tripwire®

TRIPWIRE®
INDUSTRIAL
VISIBILITY

TRIPWIRE INDUSTRIAL VISIBILITY 4.2.4
WEB API USER GUIDE

FOUNDATIONAL CONTROLS FOR
SECURITY, COMPLIANCE & IT OPERATIONS

Web API User Guide Versions

Version	Date	Owner	Author	Revisions
Rev 05	December 2020	Daniel Ashual	Beth Stolper	Clarified the Assets API
Rev 04	November 2020	Daniel Ashual	Beth Stolper	Screens updated for v4.2.1. Insights APIs added
Rev 03	July 2020	Daniel Ashual	Beth Stolper	Included general statement re permissions for methods. Added description of Models. Removed list and description of Methods. Revised Alerts example
Rev 02	July 2020	Daniel Ashual	Beth Stolper	Added Note re user/admin usage
Rev 01	June 2020	Daniel Ashual	Beth Stolper	Initial Release

Contents

1 Introduction

This document provides instructions for using the API Explorer for version 4.2.4 of Tripwire Industrial Visibility (TIV).

This web API reference provides all the information needed for application developers to implement calls for TIV's endpoints. Use these RESTful APIs to access Tripwire routes for enhancing your system.

2 Accessing the API Explorer

Log in to TIV with your username and password.

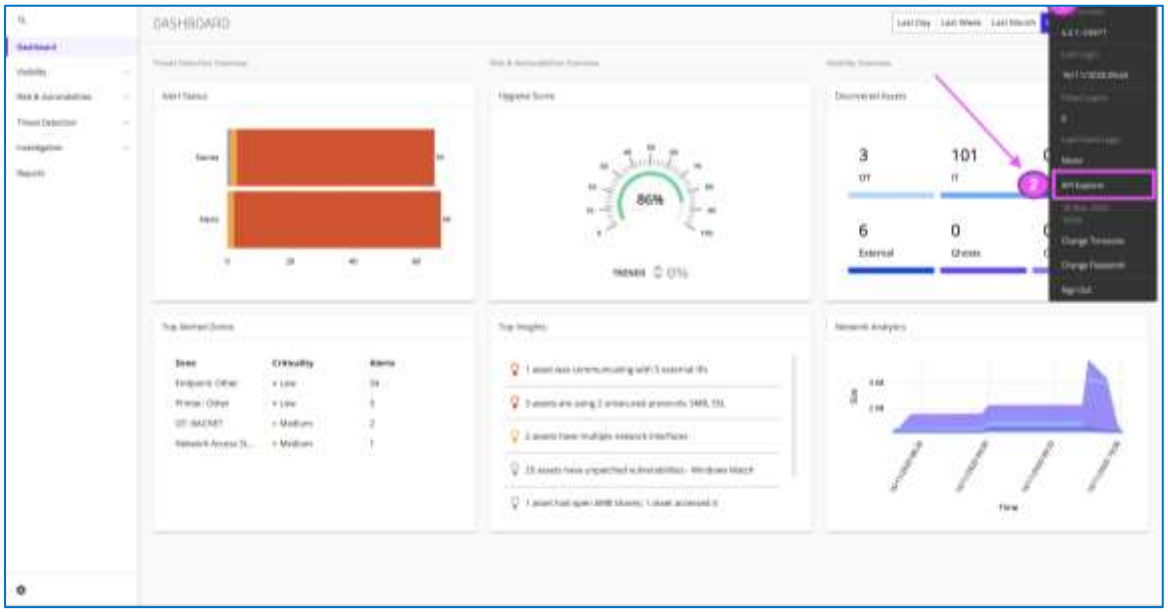


Figure 1 API Explorer

1. On the far right edge of the Activity bar at the top of the screen, click on your username to open the user dialog
2. Select the **API Explorer** button.
3. The API Explorer opens in *Swagger* as shown in Figure 2 below.

3 Using the API Explorer

The API Explorer opens in a separate tab as follows:

Figure 2 API Explorer: Opening page and Methods

Figure 3 API Models

4 Authentication

Most of the Tripwire routes are token-based. You need to have the token and be logged into the system to use these calls.

Before using these APIs, authenticate yourself as a user, using a JSON web token, as described below.

4.1 Authentication steps

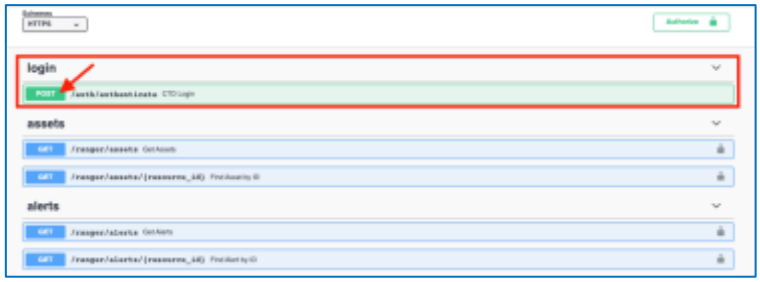


Figure 4 Authentication

1. Click on the `POST /auth/authenticate route`, and then click the **Try it out** button
2. The **Set License** parameters appear. Input your username and password credentials, and then click the **Execute** button.
3. The **Response Body** appears with the result. Select and copy to your clipboard the entire token string (between the quote marks), as shown in the example below:

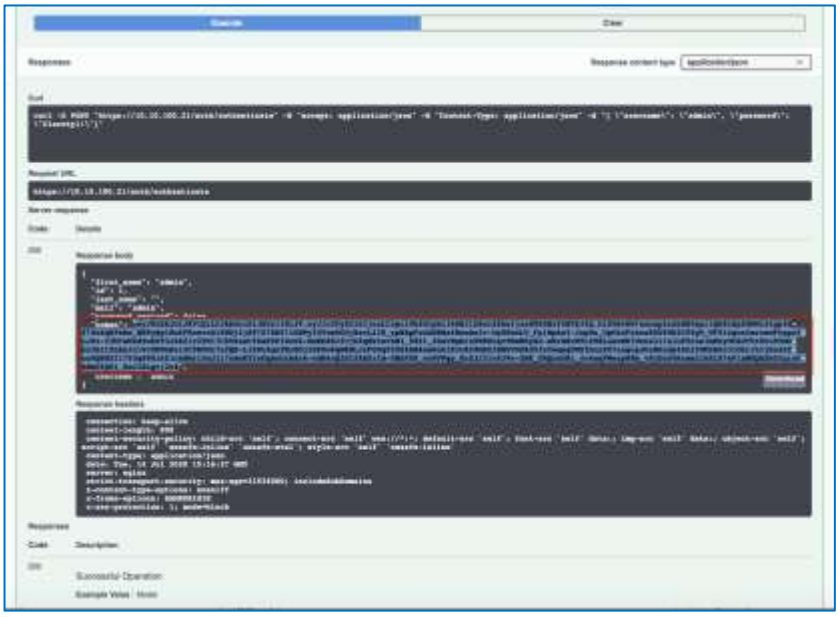


Figure 5 Copying the token from the response body

4. On the top right corner of the main API page, click the green **Authorize** button
5. The Available Authorization pops up.

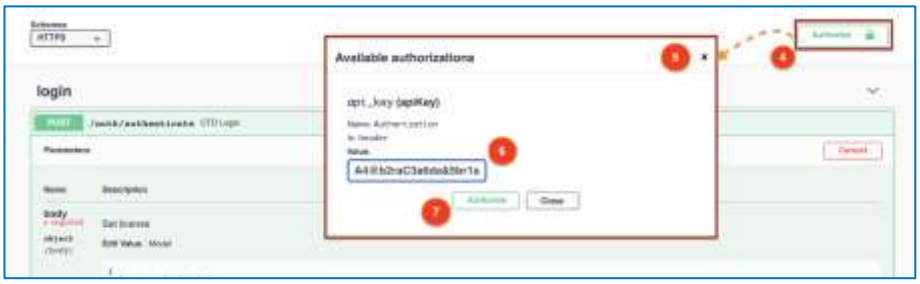


Figure 6 Authorization popup

6. Enter your token in the **Value** window
7. Click Authorize
8. Now that you have authenticated, you can use the calls.

Each function call that is available is indicated by a black lock icon.

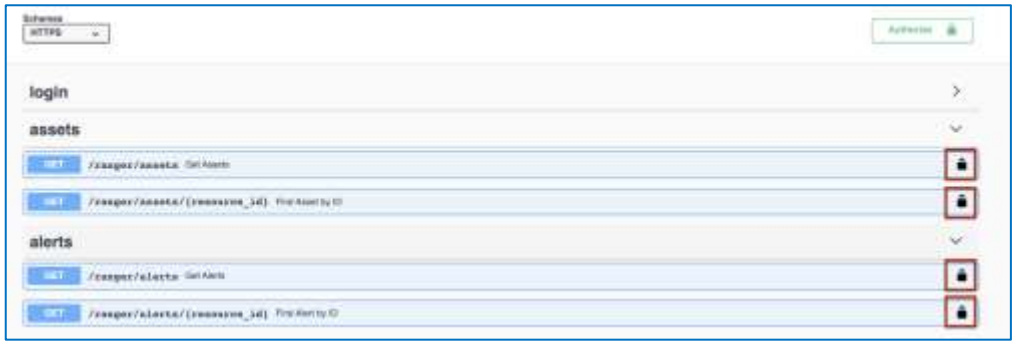


Figure 7 Black lock icons appear once the User is authenticated

5 Overview

5.1 Permissions

Admins can access all the routes. Other users can obtain all the information they can access in the TIV UI.

5.2 Separators

`,;$` is used for separating multiple values when filtering.

For example:

```
/assets?asset_type__exact=1,;$2,;$4
```

Will search for all assets but return just the assets of type 1, 2, 4.

```
/ranger/assets?fields=vendor,;$id
```

Returns just the vendor and ID fields.

5.3 Models

The Response Body is usually a JSON document, and the structure of that JSON document is defined in the **Models** provided in the API Explorer, immediately below the area of the function calls:

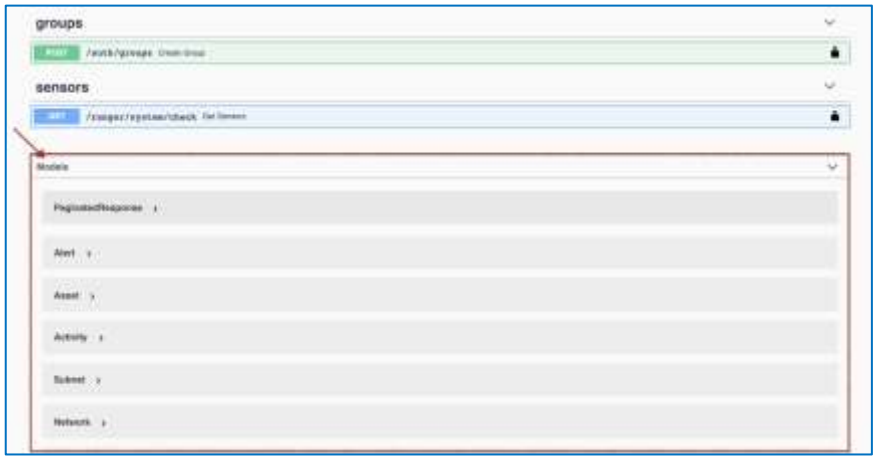


Figure 8 Models

6 Example: Get Asset Type

The following example shows how to retrieve an exact match:

Figure 9 Get Assets method

1. Select the method to use (e.g. **Get Assets**)
2. Click **Try it Out** to make the method available
3. Select the filter you want. You can search for the filter, e.g:
   - `asset_type_exact`
4. Enter the filter you want using the separator, e.g.
   - **1,;$2,;$4,;$5**

Figure 10 Specifying an Asset Type - Exact Match

5. Press **Execute**
6. The system responds with:
   - the resulting CURL statement
   - the Request URL
   - the Server Response Body, which you can copy or download as a json file or copy

7 Example: Get Alerts

The following example shows how to retrieve a specific set of alerts:

Figure 11 Get Alerts method

1. Open the **GET Alerts** method
2. Click **Try it Out**
3. Enter the filter/s you want, using the separators where needed, e.g.:

For Site ID = 1

For Severity Levels of 2 and 3

With Exact Resolved = false

With Qualified Alert = true

4. This selection runs the following command:

```
ranger/alerts?fields=
&format=alert_list
&sort=-score
&page=1
&per_page=20
&resolved__exact=false
&is_qualified__exact=true
&severity__exact=2,;$3
&site_id__exact=1
```

5. Press the **Execute** button at the bottom of the method
6. The system responds with:

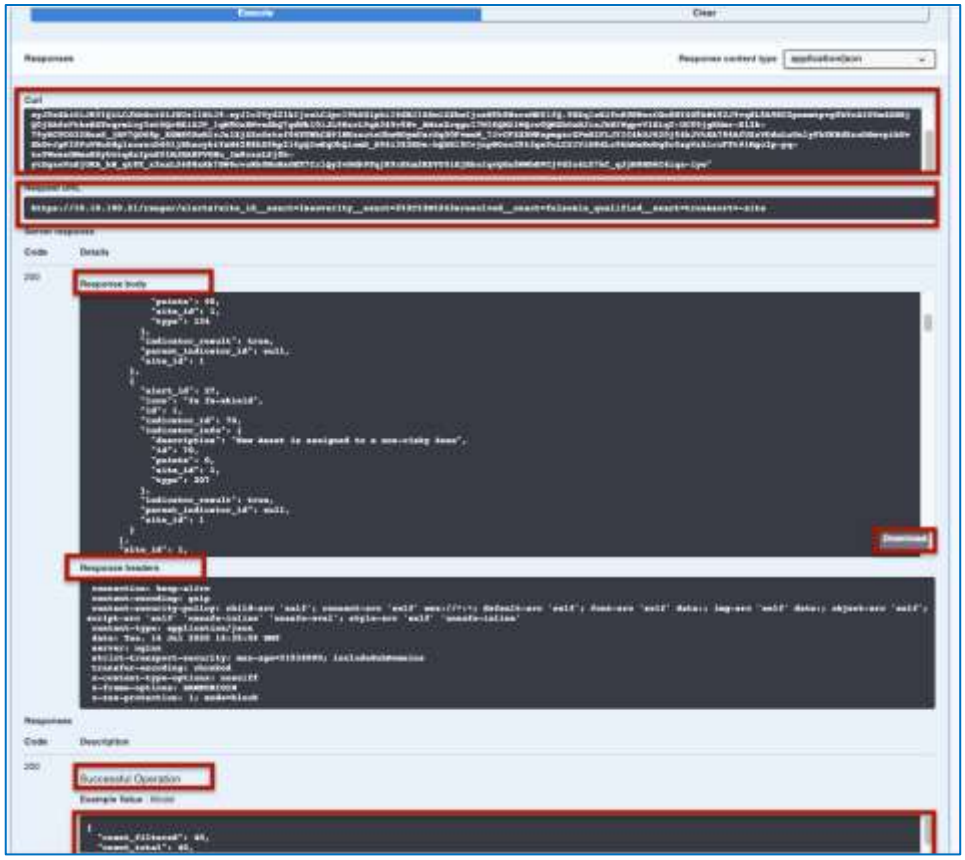


Figure 12 Get Alerts - Response

- The resulting CURL statement
- The Request URL
- The Server Response Body, which you can copy or download as a JSON file
- The Server Response Header
- The Code and Description, e.g. 200 Successful Operation
- The Example Value / Model

8 Example: Retrieve the Assets of an Insight

The following example shows how use the following sequence of functions to retrieve all the assets associated with a specific Insight, using the following sequence of methods:

- **GET /ranger/insights_summary**
- **GET /ranger/insight_details-{insight_name}**
- **GET /ranger/assets**

8.1 GET /ranger/insights_summary

1. Open the **GET /ranger/insights_summary** method
2. Click **Try it Out**
3. Apply the relevant site number in the field **site_id__exact**
4. Click **Execute**
5. From the Response body, select the insight of interest
   - e.g. `Assets Accessed SMB shares`
   - Copy the string of the **insight name**

8.2 GET /ranger/insight_details-{insight_name}

1. Open the **GET /ranger/insight_details-{insight_name}** method
2. Click **Try it Out**
3. Enter (paste) the insight name from above into the **insight name** field
4. Apply the relevant site number in the **site_id__exact** field
5. Click **Execute**
6. From the Response Body, select the **filter_key**
7. Copy the string of the **filter_key**

8.3 GET /ranger/assets

1. Open the **GET /ranger/assets** method
2. Click **Try it Out**
3. Enter (paste) the **filter_key** from above into the **INSIGHT_ROW_KEY_EXACT** field
4. Apply the relevant site number in the **site_id__exact** field
5. Enter **VALID_TRUE** to the **valid_exact** field in order to filter the assets accordingly

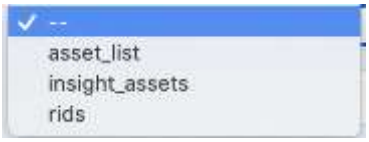
6. Enter **APPROVED_TRUE** to the **approved_exact** field in order to filter the assets accordingly
7. Enter other relevant filters as needed
8. Click **Execute**
    - The requested assets for this Insight appear in the Response Body

Figure 13 Get Assets for an Insight - Response

- The resulting CURL statement
- The Request URL
- The Server Response Body, which you can copy or download as a JSON file
- The Server Response Header
- The Code and Description, e.g. 200 Successful Operation
- The Example Value / Model

8.4 Asset Filter Information Fields

The asset filters are shown below.

Code	Means
format	Specifies the type of response (list of fields) returned by the query. If not provided, all asset properties will return and could affect performance
	Responsible for which columns will be returned by using the asset's API call: • asset_list - all columns, as if you are in the assets page • insight_assets - the set of columns that relate to insights and assets • rids - only the RIDs are returned An rid is a combination of the resource (asset) ID and the site ID
page	The actual page offset.
per_page	The number of assets to pull on each page (the maximum is 500)
ipv4__exact	The IPV4 protocol
ipv6__exact	The IPV6 protocol
mac__icontains	The MAC address of the device (free text)
vlan__exact	The VLAN/s of the device
address__exact	The IP address of this device in the network
gateway__exact	The gateway of this device in the network
asset_type__exact	The asset type for this asset
host_name__exact	The host name for this asset
os__exact	The name of the Operating System
model__icontains	The model of this asset (free text)
vendor__icontains	The vendor of this asset (free text)
state__exact	The dropdown of the asset states: 0 = Training 1 = Maintenance 2 = Guest
domain_names__exact	The exact domain name(s) of the asset
firmware__exact	The exact firmware name of the asset
serial__exact	The exact serial number for the asset

Code	Means
generic__icontains	Custom information (free text)
display_name__icontains	The name of the asset (free text)
criticality__exact	Low, Medium, or High. These TIV values represent how critical the asset itself is to the operation. TIV assigns criticality automatically to certain types of assets but enables you to edit the value
old_ip__exact	The former IPs this asset has used
protocol__exact	The list of protocols in which this asset communicates
last_seen__exact	The Timestamp of the last instance when this device was seen in the network
q__icontains	Free text for the following asset info: name, IP, version, Model and MAC fields
alert_id__exact	The IDs of the alerts triggered by this asset
baseline__exact	The baseline for this asset
arp_baselines__exact	Whether this asset has ARP baselines True = ARP
insight_status__exact integer	The status of the insight associated with this asset - this is a dropdown for choosing the integer type: 0 = Open 1 = Hidden 2 = Completed
insight_insight_name__exact	The name of the insight associated with this asset
insight_timestamp__gte	When this insight was detected; greater than or equal to a specific timestamp
insight_timestamp__lte	When this insight was detected; lower than or equal to a specific timestamp
baseline_category__exact	The category of the baseline for this asset
baseline_access_type__exact	The access type for the baseline for this asset
insight_name__exact	The name of the insight associated with this asset
insight_row_key_exact	The row key of the related insight
ghost__exact	Whether or not this is a ghost asset TRUE = Ghost
tasks__exact	The Resource ID of the Active tasks running on this asset

Code	Means
active_queries__exact	The Resource ID of the Active queries running on this asset
subnet_tag__exact	The device's network location
custom_attributes__exact	The value of the user defined custom attributes for this asset
class_type__exact	The class type of this asset (IT, OT, IoT)
domain_name__exact	The domain name of this asset
involved_in_tags_rid__exact	The tag artifact Resource IDs
hosted_tags__icontains	The artifact name (free text)
id__exact	The device's unique identifier in the system
site_id__exact	The identifier of the site in which this device resides
timestamp__exact	The timestamp of when this asset was detected
approved__exact	If this asset has been approved
valid__exact	If this asset is valid True = valid
parsed__exact	Whether or not parsed information (numerical or textual value from the protocols) have been incorporated by TIV into the asset's baseline
special_hint__exact integer	The address type 0 = unicast 1 = broadcast 2= multicast
risk_level__exact	The TIV level of risk for this device, based on its vulnerabilities, insights, alerts, policies, asset criticality, and network location
network_id__exact	The ID of the network in which this device is located
virtual_zone_id__exact	The group of related assets this device belongs to (e.g. PLC Modbus)
subnet_id__exact	The device's network location
purdue_level__exact	The Purdue model level of this specific device (0-6). This value is automatically determined based on the various characteristics of the asset and its purpose. Can be adjusted to reflect the true asset behavior. Also note that interim Purdue levels can also be used, e.g. 1.5, 2.5, 3.5