



**Understanding
Workiva Public APIs
& Workiva Scripting**



Agenda

1 Workiva Dev Platform

2 Workiva Public APIs

3 Workiva Scripting

4 Use Cases

5 Additional Resources



Workiva Dev Platform



Workiva Public APIs



Workiva Scripting



Use Cases



Additional Resources

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



Public APIs

+



Workiva Scripting

SEC IR GSR ESG +

workiva

A grid of 15 icons representing various business and technology solutions, arranged in three rows of five. The icons include a document, a pie chart, a calendar, a folder, a server rack, a speech bubble, a flask, a globe, a bar chart, a grid, a cube, a code symbol, a grid, a function symbol, and a large X. Below the grid are three dots, with the first one being blue.

Solutions Platform

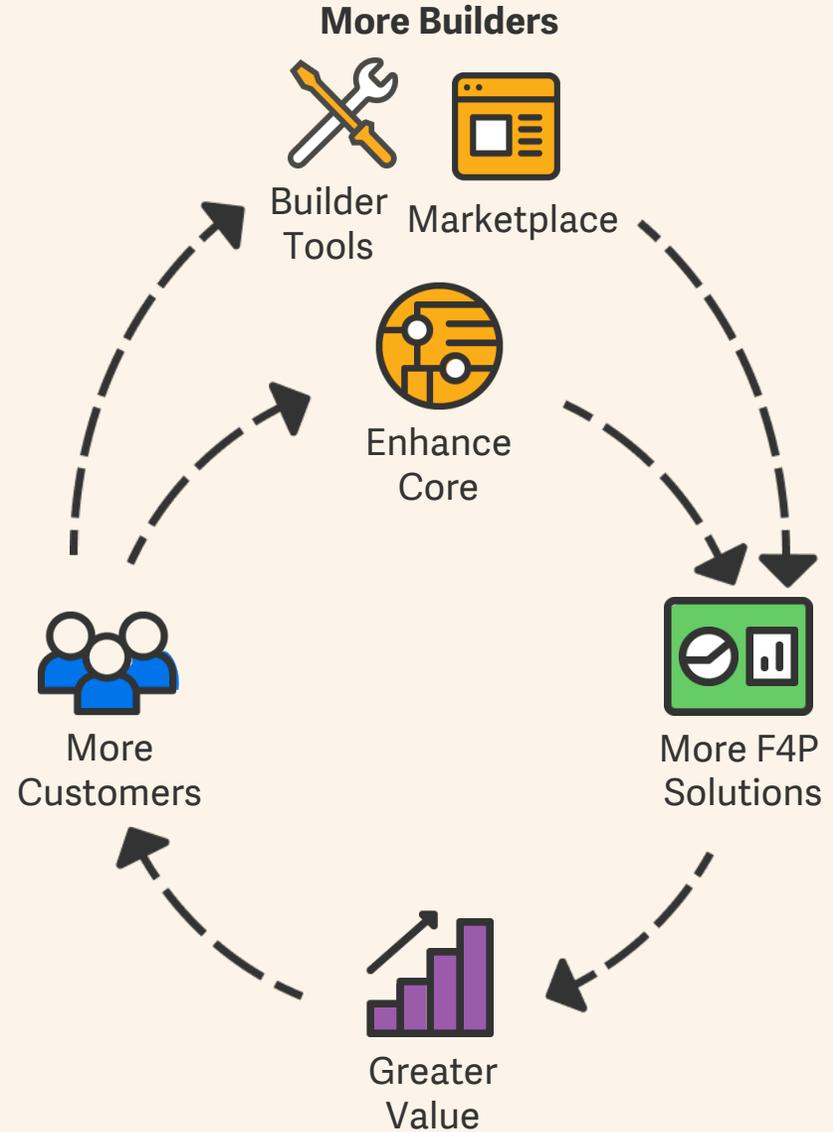
///

Our Growth Flywheel



☺

Our Growth Flywheel with Builders



Automation & Connectivity Ecosystem

ESG Data Collection Module 2020

STATUS	TITLE	ASSIGNEE	APPROVER	DUE DATE
Approval	Scope 1 - North A...	Michael Carter	Lori Vanourek	9/30/2021
Complete	Scope 1 - North A...	Michael Carter	Katie Sanderson	9/30/2021
Approval	Scope 1 - North A...	Michael Carter	Lori Vanourek	9/30/2021
Complete	Scope 1 - North A...	Michael Carter	Katie Sanderson	9/30/2021
Returned	Scope 1 - North A...	Michael Carter	Katie Sanderson	9/30/2021
Complete	Scope 1 - North A...	Michael Carter	Katie Sanderson	9/30/2021
Complete	Scope 1 - North A...	Michael Carter	Katie Sanderson	9/30/2021
Canceled	Scope 1 - North A...	Michael Carter	Lori Vanourek	9/30/2021
Sent	Scope 1 - North A...	Michael Carter	Katie Sanderson	9/30/2021

Process Status: In Progress

- 7% Approval
- 26% Complete
- 4% Returned
- 4% Canceled
- 59% Sent

Process Builder/Monitor

Add column to spreadsheet datasets

START -> List Sheets -> Get and upload s... -> Get Sheet Data -> Insert Column -> Run Chain

Available BizApps: Adaptive Insights, Amazon Redshift, Amazon S3, Anaplan, BlackLine, Box, Data Prep, Dell Boomi, Domo, Email

Chain Builder & Data Prep

Document Outline

Table of Contents

UNITED STATES SECURITIES AND EXCHANGE COMMISSION
Washington, D.C. 20549
FORM 10-Q

ABC COMPANY

Create New Automation

Automation Trigger: Outline Label Changes

Trigger Filters: All Sections

Are marked as: Any Label

By: Any Person

Action Details: Send Email Notification

To: Alice Bentley

Integrated Automations

Automation & Connectivity Ecosystem

ESG Data Collection Module 2020

STATUS	TITLE	ASSIGNEE	APPROVER	DUE DATE
Approval	Scope 1- North A...	Michael Carter	Lori Vanourek	9/30/2021
Complete	Scope 1- North A...	Michael Carter	Katie Sanderson	9/30/2021
Approval	Scope 1- North A...	Michael Carter	Lori Vanourek	9/30/2021
Complete	Scope 1- North A...	Michael Carter	Katie Sanderson	9/30/2021
Returned	Scope 1- North A...	Michael Carter	Katie Sanderson	9/30/2021
Complete	Scope 1- North A...	Michael Carter	Katie Sanderson	9/30/2021
Complete	Scope 1- North A...	Michael Carter	Katie Sanderson	9/30/2021
Canceled	Scope 1- North A...	Michael Carter	Lori Vanourek	9/30/2021
Sent	Scope 1- North A...	Michael Carter	Katie Sanderson	9/30/2021

Process Status: In Progress

- 7% Approval
- 26% Complete
- 4% Returned
- 4% Canceled
- 59% Sent

Process Builder/Monitor

1 Add column to spreadsheet datasets

Workspace: Dillon Jenkins Environment: DEV

Available BizApps: Adaptive Insights, Amazon Redshift, Amazon S3, Anypoint, BlackLine, Box, Data Prep, Dell Boomi, Domino, Email

Workflow: START -> List Sheets -> Get and upload s... -> Group Start -> Get Sheet Data -> Insert Column -> Run Chain

Chain Builder & Data Prep

Document Outline

UNITED STATES SECURITIES AND EXCHANGE COMMISSION FORM 10-Q

Automation Trigger: Outline Label Changes

Trigger Filters: All Sections

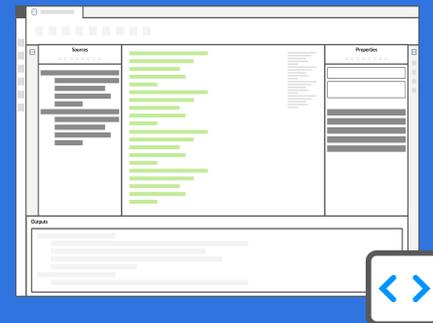
Are marked as: Any Label

Action Details: Send Email Notification, Alice Bentley

Integrated Automations



Public APIs



Workiva Scripting



1 Workiva Dev Platform

2 Workiva Public APIs

3 Workiva Scripting

4 Use Cases

5 Additional Resources

Platform APIs

Access to the core data & capabilities of Workiva



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

Platform APIs

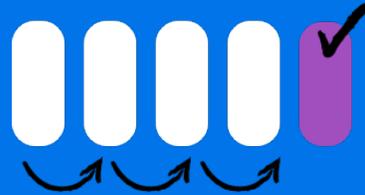
Access to the core data & capabilities of Workiva



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

Admin APIs

Automate the setup & administration of Workiva



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

Platform APIs

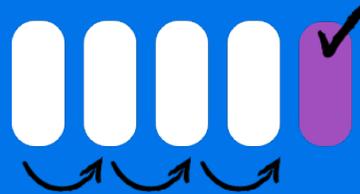
Access to the core data & capabilities of Workiva



- 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!

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

Products and Solutions

- Connect
- Automate
- Transform
- Wdesk
- Wdata

Learn

- Resource Library
- Workiva Blog
- Education
- Events

Company

- Investor Relations
- Press
- Careers
- Locations
- Contact Us

Support

- Getting Started
- Help
- Community
- Services
- Customer Support

Authorized APIs - Generate Credentials

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

1. Click Add a grant

The screenshot shows the Workiva wdesk interface. The top navigation bar includes 'Dashboard', 'Settings', 'People', 'Content', 'Permissions', 'Activities', and 'Mobile Devices'. The 'People' menu is highlighted in green. Below the navigation bar, there are tabs for 'Members', 'Groups', and 'OAuth2 Grants'. The 'OAuth2 Grants' tab is selected. The page displays the following information:

OAuth2 Grants

Organization: Workiva
Account Number: 30925364168
Account Name: Scripting Partner Workshop

[+ Add a grant](#)

Name	ID	User	Scope	Expires	IP Whitelist	Actions	...
No items found on this page.							

Red arrows in the image point to the 'People' menu item, the 'OAuth2 Grants' sub-menu, the '+ Add a grant' button, and the 'Name' column header in the table.

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)

The screenshot shows the 'wdesk' interface. The top navigation bar includes 'Dashboard', 'Settings', 'People', 'Content', 'Permissions', 'Activities', and 'Mobile Devices'. The main content area is titled 'OAuth2 Grants' and shows organization details: 'Organization: Workiva', 'Account Number: 30925364168', and 'Account Name: Scripting Partner Workshop'. A '+ Add a grant' button is visible. An 'Add An OAuth2 Grant' modal is open, displaying a form with fields for 'Grant Name', 'Username', and 'Expiration'. Below these fields is a list of scopes: 'Graph API (Read)', 'Graph API (Write)', 'SCIM (Read)', 'Graph Admin Access', 'Read Files', 'Write Files', 'Read Graph', 'Write Graph', 'Read Tasks', and 'Write Tasks'. The 'Write Tasks' scope is highlighted in blue, and a red arrow points to it from the bottom right.

The screenshot shows the 'workiva Developers' API reference page for the 'Create a new task' endpoint. The page includes a sidebar with navigation links like 'Overview', 'API Details', and 'Parameters'. The main content area has a 'Description' section and a 'Parameters' table. A red box highlights the 'Required OAuth Scopes' section, which lists 'task:write'. Below the table is a 'Body parameter example' section with a JSON snippet.

Parameter	In	Type	Required	Description
body	body	Task	true	The properties of the task to create

```
{  "assignee": {    "id": "V1ZVd2VyFzU3Nl01NDAAAJ1zNzA2Mj0"  },  "description": "Review document for spelling and grammar",  "dueDate": "2019-10-30T00:00:00Z"}
```

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

The screenshot shows the Workiva Admin console interface. The top navigation bar includes 'wdesk', 'Dashboard', 'Settings', 'People', 'Content', 'Permissions', 'Activities', 'Mobile Devices', and user information 'Scripting Partner Workshop / Jesus Bouzada'. The main content area is titled 'OAuth2 Grants' and includes a '+ Add a grant' button and a table with columns 'Name' and 'ID'. A modal window titled 'Edit OAuth2 Grant' is open, showing the following fields:

- Client ID: [Redacted]
- Client Secret: [Redacted]
- Grant Name*: Jesus Grant
- Username*: jesus.bouzada@workiva.com
- Scope*: Write Tasks
- Expiration*: 12/01/2023
- IP Whitelist: [Empty]

A tooltip points to the 'Grant Name' field with the text: 'Enter a name that will help you identify this grant'. The modal also features a '*required' label and 'Save changes' and 'Cancel' buttons.

© 2022 Workiva | [Privacy Policy](#) Last sign in: 8:13 AM Monday, November 28 | [Details](#)

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.

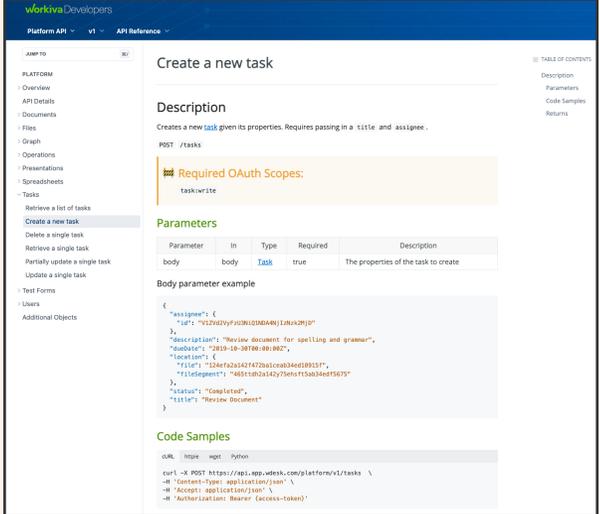
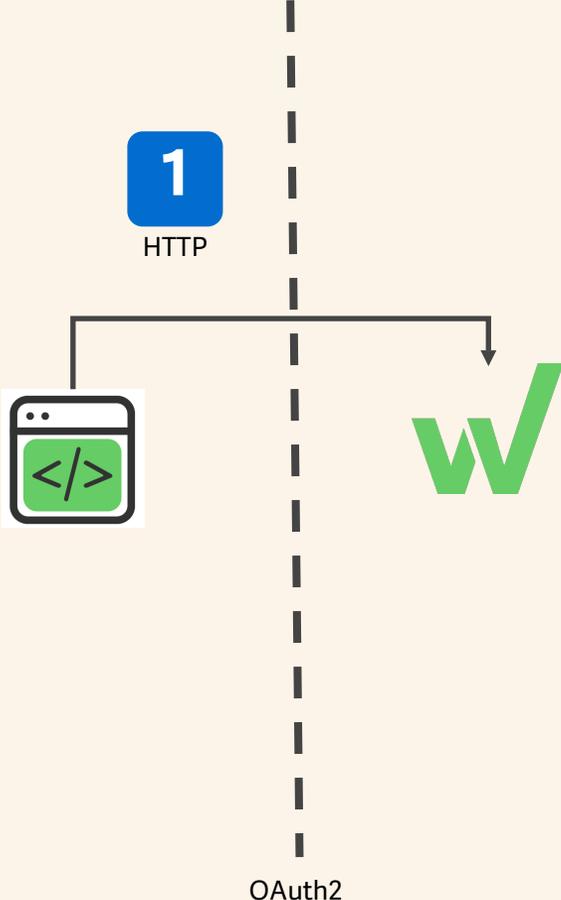
DO NOT SHARE

The image shows two screenshots of the Workiva Admin console. The top screenshot displays the 'OAuth2 Grants' page with a table containing one grant: 'Jesus Grant' with ID '1138de6f9f7f46909eef1a4354192983', User 'Jesus Bouzada (jesus.bouzada@workiva.com)', Scope 'Write Tasks', and Expires '11/30/2023'. A '+ Add a grant' button is visible. The bottom screenshot shows the 'Edit OAuth2 Grant' dialog box. The 'Client ID' and 'Client Secret' fields are highlighted with a red box. The 'Grant Name' is 'Jesus Grant', 'Username' is 'jesus.bouzada@workiva.com', and 'Scope' is 'Write Tasks'. A tooltip points to the 'Grant Name' field with the text 'Enter a name that will help you identify this grant'. The dialog has 'Save changes' and 'Cancel' buttons.

Name	ID	User	Scope	Expires	IP Whitelist	Actions
Jesus Grant	1138de6f9f7f46909eef1a4354192983	Jesus Bouzada (jesus.bouzada@workiva.com)	Write Tasks	11/30/2023		Edit Reset Secret Delete

HTTP-based REST APIs

Workiva Public APIs request-response workflow
1. Application makes request

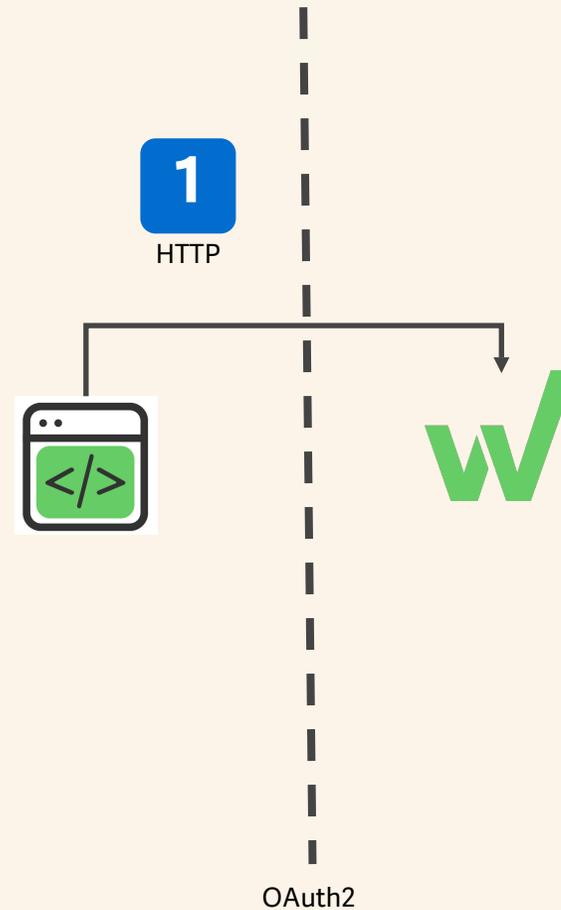


```
20 TASK_API_URL = 'https://api.sandbox.wdesk.com/platform/v1/tasks'  
21  
22 taskHeaders = {  
23     'Content-Type': 'application/x-www-form-urlencoded;charset=UTF-8',  
24     'Accept': 'application/json',  
25     'Authorization': accessToken  
26 }  
27  
28 taskData = {  
29     'assignee': {'id': 'V0Zvc2VyHzY0NDQ5MDY2NTA4NjE1Njg'},  
30     'title': 'test'  
31 }  
32  
33 taskResponse = json.loads(requests.post(TASK_API_URL, json = taskData, headers = taskHeaders).text)
```

HTTP-based REST APIs

Workiva Public APIs request-response workflow

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



workiva Developers
Platform API v1 API Reference

Create a new task

Description
Creates a new `Task` given its properties. Requires passing in a title and assignee.

Required OAuth Scopes:
task:write

Parameter	In	Type	Required	Description
body	body	Task	true	The properties of the task to create

Body parameter example

```
{  
  "assignee": {  
    "id": "V0Zvc2VyHzY0NDQ5MDY2NTA4NjE1Njg"  
  },  
  "description": "Review document for spelling and grammar",  
  "dueDate": "2019-10-30T00:00:00Z",  
  "taskStatus": {  
    "id": "1204f42442f72b01c0a34ed389151",  
    "titleSegment": "40510t0z1a2zy7501n7f5ab34e75659"  
  },  
  "status": "Completed",  
  "title": "Review Document"  
}
```

Code Samples

```
curl -X POST https://api.sandbox.wdesk.com/platform/v1/tasks \\  
-H 'Content-Type: application/json' \\  
-H 'Accept: application/json' \\  
-H 'Authorization: Bearer (access-token)'
```

```
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)
```


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"

The screenshot shows the 'Tasks' resource page in the Workiva Developers API Reference. The page title is 'Tasks' and it includes a description: 'Tasks enable users to manage project tasks in the Workiva platform. Use these endpoints to create, update, and delete tasks.' Below the description is a 'Task' section with a 'Properties' table. The table has two columns: 'Name' and 'Type'. The rows are: 'assignee' (User), 'completed' (Action), 'created' (Action), 'description' (string), 'dueDate' (string(date-time)), 'id' (string), 'location' (TaskLocation|null), 'modified' (Action), 'sourceUrl' (string), 'status' (string), and 'title' (string). A left sidebar contains a navigation menu with categories like PLATFORM, Overview, API Details, Documents, Files, Graph, Operations, Presentations, Spreadsheets, and Test Forms. The 'Tasks' category is currently selected.

The screenshot shows the 'Spreadsheets' resource page in the Workiva Developers API Reference. The page title is 'Spreadsheets' and it includes a description: 'Spreadsheets enable you to work with large, complex data in a familiar, collaborative, and controlled environment. Use these endpoints to manage spreadsheets and their sheets in the Workiva platform.' Below the description is a 'Spreadsheet' section with a 'Properties' table. The table has four columns: 'Name', 'Type', 'Description', and 'Restrictions'. The rows are: 'created' (Action, When the action was performed, and details about the user who did it, read-only), 'id' (string, The unique identifier of the spreadsheet, read-only), 'modified' (Action, When the action was performed, and details about the user who did it, read-only), 'name' (string, The name of the spreadsheet, read-only), 'sheets' (Sheet, An array of partial information about the sheets in this spreadsheet. Optionally included in the response when the \$expand query parameter is provided., read-only), and 'template' (boolean, Whether the spreadsheet is a template, read-only). Below the table is an 'Example' section with a JSON snippet:

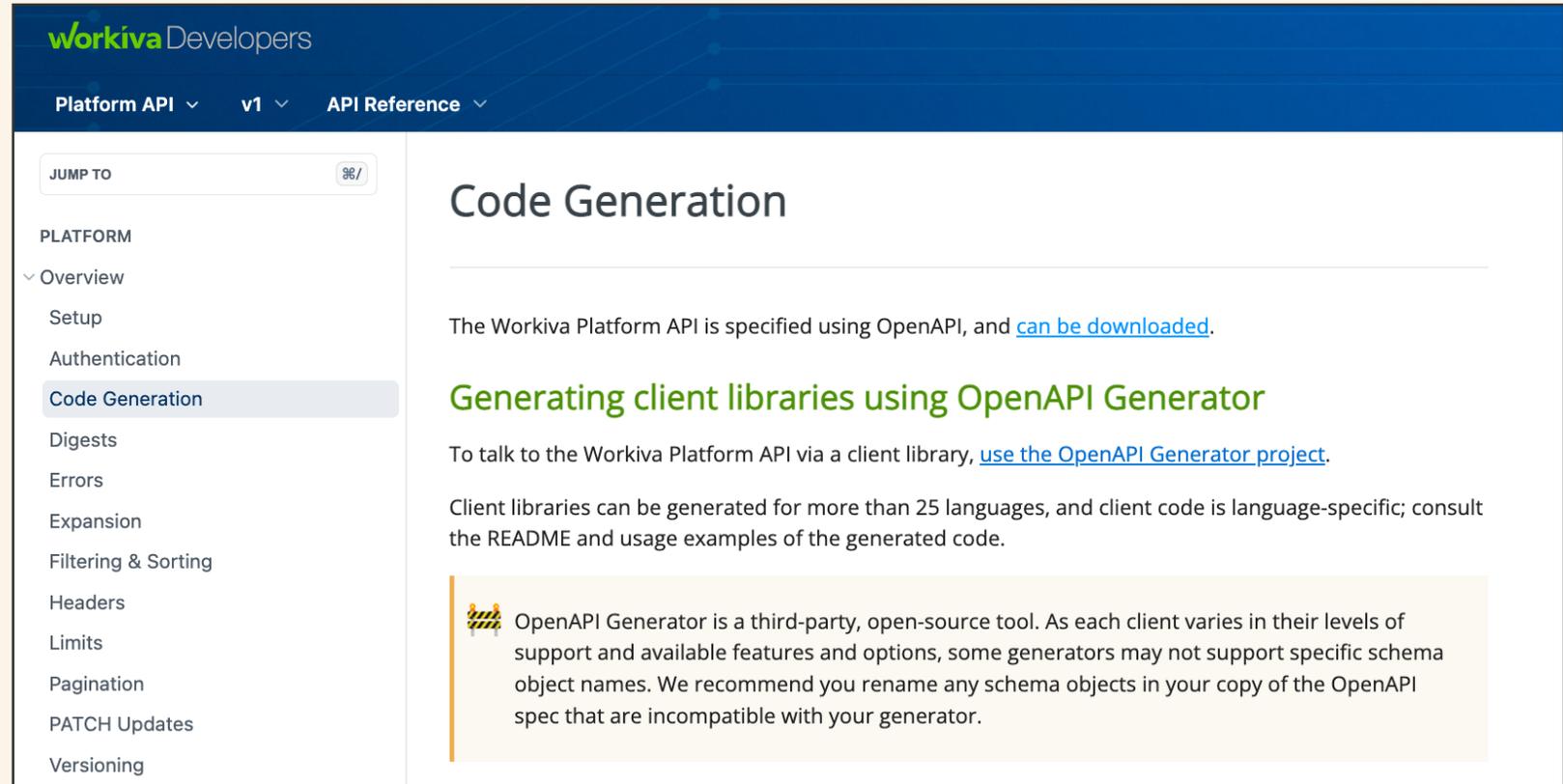
```
{  "created": {    "dateTime": "2019-10-30T15:03:27Z",    "user": {      "displayName": "John Doe"    }  }
```

 A right sidebar contains a navigation menu with categories like Spreadsheet, Properties, Example, SpreadsheetExport, SpreadsheetToPdfOptions, SpreadsheetToXlsxOptions, SpreadsheetToCsvOptions, Sheet, SheetCopy, SheetOptions, Dataset, Datasets, and Example. The 'Spreadsheets' category is currently selected.

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)



The screenshot shows the Workiva Developers API Reference page. The header includes the Workiva logo and 'Developers' text. Below the header, there are navigation links for 'Platform API', 'v1', and 'API Reference'. A 'JUMP TO' search bar is present. The left sidebar lists various API sections, with 'Code Generation' highlighted. The main content area is titled 'Code Generation' and contains the following text:

The Workiva Platform API is specified using OpenAPI, and [can be downloaded](#).

Generating client libraries using OpenAPI Generator

To talk to the Workiva Platform API via a client library, [use the OpenAPI Generator project](#).

Client libraries can be generated for more than 25 languages, and client code is language-specific; consult the README and usage examples of the generated code.

 OpenAPI Generator is a third-party, open-source tool. As each client varies in their levels of support and available features and options, some generators may not support specific schema object names. We recommend you rename any schema objects in your copy of the OpenAPI spec that are incompatible with your generator.

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

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
```

- npm
- Homebrew
- Scoop
- Docker
- JAR
- Bash Launcher Script

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



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

```
2 import openapi_client
3 from openapi_client.apis.tags import tasks_api
4 from openapi_client.model.task import Task
5
6 # Configure OAuth2 access token for authorization: oauth
7 configuration = openapi_client.Configuration(
8     host = "https://api.app.wdesk.com/platform/v1"
9 )
10
11 configuration.access_token = 'YOUR_ACCESS_TOKEN'
12
13 # Enter a context with an instance of the API client
14 with openapi_client.ApiClient(configuration) as api_client:
15     # Create an instance of the API class
16     api_instance = tasks_api.TasksApi(api_client)
17
18     body = Task(
19         assignee=User(
20             display_name="display_name_example",
21             email="email_example",
22             id="V1ZVd2VyFzU3NiQ1NDA4NjIzNzk2MjD"
23         ),
24         title="Review Document",
25     )
26
27     # Create a new task
28     api_response = api_instance.create_task(body=body)
```

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

```
2 import openapi_client
3 from openapi_client.apis.tags import tasks_api
4 from openapi_client.model.task import Task
5
6 # Configure OAuth2 access token for authorization: oauth
7 configuration = openapi_client.Configuration(
8     host = "https://api.app.wdesk.com/platform/v1"
9 )
10
11 configuration.access_token = 'YOUR_ACCESS_TOKEN'
12
13 # Enter a context with an instance of the API client
14 with openapi_client.ApiClient(configuration) as api_client:
15     # Create an instance of the API class
16     api_instance = tasks_api.TasksApi(api_client)
17
18     body = Task(
19         assignee=User(
20             display_name="display_name_example",
21             email="email_example",
22             id="V1ZVd2VyFzU3NiQ1NDA4NjIzNzk2MjD"
23         ),
24         title="Review Document",
25     )
26
27     # Create a new task
28     api_response = api_instance.create_task(body=body)
```



Workiva Dev Platform



Workiva Public APIs



Workiva Scripting



Use Cases



Additional Resources

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



Workiva Scripting

Workiva Scripting



Available today through our Early Adopter Program

Home

Files > jbouzada

NAME	TYPE	CREATED BY	ST MODIFIED
Jesus Demo Script		Jesus Bouzada	Sep 22, 11:11 AM by Jesus Bouz...
Demo		Jesus Bouzada	Sep 22, 11:02 AM by Jesus Bouz...
SP Squad Support Data		Jesus Bouzada	Jul 15, 11:32 AM by Jesus Bouz...

Scripts in Home

Search

+ Create Menu

Scripts in Files list + per-files controls



SCRIPT EDITOR

Toolbar Controls

Editor Canvas

```
1 script_requests
2 script_json
3 script_os
4
5 AUTH_URL = "https://api.sandbox.workiva.com/api/v1/oauth2/token"
6 SS_API_URL = "https://api.sandbox.workiva.com/api/forecasts/spreadsheets/"
7
8 # CLIENT_ID = os.getenv('CLIENT_ID')
9 # CLIENT_SECRET = os.getenv('CLIENT_SECRET')
10 # SPREADSHEET_ID = os.getenv('SPREADSHEET_ID')
11 # SHEET_ID = os.getenv('SHEET_ID')
12 # HELLO_WORLD = os.getenv('HELLO_WORLD')
13
14 # CLIENT_ID = "399a93cab10a4cb8941880599038a"
15 # CLIENT_SECRET = "18a6dbca786ca62c1f8b4dca9f9743a54412e740f7"
16
17 CLIENT_ID = "" # Enter your client ID here
18 CLIENT_SECRET = "" # Enter your client secret here
19
20 DOCUMENT_ID = os.getenv('DOCUMENT_ID')
21 INPUT_SHEET_ID = os.getenv('INPUT_SHEET_ID')
22 INPUT_RESOURCE_ID = os.getenv('INPUT_RESOURCE_ID')
23
24 def getSpreadsheet(suurl):
25     start = wurl.find("sheets_")
26     if (start == 0):
27         return wurl[start + len("sheets_")]
28     return None
29
30 def getSpreadsheetSection(sheetId, wurl):
31     start = wurl.find("sheets_" + sheetId + "_")
```

Sources Outline

Properties RHP

Outputs Bottom Panel



Create & Manage **Python** Scripts



Invoke from chains & Integrated Automations



Secure execution on the Workiva Platform

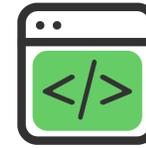
Running Scripts

Runner Credentials



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

Script Credentials



- [Generate Creds](#)
- Permissions to Workiva files

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

Run Parameters

CLIENT_ID	[REDACTED]	X
CLIENT_SECRET	[REDACTED]	X
SPREADSHEET_ID	352e1b500a549f9b55e	X
SHEET_ID	fd0ed568c12472f98d	X
HELLO_WORLD	Holaaaaa	X

+ Add Variable

Cancel Run Script

Script to Update Spreadsheets

```
1 import requests
2 import json
3 import os
4
5 CLIENT_ID = os.getenv('CLIENT_ID')
6 CLIENT_SECRET = os.getenv('CLIENT_SECRET')
7 SPREADSHEET_ID = os.getenv('SPREADSHEET_ID')
8 SHEET_ID = os.getenv('SHEET_ID')
9 HELLO_WORLD = os.getenv('HELLO_WORLD')
10
11
12 def main():
13     authToken = ApiAuth().getAuthToken()
14     ssApi = SSApi(authToken)
15     write_tier_data(ssApi)
16
17 print('Calling main')
18 main()
```

Properties

Name: Script to Update Spreadsheets

Description:

Last Run	10/11/2022 6:17:05 PM
Queuing Time	7 seconds
Run Time	4 seconds
Run Status	python-failed

History

- 10/11/2022 6:17:05 PM

Output

```
Calling main
Traceback (most recent call last):
  File "/usr/local/lib/python3.9/runpy.py", line 197, in _run_module_as_main
    return _run_code(code, main_globals, None,
```

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

The screenshot displays the Workiva interface with three main components:

- Scripting Editor:** A code editor showing a Python script for running scripts from Workiva files. The script includes functions for getting spreadsheet and sheet IDs, and logic for handling inputs from integrated automations.
- Automation Configuration Panel:** A panel on the right showing the configuration for an automation named "Running from Workiva Files". It is set to "Manual Execution" and has an action of "Execute script".
- Code Editor:** A separate window showing the following Python code:

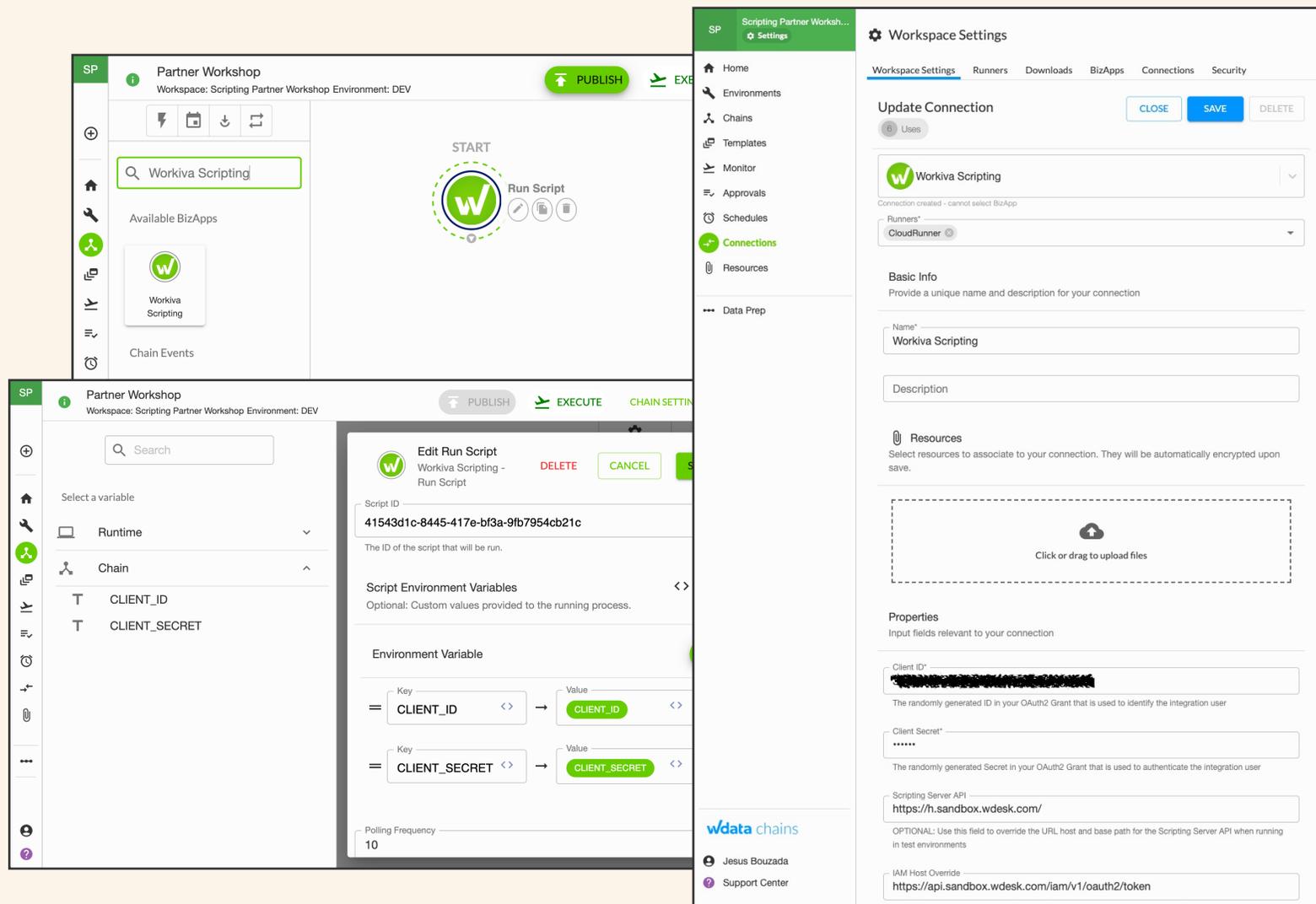
```
2 def getSpreadsheetId(wurl): ...
10
11 def getSheetId(spreadsheetId, wurl): ...
19
20 # Capture parameters (inputs) when you run the Script from the
21 # Scripting Editor, a Chain, or directly from the Scripting API
22
23 client_id = os.getenv("CLIENT_ID")
24 client_secret = os.getenv("CLIENT_SECRET")
25
26 trigger_document_id = os.getenv("DOC_ID")
27 inputs_spreadsheet_id = os.getenv("SPREADSHEET_ID")
28 inputs_sheet_id = os.getenv("SHEET_ID")
29
30 # Override inputs when you run the Script from an integrated automation
31
32 ia_inputs_spreadsheet_id = os.getenv("INPUT_SHEET_ID")
33
34 if ia_inputs_spreadsheet_id:
35     # Hardcode client id and client secret if triggered from Integrated Automation
36
37     client_id = 'YOUR_CLIENT_ID'
38     client_secret = 'YOUR_CLIENT_SECRET'
39
40     trigger_document_id = os.getenv("DOCUMENT_ID")
41
42     inputs_spreadsheet_id = getSpreadsheetId(ia_inputs_spreadsheet_id)
43
44     if document_id:
45         sheet_id = getSheetId(document_id, os.getenv("INPUT_RESOURCE_ID"))
46
```

Running Scripts from Chains

Chains Connector



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



The image displays three overlapping screenshots of the Workiva Scripting interface. The top screenshot shows the 'Partner Workshop' workspace with a 'Run Script' button and a search bar for 'Workiva Scripting'. The middle screenshot shows the 'Edit Run Script' dialog, where the script ID is '41543d1c-8445-417e-bf3a-9fb7954cb21c' and environment variables are defined: 'CLIENT_ID' and 'CLIENT_SECRET'. The bottom screenshot shows the 'Workspace Settings' for the 'Workiva Scripting' connection, including fields for Name, Description, Resources, Client ID, Client Secret, and Scripting Server API URL.

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

The screenshot displays three API endpoint panels:

- POST /v1/scripts/{scriptId}/runs**: Create a new Run for a Script. Description: "Creates a new run for a script, which will use all properties in the request object as environment variables." Parameters:

Name	Description
scriptId * required	The unique identifier of the script.

 Request body: application/json. Response: 201 Created. Example Value:

```
{ "apiKey": "Az896...", "spreadsheetId": "...", "row_number": 27, "has-changed": true }
```
- GET /v1/scripts/{scriptId}/runs/{runId}**: Retrieve the status of a Run. Description: "Retrieves a run given its ID." Parameters:

Name	Description
runId * required	The unique identifier of the run.
scriptId * required	The unique identifier of the script.
- GET /v1/scripts/{scriptId}/runs/{runId}/logs**: Retrieve a single link to download the log for a Run. Description: "Retrieve a single link to download the log for a Run given its ID." Parameters:

Name	Description
runId * required	The unique identifier of the run.
scriptId * required	The unique identifier of the script.

Scripting Demo



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

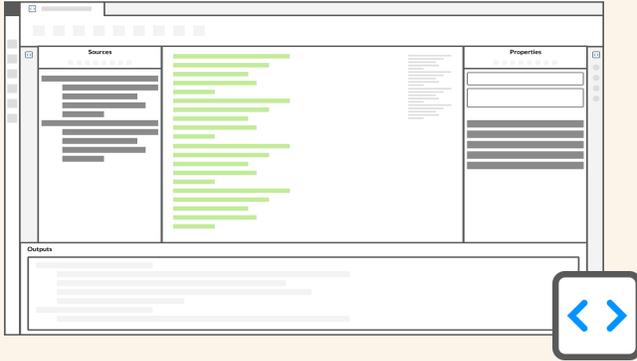


- Workiva Platform certification
- Complete the following introductory courses (Learning Hub)
 - Understanding Workiva Public APIs & Workiva Scripting
- Python skill level (beginner, intermediate, advanced, expert)
 - Intermediate required. Obtain the [PCAP](#) certification (Python Institute)
 - Advanced recommended ([PCPPI](#) 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

Deployment Scenarios

Build your own app
model

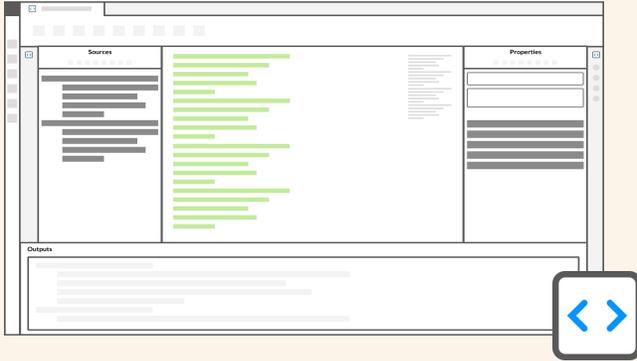


Distributed Hosting Customer Workspace

- 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

Deployment Scenarios

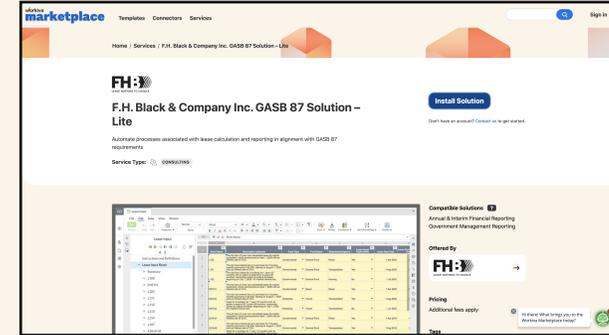
Build your own app model



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



1 Workiva Dev Platform

2 Workiva Public APIs

3 Workiva Scripting

4 Use Cases

5 Additional Resources

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

The screenshot shows a financial statement titled "Statement of Cash Flows" for "AQ 2021 IFRS entity FS". The table has columns for "Year ended December 31, 2020" and "Year ended December 31, 2021". Several rows are highlighted with red boxes, indicating empty data. These rows include "Inflation effect on cash and cash equivalents balances", "Other non-cash expense (income)", "Changes in operating assets and liabilities", "Provisions", and "Other liabilities".

	Year ended December 31, 2020	Year ended December 31, 2021
Cash and cash equivalents, beginning of year	36,807,182	801,187
OPERATING ACTIVITIES:		
Net loss	(21,234,689)	(34,830,929)
Adjustments to reconcile net loss to net cash from operating activities:		
Depreciation and amortization of property and equipment and other	457,582	4,654,507
Stock-based compensation	37,715,306	58,902,281
Finance income, finance costs and other expense (income), net	(92,713)	189,541
Inflation effect on cash and cash equivalents balances		
Other non-cash expense (income)		
Changes in operating assets and liabilities:		
Inventories		
Amounts due from / to related parties, net	(66,768,124)	184,246,704
Accounts receivable, net and other	(970,802)	(208,169,905)
Prepayments	296,761	(242,358)
Accounts payable	2,578,871	(105,428)
Accrued expenses and other	15,022,810	44,248,887
Provisions		
Unearned revenue	(34,813)	13,003,867
Other assets		(829,698)
Other liabilities		
Dividends received		
Cash received for interest	63,503	83,593
Cash paid for interest		(593)
Cash paid for income taxes	(2,684,240)	(5,370,247)
Net cash provided by (used in) operating activities	(35,650,552)	55,780,232



After

The screenshot shows the same financial statement as the 'Before' state, but with the empty rows suppressed. The table now only displays rows with data. The 'Net cash provided by (used in) operating activities' is now \$35,650,552 for 2020 and \$55,780,232 for 2021.

	Year ended December 31, 2020	Year ended December 31, 2021
Cash and cash equivalents, beginning of year	36,807,182	801,187
OPERATING ACTIVITIES:		
Net loss	(21,234,689)	(34,830,929)
Adjustments to reconcile net loss to net cash from operating activities:		
Depreciation and amortization of property and equipment and other	457,582	4,654,507
Stock-based compensation	37,715,306	58,902,281
Finance income, finance costs and other expense (income), net	(92,717)	189,541
Changes in operating assets and liabilities:		
Amounts due from / to related parties, net	(66,768,124)	184,246,704
Accounts receivable, net and other	(970,802)	(208,169,905)
Prepayments	296,761	(242,358)
Accounts payable	2,578,871	(105,428)
Accrued expenses and other	15,022,810	44,248,887
Unearned revenue	(34,813)	13,003,867
Other assets		(829,698)
Other liabilities		
Dividends received		
Cash received for interest	63,503	83,593
Cash paid for interest		(593)
Cash paid for income taxes	(2,684,240)	(5,370,247)
Net cash provided by (used in) operating activities	(35,650,552)	55,780,232

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

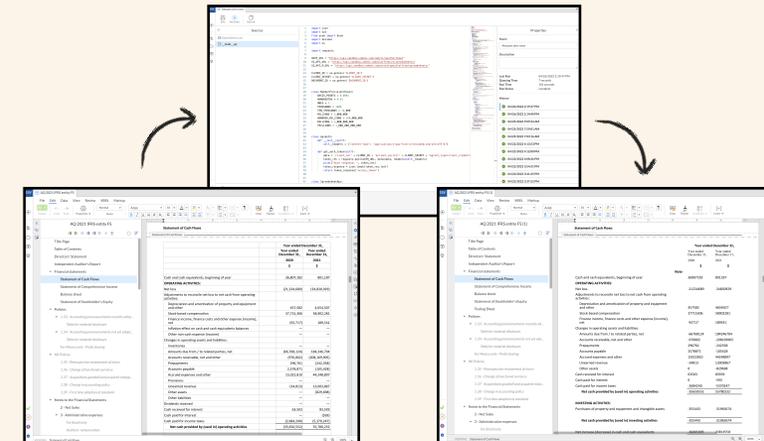
Process

- 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



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 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		CURRENT INCOME	CARRY	TOTAL DISTRIBUTION	<i>(Amount in \$)</i>				
		AFE GIVEBACK	GAIN / (LOSS)				RETURN OF CAPITAL	GAIN / (LOSS) **	CURRENT INCOME	CARRY	TOTAL DISTRIBUTION
Non-commercial asset	-	-	-	7,372.67	-	7,372.67	-	-	9,500.35	-	9,500.35
Non-commercial asset II	-	-	-	7,985.21	-	7,985.21	-	-	9,200.86	-	9,200.86
Commercial asset	45,432.65	5,925.29	35,732.16	-	(5,179.66)	81,910.44	49,787.33	55,987.12	-	(7,293.55)	98,480.9
TOTAL DISTRIBUTION	45,432.65	5,925.29	35,732.16	15,357.88	(5,179.66)	97,268.32	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 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			CURRENT INCOME	CARRY	WITHHOLDING	TOTAL DISTRIBUTION
		AFE GIVEBACK	RECOUPED LOSS	GAIN / (LOSS)				
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

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

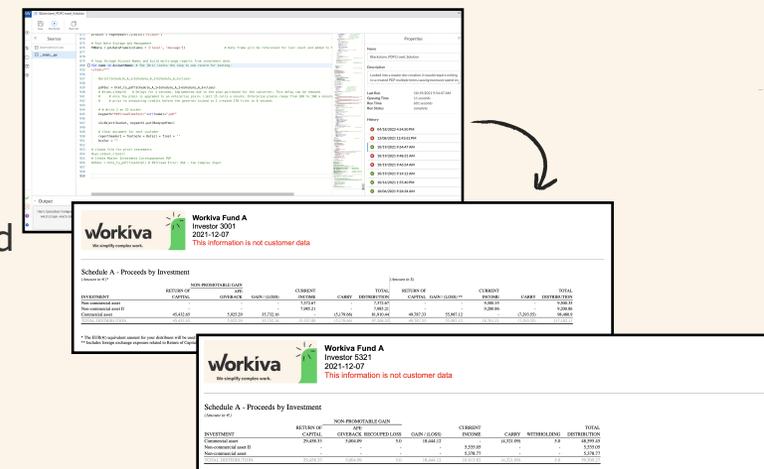
With Scripting & APIs

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

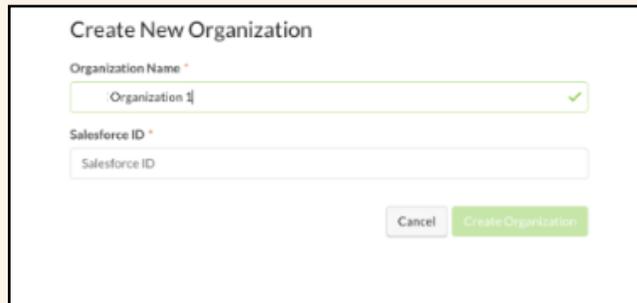
- Attract large financial institutions
- Removed hundreds of hours of manual work
- Portable



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.



Create New Organization

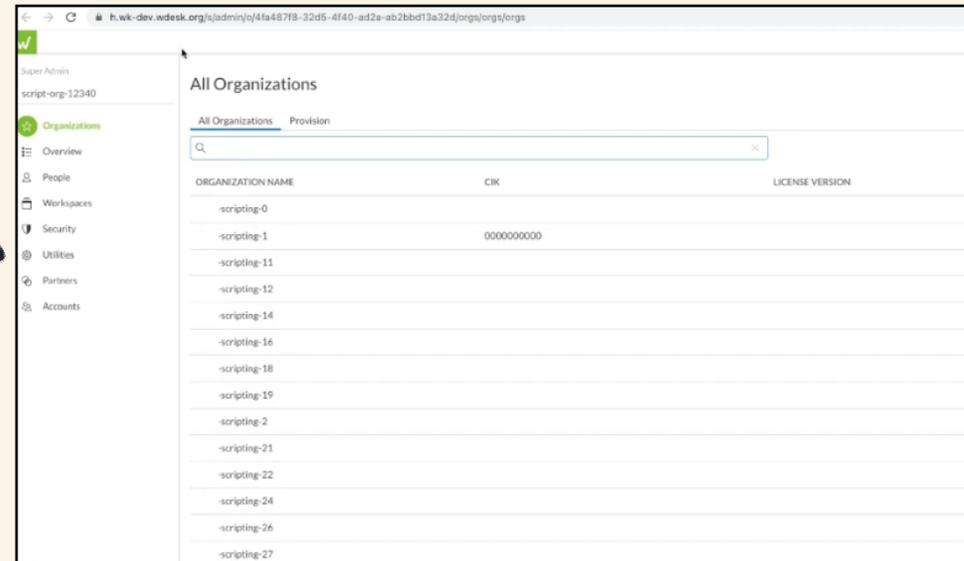
Organization Name *

Organization 1 ✓

Salesforce ID *

Salesforce ID

Cancel Create Organization



h.wk-dev.wdesk.org/s/admin/sj/4fa48718-32d5-4f40-ad2a-ab2bbd13a32d/orgs/orgs

Super Admin
script-org-12340

Organizations

All Organizations Provision

ORGANIZATION NAME CKC LICENSE VERSION

scripting-0		
scripting-1	000000000	
scripting-11		
scripting-12		
scripting-14		
scripting-16		
scripting-18		
scripting-19		
scripting-2		
scripting-21		
scripting-22		
scripting-24		
scripting-26		
scripting-27		

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

ORGANIZATION NAME	ORGANIZATION ID	ROLE
scripting 0		
scripting 1	00000000	
scripting 11		
scripting 12		
scripting 14		
scripting 16		
scripting 17		
scripting 2		
scripting 21		
scripting 22		
scripting 24		
scripting 26		
scripting 27		



1 Workiva Dev Platform

2 Workiva Public APIs

3 Workiva Scripting

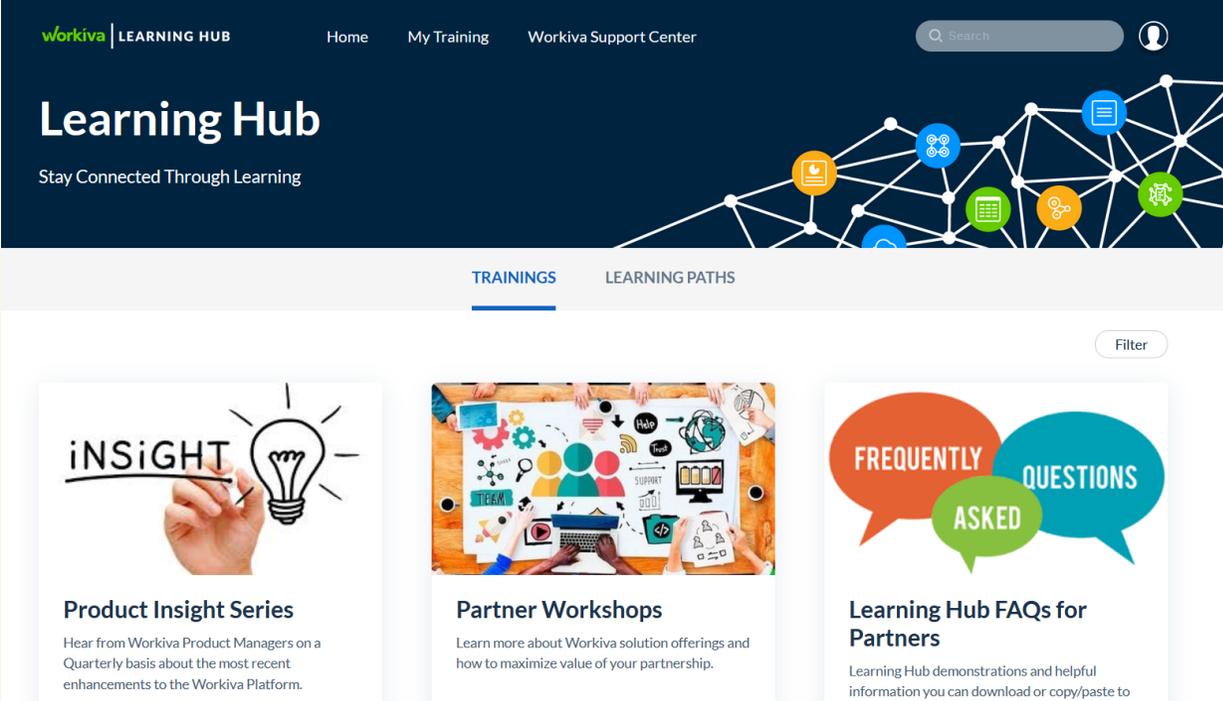
4 Use Cases

5 Additional Resources

Training Opportunities with Workiva

Learning Paths on Training tab

- Workiva Essentials
- Workiva Public APIs
- Workiva Scripting



The screenshot displays the Workiva Learning Hub interface. At the top, the navigation bar includes the Workiva logo, 'LEARNING HUB', and links for 'Home', 'My Training', and 'Workiva Support Center'. A search bar and a user profile icon are also present. The main heading is 'Learning Hub' with the tagline 'Stay Connected Through Learning'. Below this, there are tabs for 'TRAININGS' and 'LEARNING PATHS'. The 'LEARNING PATHS' tab is active, showing three featured learning paths:

- Product Insight Series**: Accompanied by an image of a hand drawing a lightbulb. Description: 'Hear from Workiva Product Managers on a Quarterly basis about the most recent enhancements to the Workiva Platform.'
- Partner Workshops**: Accompanied by an image of hands pointing to a whiteboard with various icons. Description: 'Learn more about Workiva solution offerings and how to maximize value of your partnership.'
- Learning Hub FAQs for Partners**: Accompanied by an image of three speech bubbles labeled 'FREQUENTLY', 'ASKED', and 'QUESTIONS'. Description: 'Learning Hub demonstrations and helpful information you can download or copy/paste to'

A 'Filter' button is located in the top right corner of the content area.

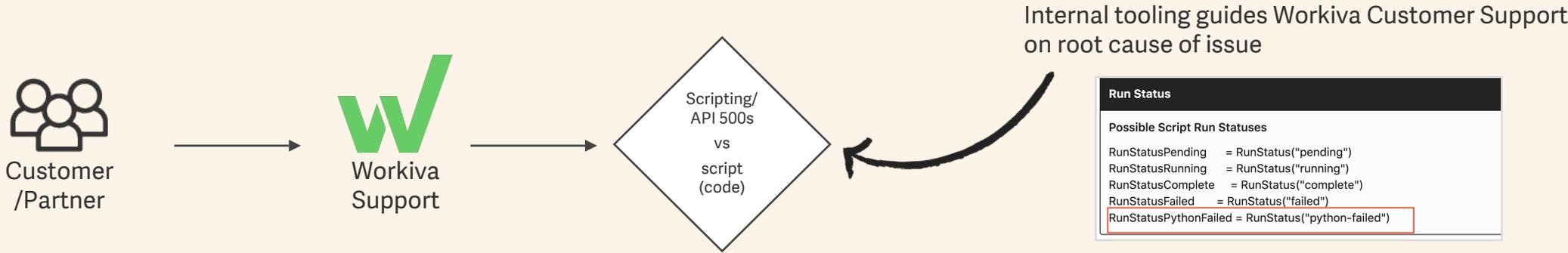
Product Support vs Code Support

(Workiva Public APIs and Workiva Scripting products)



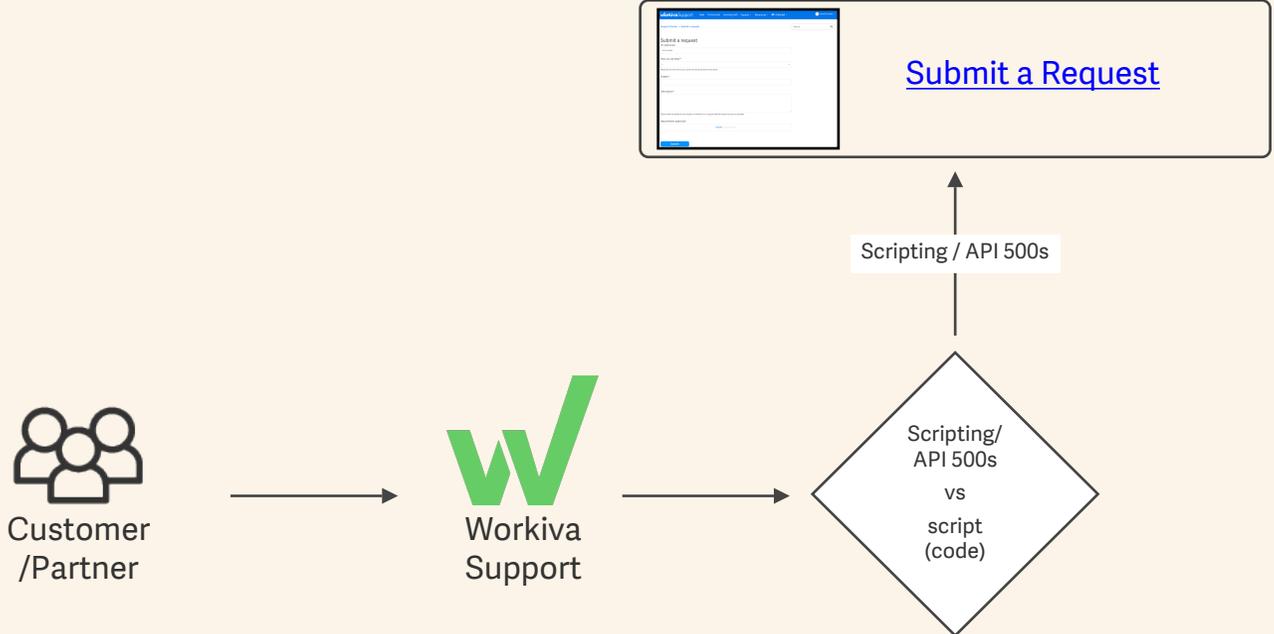
Product Support vs Code Support

(Workiva Public APIs and Workiva Scripting products)



Product Support vs Code Support

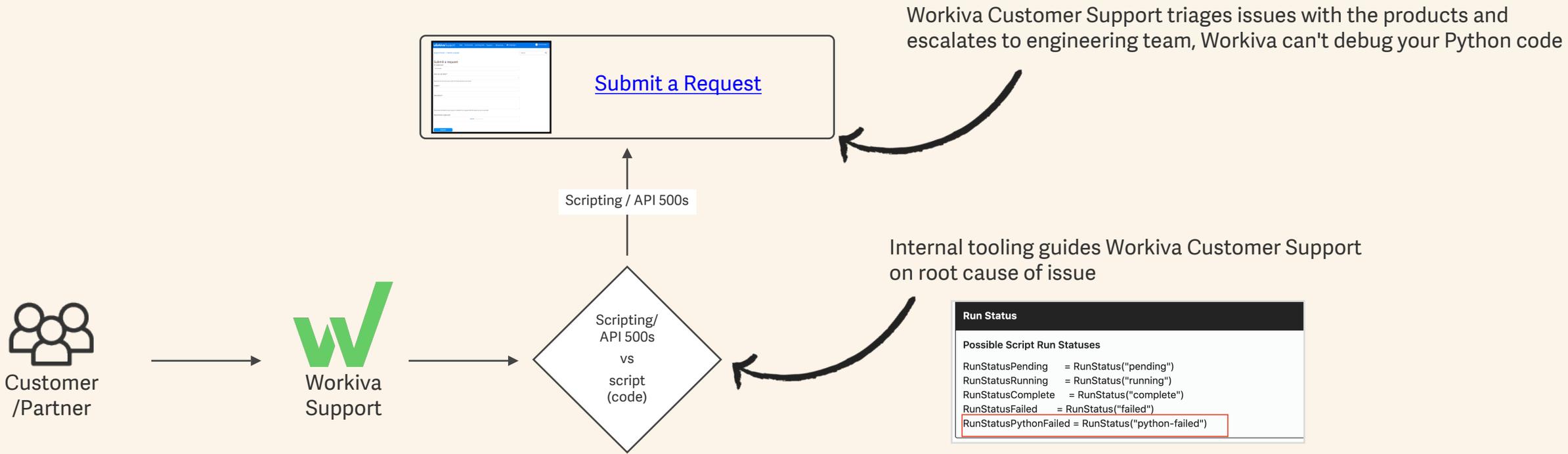
(Workiva Public APIs and Workiva Scripting products)



The screenshot shows the 'workivaSupport' website interface. The top navigation bar includes links for 'Help', 'Community', 'Learning Hub', 'Support', 'Resources', and 'Language', along with a user profile for 'Jesus Bouzada'. The main content area is titled 'Submit a request' and contains several input fields: 'CC (optional)' with an 'Add emails' button, a dropdown menu for 'How can we help?', a 'Subject' field, and a large 'Description' text area. Below these is an 'Attachments (optional)' section with an 'Add file or drop files here' button. A blue 'Submit' button is located at the bottom of the form.

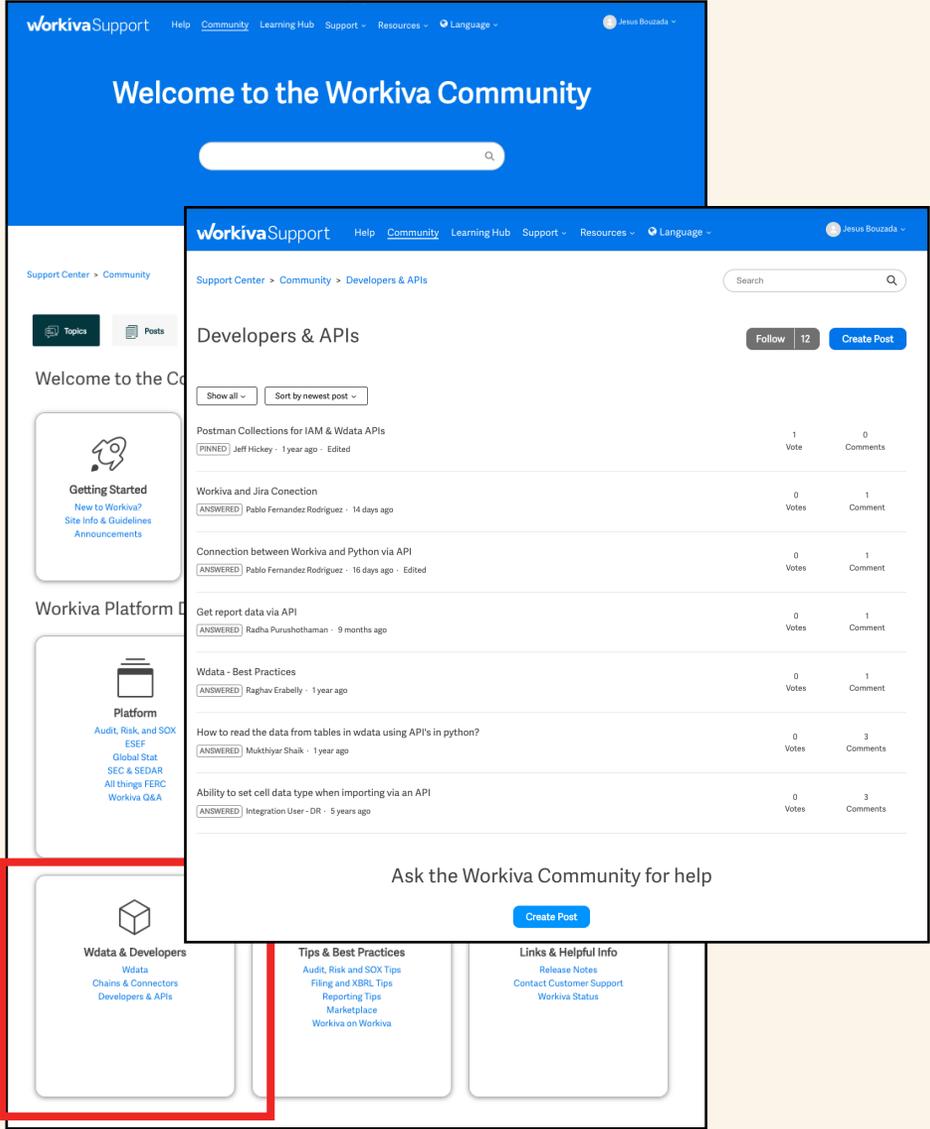
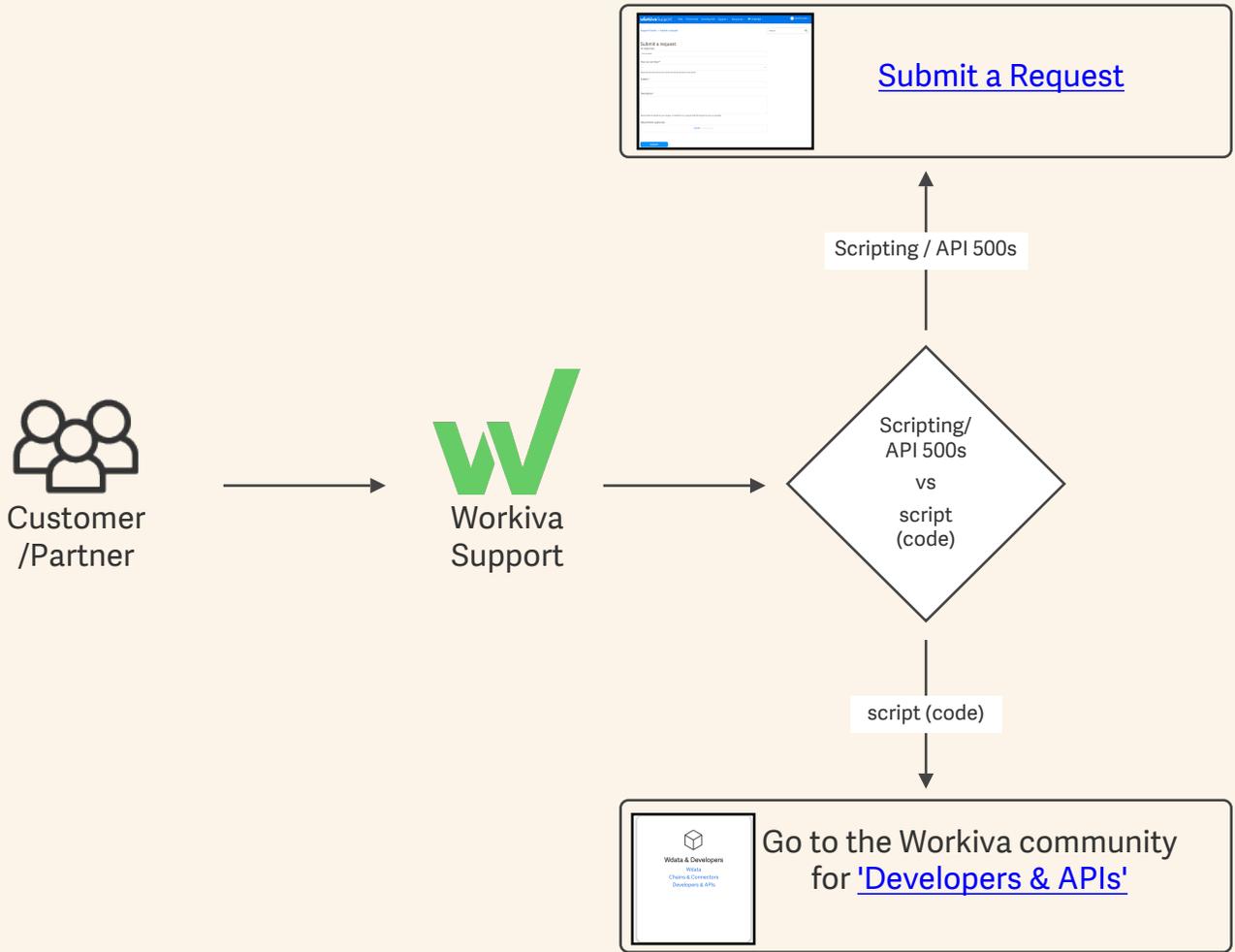
Product Support vs Code Support

(Workiva Public APIs and Workiva Scripting products)



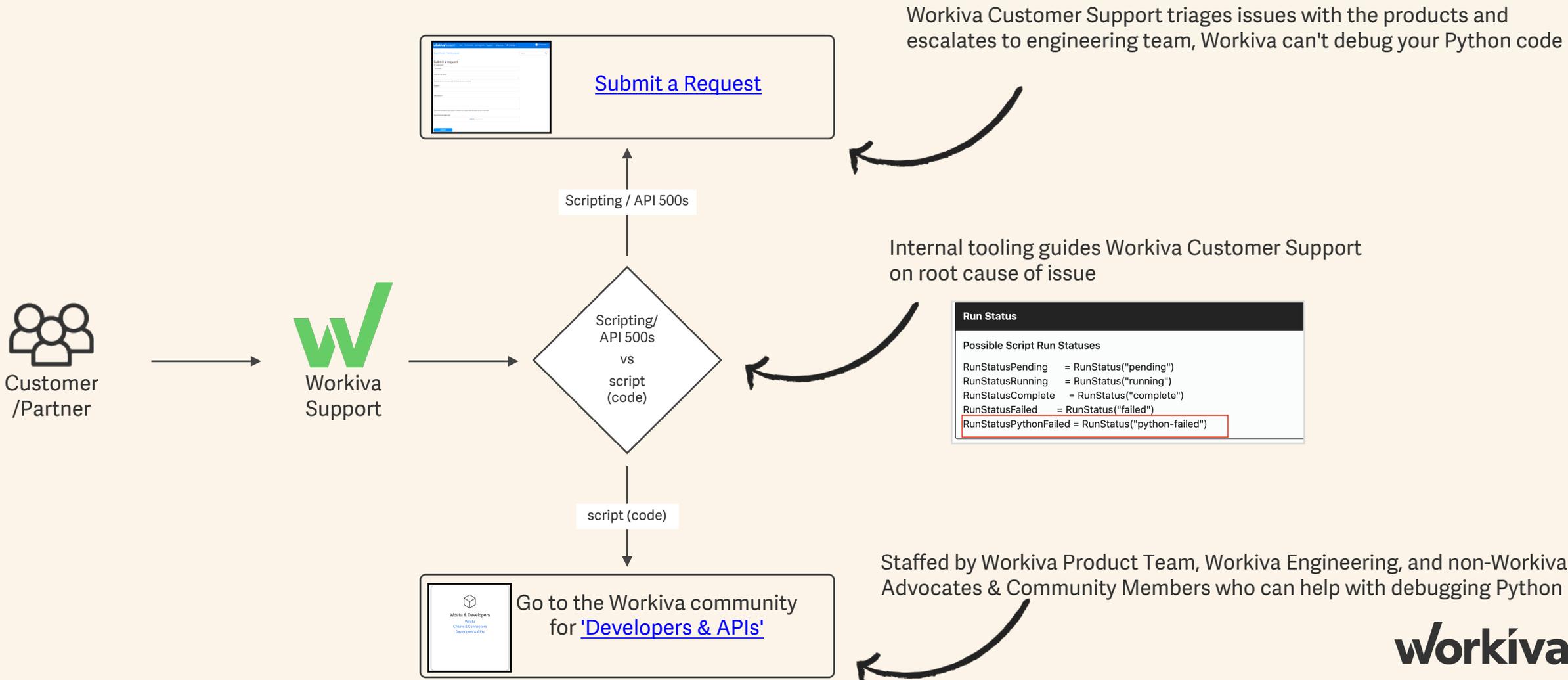
Product Support vs Code Support

(Workiva Public APIs and Workiva Scripting products)



Product Support vs Code Support

(Workiva Public APIs and Workiva Scripting products)



Run Status	
Possible Script Run Statuses	
RunStatusPending	= RunStatus("pending")
RunStatusRunning	= RunStatus("running")
RunStatusComplete	= RunStatus("complete")
RunStatusFailed	= RunStatus("failed")
RunStatusPythonFailed	= RunStatus("python-failed")



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](#)



Thank you!