Understanding Workiva Public APIs & Workiva Scripting

Agenda

1. Workiva Dev Platform
2. Workiva Public APIs
3. Workiva Scripting
4. Use Cases
5. Addtional Resources

workiva

1 Workiva Dev Platform

2 Workiva Public APIs

3 Workiva Scripting

4 Use Cases

5 Addtional Resources

workiva

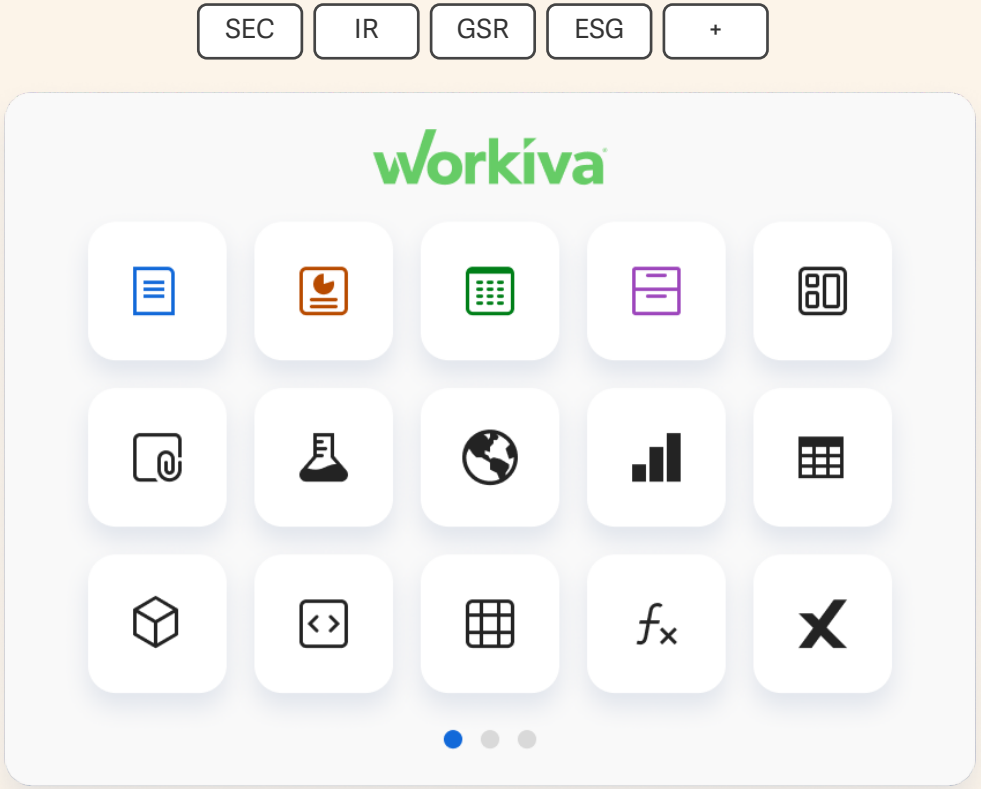
The **developer platform** equips you to extend the value of our platform with new regulatory, financial, and ESG reporting solutions.

Public APIs

+

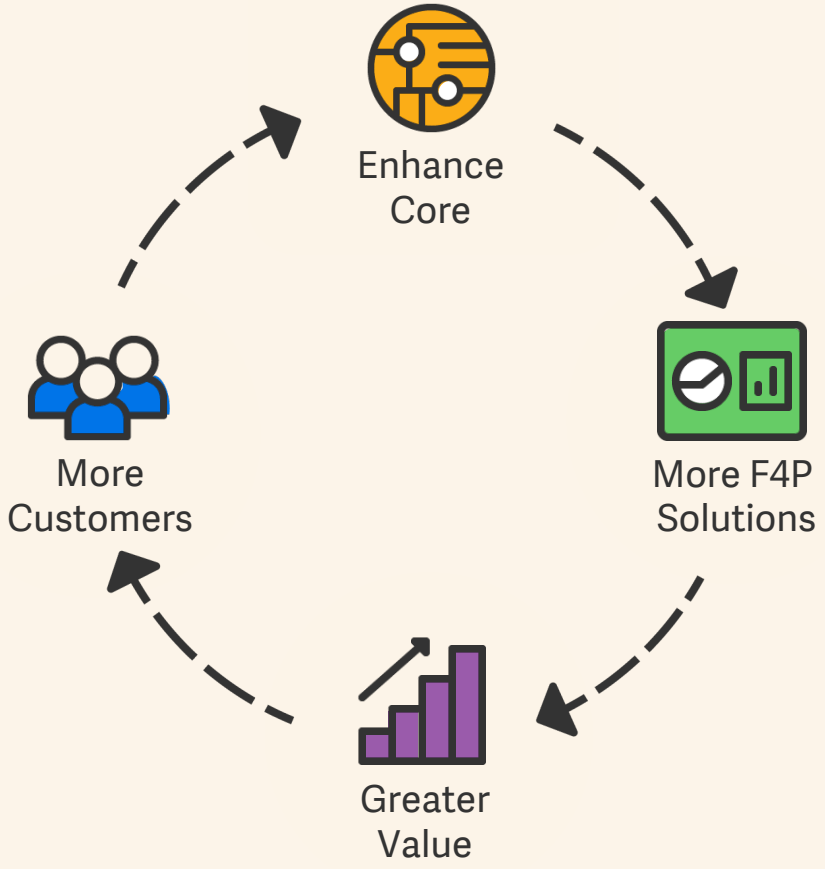
Workiva Scripting

Workiva



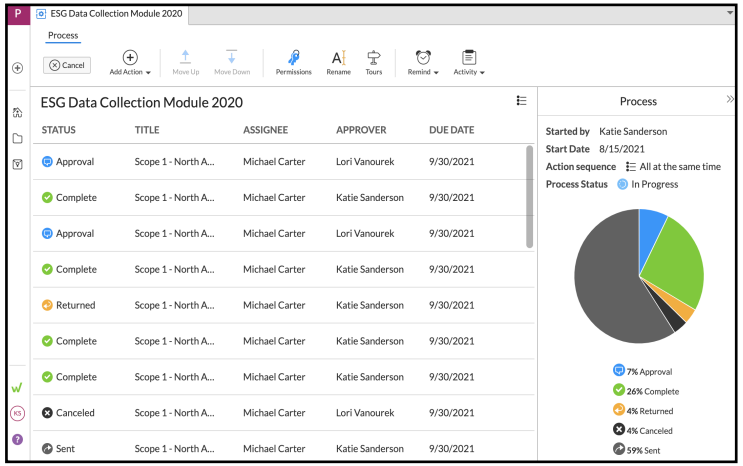
Solutions Platform

Our Growth Flywheel

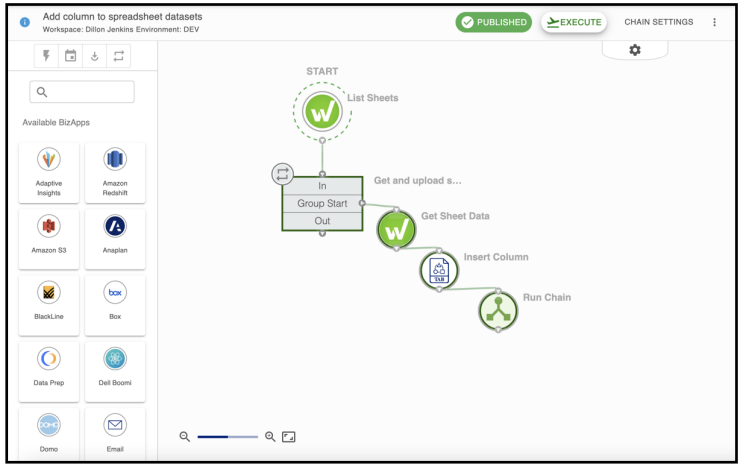


Our Growth Flywheel with Builders

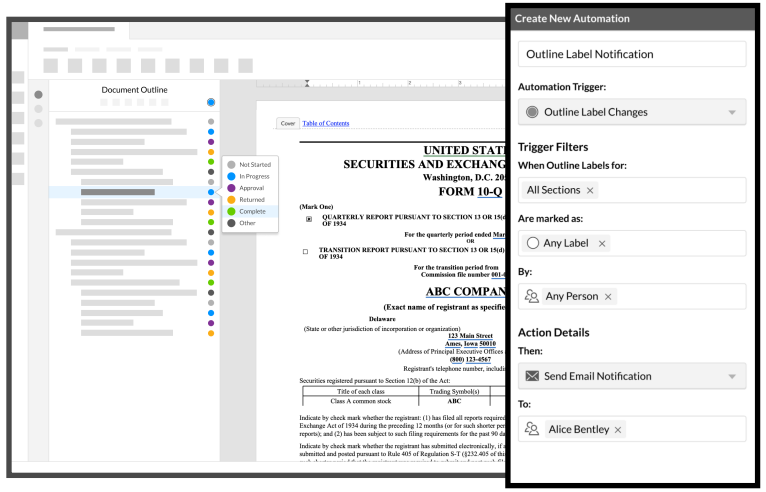
Automation & Connectivity Ecosystem



Process Builder/Monitor

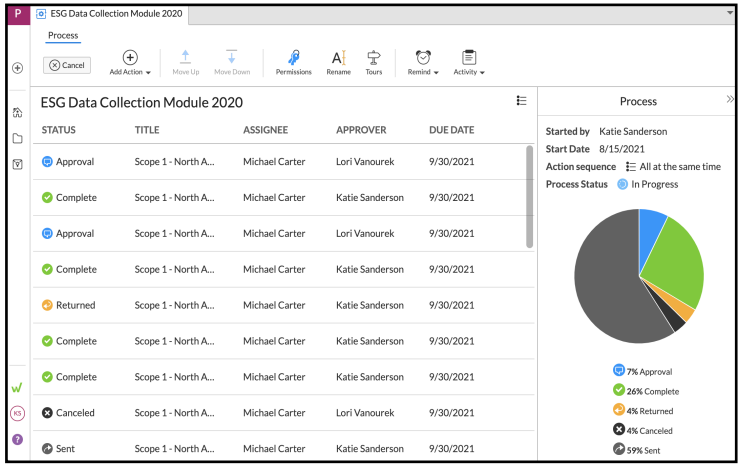


Chain Builder & Data Prep

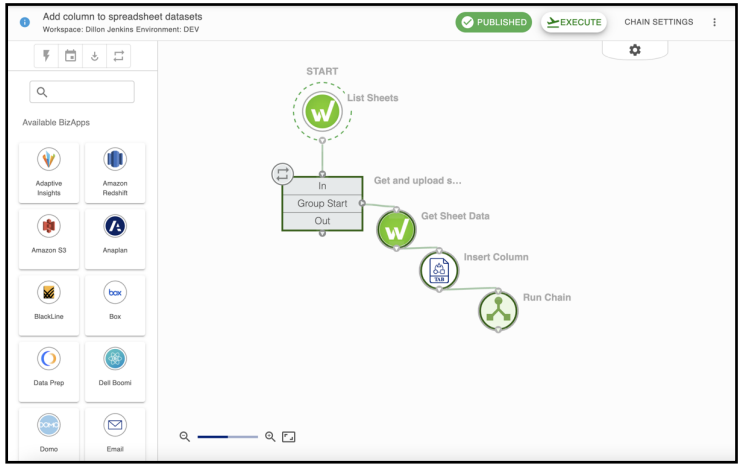


Integrated Automations

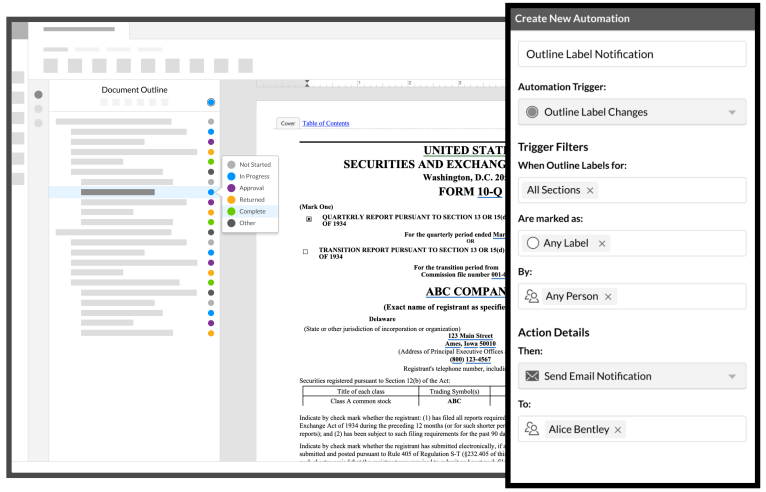
Automation & Connectivity Ecosystem



Process Builder/Monitor

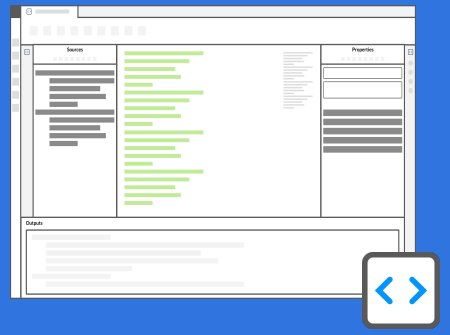


Chain Builder & Data Prep



Integrated Automations

Public APIs



Workiva Scripting

1 Workiva Dev Platform

2 Workiva Public APIs

3 Workiva Scripting

4 Use Cases

5 Addtional Resources

workiva

Platform APIs

Access to the core data & capabilities of Workiva

- Task
- Files
- Documents
- Spreadsheets
- Presentations
- Processes
- Graph
- Test Forms
- More!

workiva

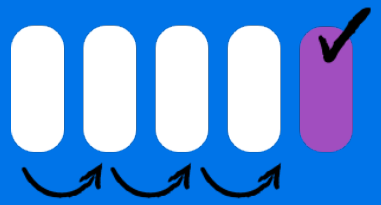
Platform APIs

Access to the core data & capabilities of Workiva

- Task
- Files
- Documents
- Spreadsheets
- Presentations
- Processes
- Graph
- Test Forms
- More!

Admin APIs

Automate the setup & administration of Workiva



- Organizations
- Workspaces
- Memberships
- Roles
- Groups
- Users

workiva

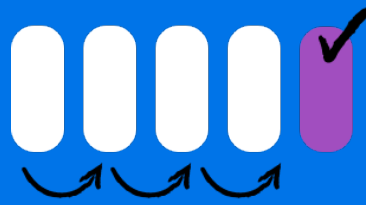
Platform APIs

Access to the core data & capabilities of Workiva

- Task
- Files
- Documents
- Spreadsheets
- Presentations
- Processes
- Graph
- Test Forms
- More!

Admin APIs

Automate the setup & administration of Workiva



- Organizations
- Workspaces
- Memberships
- Roles
- Groups
- Users

Industry Standard APIs

Leverage standards to work with existing systems

- SCIM for Identity Management
  - Active Directory
  - SailPoint
  - Okta
  - More!

Workiva

Welcome to the Workiva Developer Hub_

Platform API

Interact with core components of the Workiva platform including Documents, Spreadsheets, Presentations, Graph, and Tasks

Admin API

Manage users, organizations, and workspaces within the Workiva Platform

Identity and Access Management (IAM) API

Securely create Workiva Developer API requests

Spreadsheets API

Import, update, and fetch data from Spreadsheets

Wdata API

Connect, curate, and explore data within Wdata

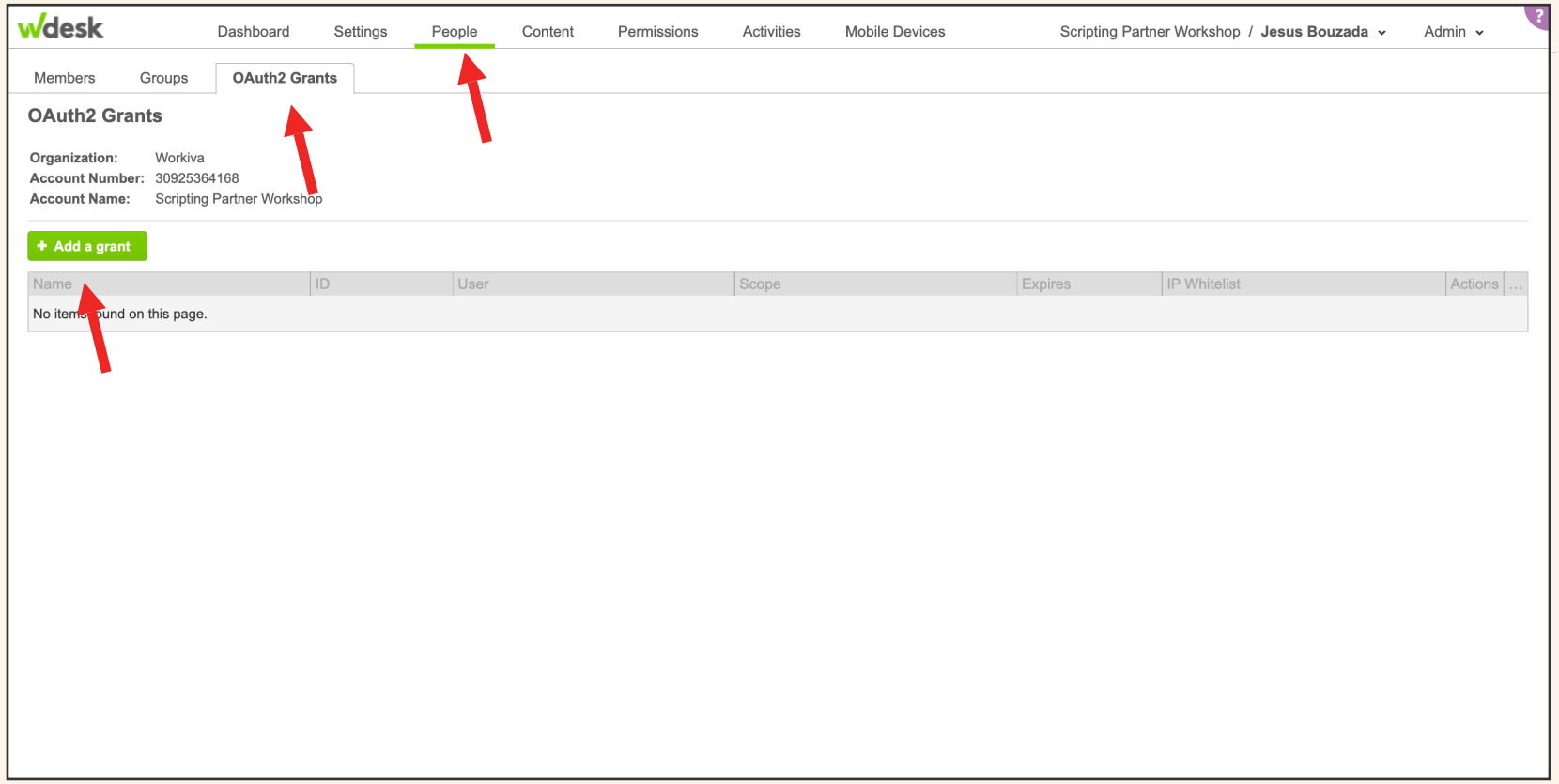
Chains API

Manage chain runs, and retrieve metadata about your workspace

Authorized APIs - Generate Credentials

Generate credentials for the **specific Workiva Workspace** you want to interact

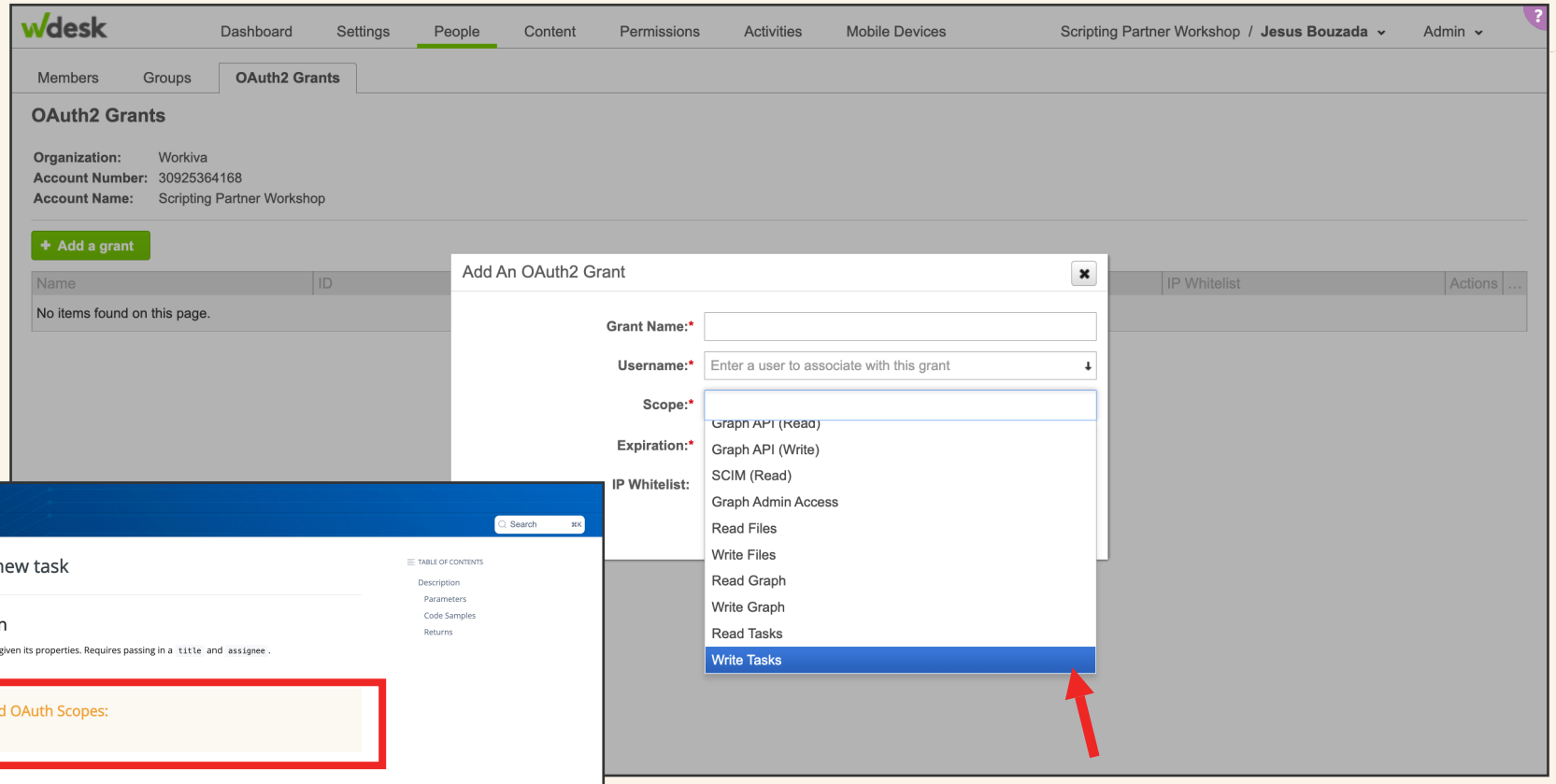
1. Click Add a grant



Authorized APIs - Generate Credentials

Generate credentials for the **specific Workiva Workspace** you want to interact

1. Click Add a grant
2. Select the scopes required for the API you want to use (ex: Write Tasks scope to create a task)

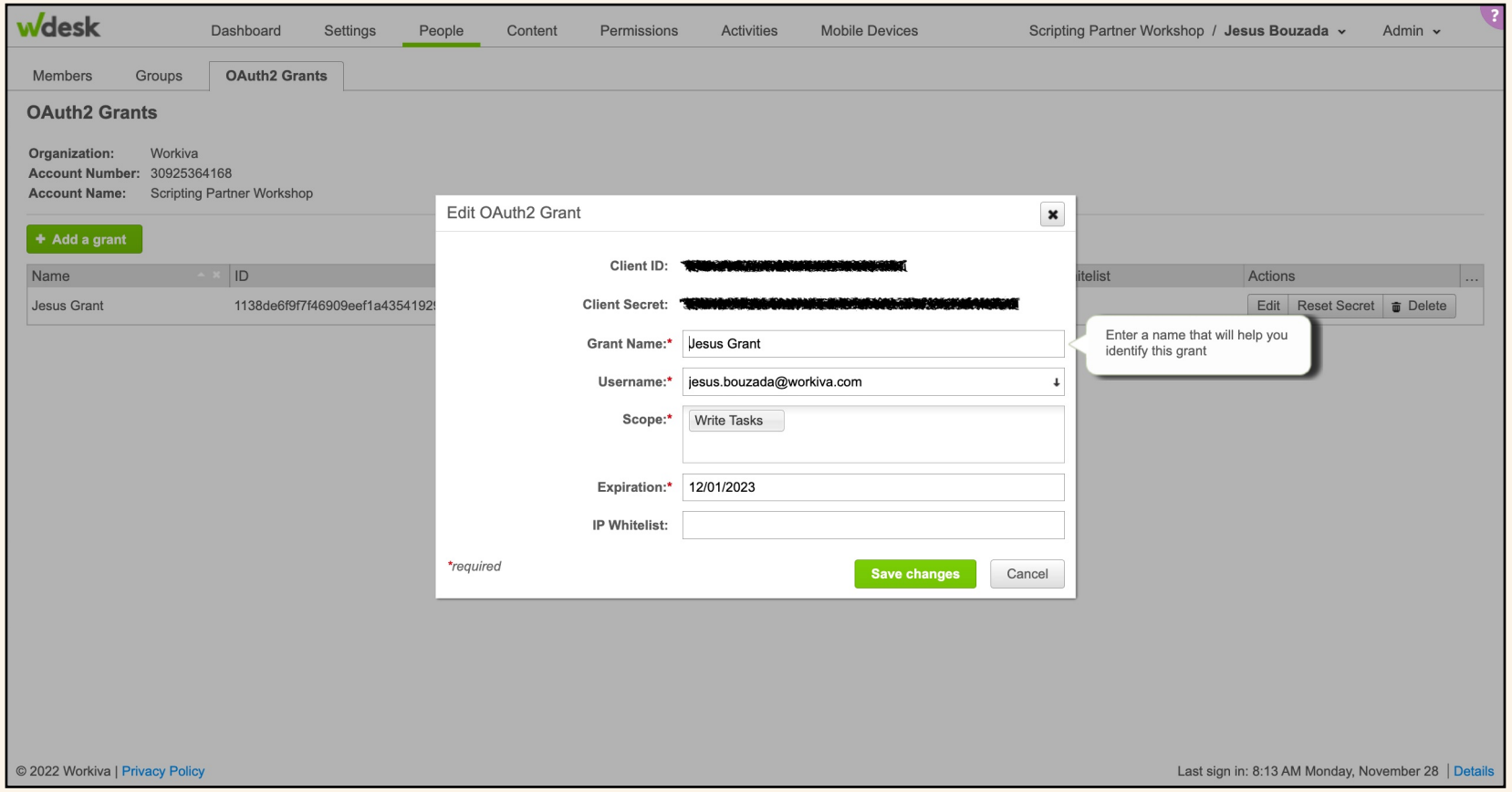


Workiva

Authorized APIs - Generate Credentials

Generate credentials for the **specific Workiva Workspace** you want to interact

1. Click Add a grant
2. Select the scopes required for the API you want to use (ex: Write Tasks scope to create a task)
3. Create the Grant

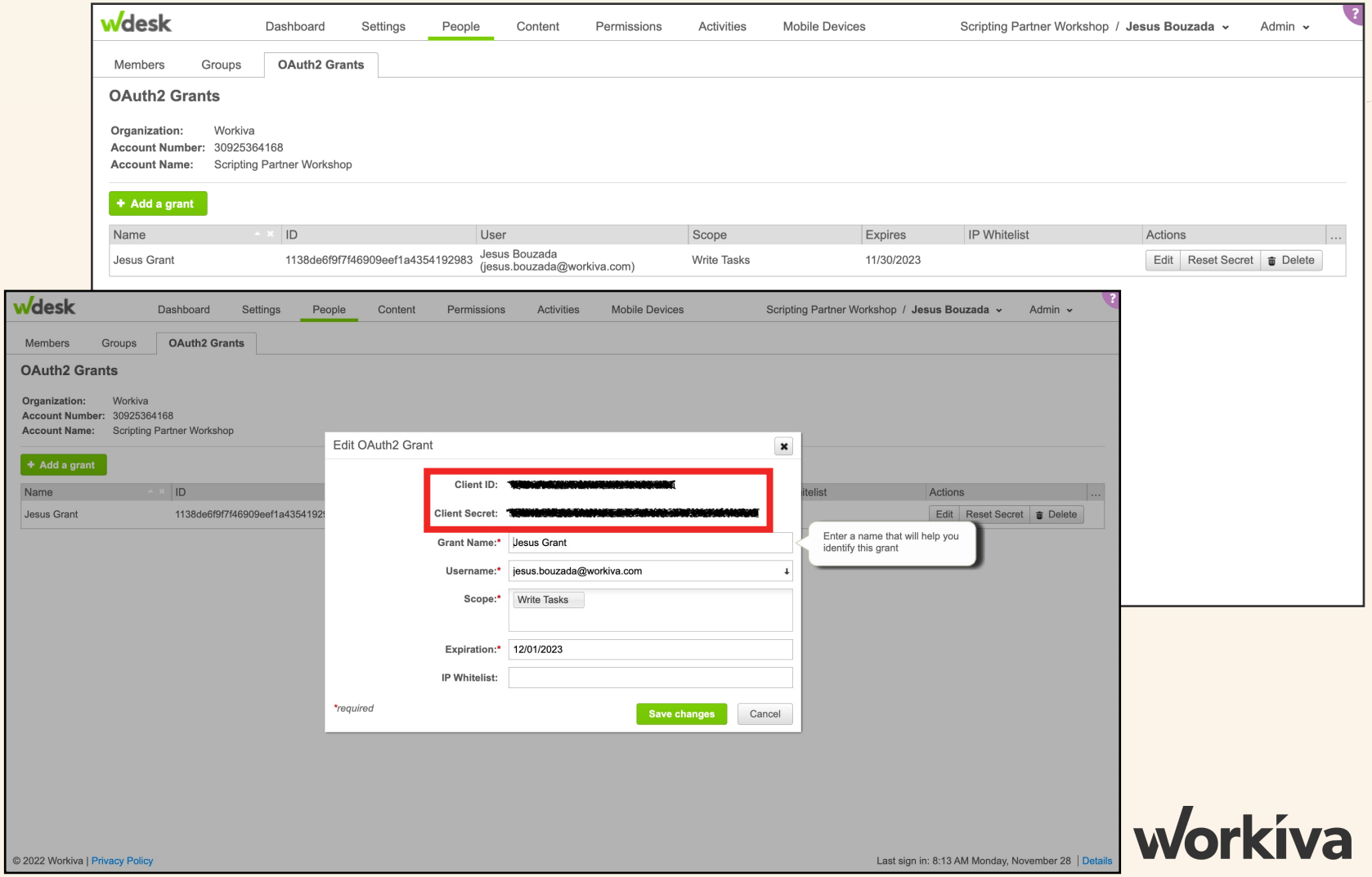


Authorized APIs - Generate Credentials

Generate credentials for the **specific Workiva Workspace** you want to interact

1. Click Add a grant
2. Select the scopes required for the API you want to use (ex: Write Tasks scope to create a task)
3. Create the Grant
4. Grab client id and client secret.

   DO NOT SHARE



Workiva

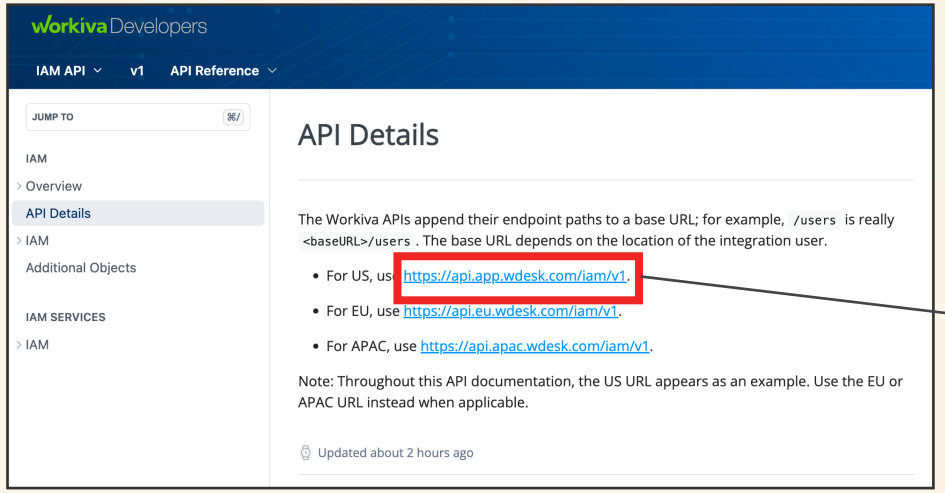
Authorized APIs - Code Sample

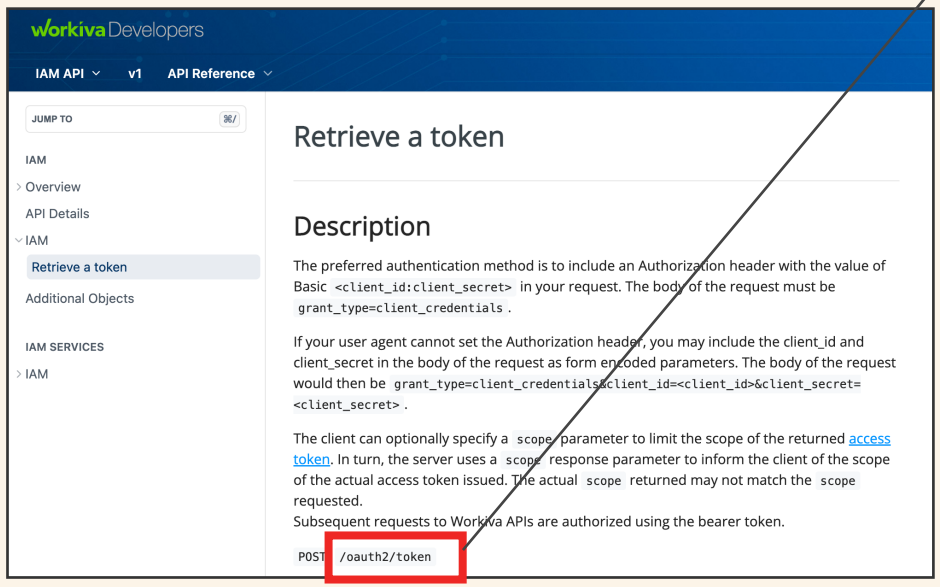
Authorized APIs - Generate Creds

```python
import requests
import json
import os

AUTH_URL = "https://api.app.wdesk.com/iam/v1/oauth2/token"

CLIENT_ID = os.getenv('CLIENT_ID')
CLIENT_SECRET = os.getenv('CLIENT_SECRET')

authHeaders = {'Content-Type': 'application/x-www-form-urlencoded;charset=UTF-8'}

tokenResponse = json.loads(requests.post(AUTH_URL, data='client_id='
                                         + CLIENT_ID + '&client_secret='
                                         + CLIENT_SECRET
                                         + '&grant_type=client_credentials',
                                         headers=authHeaders).text)

accessToken = 'Bearer ' + tokenResponse['access_token']
```

Workiva

Authorized APIs - Code Sample

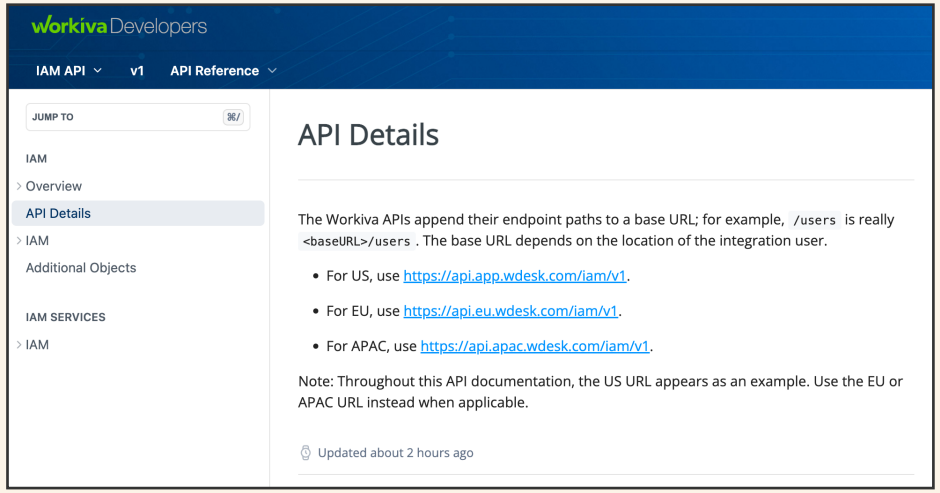


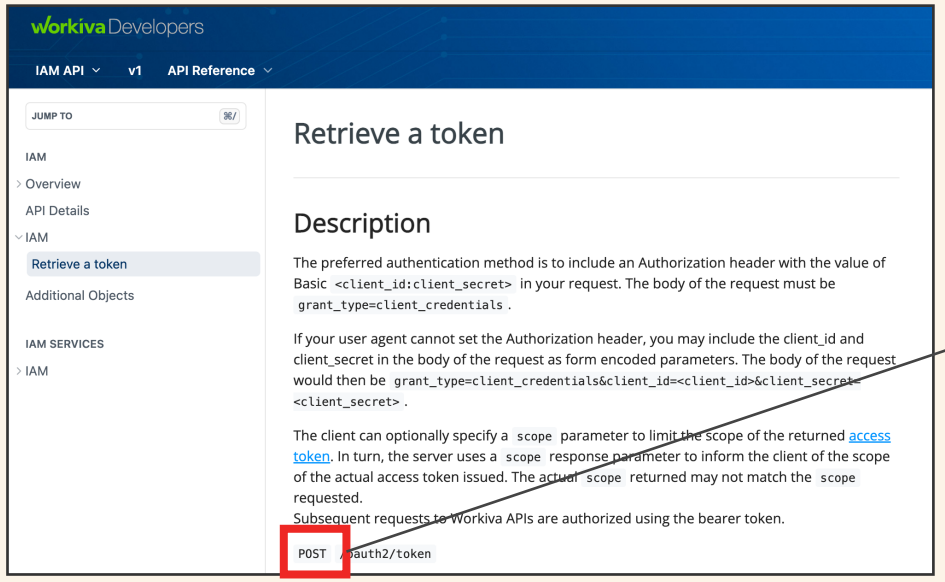



```python
import requests
import json
import os

AUTH_URL = "https://api.app.wdesk.com/iam/v1/oauth2/token"

CLIENT_ID = os.getenv('CLIENT_ID')
CLIENT_SECRET = os.getenv('CLIENT_SECRET')

authHeaders = {'Content-Type': 'application/x-www-form-urlencoded;charset=UTF-8'}

tokenResponse = json.loads(requests.post(AUTH_URL, data='client_id='
                                         + CLIENT_ID + '&client_secret='
                                         + CLIENT_SECRET
                                         + '&grant_type=client_credentials',
                                         headers=authHeaders).text)

accessToken = 'Bearer ' + tokenResponse['access_token']
```

workiva

Authorized APIs - Code Sample

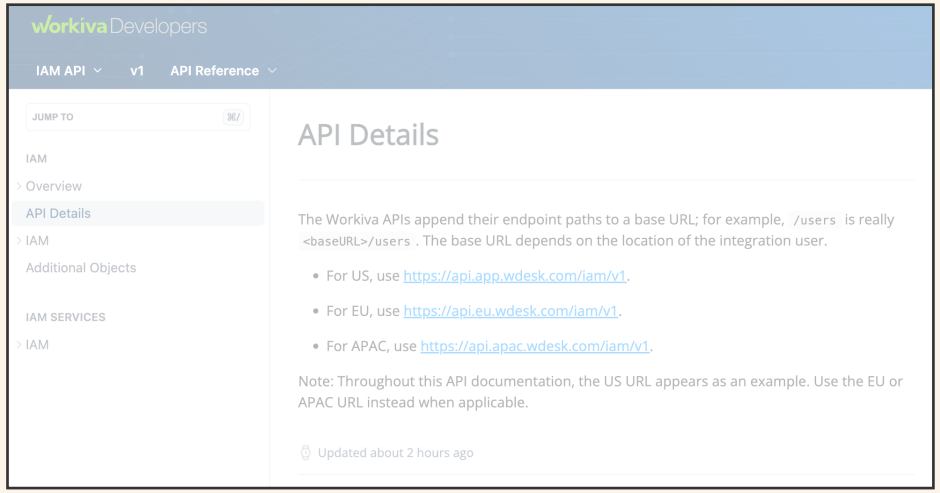


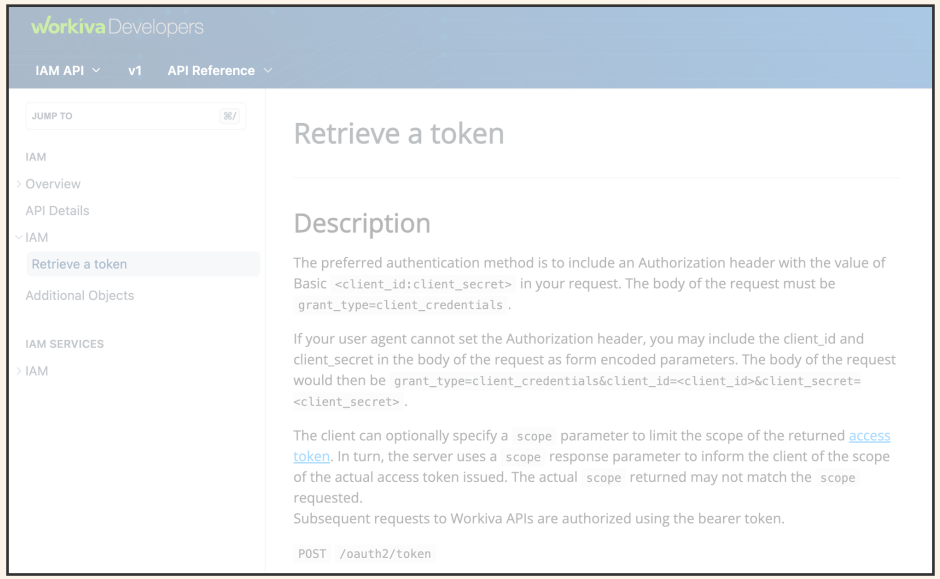



```python
import requests
import json
import os

AUTH_URL = "https://api.app.wdesk.com/iam/v1/oauth2/token"

CLIENT_ID = os.getenv('CLIENT_ID')
CLIENT_SECRET = os.getenv('CLIENT_SECRET')

authHeaders = {'Content-Type': 'application/x-www-form-urlencoded;charset=UTF-8'}

tokenResponse = json.loads(requests.post(AUTH_URL, data='client_id='
                                         + CLIENT_ID + '&client_secret='
                                         + CLIENT_SECRET
                                         + '&grant_type=client_credentials',
                                         headers=authHeaders).text)

accessToken = 'Bearer ' + tokenResponse['access_token']
```

workiva

Authorized APIs - Code Sample






```python
import requests
import json
import os

AUTH_URL = "https://api.app.wdesk.com/iam/v1/oauth2/token"

CLIENT_ID = os.getenv('CLIENT_ID')
CLIENT_SECRET = os.getenv('CLIENT_SECRET')

authHeaders = {'Content-Type': 'application/x-www-form-urlencoded;charset=UTF-8'}

tokenResponse = json.loads(requests.post(AUTH_URL, data='client_id='
                                         + CLIENT_ID + '&client_secret='
                                         + CLIENT_SECRET
                                         + '&grant_type=client_credentials',
                                         headers=authHeaders).text)

accessToken = 'Bearer ' + tokenResponse['access_token']
```

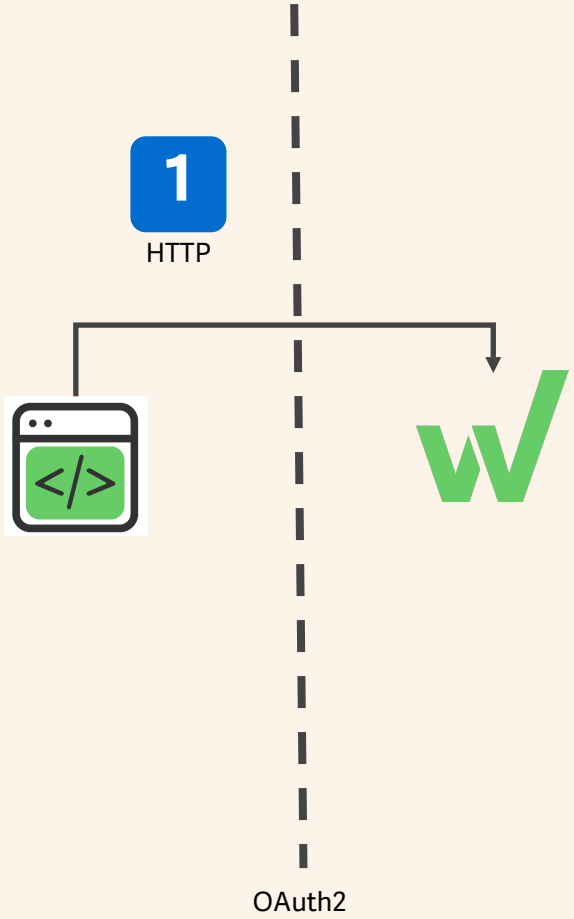
https://developers.workiva.com/workiva-iam/reference/iam-tokenrequest

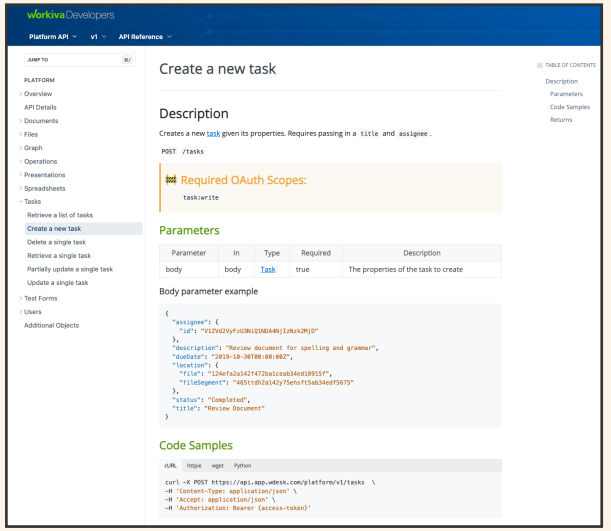
workiva

HTTP-based REST APIs

Workiva Public APIs request-response workflow

1. Application makes request





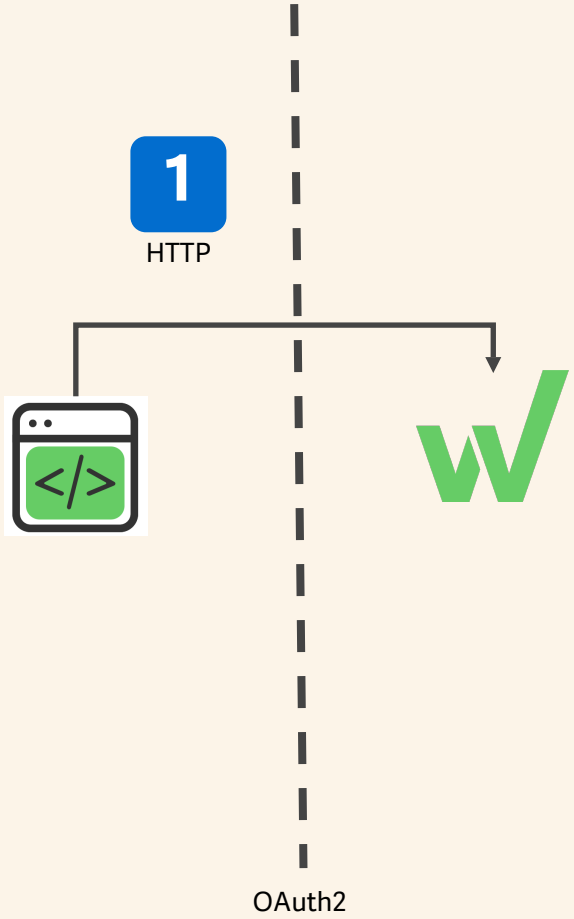

```
20  TASK_API_URL = 'https://api.sandbox.wdesk.com/platform/v1/tasks'
21
22  taskHeaders = {
23    'Content-Type': 'application/x-www-form-urlencoded;charset=UTF-8',
24    'Accept': 'application/json',
25    'Authorization': accessToken
26  }
27
28  taskData = {
29    'assignee': {'id': 'V0ZVc2VyHzY0NDQ5MDY2NTA4NjE1Njg'},
30    'title': 'test'
31  }
32
33  taskResponse = json.loads requests.post(TASK_API_URL, json = taskData, headers = taskHeaders) text)
```

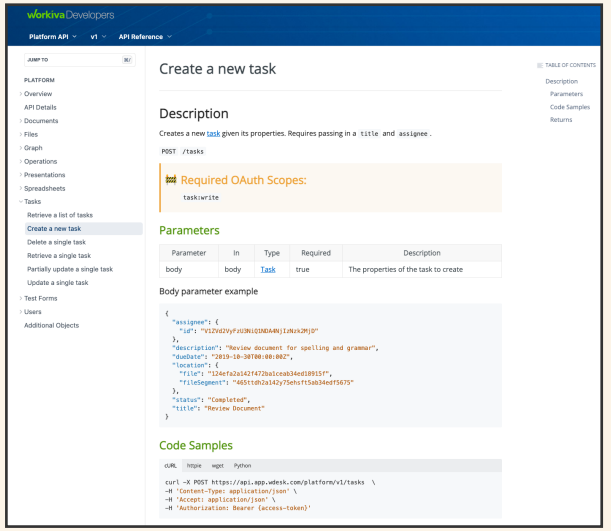
workiva

HTTP-based REST APIs

Workiva Public APIs request-response workflow

1. Application makes request
2. Service receives request
3. Process data, returns response





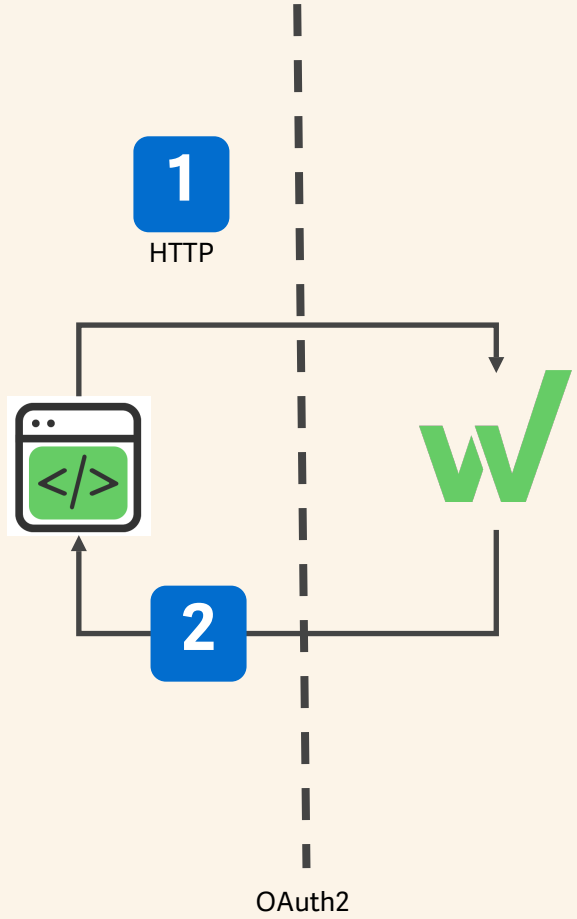

```
20  TASK_API_URL = 'https://api.sandbox.wdesk.com/platform/v1/tasks'
21
22  taskHeaders = {
23    'Content-Type': 'application/x-www-form-urlencoded;charset=UTF-8',
24    'Accept': 'application/json',
25    'Authorization': accessToken
26  }
27
28  taskData = {
29    'assignee': {'id': 'V0ZVc2VyHzY0NDQ5MDY2NTA4NjE1Njg'},
30    'title': 'test'
31  }
32
33  taskResponse = json.loads requests.post(TASK_API_URL, json = taskData, headers = taskHeaders) text)
```

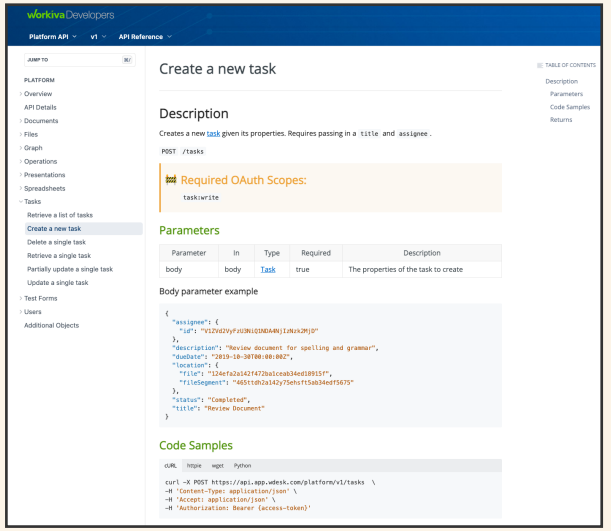
HTTP-based REST APIs

Workiva Public APIs request-response workflow

1. Application makes request
2. Service receives request
3. Process data, returns response
4. Sends results back to application

(typical client-server model)






```
20  TASK_API_URL = 'https://api.sandbox.wdesk.com/platform/v1/tasks'
21
22  taskHeaders = {
23    'Content-Type': 'application/x-www-form-urlencoded;charset=UTF-8',
24    'Accept': 'application/json',
25    'Authorization': accessToken
26  }
27
28  taskData = {
29    'assignee': {'id': 'V0ZVc2VyHzY0NDQ5MDY2NTA4NjE1Njg'},
30    'title': 'test'
31  }
32
33  taskResponse = json.loads(requests.post(TASK_API_URL, json = taskData, headers = taskHeaders).text)
```

https://developers.workiva.com/workiva-platform/reference/platform-createtask

workiva

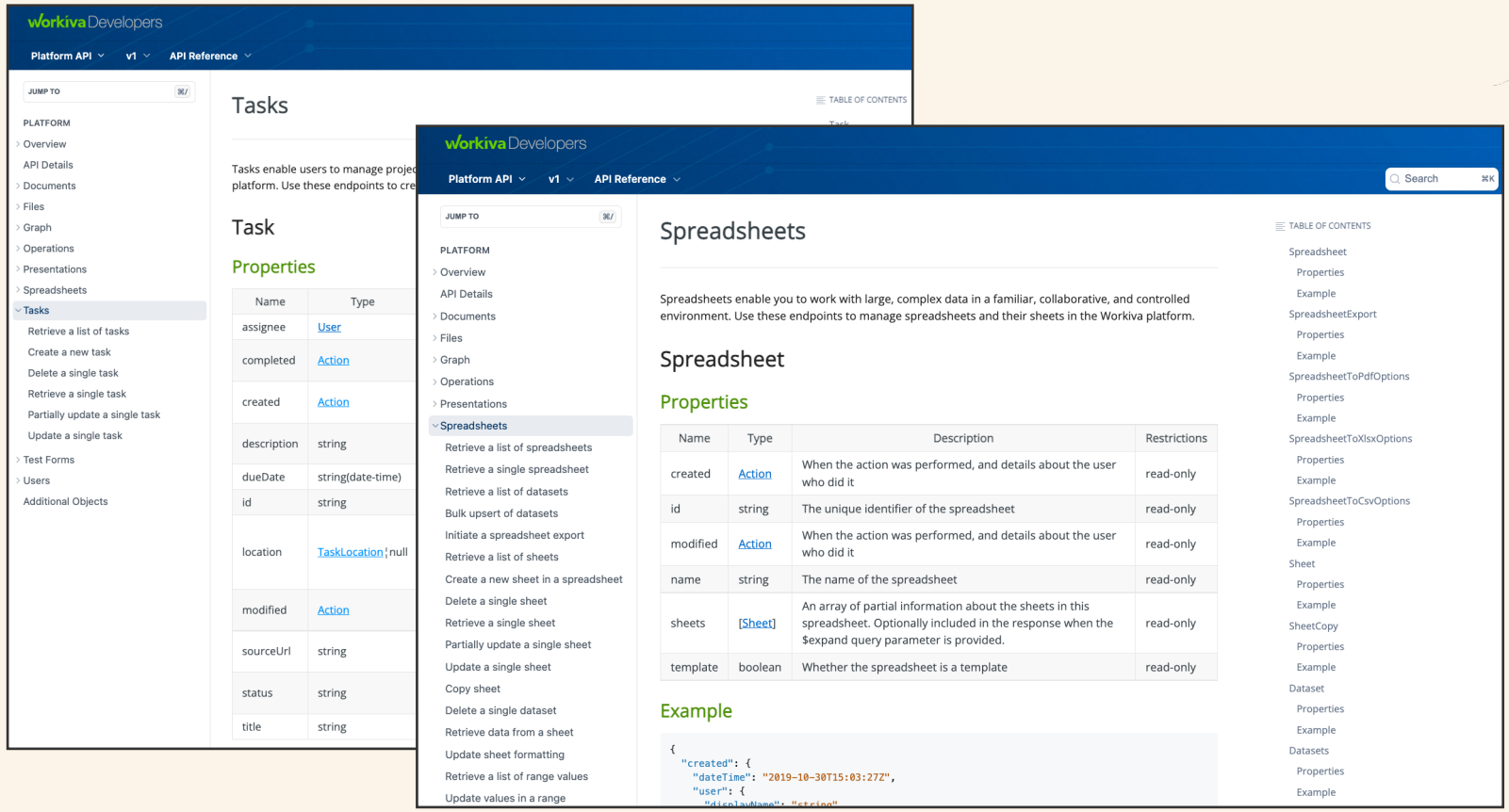
Data Model

Resource or objects

Things in the API you can interact with

Examples: User, Task, Document, File, Process, etc...

"Give me all **users**"
"Create a **task**"
"Update **file** name"



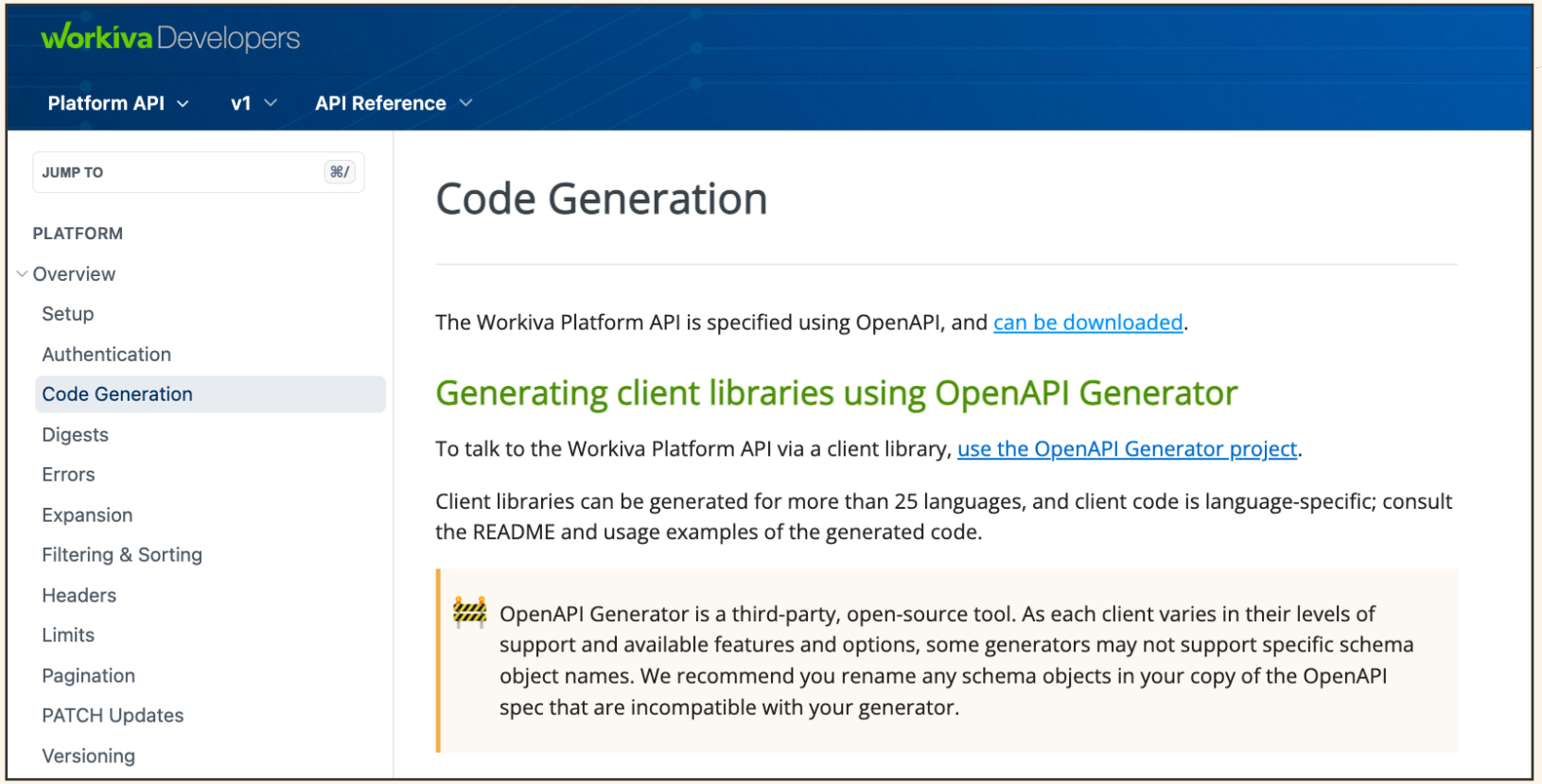
workiva

Code Generation

Today, Workiva does not provide client libraries to access Workiva Public APIs.

But you can generate client libraries for more than 25 languages with these steps:

1. Download the OpenAPI specification (yaml)



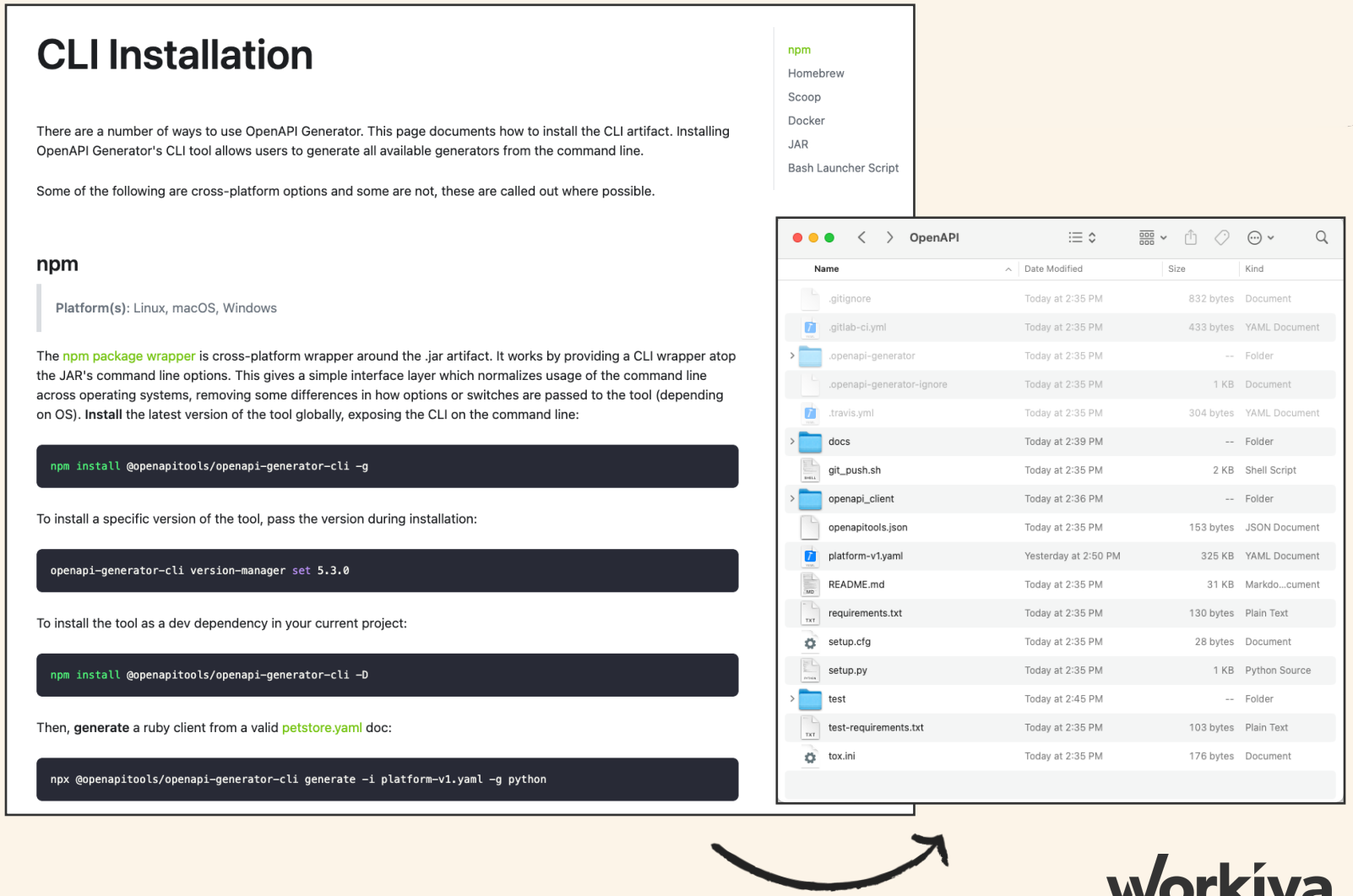
workiva

Code Generation

Today, Workiva does not provide client libraries to access Workiva Public APIs.

But you can generate client libraries for more than 25 languages with these steps:

1. Download the OpenAPI specification (yaml)
2. [Use the OpenAPI Generator project] to generate a Python client



CLI Installation

There are a number of ways to use OpenAPI Generator. This page documents how to install the CLI artifact. Installing OpenAPI Generator's CLI tool allows users to generate all available generators from the command line.

Some of the following are cross-platform options and some are not, these are called out where possible.

npm
Homebrew
Scoop
Docker
JAR
Bash Launcher Script

npm

> **Platform(s)**: Linux, macOS, Windows

The npm package wrapper is cross-platform wrapper around the .jar artifact. It works by providing a CLI wrapper atop the JAR's command line options. This gives a simple interface layer which normalizes usage of the command line across operating systems, removing some differences in how options or switches are passed to the tool (depending on OS). **Install** the latest version of the tool globally, exposing the CLI on the command line:

```
npm install @openapitools/openapi-generator-cli -g
```

To install a specific version of the tool, pass the version during installation:

```
openapi-generator-cli version-manager set 5.3.0
```

To install the tool as a dev dependency in your current project:

```
npm install @openapitools/openapi-generator-cli -D
```

Then, **generate** a ruby client from a valid petstore.yaml doc:

```
npx @openapitools/openapi-generator-cli generate -i platform-v1.yaml -g python
```

OpenAPI

Name	Date Modified	Size	Kind
.gitignore	Today at 2:35 PM	832 bytes	Document
.gitlab-ci.yml	Today at 2:35 PM	433 bytes	YAML Document
.openapi-generator	Today at 2:35 PM	--	Folder
.openapi-generator-ignore	Today at 2:35 PM	1 KB	Document
.travis.yml	Today at 2:35 PM	304 bytes	YAML Document
docs	Today at 2:39 PM	--	Folder
git_push.sh	Today at 2:35 PM	2 KB	Shell Script
openapi_client	Today at 2:36 PM	--	Folder
openapitools.json	Today at 2:35 PM	153 bytes	JSON Document
platform-v1.yaml	Yesterday at 2:50 PM	325 KB	YAML Document
README.md	Today at 2:35 PM	31 KB	Markdo...cument
requirements.txt	Today at 2:35 PM	130 bytes	Plain Text
setup.cfg	Today at 2:35 PM	28 bytes	Document
setup.py	Today at 2:35 PM	1 KB	Python Source
test	Today at 2:45 PM	--	Folder
test-requirements.txt	Today at 2:35 PM	103 bytes	Plain Text
tox.ini	Today at 2:35 PM	176 bytes	Document

workiva

Code Generation

Today, Workiva does not provide client libraries to access Workiva Public APIs.

But you can generate client libraries for more than 25 languages with these steps:

1. Download the OpenAPI specification (yaml)
2. Use the OpenAPI Generator project to generate a Python client
3. Import the generated clients into your code

```python
import openapi_client
from openapi_client.apis.tags import tasks_api
from openapi_client.model.task import Task

# Configure OAuth2 access token for authorization: oauth
configuration = openapi_client.Configuration(
    host = "https://api.app.wdesk.com/platform/v1"
)

configuration.access_token = 'YOUR_ACCESS_TOKEN'

# Enter a context with an instance of the API client
with openapi_client.ApiClient(configuration) as api_client:
    # Create an instance of the API class
    api_instance = tasks_api.TasksApi(api_client)

    body = Task(
        assignee=User(
            display_name="display_name_example",
            email="email_example",
            id="V1ZVd2VyFzU3NiQ1NDA4NjIzNzk2MjD"
        ),
        title="Review Document",
    )

    # Create a new task
    api_response = api_instance.create_task(body=body)
```

workiva

Code Generation

Today, Workiva does not provide client libraries to access Workiva Public APIs.

But you can generate client libraries for more than 25 languages with these steps:

1. Download the OpenAPI specification (yaml)
2. Use the OpenAPI Generator project to generate a Python client
3. Import the generated clients into your code
4. In this example, I use the generated clients to create a task

```python
import openapi_client
from openapi_client.apis.tags import tasks_api
from openapi_client.model.task import Task

# Configure OAuth2 access token for authorization: oauth
configuration = openapi_client.Configuration(
    host = "https://api.app.wdesk.com/platform/v1"
)

configuration.access_token = 'YOUR_ACCESS_TOKEN'

# Enter a context with an instance of the API client
with openapi_client.ApiClient(configuration) as api_client:
    # Create an instance of the API class
    api_instance = tasks_api.TasksApi(api_client)

    body = Task(
        assignee=User(
            display_name="display_name_example",
            email="email_example",
            id="V1ZVd2VyFzU3NiQ1NDA4NjIzNzk2MjD"
        ),
        title="Review Document",
    )

    # Create a new task
    api_response = api_instance.create_task(body=body)
```

workiva

1 Workiva Dev Platform

2 Workiva Public APIs

3 Workiva Scripting

4 Use Cases

5 Addtional Resources

workiva

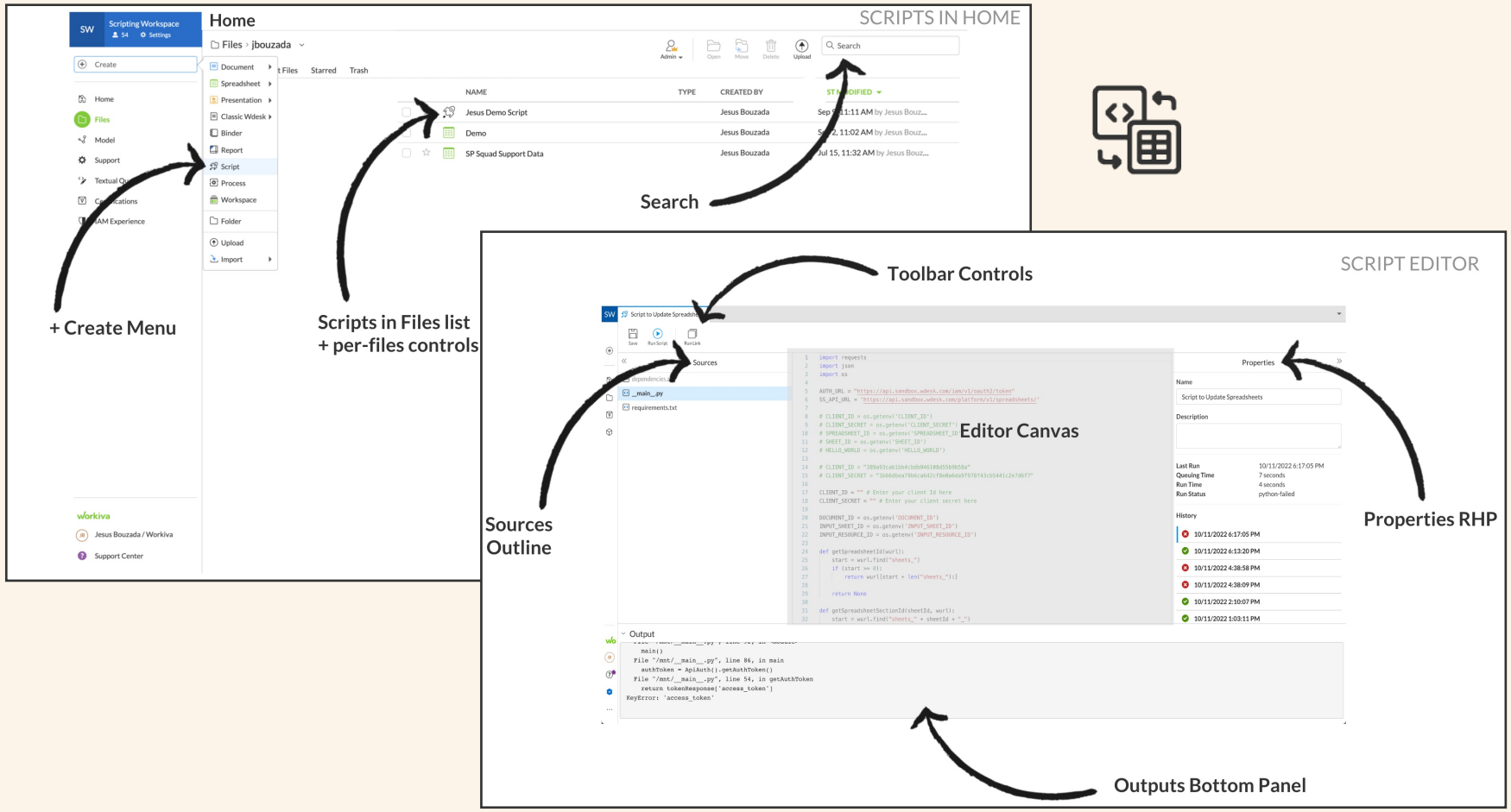
Scripting allows customers to meet unique needs by running custom code efficiently where they trust their data is secured

Workiva Scripting

Workiva

Workiva Scripting

Available today through our Early Adopter Program



Create & Manage **Python** Scripts

Invoke from chains & Integrated Automations

Secure execution on the Workiva Platform

workiva

Running Scripts

Runner Credentials

- Generate Creds (Scripting API)
- Scripting role (Editor/Viewer/ Runner)
- Script permissions

Script Credentials

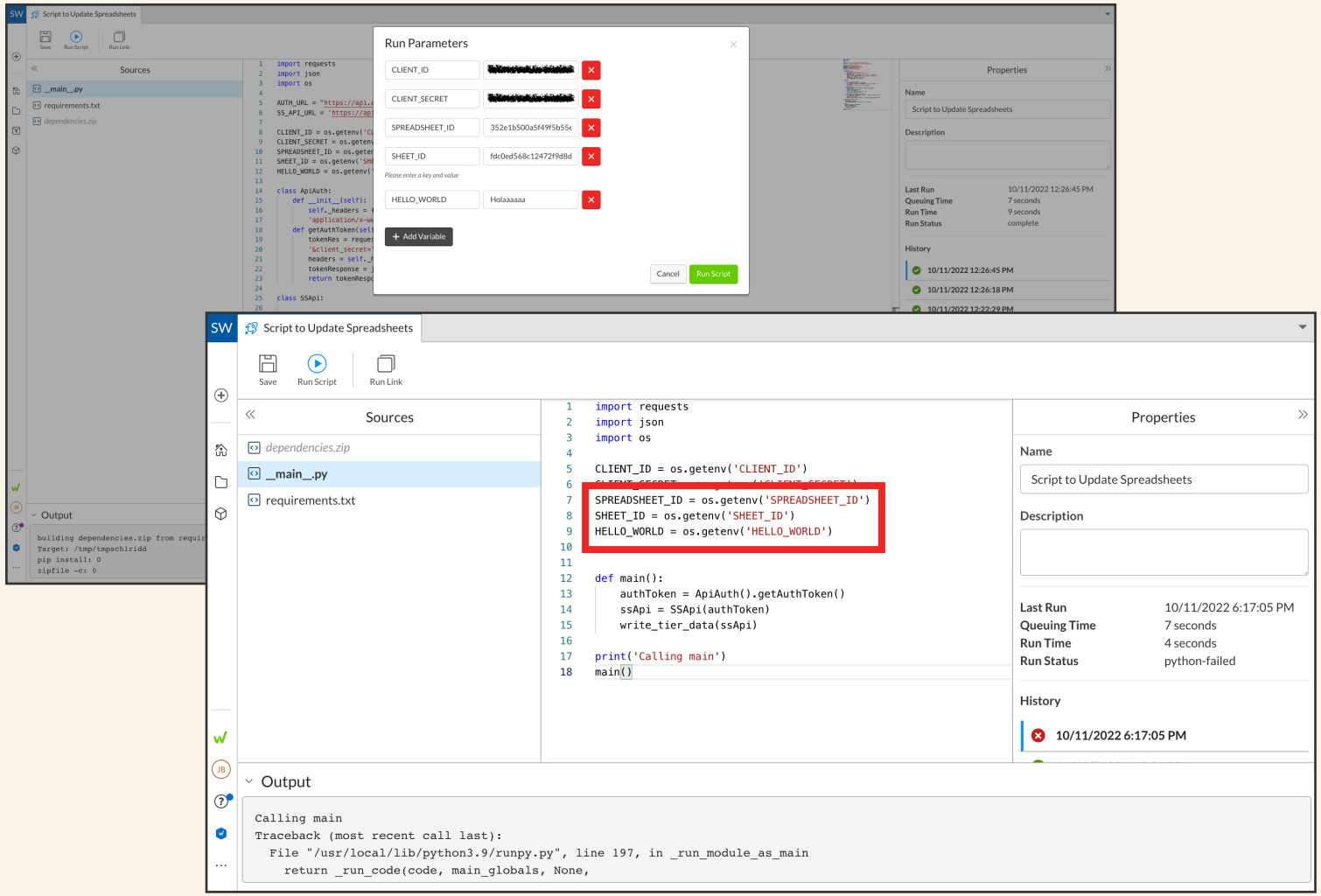
- Generate Creds
- Permissions to Workiva files

Workiva

Running Scripts with the Editor

Scripting Editor

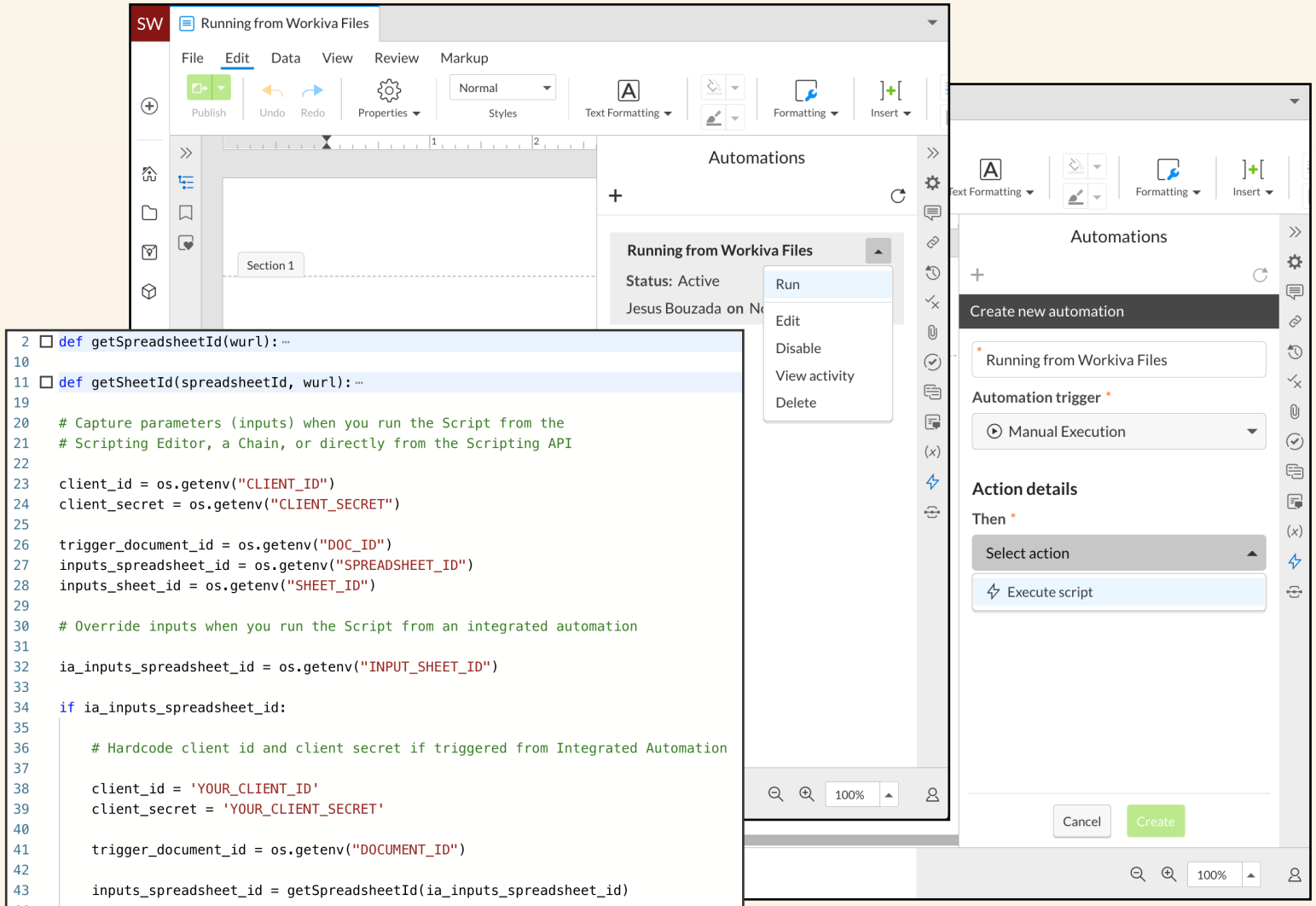
- Used for the **dev-testing cycle**
- Enter input parameters on the screen
- Read inputs as environment variables in the code



Running Scripts from Workiva Files

Integrated Automations

- Ideal for **non-tech customers**
- Setup Scripting-based automations in Workiva files
- Manually run from the file
- Define your inputs in a companion spreadsheet

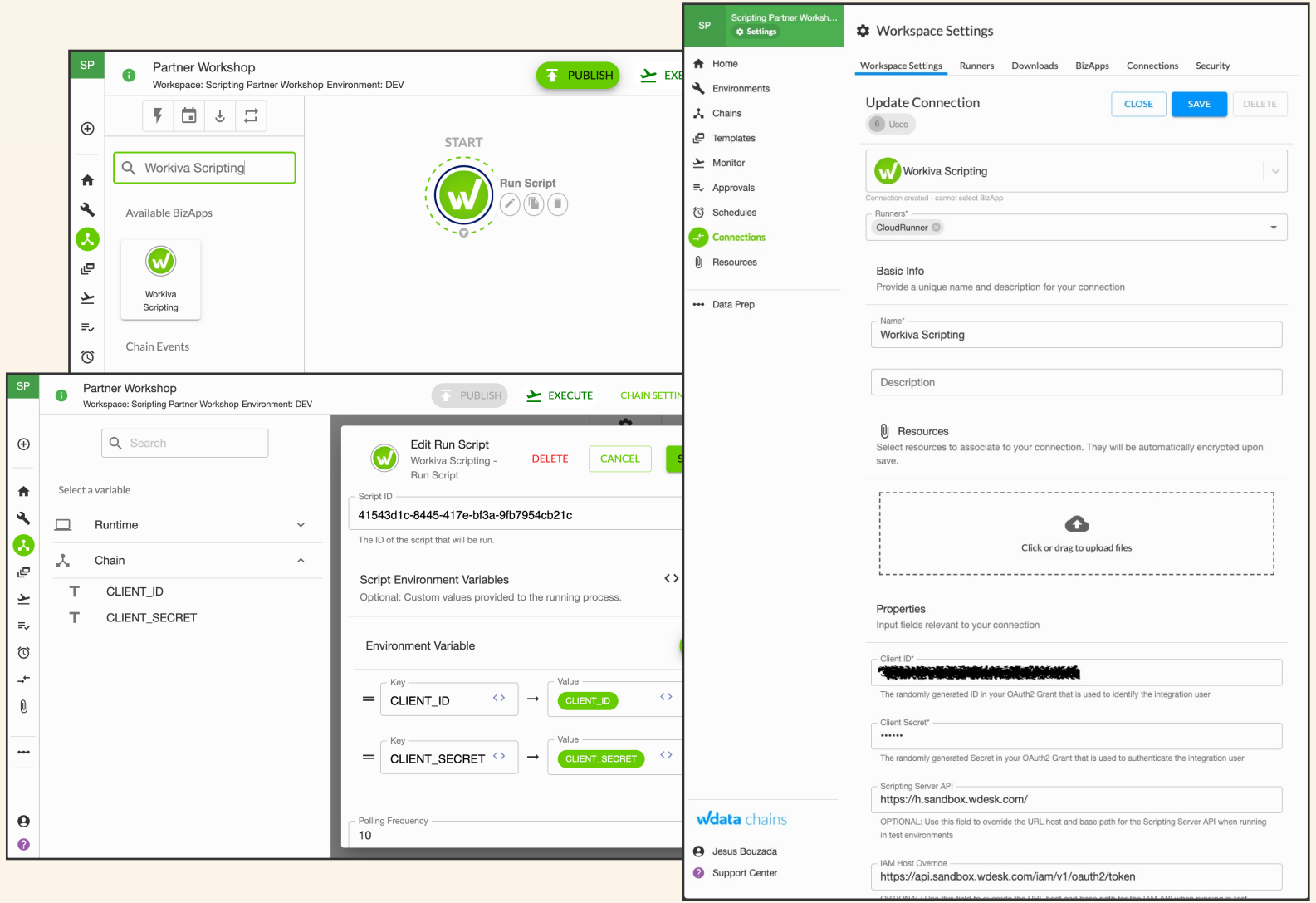



```python
def getSpreadsheetId(wurl):

def getSheetId(spreadsheetId, wurl):

# Capture parameters (inputs) when you run the Script from the
# Scripting Editor, a Chain, or directly from the Scripting API

client_id = os.getenv("CLIENT_ID")
client_secret = os.getenv("CLIENT_SECRET")

trigger_document_id = os.getenv("DOC_ID")
inputs_spreadsheet_id = os.getenv("SPREADSHEET_ID")
inputs_sheet_id = os.getenv("SHEET_ID")

# Override inputs when you run the Script from an integrated automation

ia_inputs_spreadsheet_id = os.getenv("INPUT_SHEET_ID")

if ia_inputs_spreadsheet_id:

    # Hardcode client id and client secret if triggered from Integrated Automation

    client_id = 'YOUR_CLIENT_ID'
    client_secret = 'YOUR_CLIENT_SECRET'

    trigger_document_id = os.getenv("DOCUMENT_ID")

    inputs_spreadsheet_id = getSpreadsheetId(ia_inputs_spreadsheet_id)

    if document_id:
        sheet_id = getSheetId(document_id, os.getenv("INPUT_RESOURCE_ID"))
```

Running Scripts from Chains

Chains Connector

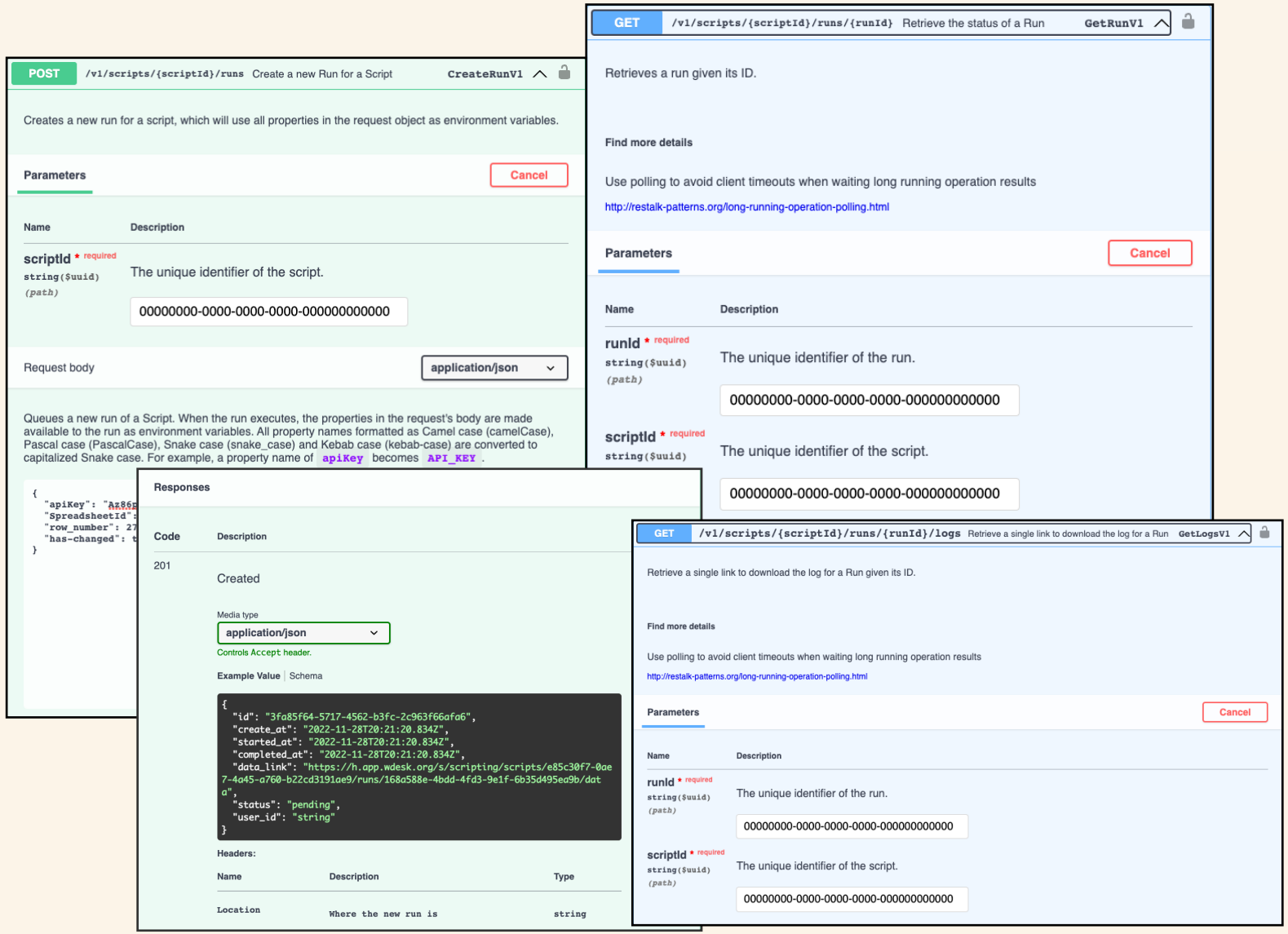
- Ideal to **optimize your Chains**
- Easily define your input parameters
- Leverage Chains' scheduling and monitoring



Running Scripts with the Scripting API

Scripting API

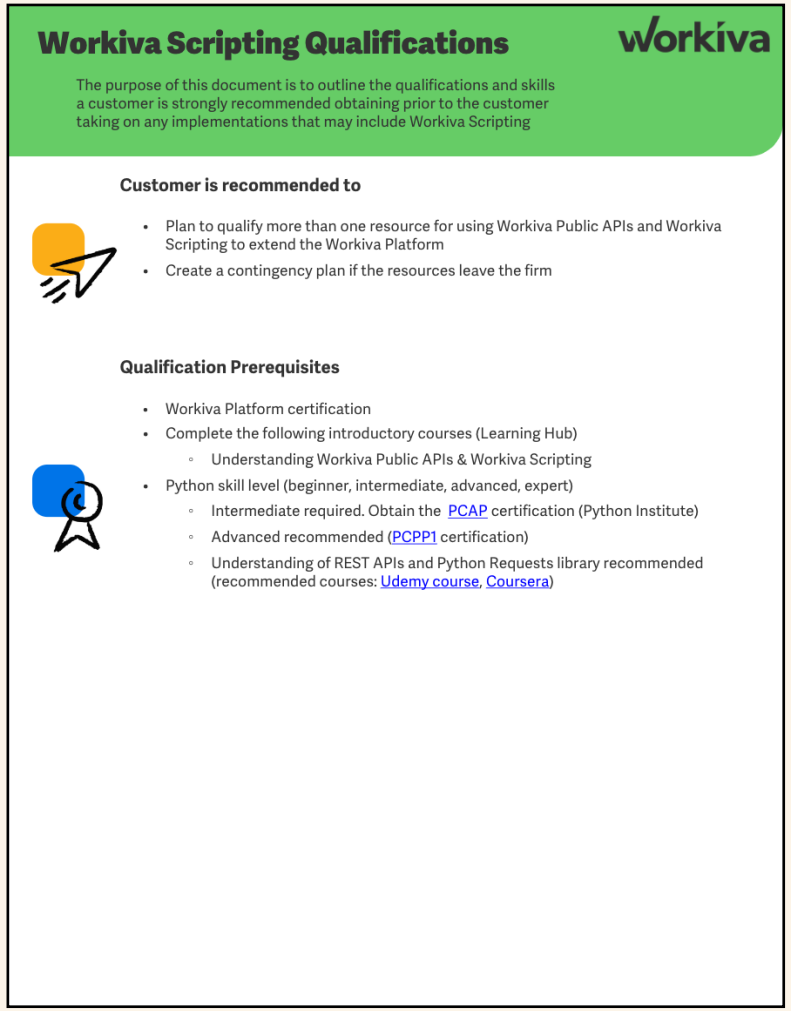
- Ideal to **integrate with 3rd party systems**
- Send input parameters with the body of the request
- Use specific endpoints to check status and get output



Scripting Demo

Workiva

Qualification Prerequisites



Workiva Scripting Qualifications

Workiva

The purpose of this document is to outline the qualifications and skills a customer is strongly recommended obtaining prior to the customer taking on any implementations that may include Workiva Scripting

Customer is recommended to

- Plan to qualify more than one resource for using Workiva Public APIs and Workiva Scripting to extend the Workiva Platform
- Create a contingency plan if the resources leave the firm

Qualification Prerequisites

- Workiva Platform certification
- Complete the following introductory courses (Learning Hub)
  - Understanding Workiva Public APIs & Workiva Scripting
- Python skill level (beginner, intermediate, advanced, expert)
  - Intermediate required. Obtain the PCAP certification (Python Institute)
  - Advanced recommended (PCPP1 certification)
  - Understanding of REST APIs and Python Requests library recommended (recommended courses: Udemy course, Coursera)

- Strongly recommended qualifying two people or more
- Workiva Public APIs and Scripting training (Learning Hub)
- Intermediate Python skills (PCAP)
- Understanding of REST APIs

Workiva

Deployment Scenarios

Build your own app model

Distributed Hosting Customer Workspace

- A customer has the ability to create, modify and execute scripts
- Customer (or partner) creates and owns scripts to meet their unique needs
- Accelerators (i.e. certified examples) available via marketplace

workiva

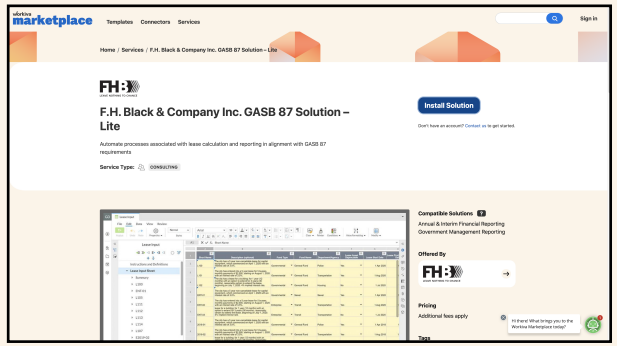
Deployment Scenarios

Build your own app model

Distributed Hosting
Customer Workspace

- A customer has the ability to create, modify and execute scripts
- Customer (or partner) creates and owns scripts to meet their unique needs
- Accelerators (i.e. certified examples) available via marketplace

iPhone AppStore model



Centralized Hosting
Partner Workspace

- Customers install or purchase a packaged solution or extension
- Customer has no knowledge of implementation details
- Customer cannot modify, extend, change solution beyond exposed configuration options

Workiva

1 Workiva Dev Platform

2 Workiva Public APIs

3 Workiva Scripting

4 Use Cases

5 Addtional Resources

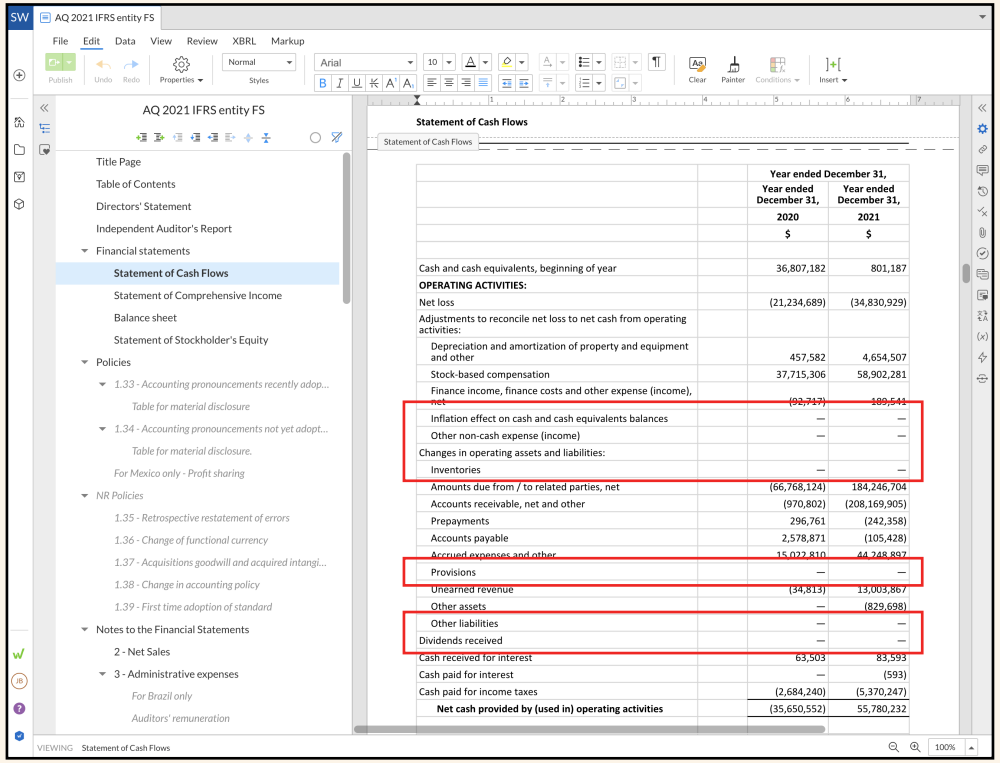
workiva

Zero Suppression

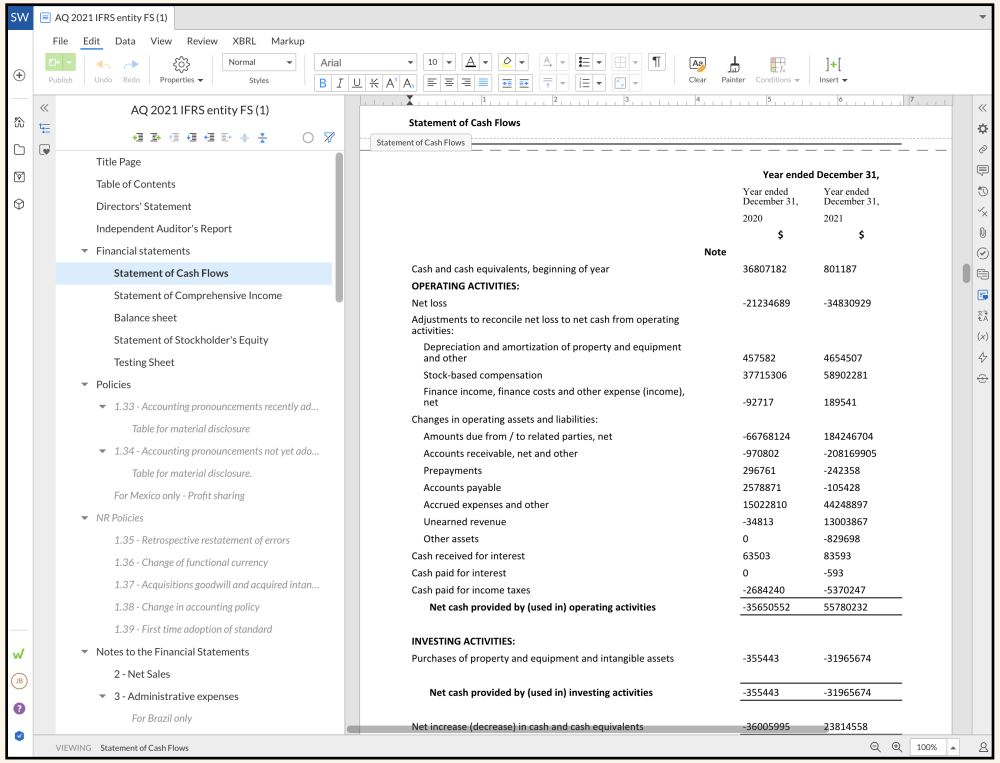
!!! The Challenge

Our customers leverage the Workiva GSR solution for hundreds of entities around the globe. They create documents with linked tables that often end up with rows with empty data due to the entity. These empty rows are not relevant and must be hidden in each entity document.

Before



After



Zero Suppression

Prior to Scripting / APIs

Process

- Manually open each document
- Scroll through document and find tables
- Unhide rows in each table
- Hide rows in each table (filter or manual selection)
- Hide rows in each document takes roughly 10-15 minutes (times 400)

Impacts

- Numerous hours of manual work
- Error prone
- Scalability concerns

With Scripting & APIs

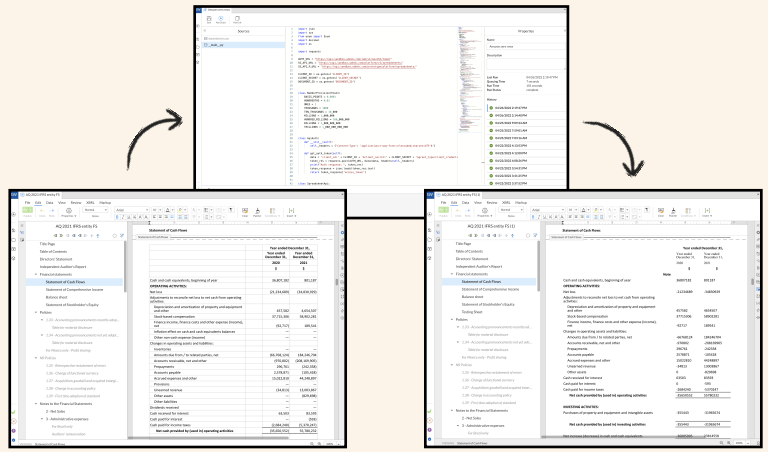
Process

- Few hours of work to create reusable script
- Script can be run via chain or as needed
- Automatically scans documents and formats tables
- Hiding rows in all 400 documents done in minutes

Impacts

- Small investment to save countless hours
- Provides a scalable path to more entities
- Can be customized for other unique needs (i.e. logic on which rows to hide)
- Portable

770 rows automatically hidden, just one report out of 400

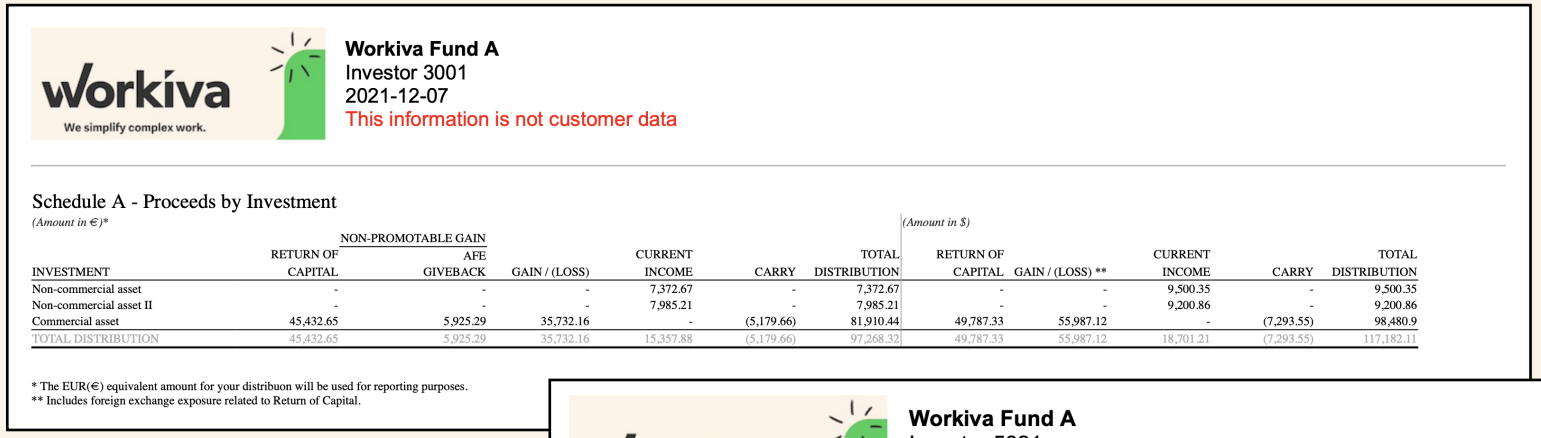


workiva

Investor Correspondence

!!! The Challenge

Large financial service firms need to report to investors on a regular basis, this largely is the generation of hundreds to thousands of documents that match a template but fill in data specific to the investor



Workiva
We simplify complex work.

Workiva Fund A
Investor 3001
2021-12-07
This information is not customer data

Schedule A - Proceeds by Investment

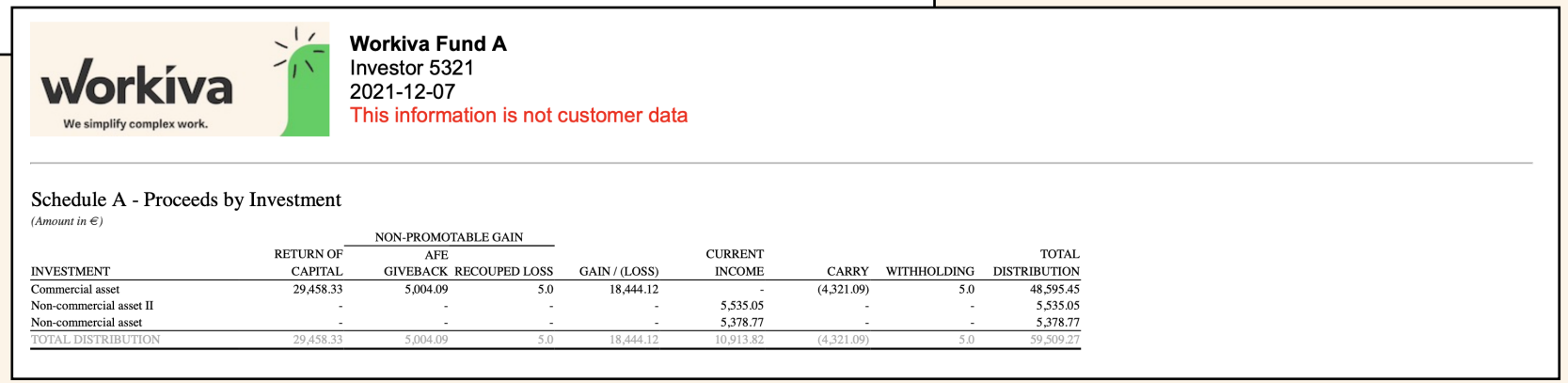
*(Amount in €)**

INVESTMENT	RETURN OF CAPITAL	NON-PROMOTABLE GAIN AFE GIVEBACK	GAIN / (LOSS)	CURRENT INCOME	CARRY	TOTAL DISTRIBUTION
Non-commercial asset	-	-	-	7,372.67	-	7,372.67
Non-commercial asset II	-	-	-	7,985.21	-	7,985.21
Commercial asset	45,432.65	5,925.29	35,732.16	-	(5,179.66)	81,910.44
TOTAL DISTRIBUTION	45,432.65	5,925.29	35,732.16	15,357.88	(5,179.66)	97,268.32

(Amount in $)

RETURN OF CAPITAL	GAIN / (LOSS) **	CURRENT INCOME	CARRY	TOTAL DISTRIBUTION
-	-	9,500.35	-	9,500.35
-	-	9,200.86	-	9,200.86
49,787.33	55,987.12	-	(7,293.55)	98,480.9
49,787.33	55,987.12	18,701.21	(7,293.55)	117,182.11

* The EUR(€) equivalent amount for your distribuon will be used for reporting purposes.
** Includes foreign exchange exposure related to Return of Capital.



Workiva
We simplify complex work.

Workiva Fund A
Investor 5321
2021-12-07
This information is not customer data

Schedule A - Proceeds by Investment

(Amount in €)

INVESTMENT	RETURN OF CAPITAL	NON-PROMOTABLE GAIN AFE GIVEBACK	RECOUPED LOSS	GAIN / (LOSS)	CURRENT INCOME	CARRY	WITHHOLDING	TOTAL DISTRIBUTION
Commercial asset	29,458.33	5,004.09	5.0	18,444.12	-	(4,321.09)	5.0	48,595.45
Non-commercial asset II	-	-	-	-	5,535.05	-	-	5,535.05
Non-commercial asset	-	-	-	-	5,378.77	-	-	5,378.77
TOTAL DISTRIBUTION	29,458.33	5,004.09	5.0	18,444.12	10,913.82	(4,321.09)	5.0	59,509.27

Workiva

Investor Correspondence

400 pdf reports generated and distributed within minutes

Prior to Scripting / APIs

Process

- Manually create each document
- Identify investor needs and populate content and apply format accordingly
- Manually export each document
- Distribute each document
- A chain cannot be used to meet formatting and speed requirements

Impacts

- Large financial institutions would look for alternatives to Workiva's solutions
- Error prone
- Scalability concerns

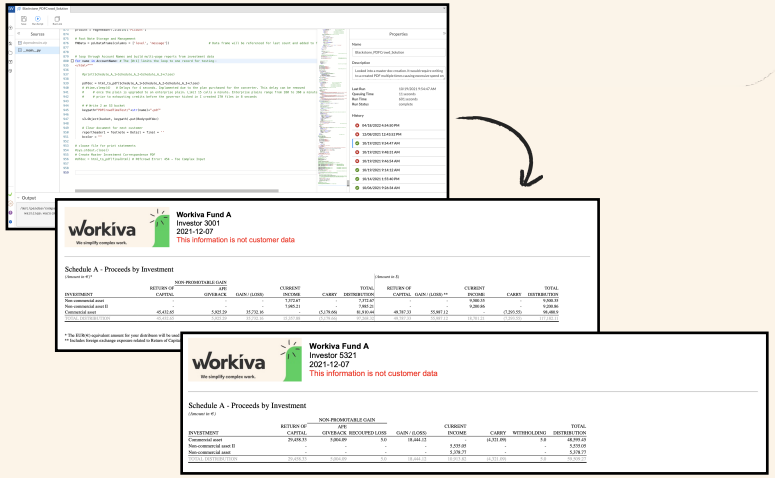
With Scripting & APIs

Process

- A power user with Python experience can create custom logic to automate the process
- The unique needs of each investor are specified in a spreadsheet that serves as a control dashboard
- Script is run via chain or from a workiva file
- Generation of documents and data population automated with a click of a button

Impacts

- Attract large financial institutions
- Removed hundreds of hours of manual work
- Portable

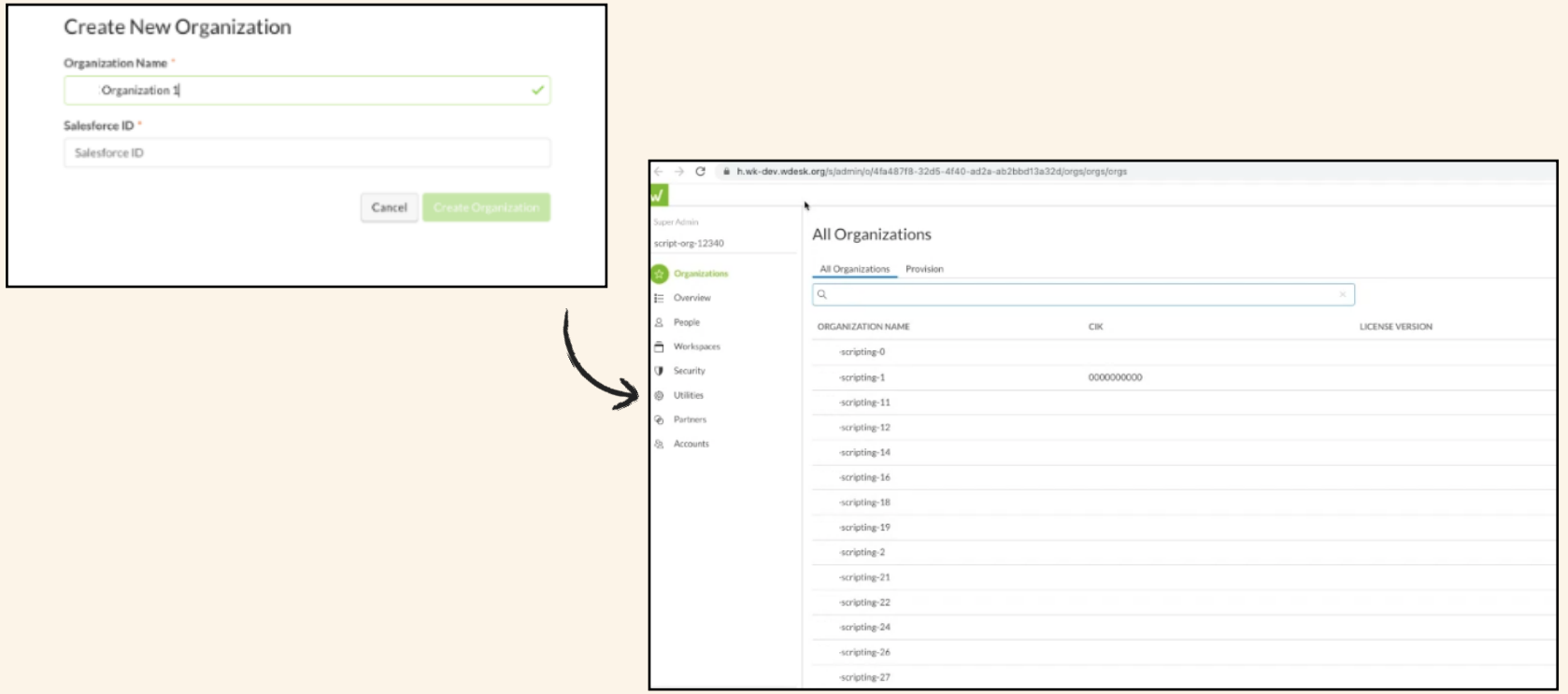


workiva

Onboarding At Scale

!!! The Challenge

When it comes to onboarding, there is a significant amount of manual effort involved to support the creation of, not just one, but sometimes hundreds of organizations and workspaces, and potentially thousands of users. This is very common with Managed Service Providers (MSPs) partners.



Onboarding At Scale

Prior to Scripting / APIs

Process

- Manually create each organization
- Manually create each workspace
- Manually assign each user to an organization and to a workspace
- Manually assign each user a role in each workspace

Impacts

- Numerous hours of manual work
- Error prone
- Dramatic increase in time from contract to onboard

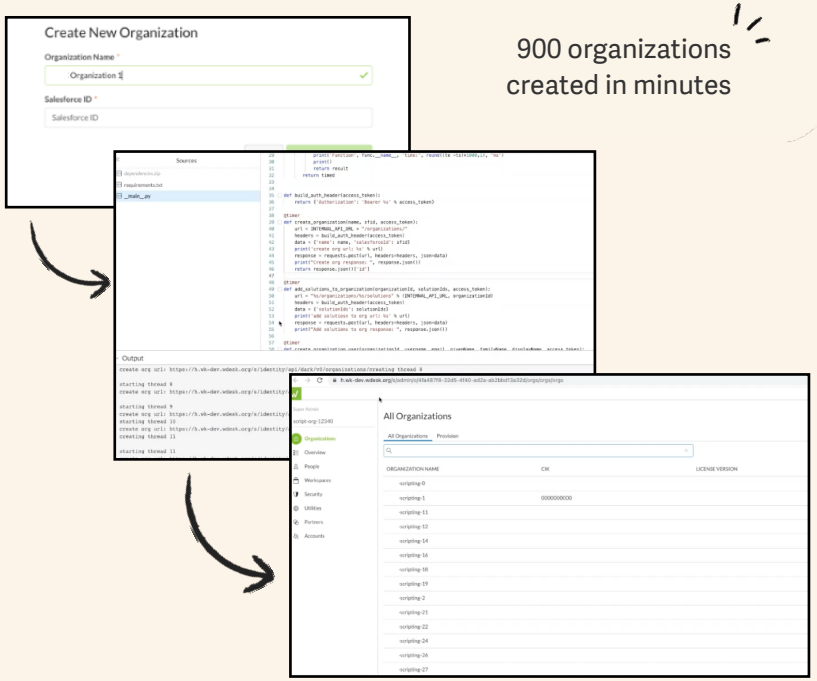
With Scripting & APIs

Process

- Customer provides the number of organizations and workspaces required as well as list of users and roles per workspace
- Few hours of work to create reusable script
- Use of a SuperAdmin token in production to run the script with necessary privileges

Impacts

- Significant reduction in onboarding time for MSPs
- Ability to break the script down into single transaction (e.g. add user to workspace with role) to automate individual processes for other clients



900 organizations created in minutes

workiva

1 Workiva Dev Platform

2 Workiva Public APIs

3 Workiva Scripting

4 Use Cases

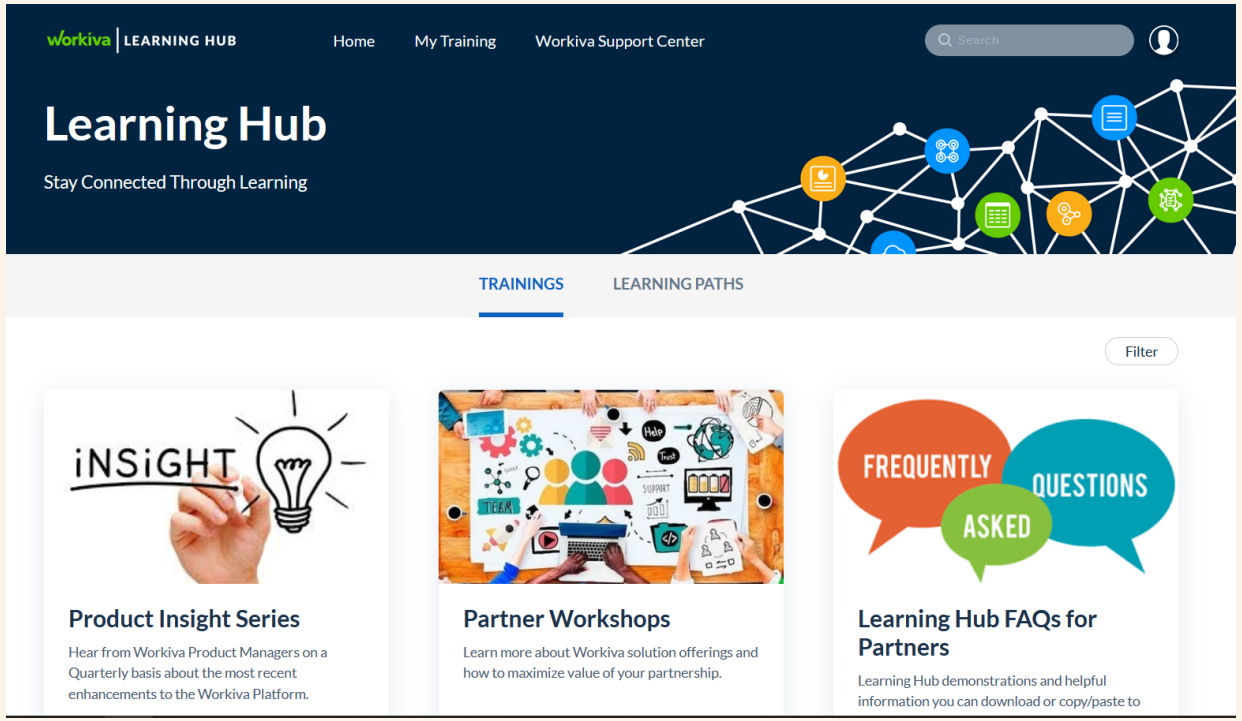
5 Addtional Resources

workiva

Training Opportunities with Workiva

Learning Paths on Training tab

- Workiva Essentials
- Workiva Public APIs
- Workiva Scripting



workiva

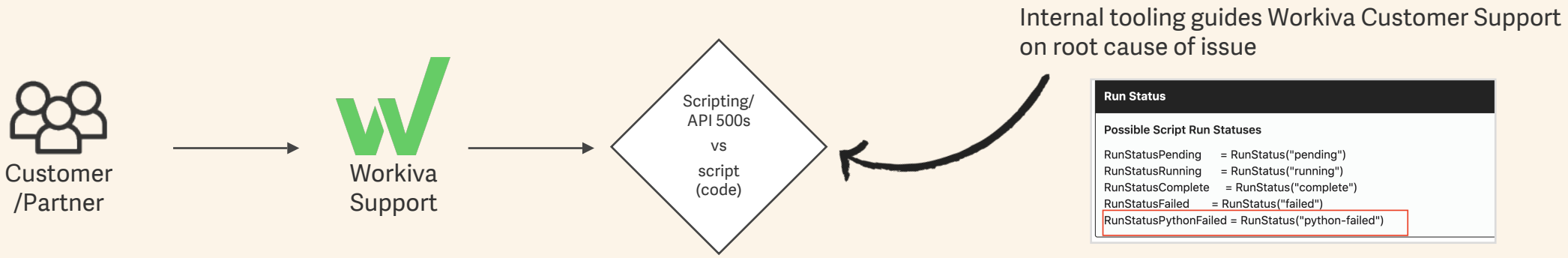
Product Support vs Code Support

(Workiva Public APIs and Workiva Scripting products)

workiva

Product Support vs Code Support

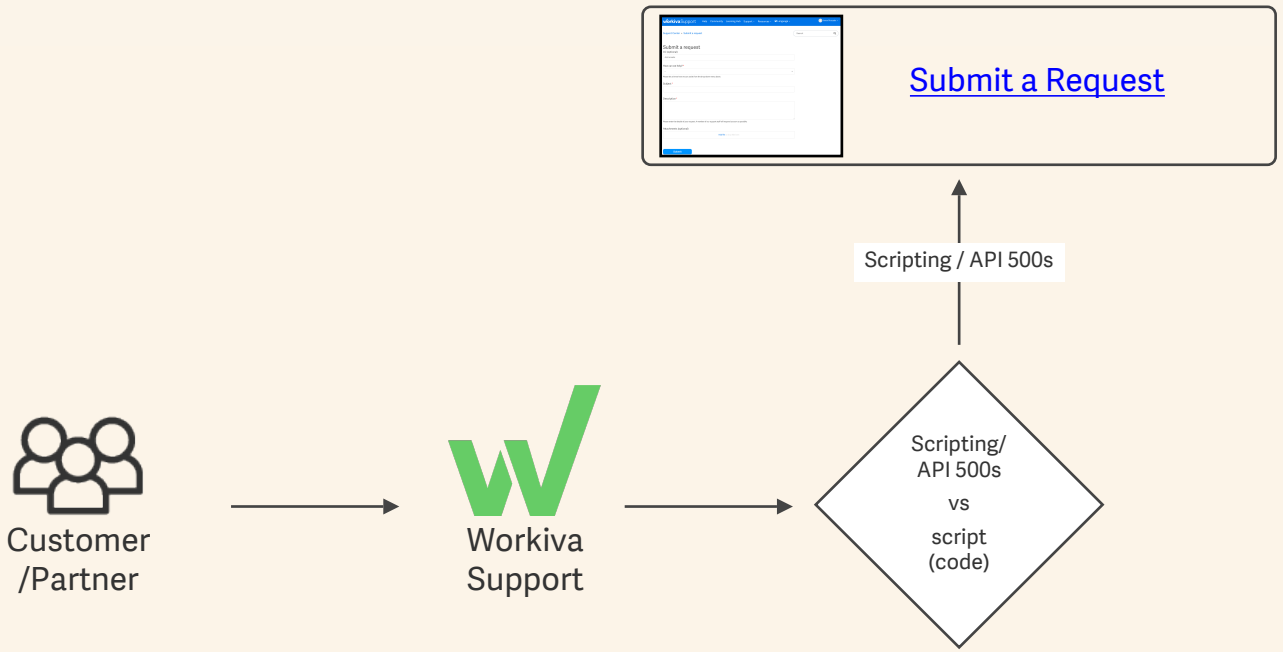
(Workiva Public APIs and Workiva Scripting products)

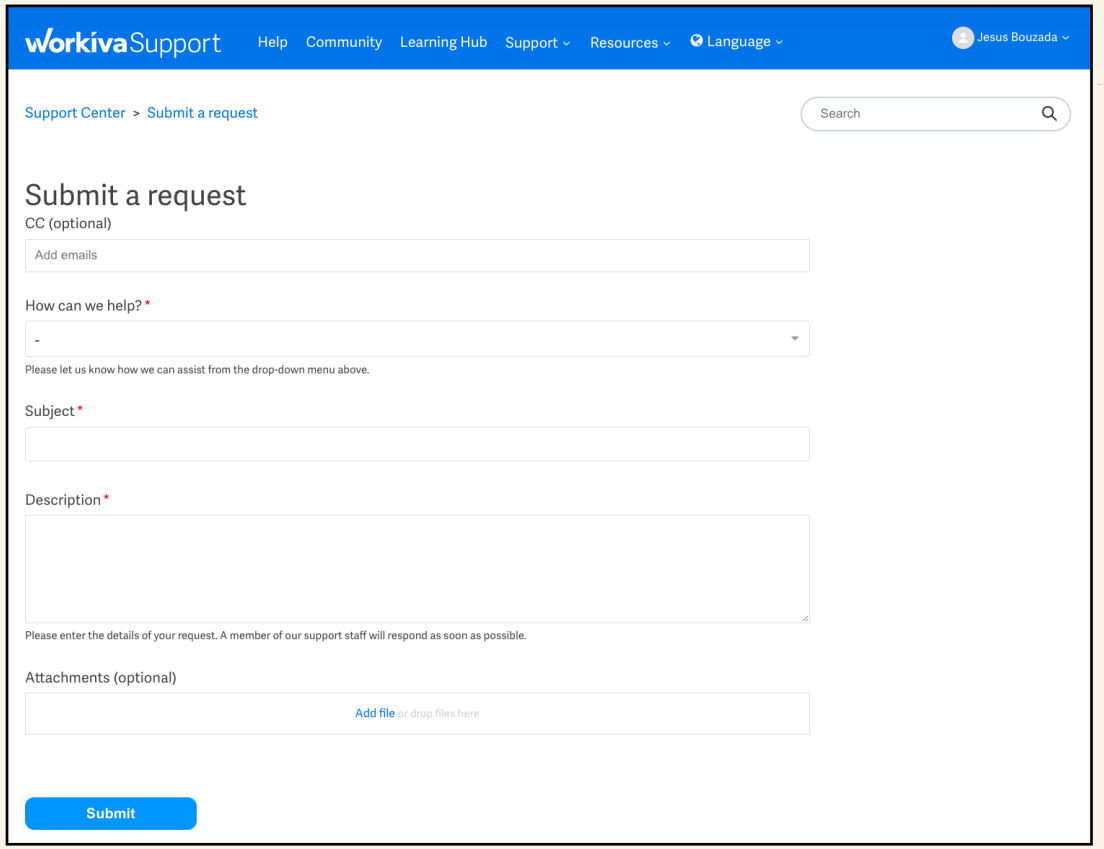


workiva

Product Support vs Code Support

(Workiva Public APIs and Workiva Scripting products)



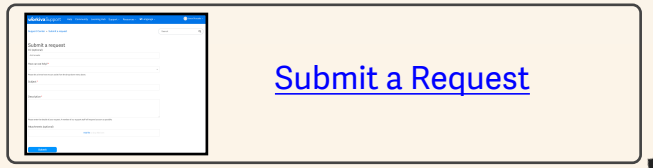


workiva

Product Support vs Code Support

(Workiva Public APIs and Workiva Scripting products)

Workiva Customer Support triages issues with the products and escalates to engineering team, Workiva can't debug your Python code



Submit a Request

Scripting / API 500s

Internal tooling guides Workiva Customer Support on root cause of issue

Customer /Partner

Workiva Support

Scripting/ API 500s vs script (code)

Run Status

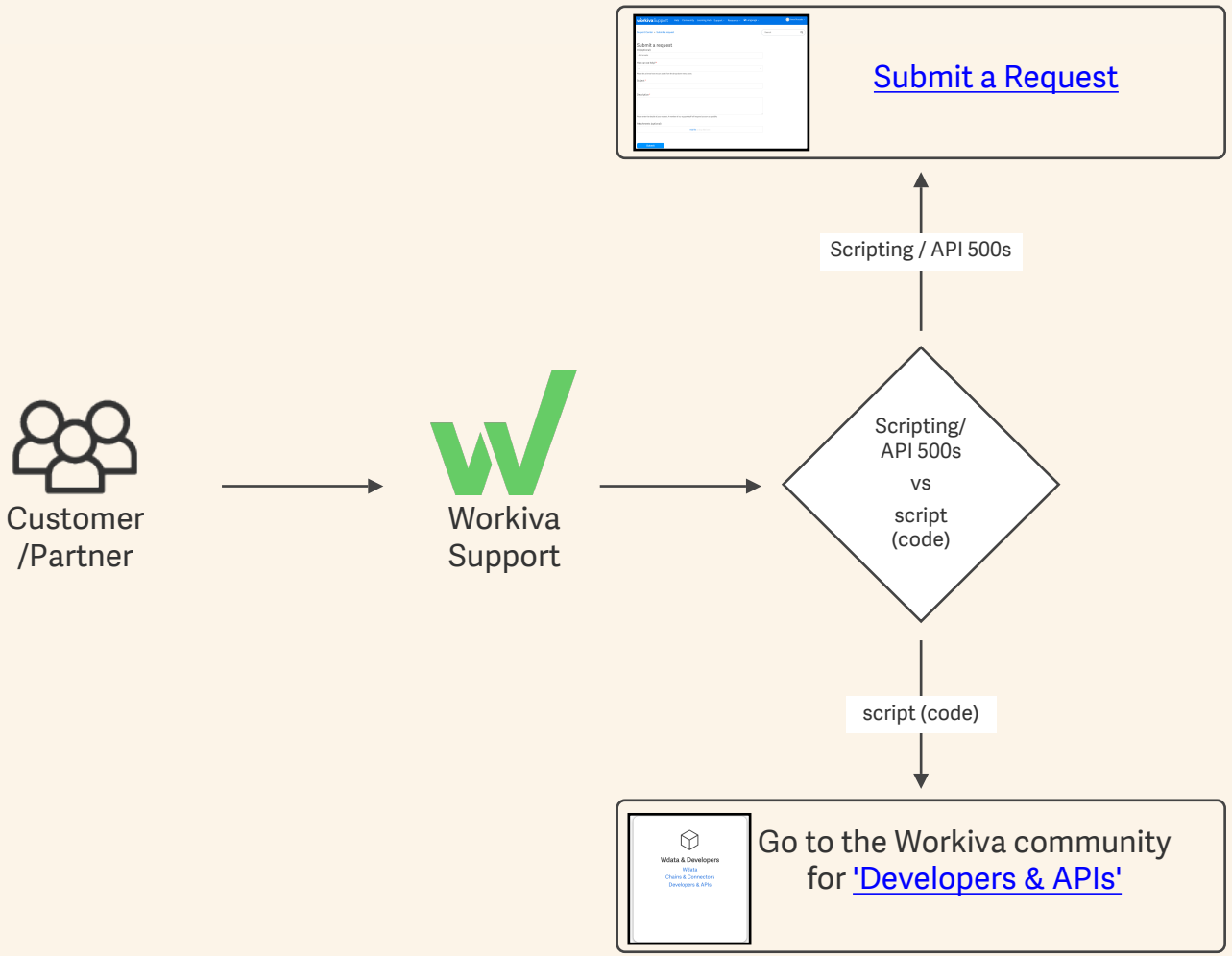
Possible Script Run Statuses

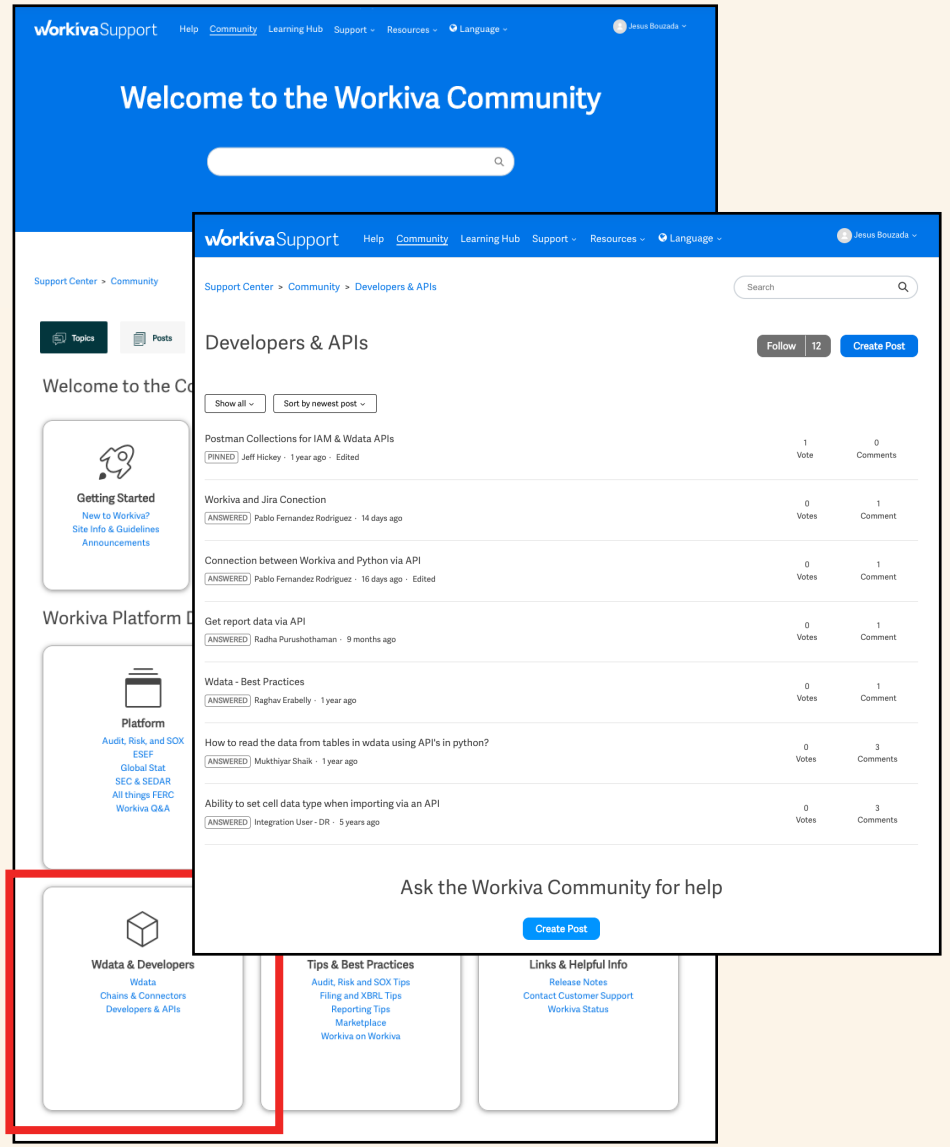
```
RunStatusPending      = RunStatus("pending")
RunStatusRunning      = RunStatus("running")
RunStatusComplete     = RunStatus("complete")
RunStatusFailed       = RunStatus("failed")
RunStatusPythonFailed = RunStatus("python-failed")
```

workiva

Product Support vs Code Support

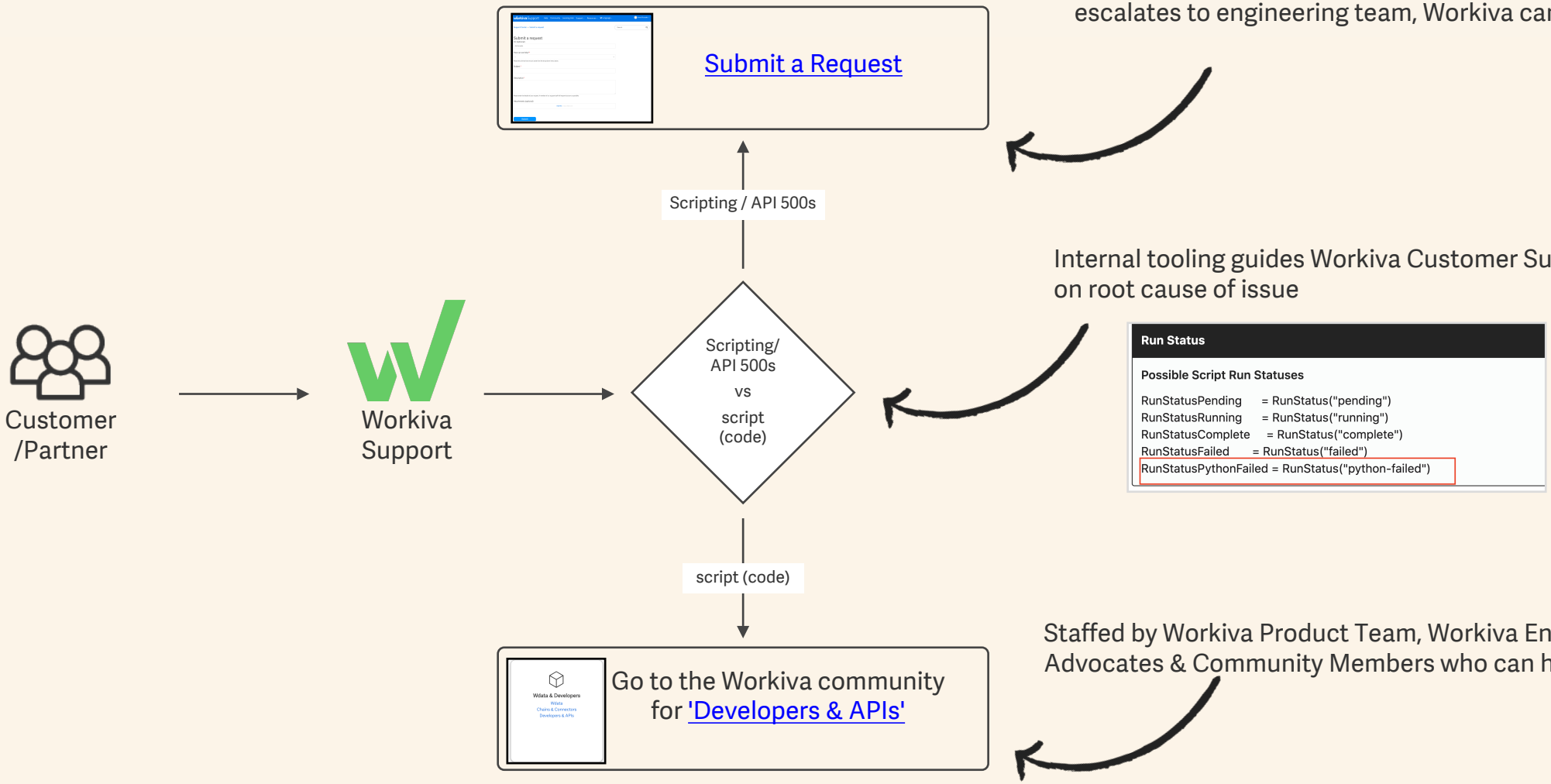
(Workiva Public APIs and Workiva Scripting products)





Product Support vs Code Support

(Workiva Public APIs and Workiva Scripting products)



Workiva Customer Support triages issues with the products and escalates to engineering team, Workiva can't debug your Python code

Internal tooling guides Workiva Customer Support on root cause of issue

Run Status

Possible Script Run Statuses

```
RunStatusPending      = RunStatus("pending")
RunStatusRunning      = RunStatus("running")
RunStatusComplete     = RunStatus("complete")
RunStatusFailed       = RunStatus("failed")
RunStatusPythonFailed = RunStatus("python-failed")
```

Staffed by Workiva Product Team, Workiva Engineering, and non-Workiva Advocates & Community Members who can help with debugging Python

workiva

A Ton of Coding Resources

Certifications (Python Institute)

- PCAP - Certified Associate in Python Programming
- PCPP1 - Certified Professional in Python Programming 1
- PCPP2 – Certified Professional in Python Programming 2

Courses

- Learn to Program: The Fundamentals (Coursera)
- Learn to Program: Crafting Quality Code (Coursera)
- Data Collection and Processing with Python (Coursera)
- Learn Python Requests (Udemy)

Blogs

- Python and REST APIs: Interacting With Web Services (Real Python)
- How to use an API with Python (Beginner's Guide)

Books

- Dive into Python 3

Community

- Python community at Stack overflow

workiva

Thank you!