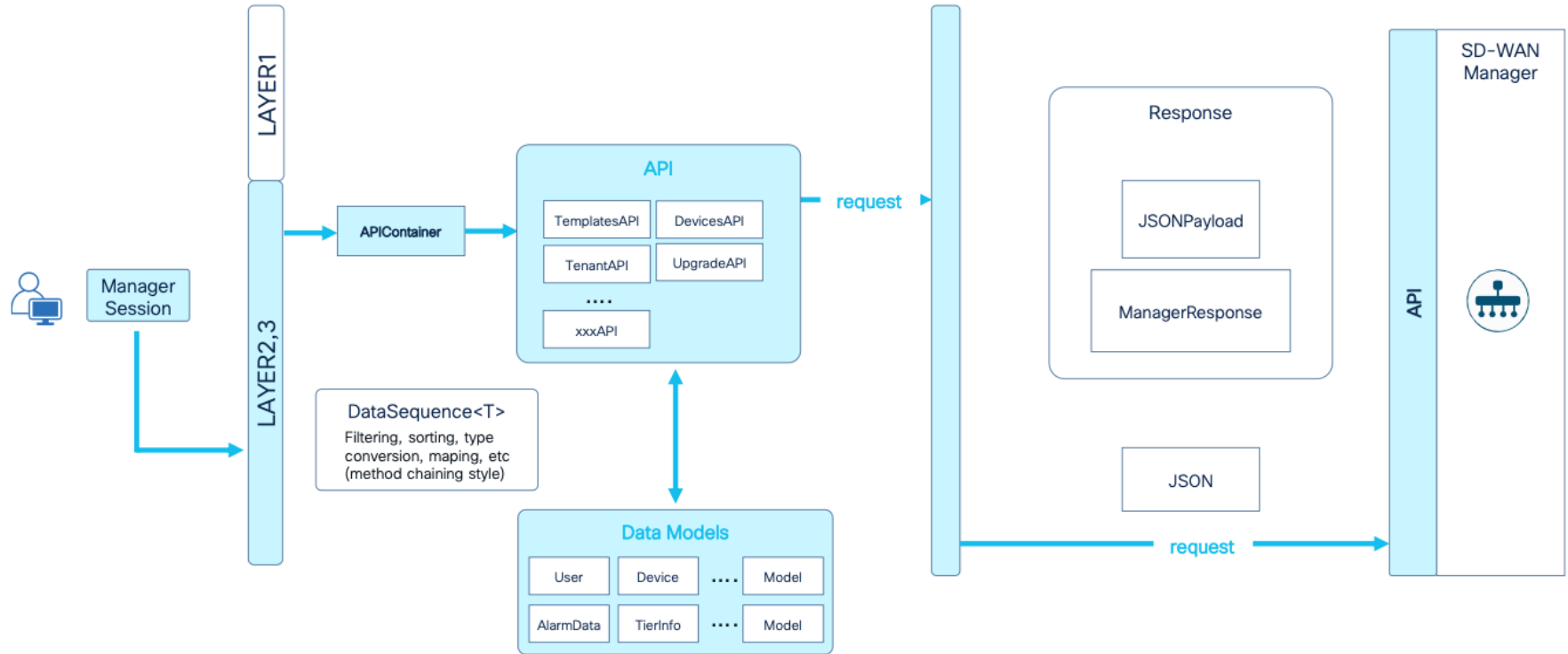
Profile



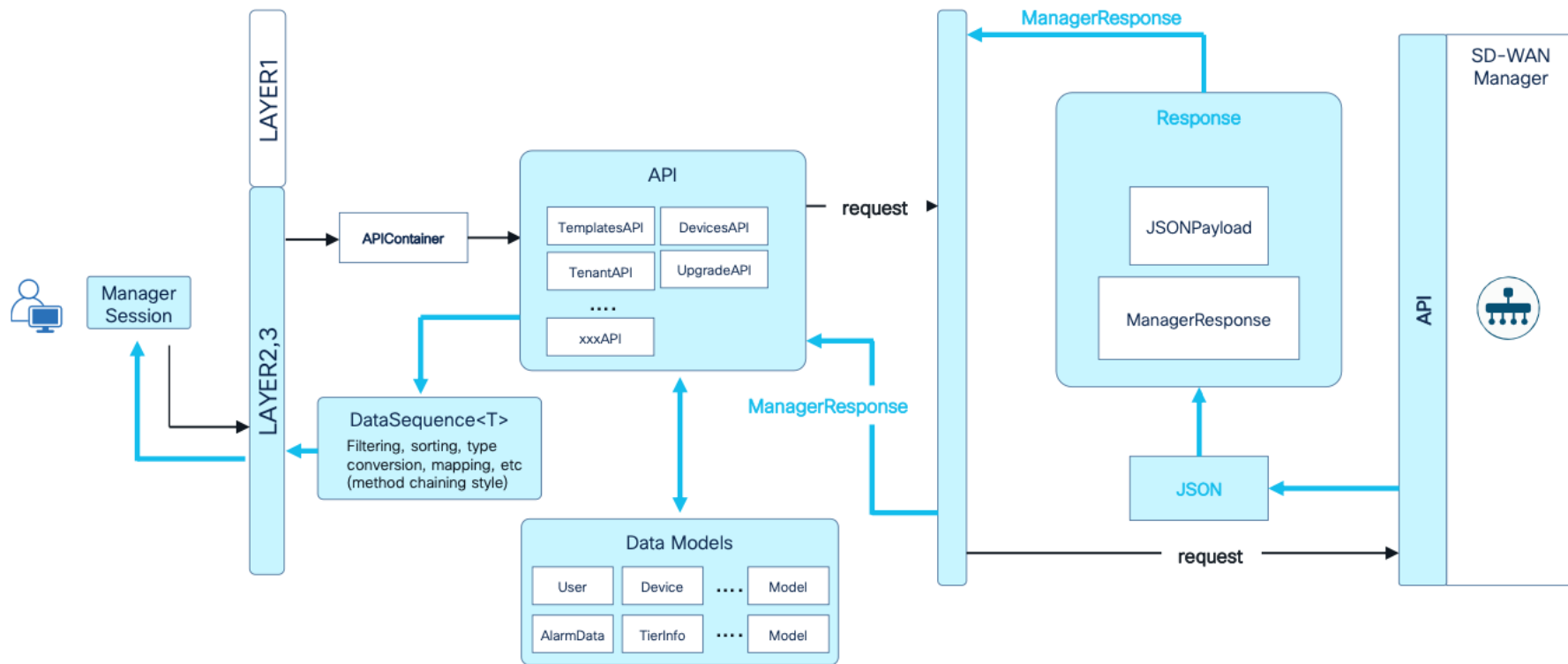
```
{
  "id": "57a3e3f9-ec2c-4a59-8ad8-79e647a6fda9",
  "name": "edge_basic_wan",
  "solution": "sdwan",
  "type": "transport",
  "description": null,
  "lastUpdatedBy": "jmb",
  "lastUpdatedOn": 1710406367740,
  "createdBy": "jmb",
  "createdOn": 1710406364543
},
{
  "id": "416090a3-37c3-49a2-a88d-90c8cde9a89f",
  "name": "edge_basic_lan",
  "solution": "sdwan",
  "type": "service",
  "description": null,
  "lastUpdatedBy": "jmb",
  "lastUpdatedOn": 1710406371688,
  "createdBy": "jmb",
  "createdOn": 1710406368185
}
```

a JSON Payload response
that you have to parse and
deserialise

(2) Using User Prepared APIs - Request



(2) Using User Prepared APIs - Response



Output

```
--- Session Information
vManage: 10.54.60.30
Version: 20.13.1
API Version: 20.13
Application Version: 20.13R-vbamboo-16-Dec-2023 00:55:38 PST
```

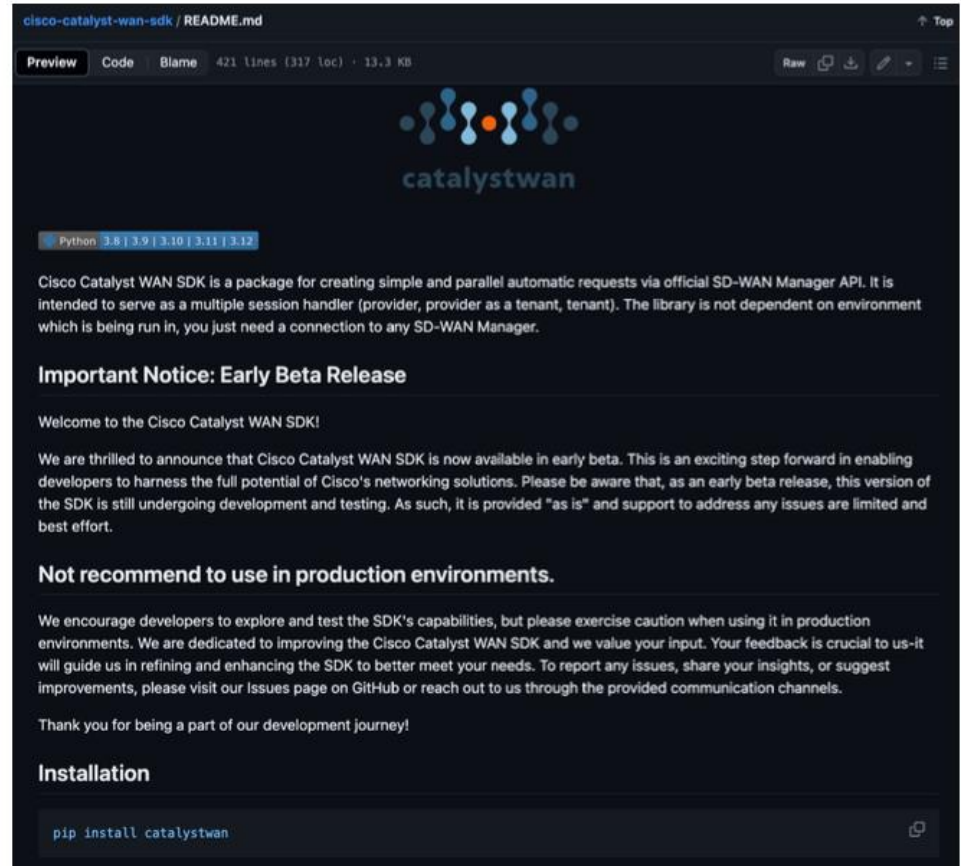
```
~~~~~
Config Group Name: edge_basic
- Description: Do not modify!
- Solution: sdwan
- Profile Id: 4a53c92c-17b2-4804-9563-b9144a8fac22
  - Name: edge_basic_system
  - Type: system
  - Solution: sdwan
  - Created by: jmb
  - Last updated by: jmb
  - Last updated on: 2024-03-14 13:19:27.658000+00:00
- Profile Id: 57a3e3f9-ec2c-4a59-8ad8-79e647a6fda9
  - Name: edge_basic_wan
  - Type: transport
  - Solution: sdwan
  - Created by: jmb
  - Last updated by: jmb
  - Last updated on: 2024-03-14 08:52:47.740000+00:00
- Profile Id: 416090a3-37c3-49a2-a88d-90c8cde9a89f
  - Name: edge_basic_lan
  - Type: service
  - Solution: sdwan
  - Created by: jmb
  - Last updated by: jmb
  - Last updated on: 2024-03-14 08:52:51.608000+00:00
- Profile Id: e85e2c1f-b801-4f3f-a9ad-df502552e790
  - Name: edge_basic_cli
  - Type: cli
  - Solution: sdwan
  - Created by: jmb
  - Last updated by: jmb
  - Last updated on: 2024-03-14 08:52:44.211000+00:00
```

SDK Repo and Sample Workflow



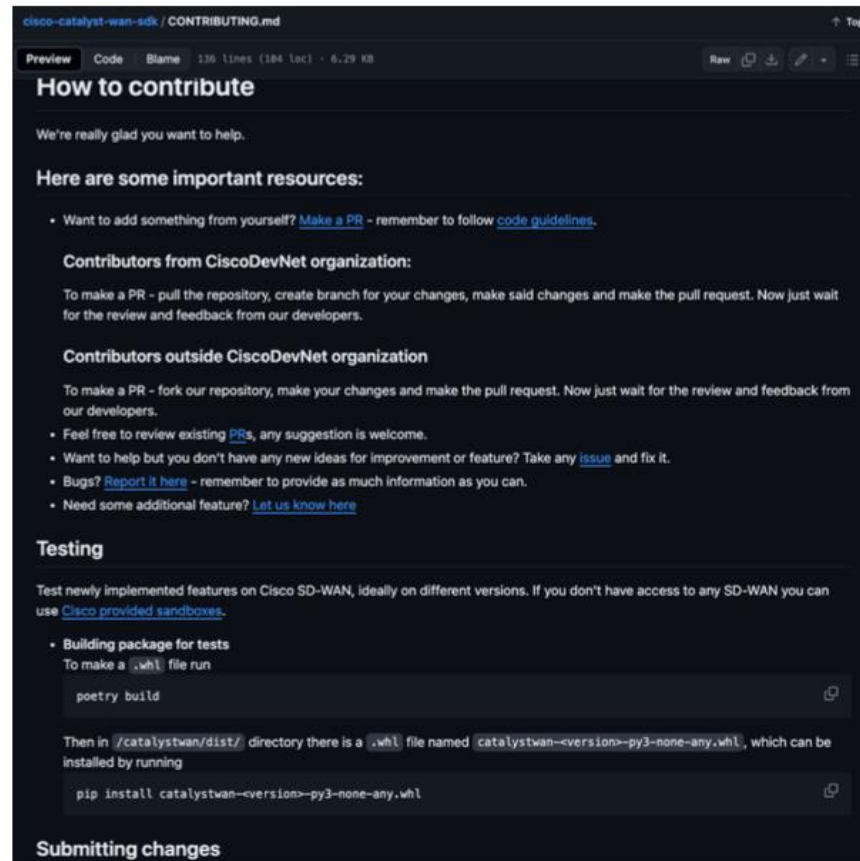
Usage and contribution

- README file
- Contribution guide
- Endpoints supported
- Usage examples in class docstrings
- Static typing
- Data models



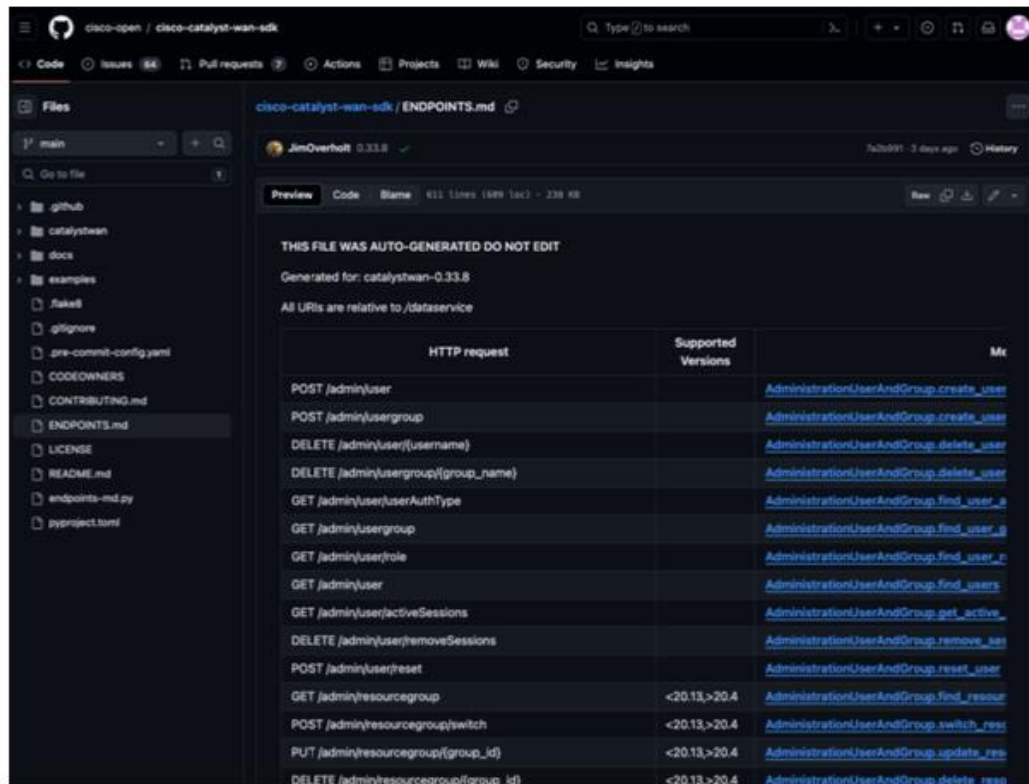
Usage and contribution

- README file
- Contribution guide
- Endpoints supported
- Usage examples in class docstrings
- Static typing
- Data models



Usage and contribution

- README file
- Contribution guide
- **Endpoints supported**
- Usage examples in class docstrings
- Static typing
- Data models



THIS FILE WAS AUTO-GENERATED DO NOT EDIT

Generated for: catalystwan-0.33.8

All URIs are relative to /dataservice

HTTP request	Supported Versions	Me
POST /admin/user		AdministrationUserAndGroup.create_user
POST /admin/usergroup		AdministrationUserAndGroup.create_user
DELETE /admin/user/{username}		AdministrationUserAndGroup.delete_user
DELETE /admin/usergroup/{group_name}		AdministrationUserAndGroup.delete_user
GET /admin/user/userAuthType		AdministrationUserAndGroup.find_user_a
GET /admin/usergroup		AdministrationUserAndGroup.find_user_g
GET /admin/user/role		AdministrationUserAndGroup.find_user_r
GET /admin/user		AdministrationUserAndGroup.find_users
GET /admin/user/activeSessions		AdministrationUserAndGroup.get_active
DELETE /admin/user/removeSessions		AdministrationUserAndGroup.remove_ses
POST /admin/user/reset		AdministrationUserAndGroup.reset_user
GET /admin/resourcegroup	<20.13,>20.4	AdministrationUserAndGroup.find_resou
POST /admin/resourcegroup/switch	<20.13,>20.4	AdministrationUserAndGroup.switch_res
PUT /admin/resourcegroup/{group_id}	<20.13,>20.4	AdministrationUserAndGroup.update_res
DELETE /admin/resourcegroup/{group_id}	<20.13,>20.4	AdministrationUserAndGroup.delete_res

Usage and contribution

- README file
- Contribution guide
- Endpoints supported
- Usage examples in class docstrings
- Static typing
- Data models

```
34 session.api.device_state.  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48
```

- get_device_control_connections_info
- get_colors
- get_bfd_sessions
- enable_data_stream
- session
- get_device_crash_info
- get_device_orchestrator_connections_info
- get_device_reboot_history
- get_device_wan_interfaces
- get_system_status
- wait_for_bfd_session_up
- wait_for_device_state

get_device_control_connections_info(device_id: str) → List[Connection]

Gets control connections for a device.

Args:
device_id: device ID (usually system-ip)

Returns:
list of Connection objects

```
34 session.api.templates.  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52
```

- compare_with_running
- create
- create_feature_template
- delete
- device_to_cli
- get
- get_feature_templates
- get_id
- get_single_feature_template
- template_validation
- __annotations__
- __class__

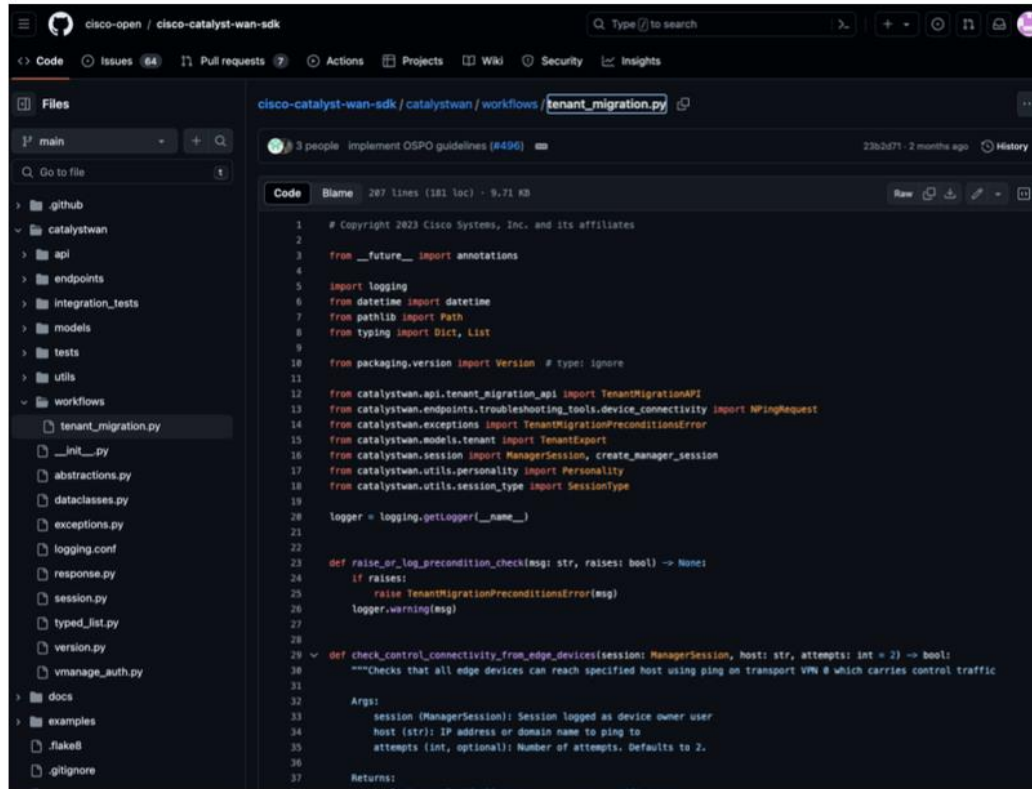
create(device_model: DeviceModel, name: str, description: str, config: CiscoConfParse) → bool

Args:
device_model (DeviceModel): Device model to create template.
name (str): Name template to create.
description (str): Description template to create.
config (CiscoConfParse): The config to device.

Raises:
TemplateAlreadyExistsError: If such template name already exists.

Returns:
bool: True if create template is successful, otherwise - False.

Sample Workflow



The screenshot shows a GitHub repository for `cisco-catalyst-wan-sdk`. The file `tenant_migration.py` is selected in the `workflows` directory. The code is a Python script for implementing OSPF guidelines. It includes imports for `__future__`, `logging`, `datetime`, `pathlib`, `typing`, `packaging.version`, and various modules from the `catalystwan` package. The script defines a `raise_or_log_precondition_check` function and a `check_control_connectivity_from_edge_devices` function.

```
1 # Copyright 2023 Cisco Systems, Inc. and its affiliates
2
3 from __future__ import annotations
4
5 import logging
6 from datetime import datetime
7 from pathlib import Path
8 from typing import Dict, List
9
10 from packaging.version import Version # type: ignore
11
12 from catalystwan.api.tenant_migration_api import TenantMigrationAPI
13 from catalystwan.endpoints.troubleshooting_tools.device_connectivity import NpingRequest
14 from catalystwan.exceptions import TenantMigrationPreconditionsError
15 from catalystwan.models.tenant import TenantExport
16 from catalystwan.session import ManagerSession, create_manager_session
17 from catalystwan.utils.personality import Personality
18 from catalystwan.utils.session_type import SessionType
19
20 logger = logging.getLogger(__name__)
21
22
23 def raise_or_log_precondition_check(msg: str, raises: bool) -> None:
24     if raises:
25         raise TenantMigrationPreconditionsError(msg)
26     logger.warning(msg)
27
28
29 def check_control_connectivity_from_edge_devices(session: ManagerSession, host: str, attempts: int = 2) -> bool:
30     """Checks that all edge devices can reach specified host using ping on transport VNM # which carries control traffic
31
32     Args:
33         session (ManagerSession): Session logged as device owner user
34         host (str): IP address or domain name to ping to
35         attempts (int, optional): Number of attempts. Defaults to 2.
36
37     Returns:
```

Demo



Resources

References

- Catalyst SDWAN Python SDK: <https://github.com/cisco-open/cisco-catalyst-wan-sdk>
- Catalyst SDWAN Terraform Provider
 - <https://registry.terraform.io/providers/CiscoDevNet/sdwan/latest>
 - <https://github.com/CiscoDevNet/terraform-provider-sdwan>
- Ansible collection: <https://github.com/cisco-open/ansible-collection-sdwan>
 - Which uses to sub repos:
 - Ansible collection: <https://github.com/cisco-open/ansible-collection-catalystwan>
 - Deployment aws (controllers+c8kv): <https://github.com/cisco-open/ansible-collection-sdwan-deployment>
- Sastre: <https://github.com/CiscoDevNet/sastre>
- sdwan-lab deployment tool: <https://github.com/cisco-open/sdwan-lab-deployment-tool>
 - <https://youtu.be/WxiZ5bxIDk8>
- Deployment - sdwan-devops: <https://github.com/CiscoDevNet/sdwan-devops>

Catalyst SD-Routing/SD-WAN – Ansible Collection

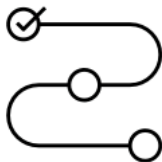
Foundation

Establish a foundation for automated network management based on SDK, Ansible modules and playbooks

- **Infrastructure Modules**
 - Ansible collections
 - Terraform Provider
- **SD-WAN Network deployment**
 - Controllers (Cloud – AWS)
 - C8Kv (Cloud – AWS)
- **Device Onboarding**
 - PNP/Zero Touch Provisioning
 - Device Onboarding
 - Day-0 device configuration
- **Device Health and Status**
- **SWIM**

Assurance and Operations

Ensure optimal performance with network health checks and monitoring



Interconnect

Extend reachability and visibility beyond basics



Integrations

Expand functionality to increase value and experiences



Ansible Collection: <https://github.com/cisco-open/ansible-collection-sdwan>

Cisco Catalyst SD-WAN Automation Awareness Series

Learn about Catalyst SD-WAN Automation with webinars and workshops

Scan the QR code or click on the following link to access the training videos:

- <https://www.youtube.com/playlist?list=PL2k86RIAekM-DFAp9NMtmrarjEC0wcwBs>

Catalyst SD-WAN REST API Basics

Catalyst SD-WAN Automation Use Cases

Sastre and Automation Toolset

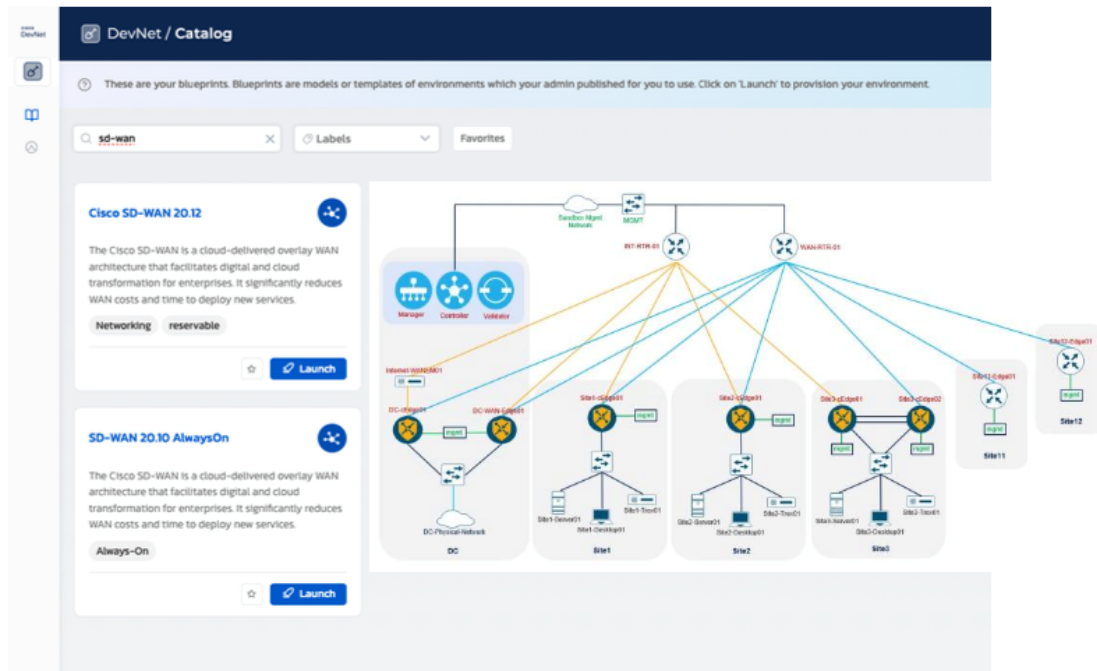


Empower engineers and developers to quickly and easily explore and create solutions based on Cisco Catalyst SD-Routing/SD-WAN by providing a ready, robust development lab environment.

- Network / infrastructure / DevOps engineers, network admins, developers

- Learn API / programmability
- Create & test your automation code

- [Catalyst SD-WAN Sandbox](#)
- [SD-WAN Programmability Learning Lab](#)



Complete Your Session Evaluations



Complete a minimum of 4 session surveys and the Overall Event Survey to claim a **Cisco Live T-Shirt**.



Complete your surveys in the **Cisco Live mobile app**.



Continue your education

- Visit the Cisco Stand for related demos
- Book your one-on-one Meet the Expert meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

Contact me at: @aidevnet



Thank you

CISCO *Live!*

#CiscoLiveAPJC

CISCO *Live!*

GO BEYOND

#CiscoLiveAPJC