Outpost24 REST API Examples

# Purpose:

This document is intended to guide a user of Outpost24 services to do the fundamentals for accessing their data through the Outpost24 REST API.

# Who This Is For:

Users with beginner to intermediate scripting skills, experience with Linux Bash and/or Powershell, or for obtaining an API key for use in third-party integration services.

# What This Is Not:

This guide is not intended to substitute for or act as full documentation of the Outpost24 REST API. Please visit [Outpost24 API](https://outscan.outpost24.com/opi/rest/openapi.json) for full documentation.

# Gotcha’s:

- Before interacting with the Outpost24 API, ensure you are setting yourself up to use TLS1.2 or greater.

- “API key” and “token” refer to the same thing and are interchangeable.

- Submitted commands must use URL encoding, which includes both passwords and date values.

* Be sure all characters are URL-encoded, or it will fail. To ensure this, check if all special characters like ‘;’ or ‘~’ have converted into new encoding such as ‘%7’
* Lots of tools exist online to convert these strings

- When sending commands, always ensure that you are using ' instead of ‘ to wrap strings.

- Text highlighted in grey are to be edited by the user.

- Hostname used in example commands is demo, but production/end users will want to use outscan.

Sandbox/Trial: demo.outpost24.com

Production: outscan.outpost24.com

# Questions?

If you have any questions about the content in this document or would like to report an error, please email the author at [eci@outpost24.com](mailto:eci@outpost24.com)

# curl/wget/invoke-webrequest in Powershell

#### Initialization:

How to use TLS 1.2 in Powershell:

[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12

#### Authentication:

Basic logic:

$response = curl -uri 'https://demo.outpost24.com/opi/rest/auth/login' -Method post -Body 'username=your\_username&password=your\_url\_encoded\_password'

How to obtain token/API Key with specified expiration:

$response = curl -uri 'https://demo.outpost24.com/opi/rest/auth/login' -Method post -Body 'username=your\_username&password=your\_url\_encoded\_password&validUntil=YYYY-MM-DDTHH%3AMM%3ASSZ'

* Date values are interpreted as GMT-0000
  + 2019/08/31 at 11:59:59pm in **New York City (GMT-0400)**

2019-09-01T03%3A59%3A59Z

* + 2019/08/31 at 11:59:59pm in **Chicago (GMT-0500)**

2019-09-01T04%3A59%3A59Z

* + 2019/08/31 at 11:59:59pm in **Denver (GMT-0600)**

2019-09-01T05%3A59%3A59Z

* + 2019/08/31 at 11:59:59pm in **San Francisco (GMT-0700)**

2019-09-01T06%3A59%3A59Z

* + 2019/08/31 at 11:59:59pm in **London (GMT+0100)**

2019-08-31T22%3A59%3A59Z

* + 2019/08/31 at 11:59:59pm in **Copenhagen (GMT+0200)**

2019-08-31T21%3A59%3A59Z

Save token to variable:

$token = $response.Content

How to invalidate/expire an existing token:

curl -uri 'https://demo.outpost24.com/opi/rest/auth/logout’ -Method post -ContentType application/json -Header @{Authorization = "Bearer "+ $token }

#### Obtain Data:

Using token/API Key, request Outscan findings:

curl -uri 'https://demo.outpost24.com/opi/rest/findings' -Header @{Authorization = "Bearer "+ $token }

Using token/API Key, request SWAT findings:

curl -uri 'https://demo.outpost24.com/opi/rest/webfindings' -Header @{Authorization = "Bearer "+ $token }

Findings with a specified filter:

curl -uri 'https://demo.outpost24.com/opi/rest/findings?limit=50&offset=0&fields=id%2Caccep%2CcvssScore%2Cname%2Chostname%2Cport%2CfirstSeen%2ClastSeen%2Csource&filter=%5B%7B%22field%22%3A%22name%22%2C%22value%22%3A%22cookie%22%2C%22comparison%22%3A%22eq%22%7D%5D&sort=-source' -Header @{Authorization = "Bearer "+ $token }

# curl in Linux

#### Initialization:

How to use TLS 1.2 in Bash:

Update OpenSSL if unsupported.

#### Authentication:

Basic logic:

curl -k 'https://demo.outpost24.com/opi/rest/auth/login' --data 'username=your\_username&password=your\_url\_encoded\_password'

How to obtain token/API Key with specified expiration:

curl -X POST \

https://demo.outpost24.com/opi/rest/auth/login \

-d 'username=your\_username' \

-d 'password=your\_password' \

-d 'validUntil=yyyy-mm-ddTHH%3AMM%3ASSZ'

* Date values are interpreted as GMT-0000
  + 2019/08/31 at 11:59:59pm in **New York City (GMT-0400)**

2019-09-01T03%3A59%3A59Z

* + 2019/08/31 at 11:59:59pm in **Chicago (GMT-0500)**

2019-09-01T04%3A59%3A59Z

* + 2019/08/31 at 11:59:59pm in **Denver (GMT-0600)**

2019-09-01T05%3A59%3A59Z

* + 2019/08/31 at 11:59:59pm in **San Francisco (GMT-0700)**

2019-09-01T06%3A59%3A59Z

* + 2019/08/31 at 11:59:59pm in **London (GMT+0100)**

2019-08-31T22%3A59%3A59Z

* + 2019/08/31 at 11:59:59pm in **Copenhagen (GMT+0200)**

2019-08-31T21%3A59%3A59Z

How to invalidate/expire an existing token:

curl 'https://demo.outpost24.com/opi/rest/auth/logout' -H 'authorization: Bearer <token>' -H 'Content-Type:application/json' -X POST

#### Obtain Data:

Using token/API Key, request Outscan findings:

curl 'https://demo.outpost24.com/opi/rest/findings' -H 'authorization: Bearer <token>'

Using token/API Key, request SWAT findings:

curl 'https://demo.outpost24.com/opi/rest/webfindings' -H 'authorization: Bearer <token>'

Findings with a specified filter:

curl -k 'https://demo.outpost24.com/opi/rest/findings?limit=50&offset=0&fields=id%2Caccep%2CcvssScore%2Cname%2Chostname%2Cport%2CfirstSeen%2ClastSeen%2Csource&filter=%5B%7B%22field%22%3A%22name%22%2C%22value%22%3A%22cookie%22%2C%22comparison%22%3A%22eq%22%7D%5D&sort=-source' -H 'authorization: Bearer <token>'

# Domo Integration

Outpost24 supports Domo, a business intelligence tool used by our customers and companies around the world. Domo provides software to connect to services like Outpost24’s full stack security suite and use these as sources to provide detailed information for reporting purposes.

## How to Begin

Before we connect to Domo, we will need an API key to be generated. This key will need to be configured with a specified duration which will depend on your tolerances, so let’s just say 1 year to start with. Copy and paste this script into Windows Powershell ISE, updating values as appropriate:

[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12

$response = curl -uri 'https://demo.outpost24.com/opi/rest/auth/login' -Method post -Body 'username=your\_username&password=your\_url\_encoded\_password&validUntil=2020-07-29T04%3A59%3A59Z'

$response.Content

This code snippet will authenticate with your user account and return a long text string. The response variable stores this within its $\_.content, which you might have noticed is saved into $token in the original examples.

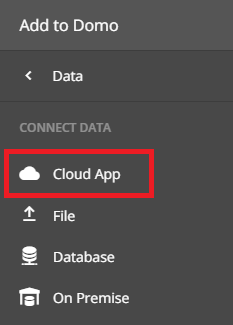
**Be sure to save this token somewhere as this will be the only time it is visible**. If you need to invalidate the token for any reason, please see above for Powershell/Bash logic to log out with the token.

## How to Connect

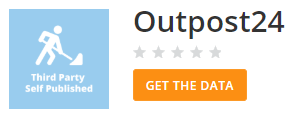
Within Domo, select the “Add to Domo” button that looks like this



Select “Data” and find the “Cloud App” option.



Find the Outpost24 connector labeled “Third Party Published”. This will change to show the Outpost24 logo after Domo has approved the connector.



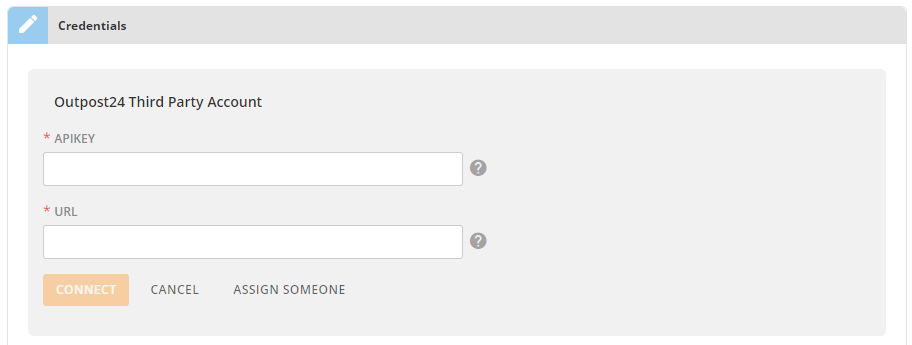
From here you need to configure the connection. We require two items from the user:

The API key generated earlier

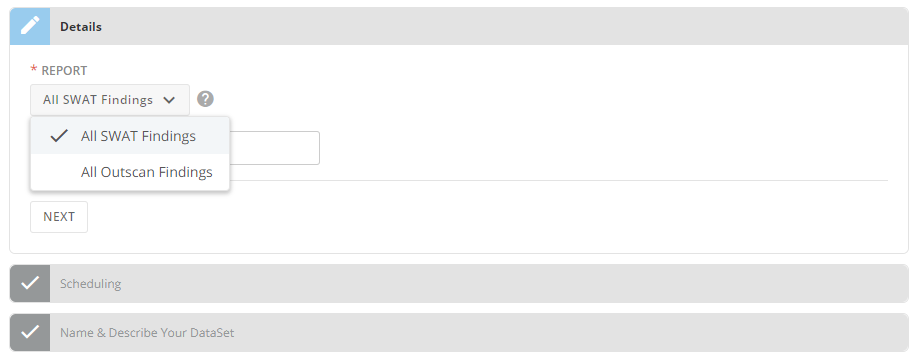
The URL to connect to. **By default, the URL will be outscan.outpost24.com**

Please note URL in this case is known also as Hostname. Do not include https://...

This option allows the user to specify an internal HIAB (Hacker-in-a-Box) system as a similar data source, or in case they are demoing Outpost24 it can also point to demo.outpost24.com.

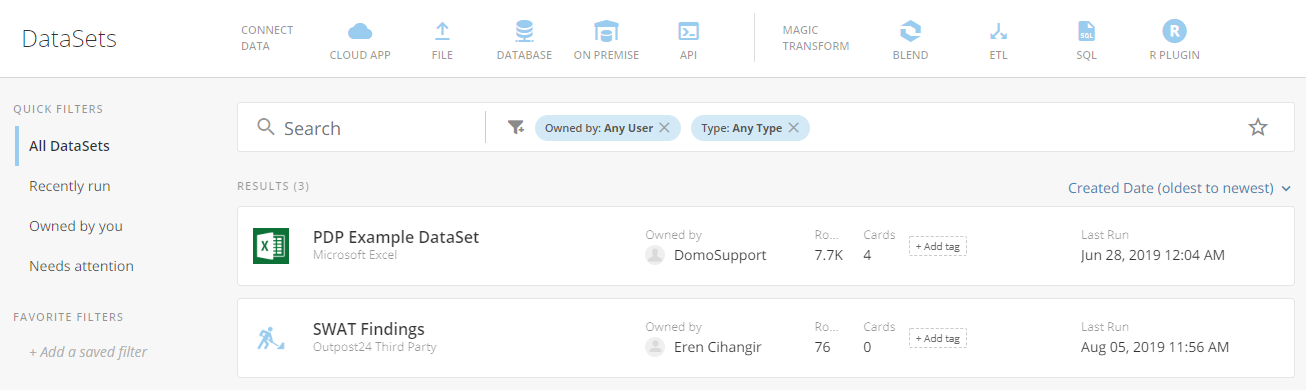


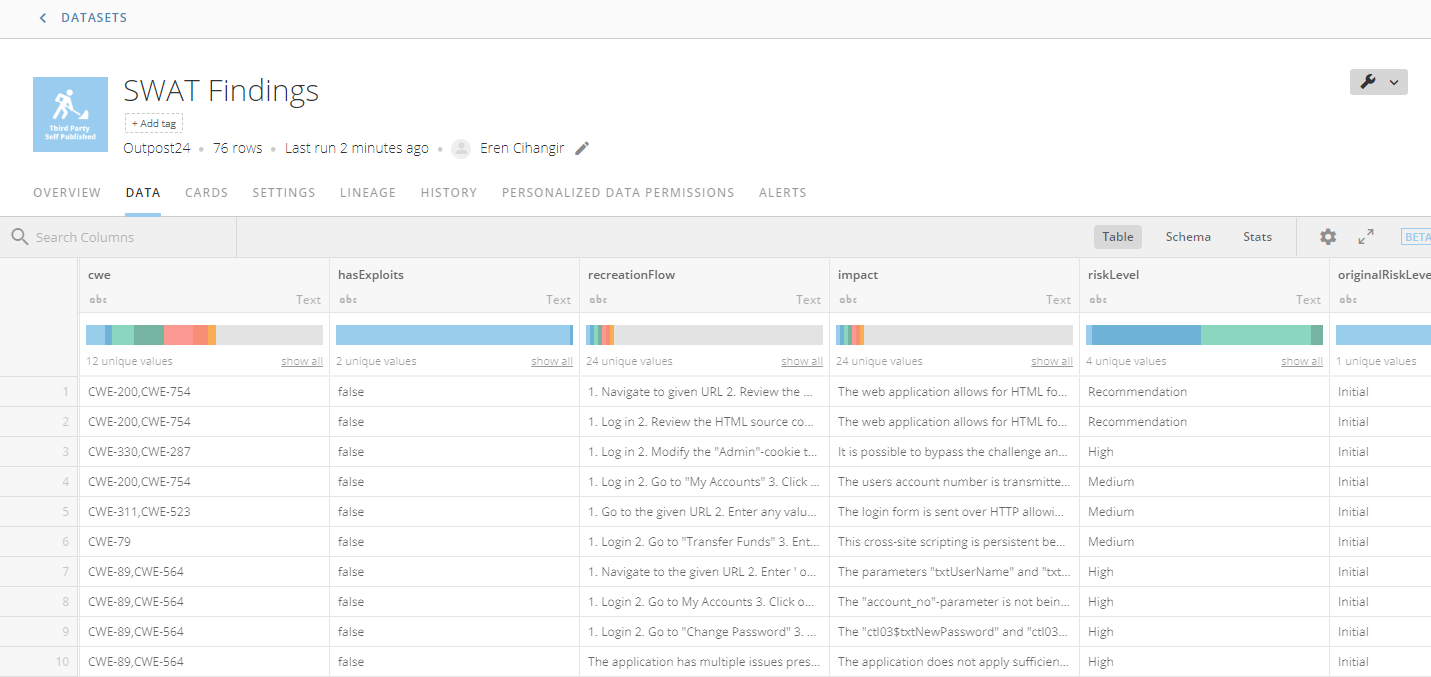
Once configured, select the report to use. From here scheduling and report name are also configured.



## How to View Data

After connecting your data source and establishing the update schedule, your Data Dashboard will show the Outpost24 data with the label you gave it.



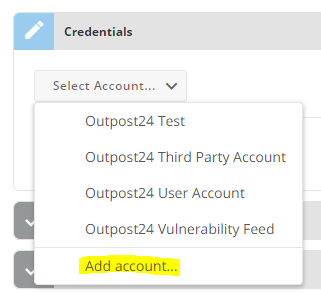


## Known Issues

Domo requires submitted apps to go through an approval process that can take up to 30 days. During this time the code cannot be updated or changed, but after approval will be given version updates as usual for software.

Unfortunately, this means that a few bugs will need to be ironed out after this document’s creation and the approval by Domo. So far, the connector works as intended, but some error conditions are not appropriately accounted for. In addition, the label of “URL” should be changed to “Hostname” or should support strings including https://...

For the time being, use the “Add Account” function when setting up the connector if a previously configured account returns incorrect rows/data. Follow the guide above to double-check that the API key is valid and ensure the URL is **outscan.outpost24.com.**



Please reach out to Eren Cihangir at [eci@outpost24.com](mailto:eci@outpost24.com) if you have any questions or run into trouble!