

Cyobstract is a SaaS web app that:

1. Creates full text RSS feed (some rss providers dont give full data)
  - a. <https://github.com/pictuga/morss> (turns website into rss)
2. polls RSS feeds on a defined schedule
  - a. <https://github.com/miniflux/v2> (feed reader)
3. extracts indicators (e.g. malicious IP addresses reported in the page) from body of the article
  - a. <https://github.com/cmu-sei/cyobstract> (extracts data)
4. exposes the extracted data to users in the UI (to export by JSON) or via JSON API.
  - a. <https://github.com/PostgREST/postgrest> (exposes api)

Note, above links are designed to show functionality. They can be used if desired, or functionality rewritten.

## Terms

To clarify any confusion, here are the terms used in this document.

- Cyobstract: The company name
- Feed: RSS feed
- Feed item: Individual items (e.g. articles) in a feed
- User: someone who logs into the platform / access the API to perform functions
- Intel group: a group of users belonging to a common group (usually the same company)
- Intel Group admin: a user who has admin rights to an Intel Group
- Staff user: Cyobstract users who manage the product
- Staff area: Area in app only visible to staff users to manage the app
- Feed: this is a RSS channel (unique RSS URL)
- Intel report: this is an item reported by an RSS channel

## Supporting Docs

UI / Workflow

[https://docs.google.com/presentation/d/1QWEWQ7JtzqxYHNRI0MRpdUpbw09WR7xRoynsg\\_jPNQ8/edit#slide=id.gada69f6b79\\_0\\_594](https://docs.google.com/presentation/d/1QWEWQ7JtzqxYHNRI0MRpdUpbw09WR7xRoynsg_jPNQ8/edit#slide=id.gada69f6b79_0_594)

## Deployment

- Will be hosted on Ubuntu droplet on Digital Ocean.
- Requires staging and production server
- Needs to run on SSL (using Let's Encrypt)
- Will be hosted at <https://app.cyobstract.com>

## Intel groups

Intel groups can be thought of like organisations. They might contain one user, they might contain many.

User is forced to create a new intel group at sign up. If they abandon sign up without creating intel group, when they attempt to login (and they do not belong to an intel group) they will be forced to create it.

Intel group has:

- UUID
  - Automatically assigned
  - *Value is fixed*
- Name:
  - Default name (for new user sign up is "Intel Group XXXXXX" (where is random number). Intel group name does not need to be unique.
  - *Can be edited after creation*
- Description (optional)
  - Default is none
  - *Can be edited after creation*
- Plan
  - Default is Unlimited free trial
    - Downgraded to free plan after 30 days, if no payment information for paid plan selected
    - See payment plans
  - *Can be edited after creation*
- Users
  - Admins
  - Members
  - *Can be edited after creation*

Users can create more than one intel group. Users can belong to more than one intel group.

Intel group has:

- Admin user (By default the user who created the group)
  - who
    - Can manage users
      - Add/remove/make admin
    - Can manage feeds
      - Enable/create/disable
    - Can manage observables
      - Enable/create/disable

- Can manage plan
    - Upgrade/downgrade
- Members
  - Can view enabled feeds
  - Can view intelligence reports (items) for enabled feeds
    - Cannot see or perform any admin actions or see any admin areas

If users belong to more than one intel group, user can switch between them in the app (dropdown top right to select intel group view, or add intel group).

When they move to group they are admin of, will show admin actions menu.

Admin user can invite other users to join group using email.

User will be prompted to confirm invite in the app to confirm they want to join the group. Will also receive email to inform them they have receive an invite (even if not a member). User can also decline invite.

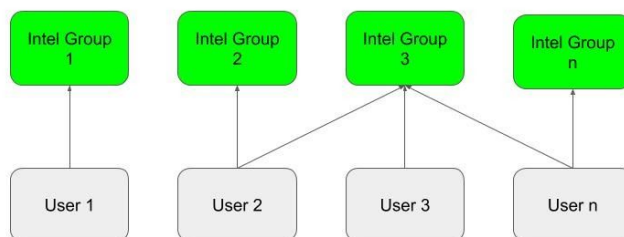
If not a member, user will be prompted to complete sign up before is able to join intel group.

Admin can invite and remove a user from intel group at any time. Admin can also assign admin privileges to other users in UI. Intel group has at least one admin user. Default admin is user who created group.

At this time, admin cannot completely delete an intel group.

What is a user?

### User / Intel Group Relationship



Users have email, password and 0 or more API keys.

Email and password can be updated (need to confirm with existing password).

API key can be created at anytime. API key is only useful if intel group they belong to has a paid API plan. Users can have multiple API keys.

API keys only show data for intel groups user belongs to. UI only shows intel groups selectable that user belongs to.

Passwords must be strongly encrypted. password (use salting/hashing/secure password). This will be tested thoroughly.

Users will be prompted to enable two factor authentication. Until enabled, they will see warning symbol in nav bar of app, when hover will show "Security warning: 2FA not enabled. Click here to enable".

For logged out users, should have password reset function.

Users can also completely delete their account. If they are only admin of an intel group with >1 other member, they must first make another user admin of those intel groups before they will be allowed to delete their account.

## Payments and plans

### Plans

The app will be charged on RSS feeds enabled in intel group.

Plans are per intel group level and managed by intel group admin.

An admin could theoretically be an admin to many intel groups and manage many payment plans (but is unlikely).

Payment plans should be able to be created by staff users in the staff area. Staff cannot edit existing or delete plans, they can only hide it. This means existing intel groups will remain on original plans.

Plans have

- Name
- Annual price
- Monthly price
- Max feeds
- Max users
- Custom feeds enabled?
- API enabled?
- Custom extractions enabled?

Plans limit intel group functionality by

- Number of enabled feeds
- If they can create custom feeds
- If they can use custom extractions
- If user can access intel group feed data by the API
- Maximum users in intel sharing group

Default plans to start:

- Small (default after free plan)
  - \$0 /mo
  - 1-5 feeds
  - No custom feeds
  - No custom extractions
  - No API access
  - 3 users maximum
- Medium
  - \$300 /yr (\$50 /mo)
  - 10-50 feeds
  - Custom feeds
  - No custom extractions

- No API access
- 20 users maximum
- Large (assigned on sign up)
  - \$500 /yr (\$75 mo)
  - 200 feeds
  - Custom feeds
  - Custom extractions
  - API access
  - 100 users maximum

All new intel groups get a free trial of the large plan for 30 days.

Whilst on free trial, everytime admin / user access intel group page, the page will show a banner message, "your plan will be downgraded and limited on DATE, to keep all existing features, you must select a plan before this date".

If not done by date, intel group will be downgraded to free plan on free-trial end date. See change plan behaviour.

### **Changing plan behaviour**

Intel group admin can upgrade or downgrade subscription at any time.

When downgrading to a lower plan we will run a check first to see if any of the allowed allowances are breached.

If automatic downgrade to free plan after free trial and any quota exceeded:

- Block non-admin users from accessing the intel group showing the message "Please contact the feed group administrator to manage intel group plan payment".
- Block API access
- Continue collecting feed data and running extractions

If manual downgrade and any quota exceeded:

- Do not let user downgrade plan until they correct the features where quota is exceeded
  - Force user to disable feeds before continuing (inc. any custom feeds)
  - Force user to remove users from group before continuing
- Then add confirm step warning admin user they will
  - Lose custom mappings functionality
    - Custom mappings will be retained and still calculated in case user upgrades late (see disabling feeds)
  - Lose API access

On downgrade all features will still be available for old plan until the end of the billing cycle they have paid for. E.g if subscription ends on 22nd, they will have access to all features until the 22nd.

## Payments

Payments handled by Stripe. We must send name, address and credit card info for each transaction to Stripe for maximum security validation.

<https://stripe.com/docs/payments/save-and-reuse>

Payments handled under Intel Group settings page where user can upgrade, downgrade a plan or delete the account.

Admin user required to enter card data when any change made.

Also, admin user should be able to update card data at any time for future payments.

Plans can be billed on yearly or annual basis. Yearly payment provide significant discount. User can select monthly or annual when changing plans.

## Feed polling

What RSS data looks like

<https://validator.w3.org/feed/docs/rss2.html>

- <channel> level data: information about the source (e.g BBC)
- <item> level data: each news article (e.g man was shot)

The database should store all raw RSS data provided.

### <channel>

<https://validator.w3.org/feed/docs/rss2.html#requiredChannelElements>

Element	Description	Example	Is always required
title	The name of the channel. It's how people refer to your service. If you have an HTML website that contains the same information as your RSS file, the title of your channel should be the same as the title of your website.	GoUpstate.com News Headlines	TRUE
link	The URL to the HTML website corresponding to the channel.	<a href="http://www.goupstate.com/">http://www.goupstate.com/</a>	TRUE
description	Phrase or sentence describing the channel.	The latest news from GoUpstate.com, a Spartanburg Herald-Journal Web site.	TRUE
language	The language the channel is written in. This allows aggregators to group all Italian language sites, for example, on a single page. A list of allowable values for this element, as provided by Netscape, is <a href="#">here</a> . You may also use <a href="#">values defined</a> by the W3C.	en-us	FALSE



copyright	Copyright notice for content in the channel.	Copyright 2002, Spartanburg Herald-Journal	FALSE
managingEditor	Email address for person responsible for editorial content.	geo@herald.com (George Matesky)	FALSE
webMaster	Email address for person responsible for technical issues relating to channel.	betty@herald.com (Betty Guernsey)	FALSE
pubDate	The publication date for the content in the channel. For example, the New York Times publishes on a daily basis, the publication date flips once every 24 hours. That's when the pubDate of the channel changes. All date-times in RSS conform to the Date and Time Specification of <a href="#">RFC 822</a> , with the exception that the year may be expressed with two characters or four characters (four preferred).	Sat, 07 Sep 2002 0:00:01 GMT	FALSE
lastBuildDate	The last time the content of the channel changed.	Sat, 07 Sep 2002 9:42:31 GMT	FALSE
category	Specify one or more categories that the channel belongs to. Follows the same rules as the <item>-level <a href="#">category</a> element. More <a href="#">info</a> .	<category>Newspapers</category>	FALSE
generator	A string indicating the program used to generate the channel.	MightyInHouse Content System v2.3	FALSE
docs	A URL that points to the documentation for the format used in the RSS file. It's probably a pointer to this page. It's for people who might stumble across an	<a href="http://backend.userland.com/rss">http://backend.userland.com/rss</a>	FALSE

	RSS file on a Web server 25 years from now and wonder what it is.		
cloud	Allows processes to register with a cloud to be notified of updates to the channel, implementing a lightweight publish-subscribe protocol for RSS feeds. More info <a href="#">here</a> .	<cloud domain="rpc.sys.com" port="80" path="/RPC2" registerProcedure="pingMe" protocol="soap"/>	FALSE
ttl	ttl stands for time to live. It's a number of minutes that indicates how long a channel can be cached before refreshing from the source. More info <a href="#">here</a> .	<ttl>60</ttl>	FALSE
image	Specifies a GIF, JPEG or PNG image that can be displayed with the channel. More info <a href="#">here</a> .		FALSE
textInput	Specifies a text input box that can be displayed with the channel. More info <a href="#">here</a> .		FALSE
skipHours	A hint for aggregators telling them which hours they can skip. More info <a href="#">here</a> .		FALSE
skipDays	A hint for aggregators telling them which days they can skip. More info <a href="#">here</a> .		FALSE

### <item>

<https://validator.w3.org/feed/docs/rss2.html#hrelementsOfLtitemgt>

A channel may contain any number of <item>s. An item may represent a "story" -- much like a story in a newspaper or magazine; if so its description is a synopsis of the story, and the link points to the full story. An item may also be complete in itself, if so, the description contains the text (entity-encoded HTML is allowed), and the link and title may be omitted. All elements of an item are optional, however at least one of title or description must be present.

Element	Description	Example	Is always required
---------	-------------	---------	--------------------

title	The title of the item.	Venice Film Festival Tries to Quit Sinking	FALSE
link	The URL of the item.	<a href="http://www.nytimes.com/2002/09/07/movies/07FEST.html">http://www.nytimes.com/2002/09/07/movies/07FEST.html</a>	FALSE
description	The item synopsis.	Some of the most heated chatter at the Venice Film Festival this week was about the way that the arrival of the stars at the Palazzo del Cinema was being staged.	FALSE
author	Email address of the author of the item. <a href="#">More.</a>	oprah@oxygen.net	FALSE
category	Includes the item in one or more categories. <a href="#">More.</a>	Simpsons Characters	FALSE
comments	URL of a page for comments relating to the item. <a href="#">More.</a>	<a href="http://www.myblog.org/cgi-local/mt/mt-comments.cgi?entry_id=290">http://www.myblog.org/cgi-local/mt/mt-comments.cgi?entry_id=290</a>	FALSE
enclosure	Describes a media object that is attached to the item. <a href="#">More.</a>	<enclosure url="http://live.curry.com/mp3/celebritySCms.mp3" length="1069871" type="audio/mpeg"/>	FALSE
guid	A string that uniquely identifies the item. <a href="#">More.</a>	<guid isPermaLink="true"> <a href="http://inessential.com/2002/09/01.php#a2">http://inessential.com/2002/09/01.php#a2</a> </guid>	FALSE
pubDate	Indicates when the item was published. <a href="#">More.</a>	Sun, 19 May 2002 15:21:36 GMT	FALSE
source	The RSS channel that the item came from. <a href="#">More.</a>	<source url="http://www.quotationspage.com/data/qotd.rss">Quotes of the Day</source>	FALSE

## Limitation of RSS feeds

A lot of feeds only provide a small intro of their articles in their RSS feeds in item.description.

As we want the full text from the URL item, we can use <https://github.com/pictuga/morss> to turn those shortened RSS item.description into the full text of the page.

To visualize the result, have a look at the before-after illustration:

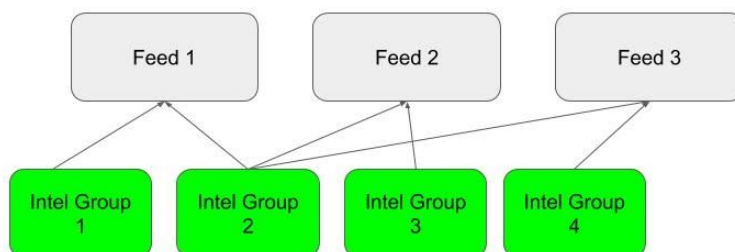
<https://morss.it/before-after.png>

This allows us to ingest complete text articles for processing.

This means we will store a new field in addition to the raw RSS, item.fullText. This will contain the full text (and formatting) of the article.

## Adding a new feed source

### Feed subscription



Staff or intel group admins can add new feed to intel group by entering:

Type	Field	Description	Can be entered by intel group admin
Fixed field	URL		TRUE
Fixed field	Poll time (seconds)	How often feed should be polled for new data	FALSE
Fixed field	Is in feed store	Boolean if feed should be displayed in the feed store globally for selection	FALSE

Per intel group field	Feed title	Name to be displayed in UI	TRUE
Per intel group field	Feed description	Description to be displayed in UI	TRUE
Per intel group field	Confidence	Value between 1 and 100 for how reliable is source	TRUE
Per intel group field	Category	User can select from a fixed list set by admin	TRUE
Per intel group field	Tags	User can assign 0 or more tags (manual entry, auto identify existing tags, letters, numbers and - only)	TRUE

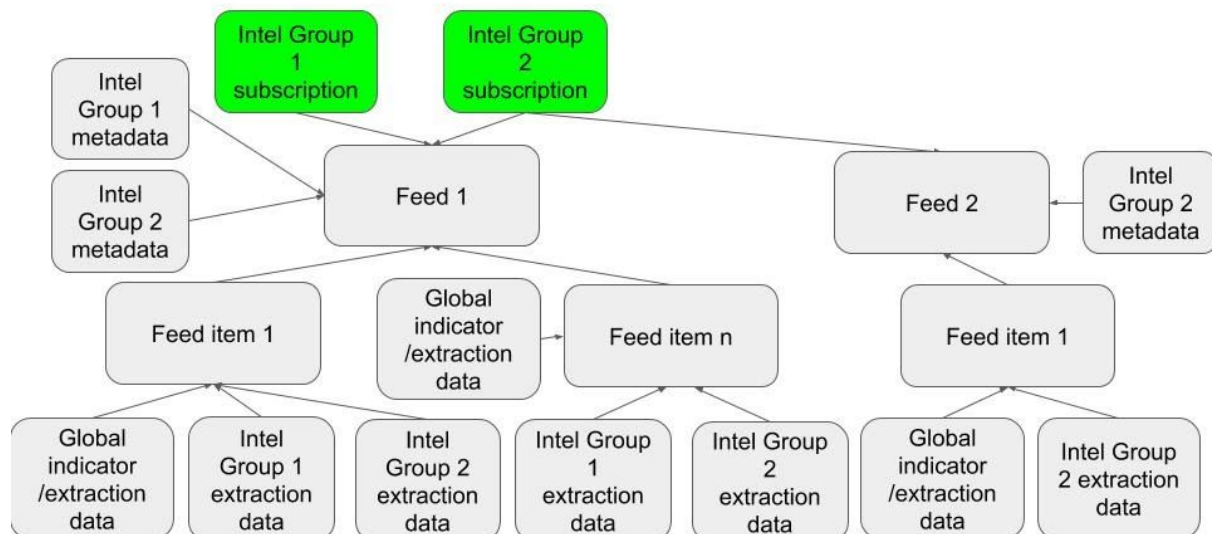
User (staff and intel group admin) will go through 2 steps to add feed

1. enters the details described above and
2. then confirm preview of their feed (what the latest entries look like) before confirming to enable the data ingest.

Only paid intel group admins can add their own custom feeds to the intel group.

Feed database design

## Feed data (extractions / intel)



When a new feed (URL) is added, a UUID is generated for it.

Note, if 2 intel groups add the same URL, the UUID will be the same, and no new feed ingest will be created. In such case, only the per intel group fields attached to the feed UUID will be stored. There is no need to ingest feed data twice -- it will be the same.

An example:

- Staff user adds feed url: x.com/rss.xml with UUID 1
- Admin of intel group 1 adds x.com/rss.xml
- Intel group 1 gets access to historic feed data for UUID 1 (with intel group extractions and observables applied)
- Admin of intel group 2 adds x.com/rss.xml
- Intel group 2 gets access to historic feed data for UUID 1 (with intel group extractions and observables applied)

In this example only one feed ID exists, and both intel groups are subscribed to it.

However, as you have seen (confidence, tags, category) and will see later (extractions) each intel specific group can assign its own metadata to feed, which is only visible to that intel group. The actual content of the report, the feed items always remain the same.

## feed store

By default there will be a library of existing feeds (created by admin) that appear in a feed store.

The library is essentially a list of all the admin created feeds (marked as library) they can quickly enable. For feed store items, intel group admin can simply enable with default settings, or modify the per intel group fields to their liking (e.g change title, description, tags, etc)

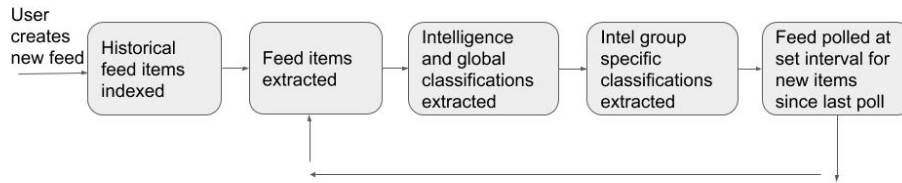
## Feed item download workflow

On first import, we must collect as much old data from feed as possible. Generally however this is a low amount of data as most feeds limit to a maximum of 25 items

(<https://stackoverflow.com/questions/576552/how-do-i-fetch-all-old-items-on-an-rss-feed>).

Feed items update periodically. We must pull in any new data when feed polled (based on time defined by staff).

## Feed item download



We should only pull items with newer timestamps from the last poll time to ensure we do not pull in old (duplicate) content (after first run)

All <channel> and <item> objects should be captured and stored in the DB as individual columns (that is, do not store raw RSS) in addition to captured item.fullText data.

## Feed processing

As new entries are polled and RSS feed data pulled into the platform, the software reads the `item.title` and `item.description` values of the post. This is done to enrich the feed.

The logic for extracting these values from feed data is here:

<https://github.com/cmu-sei/cyobstract>

Here's a good explained of this code:

<https://www.first.org/resources/papers/conf2017/Improving-Useful-Data-Extraction-from-Cybersecurity-Incident-Reports.pdf>

This is done using regex extractions over the raw feed item data:

<https://github.com/cmu-sei/cyobstract/blob/master/cyobstract/extract/regex.py>

### Extract indicators

For all feeds, cyber security indicators are extracted from text. Indicators are deduplicated and added to the database with a relationship to the RSS feed.

Extractions are assigned globally to the feed (all users see them).

Type	Type API	Value	Value API	Example
IP address	ip	IPv4	ipv4	1.1.1.1
IP address	ip	IPv4 CIDR	ipv4_cidr	192.168.100.14/24
IP address	ip	IPv4 range	ipv4_range	192.168.100.14 - 192.168.100.15
IP address	ip	IPv6	ipv6	2001:0db8:85a3:0000:000 0:8a2e:0370:7334
IP address	ip	IPv6 CIDR	ipv6_cidr	2002::1234:abcd:ffff:c0a8: 101/64
IP address	ip	IPv6 range	ipv6_range	2001:0db8:85a3:0000:000 0:8a2e:0370:7334 - 2001:0db8:85a3:0000:000 0:8a2e:0370:7335
Hash	hash	MD5	md5	79054025255fb1a26e4bc4 22aef54eb4
Hash	hash	SHA1	sha1	86F7E437FAA5A7FCE15D1 DDCB9EAEAEA377667B8



Hash	hash	SHA256	sha256	F4BF9F7FCBEDABA0392F108C59D8F4A38B3838EFB64877380171B54475C2ADE8
Hash	hash	Ssdeep SHA1	ssdeep_sha1	24:OI9rFBzwjx5ZKvBF+bi8RuM4Pp6rG5Yg+q8wIXhMC:qrFBzKx5s8sM4grq8wIXht
System	system	FQDN	fqdn	mymail.somecollege.edu
System	system	URL	url	http://x4z9arb.cn/4712/
System	system	user agent strings	user_agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.67 Safari/537.36
System	system	email address	email_addresses	example@example.com
System	system	filename	filename	file.txt
System	system	filepath	filepath	my/directory/test.txt
System	system	registry key	registry_key	HKEY_LOCAL_MACHINE\S
Infrastructure	infrastructure	ASN	asn	65525
Infrastructure	infrastructure	ASN owner	asn_owner	Some provider
Infrastructure	infrastructure	country	country	United Kingdom
Infrastructure	infrastructure	ISP	isp	Some provider
Analysis	analysis	CVE	cve	<b>CVE</b> -1999-0067
Analysis	analysis	malware	malware	WastedLocker
Analysis	analysis	attack type	attack_type	Phishing

#### A note on multiple extractions

Some values might match to multiple values. For example “192.168.100.14 - 192.168.100.15” would create 1 IPv4 range indicator and 2 IPv4 indicators.

### A note on defanged extraction

In some cases, malicious entries (e.g. URL's) are fanged in data from feeds. Defanging obfuscates indicators into a safer representations so that a user doesn't accidentally click on a malicious URL or inadvertently run malicious code. They look like so:

- www dot cert dot org ([www.cert.org](http://www.cert.org))
- www[.]cert[.]org
- www[.cert].org
- www{.}cert{.}org
- incidents at cert dot org (incidents@cert.org)

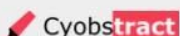
Typical types of defanged data include IP addresses, fully qualified domain names (FQDNs), email, and file extensions.

The above code extraction module successfully recognises and extracts many forms of defanged indicator values.

### A note on intel group specific visibility of indicators by type

Indicators will always be extracted from feed items and stored into the DB.

However, intel group admins will be able to enable or disable visibility of indicators shown to users in their intel group. By default all are enabled, but admin can disable as required.



[Advanced search](#)

**Member actions:** [View intel](#) **Admin actions:** [Manage feeds](#) [Manage observables](#) [Manage users](#)


#### Manage Indicator visibility

Type ?	Value ?	Example ?	
Ip (ip)	Ipv4 (ipv4)	"1.1.1.1"	<a href="#">Disable</a>
Hash (hash)	Md5 (md5)	"9384jdy84"	<a href="#">Enable</a>
System (system)	Filename (filename)	test.txt	<a href="#">Disable</a>

### A note on intel group specific visibility of indicators by whitelist

Users can also whitelist specific values.

For example, google.com might be whitelisted from an extraction domain=google.com because it is obviously a safe website. In the case of Cyobstract users, they might want to include their own custom domains.



[Advanced search](#)

Member actions: [View intel](#)

Admin actions: [Manage feeds](#) [Manage observables](#) [Manage users](#)

### Manage Indicator visibility

Manage by type

Type ?	Value ?	
Ip (ip)	Ipv4 (ipv4)	<a href="#">Disable</a>
Hash (hash)	Md5 (md5)	<a href="#">Enable</a>
System (system)	Filename (filename)	<a href="#">Disable</a>

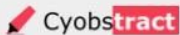
Manage by whitelist

Type	Value	Value to whitelist	
Ip (ip)	Ipv4 (ipv4)	"1.1.1.1"	<a href="#">Disable</a>
Hash (hash)	Md5 (md5)	"9384jdy84"	<a href="#">Enable</a>

[Add custom whitelist entry](#)

Page 1, 2, 3...

By default admin can add global whitelist values (disabled by default) that show in all intel groups.



[Advanced search](#)

Member actions: [View intel](#)

Admin actions: [Manage feeds](#) [Manage observables](#) [Manage users](#)

### Add to whitelist

Indicator type

?

Indicator value to exclude

?

Enable

Admin of intel group can also do this by selecting

- Indicator type parent (optional) (allow all) (e.g. ip)
- Text Value to whitelist

Whitelists do not affect how the app process the feed items. E.g. google.com still identified as an indicator in the DB for above example. The values in enabled whitelist entries are simply hidden from users of that intel group.

#### Observable matches

In addition to indicators, we should also attempt to classify the item (feed report) based on word matches in the item.fulltext and item.title

For example if <item> title or text values contained the word “malware” it would be classified as Threat Type = malware (the word malware matched in the article -- see table below)


Multiple observables can be assigned e.g Threat Type = Malware, Threat Type = Data breach, Threat actor = North Korea....

New observables mappings are applied to all feed items retrospectively when added.

Word matches are case insensitive.

observable	API observable	Value Assigned	API value	Words to match (separated by comma)
Threat type	threat_type	Malware	malware	malware
Threat type	threat_type	Data Breach	data_breach	data breach, leak, exposed
Country	country	North Korea	north_korea	North korea, KP, PRK, Pyongyang
Country	country	Russia	russia	Russia, RU, Moscow
Threat actor	threat_actor	Fancy Bear	fancy_bear	APT-21, Fancy Bear
Product	product	Adobe	adobe	Adobe, Flash
Product	product	Microsoft	microsoft	Outlook, Office365

Sector	sector	Healthcare	healthcare	Health, healthcare, hospital
Sector	sector	Energy	energy	Nuclear, energy, power plant



[Advanced search](#)

**Member actions:** [View intel](#)
**Admin actions:** [Manage feeds](#) [Manage observables](#) [Manage users](#)

## Manage Observable extractions

Custom extractions

Add custom extraction

Observable Type	Observable Value	Words to match on	
Threat type (threat_type)	Phishing (phishing)	"phishing"	<a href="#">Disable</a>

Global extractions

Observable Type ?	Observable Value ?	Words to match on ?	
Threat type (threat_type)	Malware (malware)	"malware"	<a href="#">Disable</a>
Country (country)	United Kingdom (united_kingdom)	"United Kingdom", "UK"	<a href="#">Disable</a>
Product (product)	Microsoft (microsoft)	"Outlook", "Office365"	<a href="#">Disable</a>

Page 1, 2, 3...

## Global observable extractions

Staff can add new observables, values and word matches in the staff UI..

Staff created values are assigned globally to the feeds/feed items (all users see them). These extractions are applied all feeds and feed items and global data stored in DB.

However, intel group admin can disable (hide) these extractions for being shown in their intel group. For example, the disable "malware" global extraction. The app will still extract the "malware" information and store it in DB at a global level, but for all users in this intel group it will not be shown. Other groups that have it enabled will see such extractions.

## Per intel group observable extractions

Paid users can also create their own observables and mappings in the same way as staff at an intel group level. However, only users in the intel group see the additional custom mappings.

If admin is not in a paid plan that allows for custom observables, when they click add custom observable message, modal error message will be shown "Sorry, your plan does not

currently cover custom observable abstractions. You can upgrade now to enable this feature [here](#).”

Note, the behaviour of observable extractions created by admin is different to those created by staff. Admin intel group created extractions belong to an intel group and are assigned only to feeds and feed items this intel group has enabled and disabled (because even disabled feeds still collect data although is hidden to user).

In this case, 2 or more intel groups might create same extraction (e.g. both called dog that both search for the word “dog”, with all fields being the same). These are therefore treated as two distinct extractions. In this case they are two different extractions performed by app at per intel group level only on intel group feeds.

Note, for intel group observables admin can create an entry that matches staff.

E.g staff creates

<b>observable</b>	<b>Value Assigned</b>	<b>Words to match (separated by comma)</b>
Threat type	Malware	malware

And admin creates:

<b>observable</b>	<b>API observable</b>	<b>Value Assigned</b>	<b>API value</b>	<b>Words to match (separated by comma)</b>
Threat type	threat_type	Malware	malware	Malware

These are treated as two separate extractions and entries in the DB (global and per intel group), despite having the same value. This allows user to build / modify global extractions to suit their needs. E.g. might disable global admin extraction, and change the words observable/value matches on. Similarly intel group owners can create duplicate extractions where they want to update what is shown (useful because extractions applied to all data -- see next line)

Any new observable extractions added by intel group admin OR staff will apply retrospectively, all old data will be analysed and observables extracted as a result.

## Feed / intel viewing

The UI will offer a simple UI to configure, select and view feed data.

### Enabling feeds

All intel group admin added feeds will appear in “Feed Store”.

The screenshot shows the 'Feed Store' interface. At the top, there is a header bar with the 'Cyobstract' logo. Below the header, the 'Feed Store' title is followed by a 'Filter:' section with three dropdown menus: 'Confidence >', 'Category >', and 'Tag >'. To the right of these is a pink 'Filter' button. The main content area displays three feed cards. Each card has a 'Feed name' field, a 'Feed description' field, and a 'Feed URL' field. Below these fields are three buttons: 'Feed category' (blue), 'Feed Tag 1' (yellow), and 'Feed Tag 2' (yellow). To the right of each card is an 'Enabled' button (grey for the first card, green for the others) and a link that says 'See in feed list' or 'Custom settings and enable'. At the bottom of the page, there is a pagination link 'Page 1, 2, 3...'.

To enable a feed in the feed store, user simply selects the title of the feed, and optionally sets the following values to be assigned to each feed item (if left blank they default to values entered by admin):

- Feed name
- Confidence (1-100)
- Category (selectable from list)
- Tags (manual entry, auto identify existing global tags, letters, numbers and - only)

If paid intel group, then user can add custom feed, by clicking add feed button and following the steps described earlier in the document. User can modify confidence, category and tags at any time but will be applied to new items only.

Before a custom feed is ingested, user shown a preview of what rss content will look like. They must confirm it looks ok before enabling (and app start collecting data).

Configured feeds Filter:

Feed name

Feed description

Feed URL

Feed category

Feed Tag 1

Feed Tag 2

Disable

Edit settings

Feed name

Feed description

Feed URL

Feed category

Feed Tag 1

Feed Tag 2

Enable

Edit settings and enable

Feed name

Feed description

Feed URL

Feed category

Feed Tag 1

Feed Tag 2

Enable

Edit settings and enable

Page 1, 2, 3...

A user can also disable feed data from appearing in their intel group under configured feeds page

IMPORTANT: even if they disable a feed, app will continue to poll this feed for new data AND assign intel group extractions. That way we keep historical data and can provide this data if user turns feed back on OR another group adds this feed url. User cannot delete a feed for this reason.

If user attempts to add more feeds than in their plan they are prompted to upgrade.

## Viewing feeds

User can see stream of the enabled feeds items that have been ingested into that intel group.

Under there intel group they can do this by visiting /intel page. The content in this view will be unique to intel group.

It will show title, category, confidence, tags, url of channel, the date of publication, any observable values and count of extracted indicators.

User can also filter by enabled feed name, confidence, tags, indicator type, observable, or do a raw plaintext search for title / description content to filter.



**Intel reports**

Filter:

Item name	Item description	Item URL	Feed category	Feed Tag 1	Feed Tag n	Feed name	Confidence: <input type="button" value="Confidence score"/>	observables: <input type="button" value="Classification 1"/> <input type="button" value="Classification n"/>	Indicators: <input type="button" value="Indicator 1"/> <input type="button" value="Indicator n"/>
Item name	Item description	Item URL	Feed category	Feed Tag 1	Feed Tag n	Feed name	Confidence: <input type="button" value="Confidence score"/>	observables: <input type="button" value="Classification 1"/> <input type="button" value="Classification n"/>	Indicators: <input type="button" value="Indicator 1"/> <input type="button" value="Indicator n"/>
Item name	Item description	Item URL	Feed category	Feed Tag 1	Feed Tag n	Feed name	Confidence: <input type="button" value="Confidence score"/>	observables: <input type="button" value="Classification 1"/> <input type="button" value="Classification n"/>	Indicators: <input type="button" value="Indicator 1"/> <input type="button" value="Indicator n"/>
Item name	Item description	Item URL	Feed category	Feed Tag 1	Feed Tag n	Feed name	Confidence: <input type="button" value="Confidence score"/>	observables: <input type="button" value="Classification 1"/> <input type="button" value="Classification n"/>	Indicators: <input type="button" value="Indicator 1"/> <input type="button" value="Indicator n"/>

Page 1, 2, 3...

User can also click each item to view page with entire article. Each article has a unique UUID.

/intel/FEED\_UUID/ITEM\_UUID

Note, the URL is the same between intel groups as it is referencing same database entry. It is only the observable information unique to the organization (intel specific extractions) that change in the view of the page.

**Report name**

Item short description

This is the full body of the text.

All the information with **INDICATORS** are here.

**EXTRACTIONS** are also highlighted. When a user hovers over a highlighted item shows the type=value.

Confidence:

observables:

Indicators:

JSON object:

```

1 {
2   "id": {
3     "results": {
4       "type": "EmployeeDetails.Employee"
5     }
6   }
7   "UserID": "E12012",
8   "RoleCode": "35"
9 }
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

API call

Feed: <https://www.cyobstract.com/api/v1/cccccc?x>

Object: <https://www.cyobstract.com/api/v1/cccccc?x>

[API docs.](#)

Note, report page should publish with basic HTML copying paragraphs, titles, tables etc from original report. Remove any css styling though so is delivered consistent with CO design.

Article will show any indicators or observable matches highlighted. Each highlight for observable / indicator type will be different colour. On hover will show type=value (e.g. ipv4=0.0.0.0)

At bottom of the article, it will show all the data extracted (indicators, mapping, tags, category confidence).

Also show option to view data in pretty printed JSON form (see api)

## **Staff area**

### **Will be able to see (and search)**

- Intel groups
  - Feeds enabled
    - And config (admin can edit / disable)
  - Billing plan
    - Admin can cancel / change
  - Custom observables
  - Custom tags
- Users
  - Last login
  - API usage
  - Intel groups they belong too (is admin)
    - Admin can change / remove
  - Admin can send password reset
  - Delete user
- Indicators
  - List of indicators extracted, type, and items they belong to.
- Global observables
  - Add / edit
- Global tags

## **UI**

- Keep it simple. Standard bootstrap is fine.
- User must see errors with descriptive text for all errors (inc api)
- Every field where user enters some text / field has help icon (question mark next to it that when clicks explains what it does). [e.g RSS feed URL [ ] ?
- All tables should have option for multiselect > perform action. E.g. select all whitelist values and enable

## Emails

All emails should use <https://mjml.io/> HTML styling.

### Confirm email

**From:** sherlock[@mg.cyobstract.com](mailto:sherlock@mg.cyobstract.com)

**Name:** Sherlock at Cyobstract

**Reply-to:** sherlock@cyobstract.com

**Title:** [ACTION REQUIRED] Confirm your email address to use all Cyobstract features!

**Body:**

Welcome to Cyobstract!

All that's left to do to complete your registration is click the link below to confirm your email.

Link

If you have any questions, simply reply to this email to get in contact with a real person on the team.

*Sherlock and the Cyobstract Team*

### New invite to join Intel Group

**From:** sherlock[@mg.cyobstract.com](mailto:sherlock@mg.cyobstract.com)

**Name:** Sherlock at Cyobstract

**Reply-to:** sherlock@cyobstract.com

**Title:** [ACTION REQUIRED] You've been invited to join the XXXX Intel Group on Cyobstract

**Body:**

Hello!

USER\_EMAIL has invited you to join the XXXX Intel Group on Cyobstract as a USER\_ROLE.

By accepting this invitation, you'll have access to all intelligence curated by the other members of the XXXX Intel Group.

To confirm or reject this invitation, click the link below.

Link

If you have any questions, simply reply to this email to get in contact with a real person on the team.

*Sherlock and the Cyobstract Team*

2FA enabled

**From:** sherlock@[mg.cyobstract.com](mailto:sherlock@mg.cyobstract.com)

**Name:** Sherlock at Cyobstract

**Reply-to:** sherlock@cyobstract.com

**Title:** [INFORMATIONAL] Two factor authentication has been enabled on your Cyobstract account

**Body:**

Hello!

Now you've enabled two factor authentication your Cyobstract account now has an additional level of security.

This email is just to confirm it is enabled. No further action is needed.

If you did not make this change, please reply to this email immediately to get in contact with a real person on the team.

*Sherlock and the Cyobstract Team*

2FA disabled

**From:** sherlock@[mg.cyobstract.com](mailto:sherlock@mg.cyobstract.com)

**Name:** Sherlock at Cyobstract

**Reply-to:** sherlock@cyobstract.com

**Title:** [INFORMATIONAL] Two factor authentication has been disabled on your Cyobstract account

**Body:**

Hello!

This email confirms two factor authentication has been disabled on your Cyobstract account.

We strongly recommend you re-enable it to ensure your account remains secure.

If you did not make this change, please reply to this email immediately to get in contact with a real person on the team.

*Sherlock and the Cyobstract Team*

Password changed

**From:** sherlock@mg.cyobstract.com

**Name:** Sherlock at Cyobstract

**Reply-to:** sherlock@cyobstract.com

**Title:** [INFORMATIONAL] Your Cyobstract password has been successfully reset

**Body:**

Hello!

This email is just to confirm your Cyobstract password has now been changed. No further action is needed.

If you did not make this change, please reply to this email immediately to get in contact with a real person on the team.

*Sherlock and the Cyobstract Team*

Password reset

**From:** sherlock@mg.cyobstract.com

**Name:** Sherlock at Cyobstract

**Reply-to:** sherlock@cyobstract.com

**Title:** [ACTION REQUIRED] Reset your Cyobstract password

**Body:**

Hello!

You have just initiated the process to reset your password on Cyobstract.

To set a new password, click the link below.

Link

If you did not make this request, no action is needed, your account is secure.

*Sherlock and the Cyobstract Team*

Intel group invite accepted

**From:** sherlock@mg.cyobstract.com

**Name:** Sherlock at Cyobstract

**Reply-to:** sherlock@cyobstract.com

**Title:** [INFORMATIONAL] USER\_1 has accepted your invitation to join INTELGROUP

**Body:**

Hello!

This email is just to confirm USER\_1 has accepted your invitation to join INTELGROUP

To manage members in your intel group, click the link below:

Link

*Sherlock and the Cyobstract Team*

Intel group invite rejected

**From:** sherlock@[mg.cyobstract.com](mailto:mg.cyobstract.com)

**Name:** Sherlock at Cyobstract

**Reply-to:** sherlock@cyobstract.com

**Title:** [INFORMATIONAL] USER\_1 has rejected your invitation to join INTELGROUP

**Body:**

Hello!

USER\_1 has rejected your invitation to join INTELGROUP

If you think this is a mistake, you can resend the invitation to USER\_1 to join INTELGROUP

To manage members in your intel group, click the link below:

Link

*Sherlock and the Cyobstract Team*

## Onboarding

When user creates an account, we will start an onboarding process using intro.js. It shows each of the key features and prompts users to add them.

We will also automatically add them to a mailerlite group for marketing emails.

In the UI, we should hide any other functions (e.g. search) until onboarding complete (or user abandons app). See mockups for more information.

<https://introjs.com/>

User moves to next step when action completed

1. 2 minute onboarding  
Start creating intelligence feeds with maximum value by completing this onboarding.
2. Create Intel Group (sign up)  
Intel Groups are sharing communities. Create as many as you need to for individual or team use. Lot's of users have a personal and team group.
3. Invite users to your intel group (sign up)  
Optinally, you can now add users to your intel group. Other members of the group will only be able to see and download the intelligence in your space. Enter a list of comma separated emails if you want to start sharing with colleagues or industry peers
1. Enable your first feed (feed store)  
Now it's time to start ingesting date. Cyobstract takes data from RSS feeds and extracts threat intelligence (more on that soon). You can add custom RSS feeds later, for now select some major sources you're interested in.
2. Intel reports  
You can now see that the intel report list list is population with posts. See how the intelligence is reported against each report. Let's take a closer look.
3. Report highlights (item page)  
You can see the report has highlighted the threat intelligence we've extracted. There is two objects, indicators and observables. Indicators are things like IP addresses or file hashes, observables show the context (e.g. Russia).
4. Export data (bottom of item page)  
You can see below each report the JSON object for the report. You can integrate with your SIEM, SOAR, TIP or other security product using our API. Just click view API docs.
5. Add custom extractions  
You can also create your own observables by matching text in each intel report. As you've seen we've created some global indicators you can use, or create your own.
6. End of tour  
That's it from us. You can also join out






# API

## Authentication

User can generate more than one API key. This is useful if they want to integrate with more than one external product.

They can do this under their account settings.



[Advanced search](#)

Member actions: [View intel](#)

Admin actions: [Manage feeds](#) [Manage observables](#) [Manage users](#)

### User account

[Reset password](#)

[Enable 2FA](#)

**Intel groups you belong to**  
Xxxx (member) [Leave group]  
YYY (admin) [Leave group]  
ZZZ (pending) [accept invite]

**Your API keys**

XXXXX	name 1	Intel groups	<input type="button" value="Delete"/>
XXXXX	name 2	Intel groups	

**Your webhooks**

Endpoing	Description	Intel groups	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
----------	-------------	--------------	-------------------------------------	---------------------------------------



[Advanced search](#)


Member actions: [View intel](#)

Admin actions: [Manage feeds](#) [Manage observables](#) [Manage users](#)

### Create new API key

API key gives user access to all intel from each of the intel groups they belong too.

API key passed in header of request.



[Advanced search](#)

**Member actions:** [View intel](#)
**Admin actions:** [Manage feeds](#) [Manage observables](#) [Manage users](#)

## User account

Add a webhook endpoint

Endpoint URL

Description  
An optional description of what this webhook endpoint is used for.

Version

Events to send  
Select events...

☒ account.updated

## Behaviour

Api endpoint and version is: /api/v1

API returns JSON responses only.

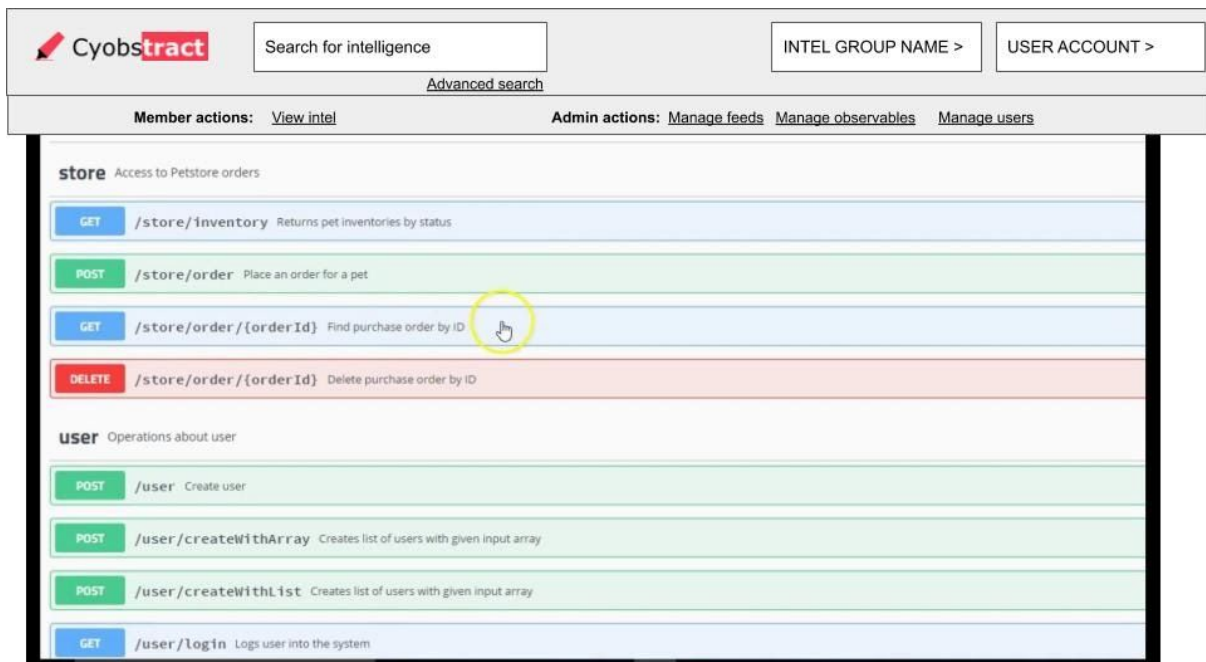
Paginate per 50 entries.

All errors must serve clear error messages (e.g. this intel group does not have an active api subscription, api key invalid...)

## API docs

All API documentation will be rendered in the app using swagger under /api/docs. Any user can view docs (even if not supported by plan).

<https://github.com/swagger-api/swagger-ui> (PostgresAPI generates swagger docs)



GET /api/v1/feeds

This endpoint reports data for enabled feeds for intel groups user belongs too.

Parameters available for filtering (if blank = all):

- Intel\_Group\_UUID: multiple allowed. Is OR statement
- Page: for pagination
- Minimum confidence (confidence must be  $\geq$  to this value)
- Tags - multiple allowed. Is OR statement
- Category - multiple allowed. Is OR statement
- Earliest date last data added (YYYY-MM-DD)
- Latest date last data added (YYYY-MM-DD)

JSON object

- Channel
  - UUID # Platform unique identifier. Is linked to feed\_url
  - Type # Always = "RSS" for v1
  - Feed\_URL # RSS feed URL
  - Intel\_Group\_UUID # Note in cases where two intel groups polling same feed we need to have two items for each because they might have custom confidence, categories or tgs
  - Confidence # Confidence score for applied to feed for intel group
  - Category # Categories applied to feed for intel group
    - []
  - Tags # Tags applied to feed for intel group
    - []
  - Datetime\_created # when feed added to db
  - Datetime\_last\_polled # when feed last polled

- Datetime\_last\_data # *when feed last produced data (not always last poll time)*
- RSS # *this object prints all original rss channel data*
  - Title
  - Link
  - description
  - language
  - copyright
  - managingEditor
  - webMaster
  - pubDate
  - category
  - lastBuildDate
  - generator
  - Docs
  - Cloud
  - itl
  - image
  - textInput
  - skipHours
  - skipDays

## GET /api/v1/reports

This endpoint reports data for intel reports belonging to intel groups they are part of.

Note we do not print the fullText field via the API for the report due to size. User can use the Report\_URL field to get platform link

Parameters available for filtering (if blank = all):

- Intel\_Group\_UUID: multiple allowed. Is OR statement
- Channel\_UUID: multiple allowed. Is OR statement
- observable types - multiple allowed. Is OR statement (e.g. threat\_actor=x, country=x)
- Indicator types - multiple allowed. Is OR statement (e.g. indicator\_type=url, indicator\_type=ipv4)
- Earliest date added (YYYY-MM-DD)
- Latest date added (YYYY-MM-DD)
- Page: for pagination

JSON object

- Item
  - UUID
  - Channel\_UUID

- Intel\_Group\_UUID # *Note in cases where two intel groups polling same feed we need to have two items for each because they might have custom observables. Hence the use of this field*
- Report\_URL # *Link to the report in the platform.*
- Datetime\_added
- RSS\_data # *this object prints all original rss item data*
  - Title
  - Link
  - Description
  - Author
  - Category
  - Comments
  - Enclosure
  - Guid
  - pubDate
  - Source
- Indicators
  - IP
    - []
  - System
    - []
  - Infrastructure
    - []
  - Analysis
    - []
  - Hash
    - []
- Observables
  - Global
    - Threat\_type
      - []
    - Product
      - []
  - Intel\_Group
    - Something
      - []
    - Else
      - []

GET /api/v1/intel\_group

This endpoint reports data for intel groups user belongs to.

JSON object

- **UUID** # shows UUID
- **Name** # shows group name
- **Description** # shows group description
- **Role** # shows authenticated users role in group

## Outgoing Webhooks

If you are new to webhooks Stripe have a good description how they work:

<https://stripe.com/docs/webhooks>

Or slack <https://api.slack.com/messaging/webhooks>

Webhooks fire when an event happens. An event is a new intel report.

User can limit the alerts to fire from events that belong to only certain intel group they belong to.

User selects the destination of the webhook when setting up

(<https://hooks.slack.com/services/T00000000/B00000000/XXXXXXXXXXXXXXXXXXXX>).

These external apps can listen for events fired by invent hooks for real time ingest. By entering the URL of the webhook.

On user account page they should be able to create webhook events with

- Webhook destination (third party url). Can be edited after creation.
  - URL of where webhook sent. Must be valid url
- Webhook description. Optional. Can be edited after creation.
  - Text description for reference
- Intel groups to listen on. Can be edited after creation.
  - Should only listen for events from selected intel groups. Must select at least one.
- Words to listen on. Can be edited after creation.
  - Optional. User can enter list of words. Treated as OR statement. Alert will only fire if a word in list appears in a report title OR fulltext.

User can also enable, disable (stop sending events) or completely delete webhooks.

The webhook sends json messages with basic report details for item (not full intel, like reported in the API).

Webhook json object:

- UUID
- Channel\_UUID
- Intel\_Group\_UUID # *Note in cases where two intel groups polling same feed we need to have two items for each because they might have custom observables. Hence the use of this field*

- Report\_URL # *Link to the report in the platform.*
- Datetime\_added
- RSS\_data # *this object prints all original rss item data*
  - Title
  - Link
  - Description

**Important: Use a proper queue:** Use RabbitMQ. This way, your interaction is limited to adding and removing “messages,” which tell you what webhooks to call. Like the DB queue, you need a separate process to consume items from the queue and send notifications. In addition to using a tool designed for this purpose, a proper queue also saves database resources for what it does best—providing data to your primary application.