



## USER GUIDE v1.0.0

# Spur Context API ThreatConnect Integration Guide

### TABLE OF CONTENTS

<b>Introduction</b>	<b>2</b>
<b>Configuration</b>	<b>2</b>
Requirements	2
App Installation	2
App Configuration	2
Sample Playbook	3
<b>Enrichments</b>	<b>6</b>
Description	6
Output	6
<b>Support</b>	<b>8</b>

# Introduction

The Spur Context API integration allows a ThreatConnect user to fetch IP Address enrichments from the remote Spur Context API using an existing Spur Context API access token.

The Spur Context API provides enrichments on anonymization infrastructure and how IP Addresses are being used on the internet. These enrichments are extremely useful for determining follow on action inside the ThreatConnect environment. For example, a ThreatConnect user can determine if an IP Address is serving as a commercial VPN exit point or residential proxy and trigger (or prevent) additional automation in a playbook. Another use-case could be identifying geo-spoofing behavior (mismatches between course IP location and actual usage location) to escalate certain threats for Address indicators. Spur Context API Enrichments also provide similar IP Addresses, estimated user counts, proxied traffic activity, wifi information and more.

The Spur Context API works by providing a default set of enrichments for an IP Address. Users can configure optional enrichments for additional data. If an enrichment exists for an IP Address it will be added to the results. If an enrichment does not exist for an IP Address it will be empty.

## Configuration

### Requirements

The following requirements must be met to include the Spur Context API App in your ThreatConnect Playbooks:

- Spur Context API Integration installed (See Installation Section)
- A valid existing Spur Context API Token
- Sufficient Spur Context API Funds

### App Installation

This integration is available from the ThreatConnect GitHub. Download the TCX package and install it into your instance. Refer to the ThreatConnect System Administration Guide (Install an App) for more information or contact your ThreatConnect Customer Success Engineer.

### App Configuration

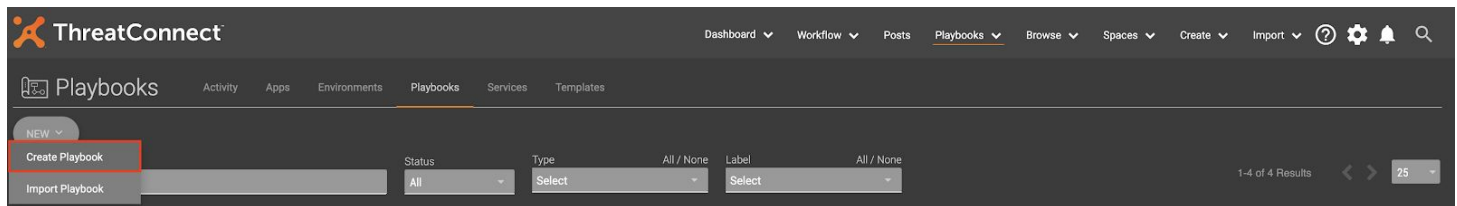
The configuration process requires a couple mandatory items:

- Action - "Context Enrichment"
- IP Address - The IP Address (IPv4 or IPv6) to enrich. This will often be a dynamic variable.
- Token - Spur Context API Token
- Fail on error - Halt execution of the app if an API error or account error occurs. This will normally be set to True.
- Fail on no results - Halt execution of the app if any of the selected "Additional Enrichments" are not present in the results. This is a strict mode that allows a user to guarantee all of their additional enrichments are returned before continuing. This will normally be set to False.

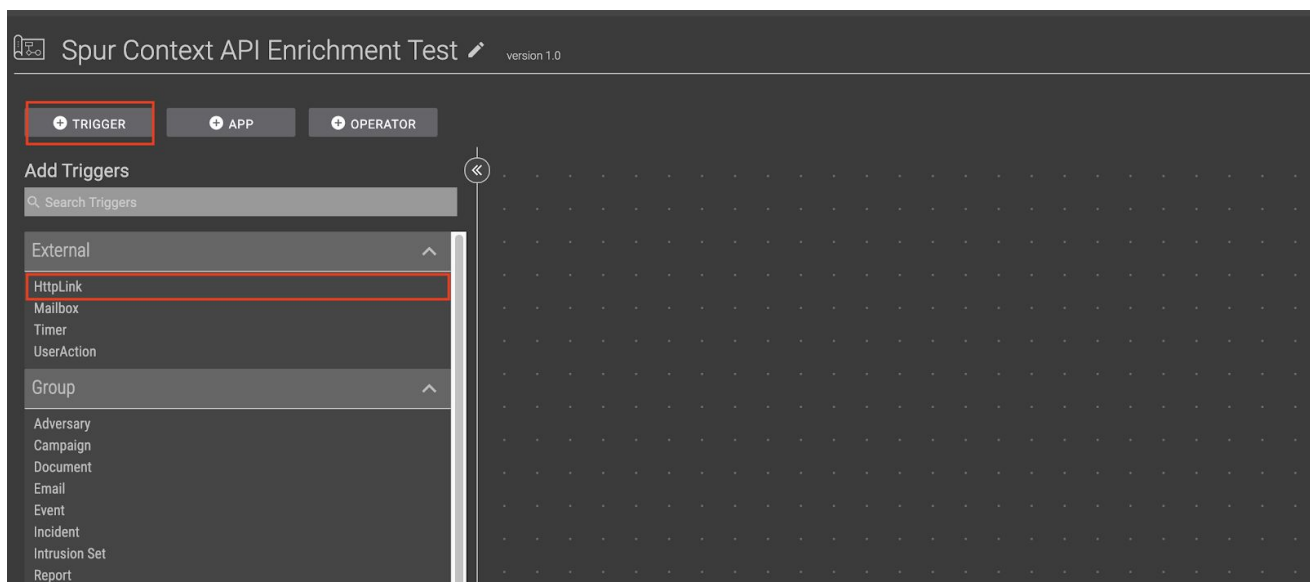
# Sample Playbook

To demonstrate the configuration of the Spur Context API integration app, we will create a sample Playbook:

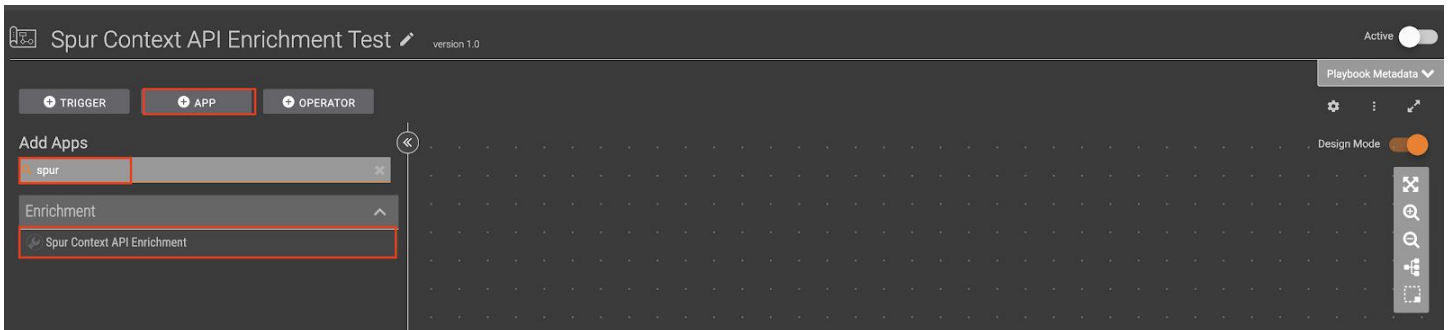
1. From the Playbooks page select **New -> Create Playbook**



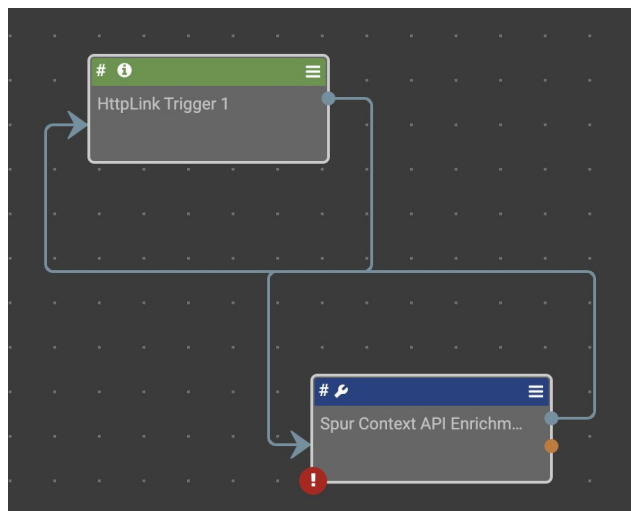
2. In the subsequent dialog box fill in the required fields and select **Save**

A 'Create Playbook' dialog box with a close button (X) in the top right. It contains two text input fields: 'Name \*' with the value 'Spur Context API Enrichment Test' and 'Description' with the value 'This is a sample playbook that uses the Spur Context API Enrichment integration'. Below these are three radio button options: 'Playbook' (selected), 'Component', and 'Workflow'. Each option has a brief description. At the bottom right are 'CANCEL' and 'SAVE' buttons. The 'SAVE' button is highlighted with an orange glow.

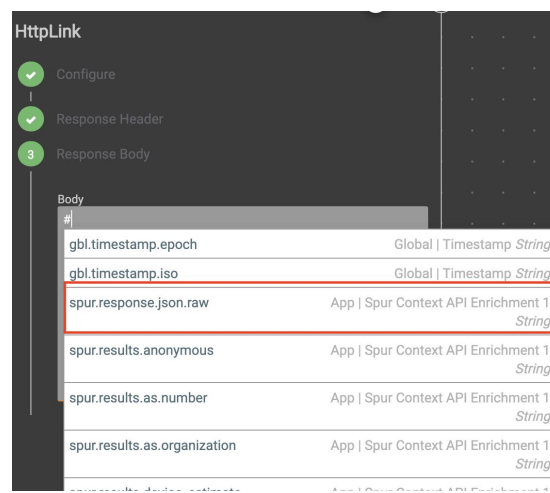
3. To test we will use an **HttpLink** Trigger to manually send an IP Address. Click on **+ TRIGGER** and then select **HttpLink**. In the future you can replace this with any Trigger that contains an IP Address.
4. Next click con **+ APP** and search for **spur**. Complete the addition by selecting **Spur Context API Enrichment**.



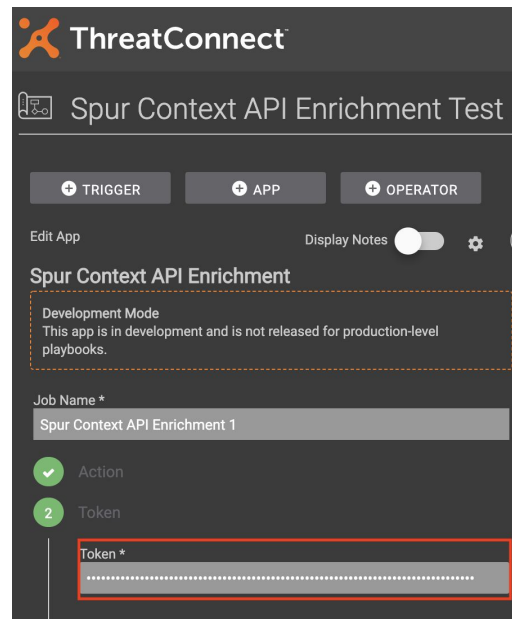
5. Next, connect the output from our **HttpLink** Trigger to our **Spur Context API Enrichment** app. Then Connect the output from the **Spur Context API Enrichment** app to the input for **HttpLink**.



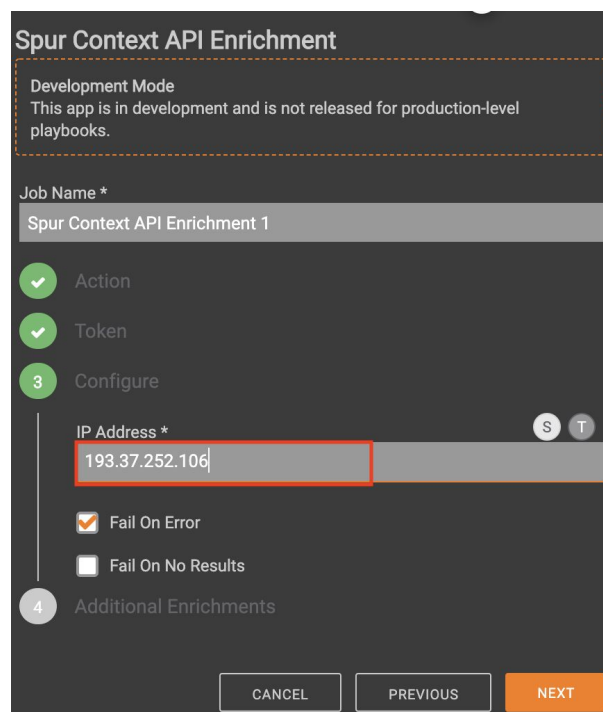
6. First, we will configure the **HttpLink** component to print the results from our **Spur Context API Enrichment**. Open the component and skip to the **Response Body** configuration step using defaults along the way. Set the response body to be **#spur.response.json.raw**. **Save** these settings.



7. Now double click on the **Spur Context API Enrichment** app to begin our Enrichment configuration. Click **Next** to select the default **Context Enrichment** action. In the second configuration stage enter your Spur Context API Token. If you need to create one, visit your account page at <https://spur.us/app/tokens>. Click on **Next** to proceed.



8. Enter an interesting IP Address to test on. This input can be configured dynamically in the future using # variables or input from other Triggers. If you would like to strictly fail if additional enrichments do not exist, check “Fail On No Results”. By default execution continues. Click **Next** to continue.



9. Select any extra data enrichments you would like to add and select **Save** to complete the configuration process. Remember, if you checked “Fail on No Results” in the previous step these fields **MUST** exist in the response for the app to complete.

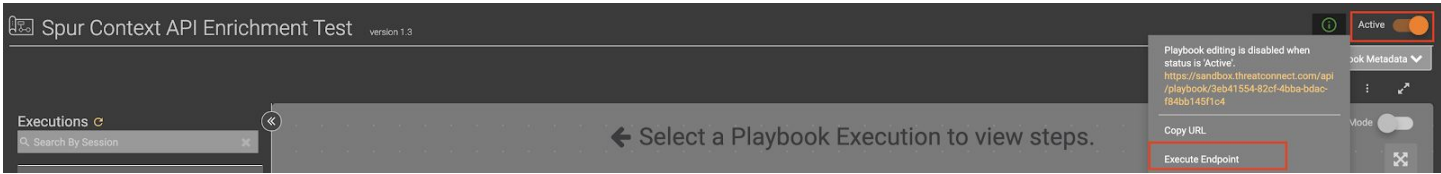
Job Name \*

Spur Context API Enrichment 1

- ✓ Action
- ✓ Token
- ✓ Configure
- 4 Additional Enrichments
  - ☐ Geo Precision
  - ☐ Proxied Traffic
  - ☒ Device Behaviors
  - ☒ Similar IPs
  - ☐ WiFi

CANCEL PREVIOUS SAVE

10. Switch the playbook to active and manually trigger the **HttpLink** endpoint to kick off the playbook. A response with your selected enrichments should return.



# Enrichments

## Description

Some enrichments are included by default. Some enrichments require their respective “Additional Enrichment” box to be checked in order to be present. If an enrichment does not exist for an IP Address it will default to it’s empty form as described in the output table.

## Output

Output Name	TC Data Type	Possible Values	Notes
#spur.results.ip	String	Valid IPv4 or IPv6 Address	The IP queried for enrichment

#spur.results.anonymous	String	"True" or "False"	Whether this IP Address contributes to anonymization infrastructure including VPN services or proxy networks.
#spur.results.vpn_operator_names	StringArray	Text strings of name	Array of VPN operators for an IP Address
#spur.results.as.number	String	0 to 4294967295	The Autonomous System Number for the IP Address
#spur.results.as.organization	String	Text organization name	The organization name for a registered ASN
#spur.results.device_estimate	String	Positive integer if confirmed. -1 if unconfirmed.	Estimated number of devices using the provided IP Address.
#spur.results.infrastructure	String	Examples: "DATACENTER", "MOBILE", "IN_FLIGHT_WIFI" or Null	Infrastructure tags for this IP Address. Only the most detailed tag is presented e.g "DATACENTER" or "MOBILE"
#spur.results.geo_lite.city	String	Example: "Orlando"	Provided by MaxMind GeoLite
#spur.results.geo_lite.state	String	Example: "Florida"	Provided by MaxMind GeoLite
#spur.results.geo_lite.country	String	A valid two Character ISO 3166-1 country code	Provided by MaxMind GeoLite
#spur.response.json.raw	String	Valid stringified JSON	Raw response data
Additional Fields (Set By Configuration)			
#spur.results.wifi.ssid	StringArray	Text Strings	Array of Wifi SSIDs associated with an IP Address. Only possible if Wifi is checked.
#spur.results.device_behavior_names	StringArray	Device behavior tags. E.g. TOR_PROXY_USER	Tags for device behaviors on the provided IP Address. Only possible if Device Behaviors is checked.
#spur.results.geo_precision.latitude	String	Float -90 to 90 or Null	The precision latitude. Only possible if Geo Precision is checked.
#spur.results.geo_precision.longitude	String	Float -180 to 180 or Null	The precision longitude. Only possible if Geo Precision is checked.
#spur.results.geo_precision.radius	String	A number 0 to 1000 or Null	The possible radius in meters for the geo point at geo_precision.latitude and geo_precision.longitude. Only possible if Geo Precision is checked.
#spur.results.geo_precision.country	String	A valid two Character ISO 3166-1 country code or Null	The precision country code. Only possible if Geo Precision is checked.
#spur.results.geo_precision.state	String	Example: "Florida" or Null	The precision state name. Only possible if Geo Precision is checked.
#spur.results.geo_precision.city	String	Example: "Orlando" or Null	The precision city name. Only possible if Geo Precision is checked.
#spur.results.geo_precision.spread	String	"xKM^2", "GLOBAL" or Null	The observed variance in relevant locations for an IP Address. Only possible if Geo Precision is checked and sufficient variance has been detected. This is different from radius which is relevant to an isolated cluster.

#spur.results.geo_precision.hash	String	Valid Geohash or Null	A Geohash that captures observable IP Location. Only possible if Geo Precision is checked.
#spur.results.proxied_traffic.residential	StringArray	Text strings of residential proxy names	Residential Proxy traffic observed on an IP Address. This field is Only possible if Proxied Traffic configuration is checked.
#spur.results.proxied_traffic.public	StringArray	Text strings of public proxy names	Public Proxy traffic observed on an IP Address. This field is Only possible if Proxied Traffic configuration is checked.
#spur.results.proxied_traffic.malware	StringArray	Text strings of malware names	Malware Proxy traffic observed on an IP Address. This field is Only possible if Proxied Traffic configuration is checked.
#spur.results.similar_ips	StringArray	Array of valid IPv4 or IPv6 Addresses	Other IP Addresses that have similar characteristics to the provided IP Address. Only possible if Similar IPs configuration is checked.

## Support

Full documentation on the enrichments available can be found on the Spur website at <https://spur.us> and in the Spur [user dashboard](#). For assistance with this app, to report a bug, or for feature requests please contact us at [support@spur.us](mailto:support@spur.us).