

CISCO *Live!*



#CiscoLive



The bridge to possible

The 12 facets of the OpenAPI standard

Getting to world-class APIs

Stève Sfartz, Principal Architect
DEVNET-1346



#CiscoLive

Cisco Webex App

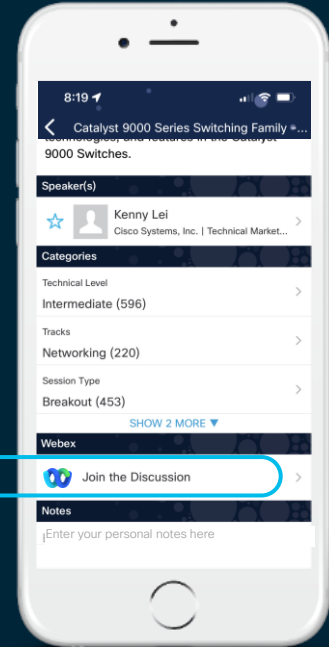
Questions?

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 June 17, 2022.



<https://ciscolive.ciscoevents.com/ciscolivebot/#DEVNET-1346>

/Cisco/DevNet/StèveSfartz

- API Architect in Cisco Developer Relations
- Technical Lead for API Experience and Cisco API Guidelines: prescriptive standards and best practices for API & SDKs
- Working to deliver a great and consistent developer experience across Cisco platforms



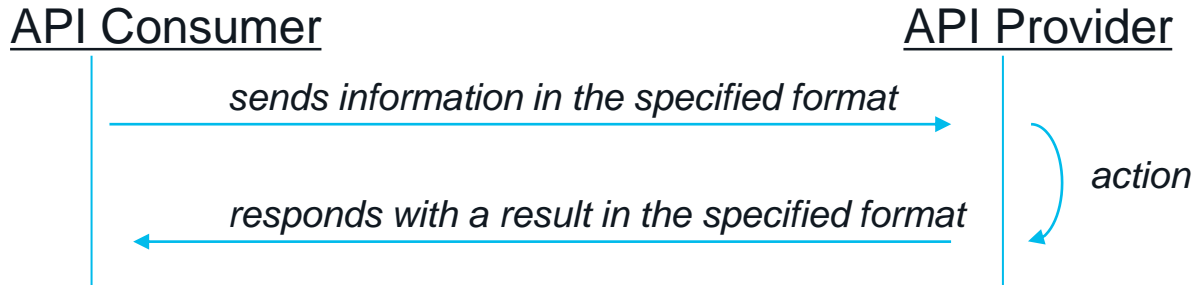
webex: stsfartz@cisco.com
github: [ObjectIsAdvantag](#)
twitter: [@SteveSfartz](#)

“vision
without
execution is
hallucination”


#1. OpenAPI to Describe API Contracts

APIs as Contracts

- An application programming interface (API) specifies how software components should interact with each other.
- As such, APIs are considered **contracts between the organization providing the API and developers consuming this API.**
- *"if you provide information in this format, I – the API - will perform a specific action and return a result in this format".*



Formalizing API Contracts

- For every [operation](#) supported by an API, its contract describes:
 - what must be provided as input
 - what will happen
 - and what, if any, data is returned as a result
- [OpenAPI Specification](#) (OAS) is a standard to define contracts for Web API
- Example of OAS document 

```
1  openapi: 3.0.0
2  info:
3    title: Deck of Cards API
4    description: |
5      An API to simulate a deck of cards. This API can be used
6      may be created using a single deck of 52 cards, or multiple
7      of cards.
8
9      This API supports the following capabilities -
10
11      * Create a new deck, shuffled or unshuffled (order preserved)
12      * Reshuffle an existing deck
13      * Draw one or more cards from an existing deck
14      * Create piles of cards that can receive a draw
15      * Create a new deck with a specific set of cards
16
17      The API offers images of each card for easy display.
18
19  contact: {}
20  version: '1.0'
21  servers:
22    - url: http://www.deckofcardsapi.com/api/
23      variables: {}
24  paths:
25    "/deck/new/": {
26      get: {
27        summary: "Create a new deck"
28      }
29    }
30    "/deck/{deck_id}/draw/": {
31      get: {
32        summary: "Draw one or more cards from an existing deck"
33      }
34    }
35    "/deck/{deck_id}/shuffle/": {
36      get: {
37        summary: "Reshuffle an existing deck"
38      }
39    }
40  components:
41    schemas:
```

#2. OAS Documents Editing

Editing with SwaggerEditor

<https://editor.swagger.io/>

The image shows the Swagger Editor interface. On the left, the OpenAPI specification is displayed in a code editor. The specification is for the Meraki Dashboard API, version 1.22.0. It includes an info section with the title 'Meraki Dashboard API', a description, and contact information. The servers section lists the API endpoint as 'https://api.meraki.com/api/v1'. The security section indicates that the API uses the 'meraki_api_key' for authentication. The components section is currently empty. The paths section defines a GET endpoint for '/organizations' with a description 'List the organizations that the user has privileges on' and a successful response (200) with a description 'Successful operation' and a schema of type '*/*'.

Swagger Editor
Supported by SMARTBEAR

File Edit Insert Generate Server Generate Client

```
1 openapi: 3.0.0
2 info:
3   version: v1.22.0
4   title: Meraki Dashboard API
5   description: >
6     The Cisco Meraki Dashboard API is a modern REST API based on the
7     OpenAPI
8     specification. Date: 01 June, 2022. [Recent
9     Updates](https://meraki.io/whats-new/)
10  contact:
11    name: Meraki Developer Community
12    url: https://meraki.io/community
13  servers:
14    - url: https://api.meraki.com/api/v1
15  security:
16    - meraki_api_key: []
17  components:
18  paths:
19    /organizations:
20      get:
21        description: List the organizations that the user has privileges
22        on
23        operationId: getOrganizations
24        responses:
25          '200':
26            description: Successful operation
27            content:
28              '*/*':
29                schema:
```

Meraki Dashboard API v1.22.0 OAS3

The Cisco Meraki Dashboard API is a modern REST API based on the OpenAPI specification.
Date: 01 June, 2022. [Recent Updates](#)

[Meraki Developer Community - Website](#)

Servers

Authorize

read

GET **/organizations** List the organizations that the user has privileges on

Demo

#3. OAS Documents to Publish Reference Documentation

API Reference Documentation

developer.cisco.com/meraki/api-v1/#!/create-organization

- Reference documentation automatically rendered from OAS documents

```
post:
  description: Create a new organization
  operationId: createOrganization
  parameters:
    - name: createOrganization
      in: body
      schema:
        type: object
        properties:
          name:
            type: string
            description: The name of the organization
        example:
          name: My organization
        required:
          - name
      required: true
  responses:
    '201':
      description: Successful operation
```



API	>
GENERAL	>
devices	>
networks	>
organizations	>
CONFIGURE	>
GET	Get Organizations
POST	Create Organization
GET	Get Organization
PUT	Update Organization
DELETE	Delete Organization
actionBatches	>
adaptivePolicy	>
admins	>
alerts	>
brandingPolicies	>
POST	Claim Into Organization
clients	>
POST	Clone Organization
configTemplates	>
devices	>

Create Organization

Operation Id: createOrganization

Description: Create a new organization

POST /organizations

Request Parameters

Body

createOrganization * | Object

Schema Definition

Example Body

```
{ "name": "My organization" }
```

Responses

Status: 201

#4. OAS Documents to Try out API Operations

Get Organizations

Operation Id: `getOrganizations`

Description: List the organizations that the user has privileges on

GET /organizations

Responses

Status: 200

Successful operation

Schema Definition

Example Body

-	array[]	+ -
-	id: string	Organization ID
-	name: string	Organization name
-	url: string	Organization URL
+	api: object	API related settings
+	cloud: object	Data for this organization
+	licensing: object	Licensing related settings

Parameters

Template

GET /organizations

Query Params

Headers

Run

Response: 200 OK

Data

Info

```
[
  {
    "id": "681155",
    "name": "DeLab",
    "url": "https://n392.meraki.com/o/49Gm_c/manage/orga
    "api": { "enabled": true },
    "licensing": { "model": "per-device" },
    "cloud": {
```

#5. OAS Documents to Generate Code

Auto-generate client code

- From the reference documentation itself
- Pick among the proposed languages
python

```
import requests

url="https://api.meraki.com/api/v1/organizations"

payload=None

headers={
    "Content-Type": "application/json",
    "Accept": "application/json",
    "X-Cisco-Meraki-API-Key": "6bec40c...9ea0"
}

response=requests.request('GET', url,
headers=headers, data = payload)

print(response.text.encode('utf8'))
```

Get Organizations

Operation Id: *getOrganizations*

Description: *List the organizations that the user has privileges on*

GET /organizations

Responses

Status: 200

Successful operation

Schema Definition

Example Body

array[]

- id: string Organization ID
- name: string Organization name
- url: string Organization URL
- + api: object API related settings
- + cloud: object Data for this organization
- + licensing: object Licensing related settings

Auto-generate client code

- Using a CLI tool
- For your preferred language

python

```
import requests

url="https://api.meraki.com/api/v1/organizations"

payload=None

headers={
    "Content-Type": "application/json",
    "Accept": "application/json",
    "X-Cisco-Meraki-API-Key": "6bec40c...9ea0"
}

response=requests.request('GET', url,
headers=headers, data = payload)

print(response.text.encode('utf8'))
```



```
/organizations:
  get:
    description: List the organizations th
    privileges on
    operationId: getOrganizations
    responses:
      '200':
        description: Successful operation
        schema:
          type: array
          items:
            type: object
            properties:
              id:
                type: string
                description: Organization
              name:
                type: string
                description: Organization
              url:
                type: string
                description: Organization
              api:
                type: object
                properties:
```


Auto-generate client or server code

- Auto-generate **client code** that consumes the API
 - Target your script or app programming language
- Auto-generate **server code** as a skeleton of the API
 - Target the programming language used by engineering groups
 - Useful to create mock servers










```
30934 tags:
30938 /organizations:
30939 get:
30940   description: List the organizations that the user has
30941               privileges on
30942   operationId: getOrganizations
30943   responses:
30944     '200':
30945       description: Successful operation
30946       schema:
```


Mocking APIs


 SMARTBEAR
SwaggerHub™


CiscoLive has 11 days left in **Enterprise Trial** [Upgrade](#)


   ObjectIsAdvantag ▾

← Inventory ▾ 1.0.0 ▾     Export ▾

 **Info**

 **Tags**



 **Servers**

 Search

admins ^
POST /inventory

developers ^
GET /inventory

Schemas ^ +
SCHEMA InventoryItem
SCHEMA Manufacturer

Aa   [SAVE](#) ▾ [SYNC](#)

```
20 paths:
21   /inventory:
22     get:
23       tags:
24         - developers
25       summary: searches inventory
26       operationId: searchInventory
27       description: |
28         By passing in the appropriate options, you can
29         search for
30         available inventory in the system
31       parameters:
32         - in: query
33           name: searchString
34           description: pass an optional search string
35             for looking up inventory
36           required: false
37           schema:
38             type: string
39         - in: query
40           name: skip
41           description: number of records to skip for
42             pagination
43           schema:
44             type: integer
45             format: int32
46             minimum: 0
47         - in: query
48           name: limit
49           description: maximum number of records to
50             return
```

Servers

https://virtserver.swaggerhub.com/CiscoLive/Mera... ▾

https://virtserver.swaggerhub.com/CiscoLive/MerakiLight/1.0.0 - SwaggerHub API Auto Mocking

admins Secured Admin-only calls ▾

developers Operations available to regular developers ^

GET /inventory searches inventory ^ ↺

By passing in the appropriate options, you can search for available inventory in the system

Parameters

Cancel

Name	Description
searchString	pass an optional search string for looking up inventory

Mocking APIs

```
responses:
  '200':
    description: search results matching
      criteria
    content:
      application/json:
        schema:
          type: array
          items:
            $ref: '#/components/schemas
              /InventoryItem'
```

```
InventoryItem:
  type: object
  required:
    - id
    - name
    - manufacturer
    - releaseDate
  properties:
    id:
      type: string
      format: uuid
      example: d290f1ee-6c54-4b01-90e6
        -d701748f0851
    name:
```



Request URL

<https://virtserver.swaggerhub.com/CiscoLive/MerakiLight/1.0.0/inventory>

Server response

Code

Details

200

Response body

```
[
  {
    "id": "d290f1ee-6c54-4b01-90e6-d701748f0851",
    "name": "Widget Adapter",
    "releaseDate": "2016-08-29T09:12:33.001Z",
    "manufacturer": {
      "name": "ACME Corporation",
      "homePage": "https://www.acme-corp.com",
      "phone": "408-867-5309"
    }
  }
]
```

Summary

OpenAPI Specification to

- #1. Describe API Contracts
- #2. Edit OAS documents
- #3. Publish Documentation
- #4. Try out API Operations
- #5. Generate Client/Server Code

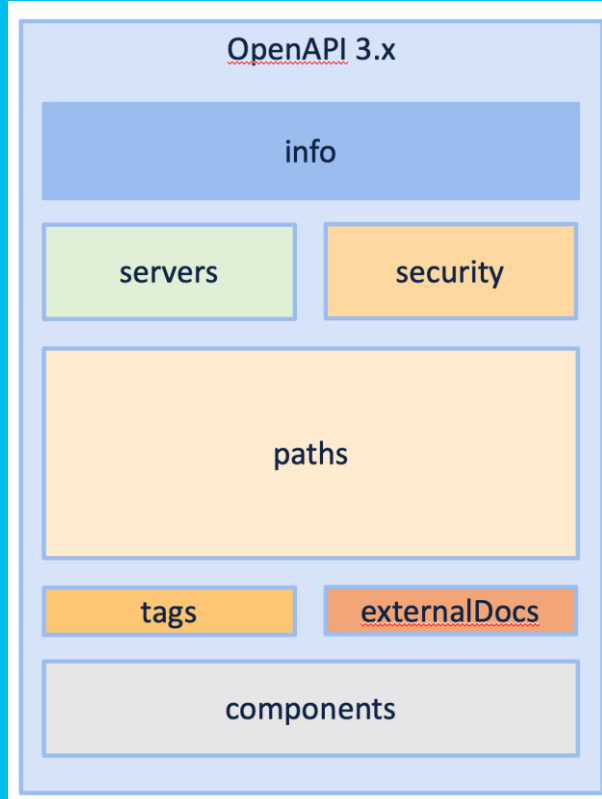
#6. The OpenAPI Specification

OpenAPI Specification (OAS)

swagger.io/specification/

- Defines a standard, language-agnostic interface to RESTful APIs which allows both humans and computers to discover and understand the capabilities of the service without access to source code, documentation, or through network traffic inspection.
- When properly defined, a consumer can understand and interact with the remote service with a minimal amount of implementation logic.
- An OpenAPI definition can then be used by documentation generation tools to display the API, code generation tools to generate servers and clients in various programming languages, testing tools, and many other use cases.

Elements of OAS



- **Info:** Provides details about the API, including a title and description
- **Security:** Specifies authorization. Required for interactive documentation
- **Paths:** What the API can do. Defined by a path + HTTP method (aka verb) + input parameters + one or more response details
- **Components:** Capture reusable elements that may be referenced within and across OAS files. Includes schema, headers, security details

OAS Elements: Info Example

info

```
openapi: 3.0.3
info:
  title: Cisco SD-WAN vManage API - Device State and Statistics Part
  description: The vManage API exposes the functionality of operations
    maintaining devices and the overlay network
  contact:
    email: vmanage-api@cisco.com
  license:
    name: Commercial License
    url: 'https://www.cisco.com/c/en/us/solutions/enterprise-networks/sd-wan
      /index.html'
  version: 2.0.0
```

OAS Elements: Servers (aka Hosts)

servers

- Describes one or more API endpoints for cloud and on-premises APIs
- They may be a complete URL or use variable substitution

```
servers:  
- url: https://development.gigantic-server.com/v1  
  description: Development server
```

```
servers:  
- url: https://{username}.gigantic-server.com:{port}/{basePath}  
  description: The production API server  
  variables:  
    username:  
      # note! no enum here means it is an open value  
      default: demo  
      description: this value is assigned by the service provider, in this example `gigantic-server.com`
```

OAS Elements: Paths


- Operation = Path
+ HTTP Method
- Query parameter
 - GET /alarms?active=true
- 200: Response with payload

```
paths:  
  /alarms:  
    get:  
      tags:  
        - Monitoring - Alarms Details  
      description: Get alarms for last 30min  
      operationId: getAlarms  
      parameters:  
        - name: query  
          in: query  
          description: Alarm query string  
          schema:  
            type: string  
          example:  
            query:  
              field: active  
              type: boolean  
              value:  
                - 'true'  
              operator: equal  
      responses:  
        '200':  
          description: Success  
          content:  
            application/json:  
              schema:  
                type: object
```


OAS Elements: Schema Components

components

- Define reusable structures for request inputs and response outputs.
- Often referenced from operations or from one schema to another



```
category:  
  $ref: '#/components/schemas/Category'
```



```
Category:  
  type: object  
  properties:  
    id:  
      type: integer  
      format: int64  
      example: 1  
    name:  
      type: string  
      example: Dogs
```

#7. OpenAPI Specification as a Standard

The OpenAPI Initiative Charter

<https://www.openapis.org/>



1. Mission of the OpenAPI Initiative

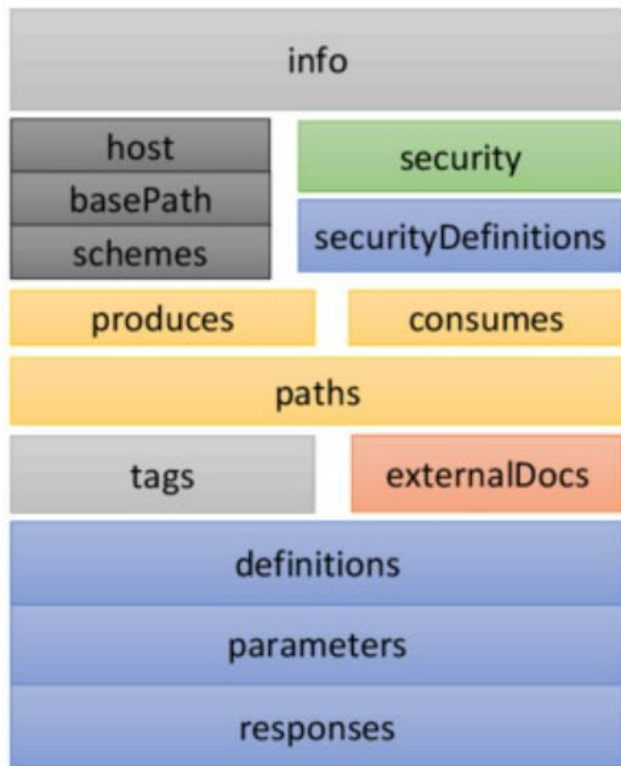
The OpenAPI Initiative (“OAI”) provides an open source community, within which industry participants may easily contribute to building vendor-neutral, portable and open specifications for providing technical metadata for APIs – such as the “OpenAPI Specification” (collectively, “Specifications”) – and supporting tooling for validating the integrity of the specifications or instantiations of it. The OAI is as such not intended to be a destination for community/consumer-focused tooling outside of the specification itself.

2. Membership

- The OpenAPI Initiative shall be composed of:
 - i. corporate members (OAI Members) that have executed an OAI Membership Agreement to sponsor the activities of the OAI Community
 - ii. a Business Governance Board (BGB);
 - iii. an open source, Technical Steering Committee (“TSC”), open to any participant, whether an OAI Member or not; and
 - iv. a Technical Oversight Board (“TOB”).

#8. OpenAPI Specification Versions

Versions of the OpenAPI Specifications



OpenAPI 2.0 - 2014



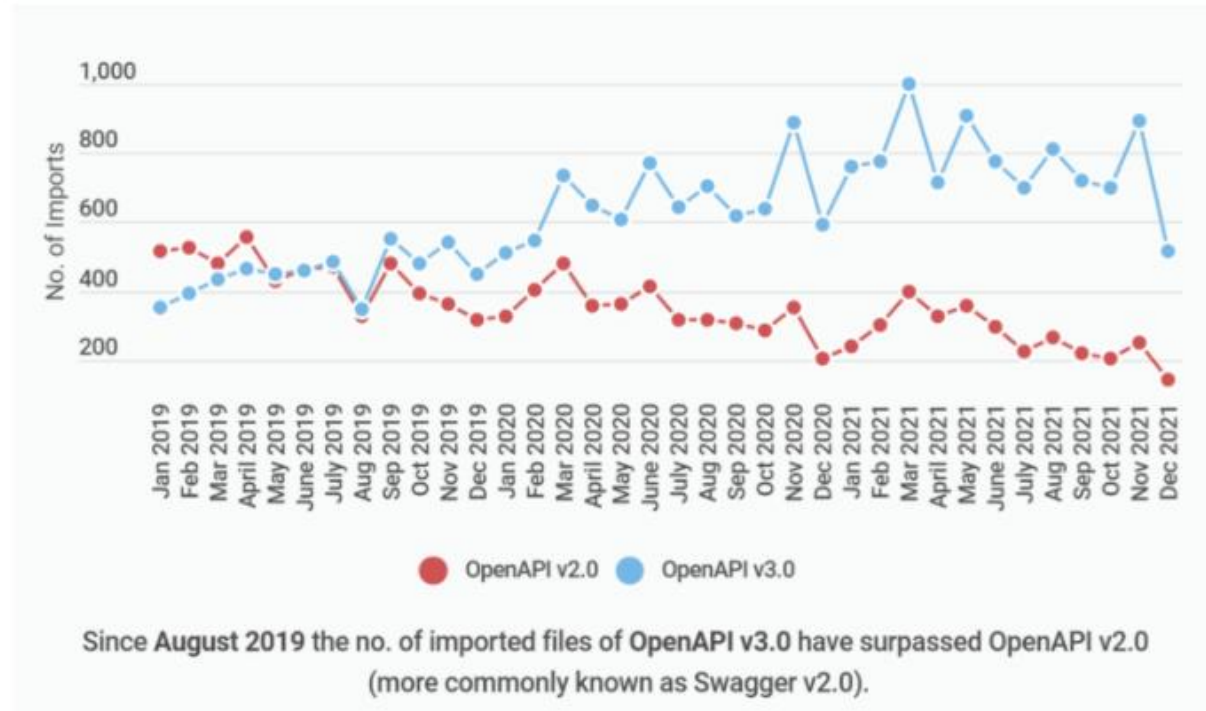
OpenAPI 3.0 - 2017

full compatibility
with modern
JSON Schema
[Draft 2020-12]

OpenAPI 3.1 - 2021

OAS v3.0 as the default import format

<https://www.apimatic.io/blog/2022/03/top-api-specification-trends-2019-2022/>

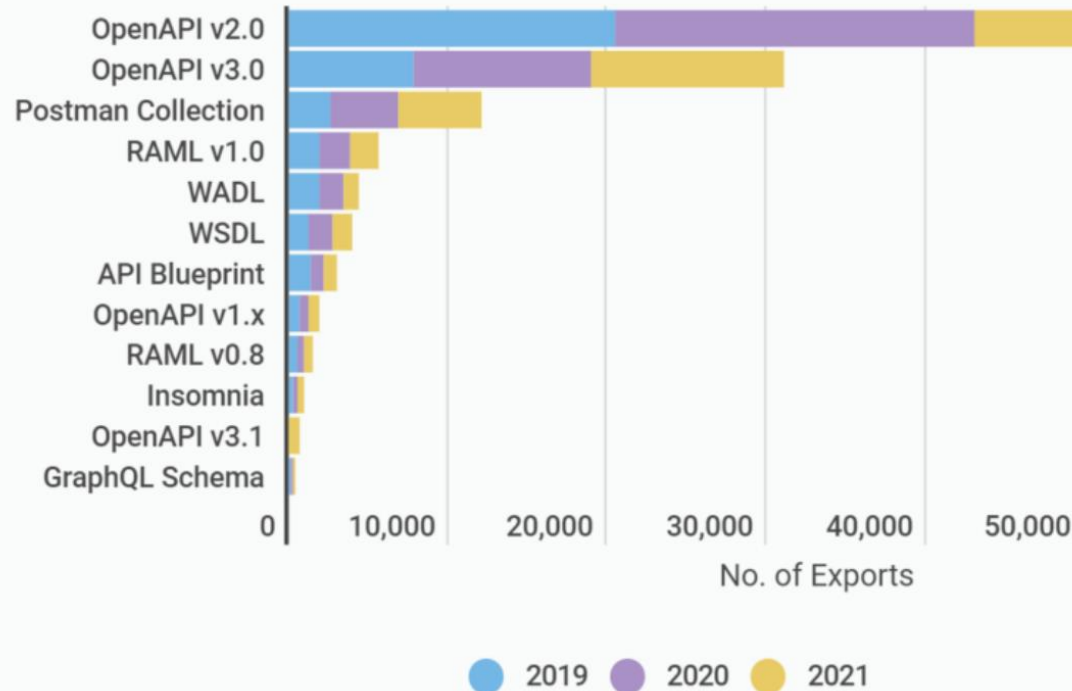


“Since August 2019, the number of imported OAS v3.0 documents has surpassed OAS v2.0”

APIMatic March 2022

but... OAS v2.0 still largely spread

API Specification Transformation Trends



“50% users preferred to convert to OpenAPI v2.0 over 3.0”

APIMatic March 2022

- quality of support for OpenAPI v3.0 in tools
- legacy tools that still supports v2.0 only.

#9. OAS Documents Formats

JSON vs. YAML: Side-by-side

```
1 {
2   "openapi": "3.0.3",
3   "info": {
4     "title": "Cisco SD-WAN vManage API - Device State and Statistics",
5     "description": "The vManage API exposes the functionality of openapi exposing the state of the devices and the overlay network",
6     "contact": {
7       "email": "vmanage-api@cisco.com"
8     },
9     "license": {
10      "name": "Commercial License",
11      "url": "https://www.cisco.com/c/en/us/solutions/enterprise-networks/whitepapers/cisco-sd-wan-vmanage-api-device-state-and-statistics.html/index.html"
12    },
13    "version": "2.0.0"
14  },
15  "servers": [
16    {
17      "url": "/dataservice"
18    }
19  ],
20  "paths": {
21    "/alarms": {
22      "get": {
23        "tags": [
24          "Monitoring - Alarms Details"
25        ],
26        "description": "Get alarms for last 30min if vManage query is successful",
27        "operationId": "getAlarms",
28        "parameters": [
29          {
30            "name": "query",
31            "in": "query",
32            "description": "Alarm query string",
```

```
1 openapi: 3.0.3
2 info:
3   title: Cisco SD-WAN vManage API - Device State and Statistics
4   description: The vManage API exposes the functionality of openapi exposing the state of the devices and the overlay network
5   contact:
6     email: vmanage-api@cisco.com
7   license:
8     name: Commercial License
9     url: 'https://www.cisco.com/c/en/us/solutions/enterprise-networks/whitepapers/cisco-sd-wan-vmanage-api-device-state-and-statistics.html/index.html'
10  version: 2.0.0
11  servers:
12    - url: /dataservice
13  paths:
14    /alarms:
15      get:
16        tags:
17          - Monitoring - Alarms Details
18        description: Get alarms for last 30min if vManage query is successful
19        operationId: getAlarms
20        parameters:
21          - name: query
22            in: query
23            description: Alarm query string
24          schema:
25            type: string
26          example:
27            query:
28              field: active
29              type: boolean
30              value: true
```

YAML Syntax

String

```
"Create Shuffled Deck"
```

Dictionary

```
post:
  summary: "Create Shuffled Deck"
  operationId: "NewShuffledDeck"
  description: "Create a shuffled deck of cards."
  tags: [ "setup" ]
```

Array (with hyphens)

```
tags:
  - name: "setup"
  - name: "play"
```

Condensed Array (square brackets)

```
tags: [ "play", "setup" ]
```

Array of objects

(notice the hyphens for each item)



```
parameters:
  - name: "deck_id"
    in: path
    description: "The Deck ID to shuffle."
    required: true
    schema:
      type: string
      example: "8obqzdtk9tau"
  - name: "count"
    in: query
    description: "The number of cards to draw.
      specified."
    required: false
    style: form
    explode: true
    schema:
      type: integer
      format: int32
      default: 1
```

#10. OpenAPI Specification Terminology

Terminology

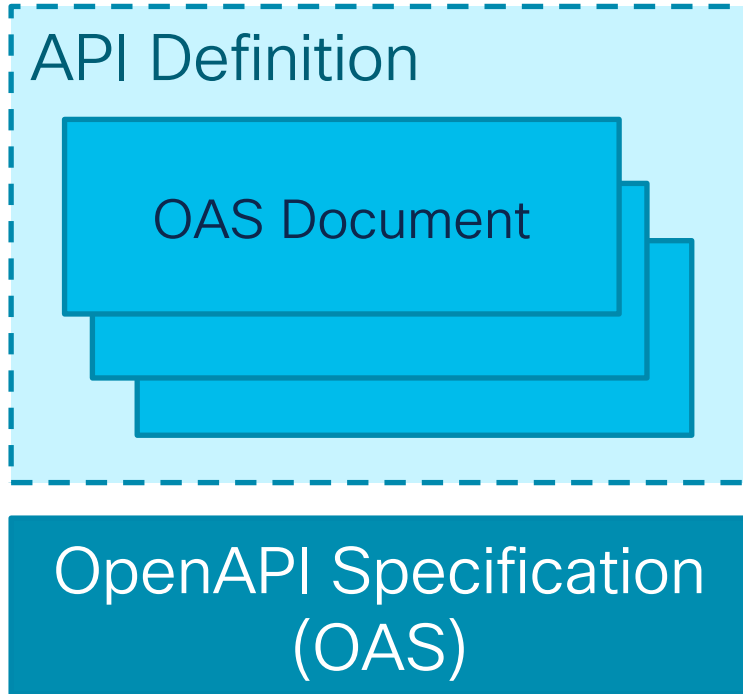
OAS Document

An **OAS Document** contains the description of the full set or a subset of the API features. OAS document should be read as in *“a document that conforms to the OpenAPI Specification”*

OpenAPI Specification
(OAS)

OAS is a standard used to document the contract for HTTP/REST APIs

Terminology



An **API Definition** corresponds to the full contract of an API. Concretely the API Definition is a set of OAS documents.

Often, one OAS document will be enough to fully define the API. In this latter case, the API definition and the OAS document could be interchanged.

An **OAS document** contains the description of the full set or a subset of the API features. OAS document should be read as in *“a document that conforms to the OpenAPI Specification”*

OAS is a standard used to document the contract for HTTP/REST APIs

OpenAPI Terminology

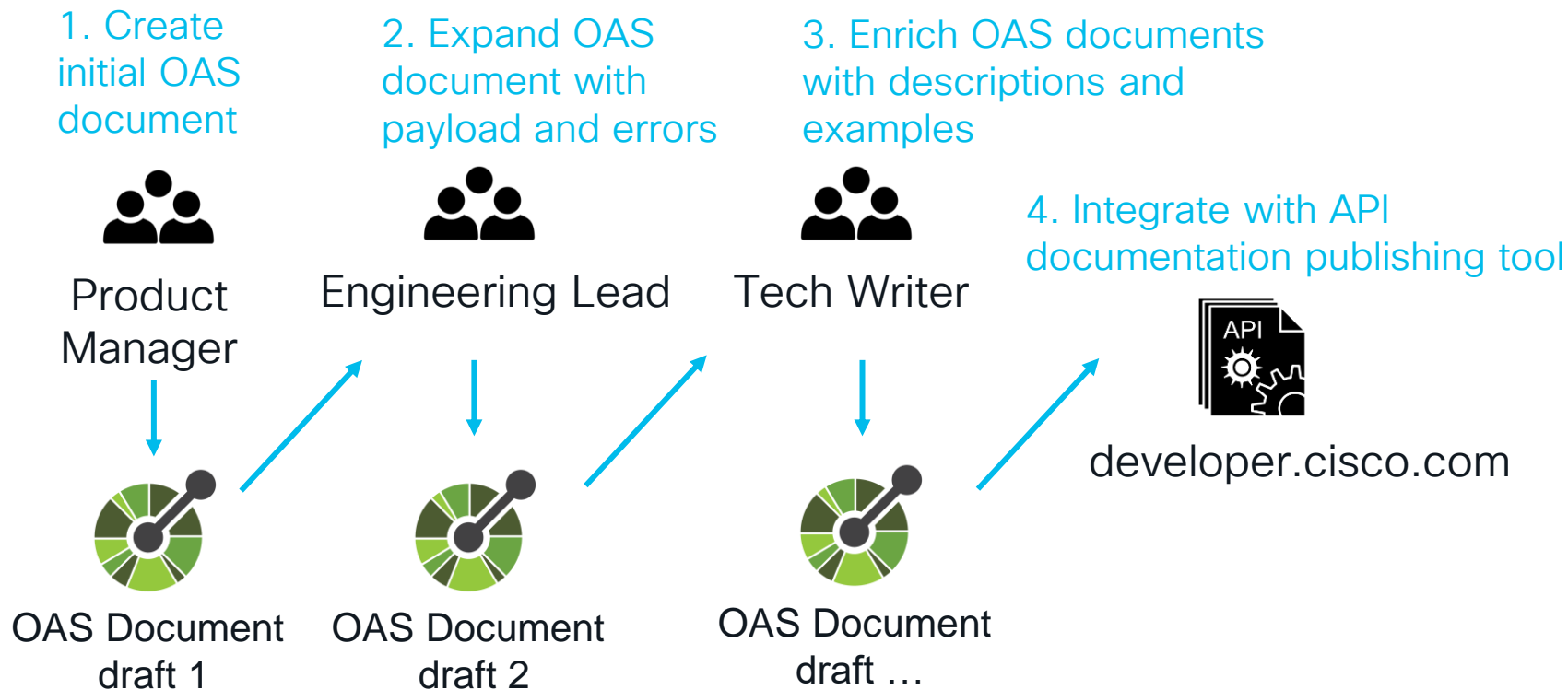
- OpenAPI Specification, OpenAPI Initiative, OpenAPI Tools, ~~OpenAPI~~
- OAS Document: describe an API using the OpenAPI Specification
- API Contract, API Definition, API Specification/Spec, API Description
- API Documentation: the reference documentation for an API, published on a web site, and kept in sync with a version of an API
- API Endpoint, ~~API~~: the URL at which an API version can be accessed, such as 'https://api.meraki.com/v1/'
- API Path, REST Resource, ~~API~~: a URL such as '/organizations'
- API Operation, ~~API~~: a Path + a method such as 'GET /organizations'
- SDK, Client Library, ~~API~~: ready-to-use code to consume an API

API Guidelines

- An API is a **network programmatic interface that a product** – may be bare metal hardware, or virtual machine or software – AND – may be cloud or on-premises – **publishes**.
- It has **versions** – it's the API lifecycle
- **For every update, an API would publish its contract** as one or multiple OAS documents for download or online browsing.
- Every API version provides a documentation which includes authentication instructions, developer guides, code samples and reference documentation... and an API changelog.

#11. OAS Documents Management & Lifecycle

OAS Document Lifecycle



OAS Workflow Best Practices

Single Artifact

Define **where** the OAS documents of your API will be stored

- Single source of truth
- OAS documents should be checked into a git repo to track changes

Clarify Strategy

Define **who** is responsible to merge changes

- Whether a product manager, technical writer or technical lead – be consistent
- Use GitHub pull requests for tracking and merging changes

Educate the Team

Educate your team members on OAS

- OAS fundamentals
- OAS workflow
- OAS toolsets (linters, code generators...)

Refine the Process

Practice and refine as needed

- Update OAS documents, review PR and merge changes
- Maintain an API Changelog
- This workflow may take time to establish

#12. Keeping OAS Documents accurate

Code as the source of truth

Convert code comments or annotations

<https://openap.is/>



```
/* @oas [post] /pets/{category}
 * description: "List all pets in a category"
 * parameters:
 * - (path) category=all* {String} Pet category
 * - (query) limit {Integer:int32} Amt returned
 * - (body) search {String} Search pet details
 * - (body) strict {Boolean} Exact matches?
 */
```

```
routes.get('/pets/:category', getPets);
```

```
paths:
  "/pets/{category}":
    post:
      description: List all pets in a category
      parameters:
        - in: path
          name: category
          required: true
          description: Pet category
          schema:
            type: string
            default: all
        - in: query
          name: limit
          description: Amt returned
          schema:
            type: integer
            format: int32
      responses:
        '200':
          description: Successful response
      requestBody:
        content:
          application/json:
            schema:
              type: object
              properties:
                search:
                  type: string
                  description: Search pet details
                strict:
                  type: boolean
                  description: Exact matches?
```

#13. OpenAPI to Automate Design Reviews

OAS Documents Static Analysis

Detecting Quality or Security Faults

```
19
20 paths:
21   /pet:
22     post:
23
24       summary: Add a new pet to the store
25       description: ''
26       operationId: addPet
27       requestBody:
28         $ref: '#/components/requestBodies/Pet'
29       responses:
30         '405':
31           description: Invalid input
32       security:
33         - petstore_auth:
34           - write:pets
35           - read:pets
```

PROBLEMS 9 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER Filter (e.g. text, **/*.ts, !**/node_modules/**)

! mini_petstore_v3.yaml 9

- ⚠ Operation "description" must be present and non-empty string. spectral(operation-description) [Ln 25, Col 20]
- ⚠ Operation must have at least one "2xx" or "3xx" response. spectral(operation-success-response) [Ln 29, Col 17]
- ⚠ Operation "description" must be present and non-empty string. spectral(operation-description) [Ln 42, Col 20]
- ⚠ Operation must have at least one "2xx" or "3xx" response. spectral(operation-success-response) [Ln 46, Col 17]

Ln 19, Col 2

**Spectral, an Open Source
JSON/YAML Linter**



#14. OpenAPI to Generate API Changelogs

+ Added: GET /catalogue/personalize

~ Modified: GET /catalogue/{id} Breaking

Parameters:

Modified: id in path

catalogue id

Code Diff

```
1 {
2   "description": "get",
3   "operationId": "Get-an-item",
4   "parameters": [
5     {
6       "description": "catalogue id",
7       "in": "path",
8       "maxLength": 255,
9       "minLength": 1,
10      "name": "id",
11      "pattern": "[^/]*",
12      "required": true,
13      "type": "string"
14    }
15  ],
16  ...

```

```
1 {
2   "description": "get",
3   "operationId": "Get-an-item",
4   "parameters": [
5     {
6       "description": "catalogue id",
7+      "format": "double",
8       "in": "path",
9+      "maximum": 255,
10+     "minimum": 1,
11      "name": "id",
12      "required": true,
13+     "type": "number"
14    }
15  ],
16  ...

```

API Changelog

- One operation got added
- A breaking change detected




#15. OAS Documents to Detect Drifts

Drifts

Comparing OAS documents to live traffic observations



3 Error 0 Warning 0 Info 0 Hint

Severity ▼	Findings	Recommendation
 Error	General diff: a general diff has been detected <div>28 items affected ▼</div>	Please update the spec accordingly.
 Error	Shadow: an undocumented API has been detected <div>11 items affected ▼</div>	Please add the API in the spec.
 Error	Zombie: a deprecated API has been detected <div>11 items affected ▼</div>	Please update clients to stop using deprecated APIs.

Shadow Paths

Shadow: an undocumented API has been detected

Solution

Please add the API in the spec.

```
1 Path: /tags
```

```
2-PathItem: null
```

```
1 Path: /tags
```

```
2+PathItem:
```

```
3+ get:
```

```
4+   produces:
```

```
5+     - application/json; charset=utf-8
```

```
6+   responses:
```

```
7+     "200":
```

```
8+       description: ""
```

```
9+       schema:
```

```
10+         properties:
```

```
11+           err:
```

```
12+             type: string
```

```
13+           tags:
```

```
14+             items:
```

```
15+               type: string
```

```
16+             type: array
```

```
17+           type: object
```

```
18+         default:
```

```
19+           description: Default Response
```

```
20+           schema:
```

```
21+             properties:
```

Undocumented Query Parameters

Solution

Please update the spec accordingly.

```
1 Path: /v1/catalogue
2 PathItem:
3   get:
4     description: Catalogue API
5     operationId: List-catalogue
6     parameters:
7       - description: query
8         format: double
9         in: query
10        maximum: 255
11        minimum: 1
12        name: max
13        type: number
14       - description: query
15         format: double
16         in: query
17         maximum: 255
18         minimum: 1
19         name: offset
20         type: number
21       - description: query
```

```
1 Path: /v1/catalogue
2 PathItem:
3   get:
4     parameters:
5+    - in: query
6+      name: size
7+      type: integer
```

Zombie: Deprecated Paths still active

Solution

Please update clients to stop using deprecated APIs.

```
1 Path: /v1/catalogue/size
2 PathItem:
3   get:
4     deprecated: true
5     description: Get size
6     operationId: Get-size
7   parameters:
8     - description: query
9       format: double
10      in: query
11      maximum: 255
12      minimum: 1
13      name: max
14      type: number
15     - description: query
16       format: double
17       in: query
18       maximum: 255
19       minimum: 1
20       name: offset
21       type: number
```

```
1 Path: /v1/catalogue/size
2 PathItem:
3   get:
4     parameters:
5+    - in: query
6+      name: tags
```


#16. OpenAPI for Lifecycle and Compliance Management

API Contract Health for Governance Officers

Trust



Highly Reliable

(Lifecycle with deprecation notices, complete & accurate definition, complete changelog)



Versioned

(API-specific lifecycle, definition published with large coverage, complete changelog)



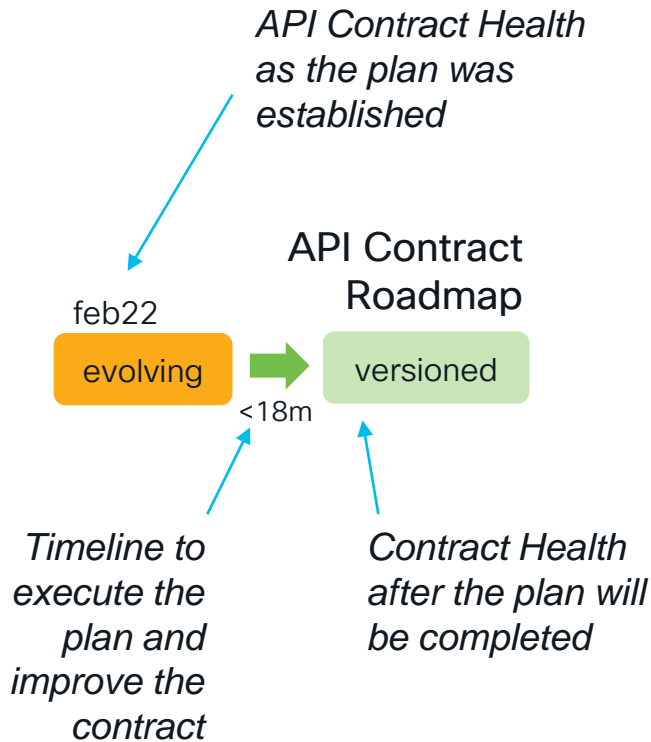
Evolving

(Product-tied versions, changelogs and contract may not be complete, ie typically UI-led design)



Unreliable

(Breaking changes, no or partial changelog, typically unstructured or UI-led design)



Organization Dashboard

+ Add New Service

DevRel Store

Legend



> 90 Excellent



80-89 Very Good



70-79 Good



60-69 At Risk



50-59 Warning



1-49 Alert

100

Catalogue Demo API v0.1 r2

Updated At Jun 06, 20:07



Catalogue microservice for DevRel Store demo application

87

Cart Demo API v0.0 r2

Updated At Jun 06, 02:48



Cart microservice for DevRel Store demo application

engineering@merchandiseshop.com

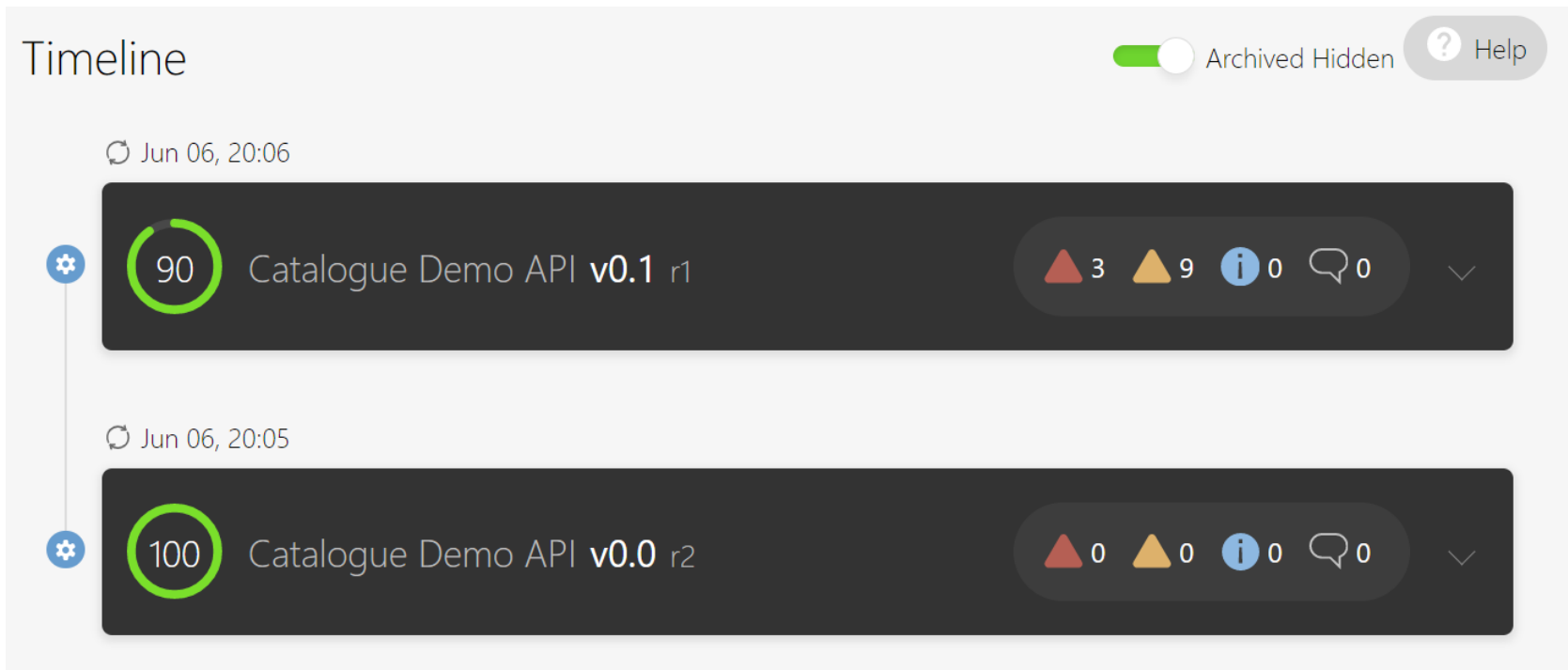
https://cs.co/devrel-wear-demo

engineering@merchandiseshop.com

https://cs.co/devrel-wear-demo

<https://developer.cisco.com/apix-manager/>

Lifecycle of OAS Documents



⚙ Analyzing spec v0.1 r1

for compliance with ☒ Completeness ☒ REST Guidelines ☒ Security ☒ Inclusive Language

▲ 3 Error ▲ 9 Warning ⓘ 0 Info 💬 0 Hint

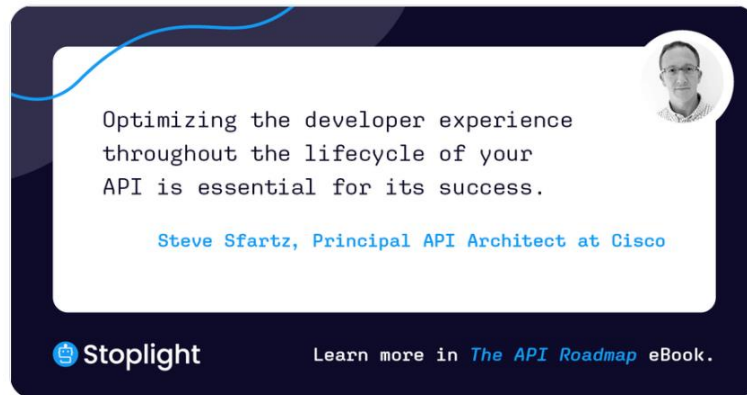
INTDEV-1250
Tuesday 4PM
Grace Francisco

Analyzer	Severity▼	Findings	Recommendation
Completeness	▲ Error	Some operations do not define errors. (Line 483)	Please add an error status code for the items identified.
REST Guidelines	▲ Error	A 401 status code is returned when authentication fails. (Line 483)	Please add a 401 status code for the items identified.
REST Guidelines	▲ Error	A 403 status code is returned if a consumer is not authorized to access an operation. (Line 483)	Please add a 403 status code for the items identified.
Inclusive Language	▲ Warning	The underlying assumption of the whitelist/blacklist metaphor is that white = good and black = bad. Because colors in and of themselves have no predetermined meaning, any meaning we assign to them is cultural: for example, the color red in many Southeast Asian countries is lucky, and is often associated with events like marriages, whereas the color white carries the same connotations in many European countries. In the case of whitelist/blacklist, the terms originate in the publishing industry – one dominated by the USA and England, two countries which participated in slavery and which grapple with their racist legacies to this day. (Line 442)	'blacklist' may be insensitive, use 'denylist', 'blocklist', 'exclusion list' instead

#17. OpenAPI Tools and Communities

- **Auto Generators:** Tools that will take your code and turn it into an OpenAPI Specification document
- **Converters:** Various tools to convert to and from OpenAPI and other API description formats.
- **Data Validators:** Check to see if API requests and responses are lining up with the API description.
- **Description Validators:** Check your API description to see if it is valid OpenAPI.
- **Documentation:** Render API Description as HTML (or maybe a PDF) so slightly less technical people can figure out how to work with the API.
- **DSL:** Writing YAML by hand is no fun, and maybe you don't want a GUI, so use a Domain Specific Language to write OpenAPI in your language of choice.
- **GUI Editors:** Visual editors help you design APIs without needing to memorize the entire OpenAPI specification.
- **Learning:** Whether you're trying to get documentation for a third party API based on traffic, or are trying to switch to design-first at an organization with no OpenAPI at all, learning can help you move your API spec forward and keep it up to date.
- **Miscellaneous:** Anything else that does stuff with OpenAPI but hasn't quite got enough to warrant its own category.
- **Mock Servers:** Fake servers that take description document as input, then route incoming HTTP requests to example responses or dynamically generates examples.
- **Parsers:** Loads and read OpenAPI descriptions, so you can work with them programmatically.
- **SDK Generators:** Generate code to give to consumers, to help them avoid interacting at a HTTP level.
- **Security:** By poking around your OpenAPI description, some tools can look out for attack vectors you might not have noticed.
- **Server Implementations:** Easily create and implement resources and routes for your APIs.
- **Testing:** Quickly execute API requests and validate responses on the fly through command line or GUI interfaces.
- **Text Editors:** Text editors give you visual feedback whilst you write OpenAPI, so you can see what docs might look like.

OpenAPI Communities



stoplight.io/api-roadmap-ebook

Wrapup

Who you are	Benefits blogs.cisco.com/developer/worldclassapis01
IT Pro or Application Developer consuming APIs	<ul style="list-style-type: none"> • OAS to discover the capabilities of an API • OAS to automatically generate client code for your preferred language • OAS as a pivot format to import/export API definitions across tools
Engineering group publishing internal or external-facing APIs	<ul style="list-style-type: none"> • OAS to define the capabilities offered for your API • OAS to publish low-level SDKs • OAS to publish accurate and interactive documentation • OAS to automate raw API Changelogs • Authoring tools to initiate/edit OAS documents (Design-First) • Source code annotations to generate OAS documents (Code-First) • OAS linters to automate design reviews and adoption of REST Guidelines • Static & dynamic analysis of API Security issues including OWASP Top 10
Security and Compliance Officers overseeing every APIs	<ul style="list-style-type: none"> • OAS to maintain an inventory of an organization's APIs • Analysis of OAS documents to identify breaking changes and ensure backward compatibility of existing API Contracts • OAS to ensure compliance of new releases along CI/CD pipelines • OAS to identify zombie & shadow operations via live traffic observations

Technical Session Surveys

- Attendees who fill out a minimum of four session surveys and the overall event survey will get Cisco Live branded socks!
- Attendees will also earn 100 points in the Cisco Live Game for every survey completed.
- These points help you get on the leaderboard and increase your chances of winning daily and grand prizes.



Cisco Learning and Certifications

From technology training and team development to Cisco certifications and learning plans, let us help you empower your business and career. www.cisco.com/go/certs

Pay for Learning with Cisco Learning Credits

(CLCs) are prepaid training vouchers redeemed directly with Cisco.



Learn

Cisco U.

IT learning hub that guides teams and learners toward their goals

Cisco Digital Learning

Subscription-based product, technology, and certification training

Cisco Modeling Labs

Network simulation platform for design, testing, and troubleshooting

Cisco Learning Network

Resource community portal for certifications and learning



Train

Cisco Training Bootcamps

Intensive team & individual automation and technology training programs

Cisco Learning Partner Program

Authorized training partners supporting Cisco technology and career certifications

Cisco Instructor-led and Virtual Instructor-led training

Accelerated curriculum of product, technology, and certification courses



Certify

Cisco Certifications and Specialist Certifications

Award-winning certification program empowers students and IT Professionals to advance their technical careers

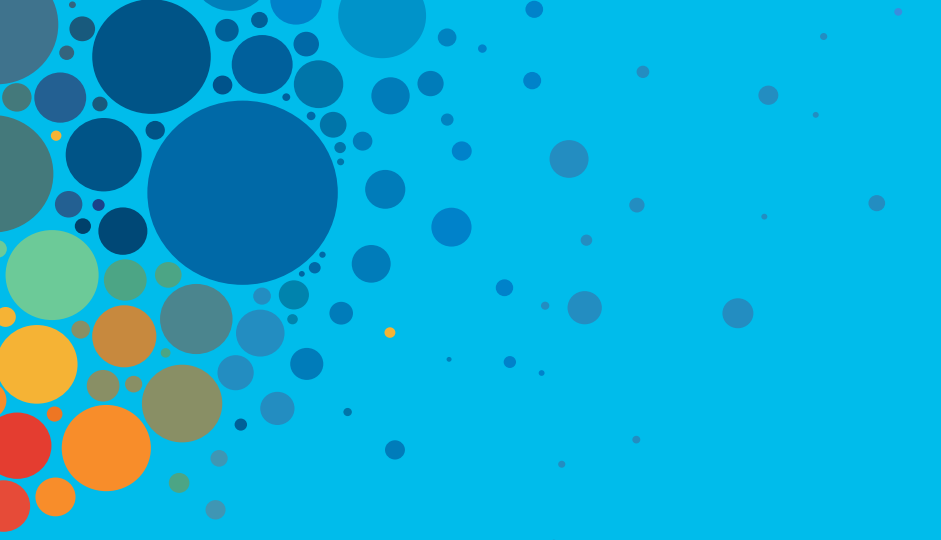
Cisco Guided Study Groups

180-day certification prep program with learning and support

Cisco Continuing Education Program

Recertification training options for Cisco certified individuals

Here at the event? Visit us at **The Learning and Certifications lounge at the World of Solutions**



Continue your education

- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Engineer 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



The bridge to possible

Thank you

CISCO *Live!*

#CiscoLive

CISCO *Live!*



#CiscoLive