

---

**pyexclient**

**Jul 26, 2021**



---

## Contents

---

<b>1</b>	<b>About the documentation</b>	<b>3</b>
<b>2</b>	<b>Getting Started</b>	<b>5</b>
2.1	Requesting an API key . . . . .	5
2.2	Installation . . . . .	5
<b>3</b>	<b>Understanding the Client</b>	<b>7</b>
3.1	Resource Objects . . . . .	7
3.2	Attributes . . . . .	8
3.3	Relationships . . . . .	8
<b>4</b>	<b>Usage and Example use cases</b>	<b>11</b>
4.1	The basics . . . . .	11
4.2	Helpers . . . . .	15
4.3	Examples . . . . .	16
<b>5</b>	<b>Workbench API Reference</b>	<b>25</b>
	<b>Python Module Index</b>	<b>99</b>
	<b>Index</b>	<b>101</b>



Expel is a technology company that built a platform (Expel Workbench) to enable delivery of our 24x7 Managed Detection and Response (MDR) service delivery. The Expel Workbench platform is used by both analysts and customers to communicate, triage, investigate and remediate incidents.

When Expel was founded, we had the goal of making our platform “flypaper for inventors.” What this means is that, if you have an idea, you should be able to safely test it out on top of our platform using existing technology levers (in this case APIs).

The Expel Workbench is [JSON API compliant](#) and uses swagger for API specification/documentation. You can find Expel’s swagger doc [here](#). Pyexclient is the pythonification of the JSON API spec. The pyexclient client code, and much of the documentation, is auto generated from swagger spec file.



---

## About the documentation

---

The pyexclient documentation is composed of the following sections:

- **Introduction** – Brief background and description of Expel and pyexclient.
- *Getting Started* – How to get started using pyexclient and walks through requesting an API key.
- *Understanding the client* – Provides intuition and background on the pythonification of JSON API spec.
- *Usage and Examples* – Reading this section should enable you to have an intuitive sense for how to accomplish your specific needs. A large number of example uses in this documentation can implement everything from creating and updating an investigation, to extracting all source/dest IPs associated with Expel alerts.
- *API Reference* – Detailed API reference.





There's one precondition to using pyexclient: you must have a Workbench user account. This means you need to be a customer or going through a POC with Expel.

If you have an Expel Workbench account, you can authenticate to *Expel using MFA* or you can request an API key.

### 2.1 Requesting an API key

To get an API key, you'll need to reach out to your Expel Engagement Manager. In the future we'll make the API key generation a self-serviceable feature.

### 2.2 Installation

Pyexclient requires Python3.7+. It has no additional dependencies.

To install pyexclient using pip, run:

```
$ pip3 install pyexclient
```



---

## Understanding the Client

---

Becoming familiar with pyexclient requires understanding three main properties of the JSON API spec and how the pyexclient implements them. They are:

- *Resource Objects* – These are implemented as python classes. Each class is documented and referenced in the API Reference section.
- *Attributes* – These are updatable/readable fields per resource object. They're implemented as python members of a class. There are certain fields that are read-only. Currently you can update them in the python class, but if you POST (try to save) those changes, they'll fail. The documentation of each attribute calls out which fields are read-only.
- *Relationships* – These are specific links from one resource object to another.

<p><b>Warning:</b> As a user of the client you can delete any object within your tenant. Expel keeps point in time as well as daily back ups, but you're using the <b>delete functionality at your own peril and risk.</b></p>
--

### 3.1 Resource Objects

It's useful to think of resource objects as logical containers for attributes and relationships. A resource object within Expel Workbench is just a data table in a database. Relationships are links to other tables and attributes are columns in a table. This intuition is helpful as we build complex tasks.

Every resource object is documented in the API reference section. Each resource object has a table containing the following:

- Field Description – A description of the given field
- Field Name – The name of the field to reference it in code
- Field Type – The type of field it is. If the field is a relationship, this will be a hyperlink to the resource object.
- Attribute – Yes / No indicates if the field is an attribute
- Relationship – Yes / No indicates if the field is a relationship.

Pyexclient has properties representing each resource object. Each documented resource (see [API Reference section](#)) has a resource type field. The value of this field is the property name. So, if for example, we want to work with investigations, we would find the *Investigations* resource object in the API reference section and see that it is called investigations. We can then use this property to do any of the creating or retrieving.

See examples on [creating](#), [retrieving](#), [listing](#) and [finding](#) resource objects.

## 3.2 Attributes

Every resource has a set of attributes. There will always be an id, and a created\_at attribute. The attributes are used by Expel Workbench UI and automated systems to reason about various activities and to reflect/updated status.

Attributes can be used to filter down to specific resource objects that you're interested in. See [Finding Open Investigations](#) for an example. Not sure what attributes are available for a resource object? Check out the Attribute column in the API docs for the resource. Rows where the Attribute column is "Y" indicate the given field is an attribute of the object, not a relationship to another object.

Attributes are accessible like any attribute on a python object. Changing them, and then calling the save method will write the changes back to Expel Workbench.

## 3.3 Relationships

Relationships describe the linkage between two different resource types There are two types of relationships an resource object can have. The first is one to one, where the relationship represents a relationship to another single resource instance. The second type of relationship is one to many, where the relationship would encompass multiple resource instances.

---

**Note:** It's not entirely clear from the documents which relationship is one to one versus one to many it's something we'll look at addressing in the future.

---

The most common task when working with relationships is to retrieve the full resource object referenced by the relationship. For example, let's say we want to grab the name of the actor that is assigned to the investigation with ID cf9445b1-a0aa-4092-af5f-ecdc136d1661.

```
inv = x.investigations.get(id="cf9445b1-a0aa-4092-af5f-ecdc136d1661")
print(f"Assigned to actor name {inv.assigned_to_actor.display_name}")
```

This pattern will retrieve the full underlying resource referenced by the relationship assigned\_to\_actor, which is a relationship between a resource type of Investigation. In this case, an Investigation instance (ID = cf9445b1-a0aa-4092-af5f-ecdc136d1661) and an instance of the *Actor* resource.

In the case of a one to many relationship, where you want to retrieve the full resource object you would do the following:

```
for ea in x.expel_alerts.search(expel_severity=neq("TESTING")):
    for va in ea.vendor_alerts:
        print(va.vendor_sig_name)
```

In the above example, the Expel alerts resource object has a one to many relationship with vendor alerts and in this situation you'd iterate over them to see every instance that is part of that relationship.

Sometimes you just want to know the identifier of the resource referenced by the relationship. In this example we're just retrieving the ID for the actor assigned to our investigation. To do this you can do the following:

```
inv = x.investigations.get(id="cf9445b1-a0aa-4092-af5f-ecdc136d1661")
print(f"Assigned to actor id {inv.relationship.assigned_to_actor.id}")
```

Note in the above code snippet how we use `relationship`, this tells pyexclient that you just want the ID for the relationship and not the full resource object.

### 3.3.1 Modify Objects

Modifying an object with pyexclient can be done by retrieving the object, updating its attributes and then saving the updated object. For example:

```
inv = x.investigations.get(id="myinvestigationid")
inv.title = "My updated investigation title"
inv.save()
```

This can also be simplified with the below syntax (which will automatically call `.save()` for you):

```
with x.investigations.get(id="myinvestigationid") as inv:
    inv.title = "My updated investigation title"
```



---

## Usage and Example use cases

---

Before diving into example use cases, it's important to grasp the basics of the pyexclient. The basics will allow you to implement your own custom use cases.

### 4.1 The basics

Every resource type supported by Expel Workbench is implemented as a python class in pyexclient. The base resource type class has four methods implemented along with a context handler and iter method. Understanding how to use these concepts will make you a rockstar (sorry had to) when it comes to building or improving your automated use cases. Let's walk through each method in detail.

All code snippets below assume you've *authenticated* and have the authenticated pyexclient in the variable `x`.

#### 4.1.1 create(...)

The create method is used to create new instances of a resource type. You can see examples of this *create comment*, or *create investigation*. You must call *save()* for changes/creations to be written back to the server. Every attribute for the given resource type can be specified (via its field name) as a named parameter to the create method. In addition to specifying the values of attributes for a specific resource type, you can also specify relationships when creating a new resource type. To specify a relationship when creating a new resource type you'll prepend `relationship_` and then relationship name. The value is going to be the identifier to the already existing resource type that the relationship will link to. Some relationships are required when creating a new instance of a resource type. Let's look at a sample:

```
ACTOR_ID = "5ac919dd-352d-4cde-a5b3-c0c3ed77a318" # Current User ID
CUSTOMER_ID = "d44fcb09-90e3-44a2-831e-f381aaec37f5" # Customer ID
inv = x.investigations.create(title="New Incident", relationship_
↳organization=CUSTOMER_ID, is_incident=True, analyst_severity="MEDIUM",
↳relationship_assigned_to_actor=ACTOR_ID)
inv.save()
```

The above snippet creates an incident with a severity of *Medium*, title of *New Incident* that is assigned to *ACTOR\_ID*. The other way to create a new instance is:

```
ACTOR_ID = "5ac919dd-352d-4cde-a5b3-c0c3ed77a318" # Current User ID
CUSTOMER_ID = "d44fcb09-90e3-44a2-831e-f381aaec37f5" # Customer ID
inv = x.investigations.create(title="New Incident", is_incident=True)
inv.relationship.organization = CUSTOMER_ID
inv.relationship.assigned_to_actor = ACTOR_ID
inv.save()
```

This snippet accomplishes the same thing as above but to some maybe easier to read.

### 4.1.2 get(...)

The get method is used when you already know the identifier of the existing resource instance you want to retrieve. Once you've retrieved the resource instance you can read and/or modify the resource instance's attributes.

```
inv = x.investigations.get(id="22adb298-1e9e-424c-a754-b8ab09f38282")
inv.title = "New Title"
inv.save()
```

The above snippet changes the title of the investigation. You must call `save()` to have changes written back to the Expel Workbench. Otherwise the changes are just local and useless.

### 4.1.3 save(...)

This method will POST any changes to the resource instance back to the Expel Workbench. If you do not call this method after making modifications the modifications will not be reflected in Expel Workbench.

### 4.1.4 search(...)

Understanding this method means you can easily access resource instances that meet complex criteria without having to iterate through tons of data. The search method pushes the filtering logic to the server side for evaluation and only returns instances that matched the criteria. There are six useful operators to be aware of when building search criteria. Let's walk through examples of each:

#### neq()

This operator will return resource instances where the specified attribute is not equal to the value provided to `filter_by`.

```
for rem_act in x.remiation_actions.search(status=neq("CLOSED")):
    print(f"Recommended remediation action is {rem_act.action} the status is {rem_act.
↪status}")
```

In the snippet above we're searching for any remediation action that is not currently closed. Then we print the remediation action text and the current status.

#### contains()

**Warning:** Partial matches are not indexed and API performance can be impacted by doing a lot of these requests. Investigative data is indexed and optimized for searching, but you must use `flag("search", "term")`.



This operator will do a substring search (“partial match”) on a given attribute’s value and return the resource instances that have a partial match. This search operation is case insensitive. This operator will return resource instances where the specified attribute is equal to the value provided to filter by.

```
for cmt in x.comments.search(comment=contains("oops")):
    print(f"Found comment with word oops in it {cmt.comment}")
```

The above snippet will search all comments in Expel Workbench and return any instance where the comment contains the word “oops.”

### startswith()

This operator will return instances of resources where the value of a specified attribute starts with the provided text.

```
for cmt in x.comments.search(comment=startswith("hey")):
    print(f"Found comment that starts with hey '{cmt.comment}'")
```

### isnull(), notnull()

It allows you to search for instances where a specified attribute is null or not null.

```
for rem_act in x.remdiation_actions.search(status=isnull()):
    print(f"Recommended remediation action is {rem_act.action} the status is null")

for rem_act in x.remdiation_actions.search(status=notnull()):
    print(f"Recommended remediation action is {rem_act.action} the status is not null
↪")
```

### gt(), lt(), window()

You can specify a field should be greater than, and/or less than a value by using the `gt()` or `lt()` operators respectively. To do searches over a range or window you’ll use the `window()` operator.

```
start_date = (datetime.datetime.now()-datetime.timedelta(days=1)).isoformat()

for cmt in x.comments.search(comment=startswith("hey"), created_at=gt(start_date)::
    print(f"Found comment in past 24 hours that starts with hey '{cmt.comment}'")
```

The above snippet looks for comments starting with the word “hey” that were created in the past 24 hours.

```
end_date = datetime.datetime.now().isoformat()

for cmt in x.comments.search(comment=startswith("hey"), created_at=lt(end_date)
    print(f"Found comment in past 24 hours that starts with hey '{cmt.comment}'")
```

The above snippet does the same thing looking for comments created at a timestamp less than the current time. Finally the window operator:

```
start_dt = (datetime.datetime.now()-datetime.timedelta(days=3)).isoformat()
end_date = (datetime.datetime.now()-datetime.timedelta(days=1)).isoformat()

for cmt in x.comments.search(created_at=window(start_dt, end_date), ↪
↪comment=startswith("hey")):
    print(f"Found comment in past 2 days that starts with hey '{cmt.comment}'")
```

This example looks for comments created in past two days that start with “hey”. The window operator supports strings, integers and datetime objects.

### flag()

Our API supports a custom query parameter called flag. Flag allows callers to pass variables to the backend. Flags are defined on a resource by resource basis, and will alter the behavior of a given API call. The most commonly used flag parameter will be “search” which will search investigative data in a highly optimized way.

```
for inv in x.investigations.search(flag("search", "ransomware")):
    print(f"Incident related to ransomware: {inv.title}")
```

### limit()

The API supports a limit operator that will limit the number of results returned by the server. This can be used when you are calling an API and you only need, or care about one result.

```
for inv in x.investigations.search(flag("search", "ransomware"), limit(1)):
    print(f"Incident related to ransomware: {inv.title}")
```

## 4.1.5 relationship(...)

Sometimes you may want to work with a resource type, but you want to filter based on criteria applied to another resource type that it has a relationship to. This is most common when you are wanting to filter resource type objects that are voluminous like investigative actions. You can specify you’re wanting to filter on a relationship resource type by using the relationship operator. Let’s look at a few examples:

```
start_date = (datetime.datetime.now()-datetime.timedelta(days=1)).isoformat()
for inv_act in x.investigative_actions.search(relationship("investigation.created_at",
↳ gt(start_date)), action_type="MANUAL"):
    print(f"Found investigative action associated with manual investigation created_
↳ in the past 24 hours {inv_act.title}")
```

This snippet applies filtering criteria to two attributes. The action\_type attribute lives in the investigative\_action resource type and filters out any investigative action that is not manually created. The next filter is applied to investigation resource type. In this case there’s a relationship between investigations and investigative actions. This scopes what search returns to investigative actions that are associated with investigations that have been created in the past 24 hours.

## 4.1.6 Context Handler

There’s a context handler implemented for all resource types. It makes it easy to save changes to existing resource instances. It can be used by specifying the resource type as a property in conjunction with a call to the get method().

```
with x.investigations.get(id="53212cd8-475e-442e-8102-28d20ca33246") as inv:
    inv.title = "New Updated Title"
```

This will update the investigation with a new title and save it back to the API.

## 4.1.7 Iteration / Pagination

Iterating over all the instances of any resource type is as simple as a for loop.

```
for expel_alert in x.expel_alerts:
    print(f"Expel Alert {expel_alert.expel_name}")
```

Pyexclient will handle the pagination of results and will yield each instance in the for loop. This allows for easy implementation of filtering logic on the client side should you so desire.

```
for expel_alert in x.expel_alerts:
    if expel_alert.expel_severity != "HIGH":
        continue
    print(f"Expel Alert {expel_alert.expel_name}")
```

The above snippet only prints Expel alerts with *HIGH* severity. You could also implement this with `search(expel_severity="HIGH")`.

## 4.2 Helpers

Pyexclient contains a number of helper methods that can be useful when performing common tasks.

Before diving into the helper methods, it's important to understand a little bit about Investigative Actions within Expel Workbench since the helper functions operate on investigative actions.

### Background on Investigative Actions

Investigative actions are most commonly actions run by Expel's automated systems or analysts during the course of alert triage and/or during investigations/incidents. The actions type and parameters specified to the investigative action tell Expel's backend integration and tasking infrastructure to go gather specific types of data.

The acquired data is usually summarized and relevant information presented to the analyst and/or customer. The raw data can be downloaded from within Workbench, or viewed using Expel Workbench's built-in data viewer.

### 4.2.1 download(...)

Sometimes when you're automating tasks or integrating systems, you'll want the ability to access the raw data that the investigative action collected. This helper function makes downloading data from an investigative action easy. This can only be called on investigative action resource types.

```
with x.investigative_actions.get(id=inv.act_id) as ia:
    with tempfile.NamedTemporaryFile() as fd:
        ia.download(fd)
        pprint.pprint(json.loads(fd.read()))
```

The above example will download and print the JSON data backing the investigative action (`inv.act_id`).

### 4.2.2 create\_auto\_inv\_action(...)

This helper function will automate the creation (subsequent execution) of an investigative action associated with a security device. This is how you can automate investigative tasks that are backed by Expel's integration with a security vendor.

```
ia = x.create_auto_inv_action(
    title='Query SIEM for activity involving 1.2.3.4',
    input_args={'query':'1.2.3.4',
               'start_time':'2020-09-03T13:38:19.539071',
               'end_time':'2020-09-03T16:38:19.539071'},
    capability_name='query_logs',
    vendor_device_id='my-vendor-device-guid',
    customer_id='my-organization-guid',
    reason='To see what else happened involving this IP.',
    created_by_id='my-actor-id',
    investigation_id='my-investigation-id',
)
```

In the above example, we ran an investigative action “Query Logs” which will query the security device for activity involving 1.2.3.4.

### 4.2.3 create\_manual\_inv\_action(...)

This helper function will create a manual investigative action associated with an investigative action. Manual actions can be used to record investigative questions and answers that analysts wish to associate with an investigation.

```
ia = xc.create_manual_inv_action(
    title = "Investigate suspicious url evil.com",
    reason = "Research evil.com to see if it is actually suspicious.",
    instructions = "Investigate open source intel to gather additional details",
    Investigation_id = "my-investigation-id")
```

In the above example, we created a manual investigative action to investigate a suspicious URL. Once created, the action can serve as a placeholder for our results once we’ve gathered the relevant data. To complete the action, we can close it with results like so:

```
ia.status = "COMPLETED"
ia.results = "I investigated this URL and found it was not suspicious."
ia.save()
```

### 4.2.4 capabilities(...)

The capabilities helper function can be used to determine what automatic actions are possible for your organization based on the currently on-boarded integrations.

```
x.capabilities("my-organization-id")
```

## 4.3 Examples

We’ve provided examples based on what we’ve heard about from customers who are wanting to further integrate with our platform. There are three types of examples we’ve documented.

1. *Snippet* - This is code self contained in the documentation. Usually just a few lines.
2. *Script* - This is a whole python script that accomplishes the use cases. A brief description on each script is provided. The scripts themselves are in examples/ directory.

3. *Notebook* - A jupyter notebook that implements, mostly experimental concepts that forward leaning customers might be interested in.

### 4.3.1 Snippet: Authentication

There are two ways to authenticate to Expel Workbench. The first is as a user with your password and MFA token, the second is with an API key. To authenticate as a user, you'll need to provide your password and your 2FA code.

```
import getpass
from pyexclient import WorkbenchClient

print("Enter Username:")
username = input()
print("Enter Password:")
password = getpass.getpass()
print("2FA Code:")
code = input()

xc = WorkbenchClient('https://workbench.expel.io', username=username,
↳password=password, mfa_code=code)
```

To authenticate with an api token:

```
xc = WorkbenchClient('https://workbench.expel.io', token='apitoken')
```

### 4.3.2 Snippet: List all open remediation actions

Sometimes it can be useful to review all open remediation actions. This is a snippet of [Open Remediation Actions](#) will list all remediation actions that are not currently completed or closed. You can optionally specify a date range to scope the search too.

Listing 1: examples/open\_remediation\_actions.py

```
# Start documentation snippet

# Search remediation actions where the status is not equal to CLOSED or COMPLETED,
↳and optionally it was created within the window of start_date and end_date.
# start and end date's can be None in which case the search will look at all
↳remediation actions.
for rem in xc.remediation_actions.search(created_at=window(start_date, end_date),
↳status=neq('COMPLETED', 'CLOSED')):
    # Calculate the number of days since the remediation action was created.
    since = (datetime.datetime.now() - datetime.datetime.strptime(rem.created_at, "%Y-
↳%m-%dT%H:%M:%S.%fZ")).days
    print(f'{rem.action} created {rem.created_at} ({since} days ago) has status {rem.
↳status} and the comment is "{rem.comment if rem.comment else ""}')

    # Count the number of assets we have
    count = xc.remediation_action_assets.search(relationship('remediation_action.id',
↳rem.id)).count()
    print(f'Found {count} remediation action assets for remediation action {rem.id}')

    # Now print all the assets we have for the parent action
    for asset in xc.remediation_action_assets.search(relationship('remediation_action.
↳id', rem.id)):
```

(continues on next page)

(continued from previous page)

```
print(f'\t{asset.status} - {asset.asset_type} - {asset.value}')
# End documentation snippet

__name__ == '__main__':
```

### 4.3.3 Snippet: Return device name of security device ID

Working with identifiers can be helpful, but also hard to mentally keep track of at times. This example is a simple function to return the human readable name of a security device ID

```
def security_device_to_name(xc, device_id):
    device = xc.security_devices.get(id=device_id)
    if device:
        return device.name
    return None

device_id = "158b031d-87f8-4c42-80ee-f9fb15796360"
device_name = security_device_to_name(xc, device_id)
```

### 4.3.4 Snippet: Return devices with a specific investigative action support

Before starting an investigative action, it is sometimes helpful to look up the capabilities of your onboarded devices to make sure you have a device that supports a particular investigative action. This example will use Capabilities to look for *ENDPOINT* devices, such as EDR or antivirus devices, that support the Query Domain capability.

```
def get_query_domain_devices(xc):
    endpoint = xc.capabilities().get("ENDPOINT")
    if endpoint:
        query_domain = endpoint.get("query_domain")
        if query_domain:
            security_devices = query_domain.get("security_devices")
            if security_devices:
                return security_devices
    return None

query_domain_devices = get_query_domain_devices(xc)
```

### 4.3.5 Snippet: Listing investigations

Iterate over all the investigations and print their title and status.

```
for inv in xc.investigations:
    s = "Investigation ID: {inv_id} Title: {inv_title} Status: {inv_status}"
    status = "OPEN" if inv.decision is not None else "CLOSED"
    print(s.format(inv_id=inv.id, inv_title=inv.title, inv_status=status))
```

### 4.3.6 Snippet: List comments

List all comments, displaying when they were created and by which user.

```

for comment in xc.comments:
    s = "[{ts}] {cmt} - {user}"
    print(s.format(ts=comment.created_at, cmt=comment.comment, user=comment.created_
↳by.display_name))

```

### 4.3.7 Snippet: create comment

Create a comment and associate it with an investigation.

```

comment = xc.comments.create(comment="Hello world!")
comment.relationship.investigation = 'my-investigation-id'
comment.save()

```

### 4.3.8 Snippet: Listing Investigative Actions

List investigative actions by type or capability name.

For example, listing all manual (human driven) investigative actions:

```

for inv_act in xc.investigative_actions.search(action_type='MANUAL'):
    print(inv_act)

```

Alternatively, you could search for all automatic actions to acquire a file like this:

```

for inv_act in xc.investigative_actions.search(capability_name='acquire_file'):
    print(inv_act)

```

### 4.3.9 Snippet: Find top automatic Investigative Actions

Find the top 10 automatic investigative actions by number of times they are issued.

```

from collections import defaultdict

# Retrieve all automatic actions
actions = defaultdict(int)
for action in xc.investigative_actions.search(action_type='TASKABILITY'):
    actions[action.capability_name] += 1

# Sort and list top 10 actions
top_actions = sorted(actions.items(), key=lambda x: x[1], reverse=True)
top_actions[:10]

```

### 4.3.10 Snippet: Creating new investigation

Create a new investigation in Workbench.

```

inv = xc.investigations.create(title='My investigation title')
inv.save()

```

### 4.3.11 Snippet: List open investigation

List open investigations in Workbench.

```
from pyexclient.workbench import notnull

for inv in xc.investigations.search(decision=notnull()):
    print(inv)
```

### 4.3.12 Snippet: Close an investigation

Update an investigation's state by closing it. Note that setting an investigation's decision to anything other than None will close it.

```
with xc.investigations.get(id='my-investigation-id') as inv:
    inv.decision = "FALSE_POSITIVE"
    inv.close_comment = "This is a false positive."
```

### 4.3.13 Snippet: Creating findings for an incident

Create new investigative findings for an incident.

```
finding = xc.investigation_findings.create(
    rank = 1, # The order in which this finding will appear in Workbench
    title = "Where else is it?", # Title of the finding
    finding = "We found it EVERYWHERE!", # Markdown body for the finding
)
finding.relationship.investigation = 'my-investigation-id'
finding.save()
```

### 4.3.14 Snippet: Modify investigation findings

Modify findings text for an investigation.

```
with xc.investigation_findings.get(id='my-finding-id') as finding:
    finding.finding = "Updated: Turns out it wasn't _everywhere_..."
```

### 4.3.15 Snippet: Create an investigative action and poll for completion

Create “auto” investigative actions, using our tasking framework. This example will use the Query Logs investigative action. After creating the investigative action shows how to download the results. Assumes the results completed. Requires knowing the following values: - Investigation ID - A user ID, can also use customer ID in place of a specific user - Vendor device ID to task - Input arguments to the “task” defined per capability - Query that is specific to the SIEM we are talking too. This example works on Sumo Logic.

```
import time
from io import BytesIO
from datetime import datetime, timedelta

input_args = dict(
```

(continues on next page)



(continued from previous page)

```

query="evil.exe",
start_time=(datetime.now() - timedelta(days=1)).isoformat(),
end_time=datetime.now().isoformat(),
)

action = xc.create_auto_inv_action(
    vendor_device_id='my-vendor-device-id',
    capability_name='query_logs',
    input_args=input_args,
    title="Query Sumo Logic for some logs",
    reason="I want to see if I can find some logs...",
    investigation_id='my-investigation-id'
)

while action.status == 'RUNNING':
    print("Waiting for results...")
    time.sleep(3)
    action = xc.investigative_actions.get(id=action.id)

if action.status == 'READY_FOR_ANALYSIS':
    results = io.BytesIO()
    action.download(results)
    results.seek(0)

    with open("results.json", 'wb') as fd:
        fd.write(results.read())
    print("Got results! Saved to results.json")
else:
    print("No results... {status}".format(status=action.status))

```

#### 4.3.16 Snippet: Upload investigative data

While uncommon, it can happen that a customer has access to logs or data that we don't. In that case it's important Expel gain access to that data to help complete an investigation. In this example we'll show how you can upload arbitrary to an investigation.

```

# create an manual investigative action
action = xc.investigative_actions.create(
    action_type='MANUAL',
    title='Upload file',
    reason='To provide a file to Expel for analysis',
    status='READY_FOR_ANALYSIS',
)
action.save()

# read an upload a file
fname = 'evil.exe'
with open(fname, 'rb') as fd:
    action.upload(fname, fd.read())

```

### 4.3.17 Snippet: Return Expel Alerts closed as PUP/PUA

Expel Alert close decisions can be helpful to identify certain types of alerts in your organization. This example will find alerts with a close decision of PUP/PUA.

```
for ea in xc.expel_alerts.search(close_reason='PUP_PUA'):  
    print(ea)
```

### 4.3.18 Snippet: Interacting with Expel hunting investigations

Note: Hunting investigations are specific to the Expel Hunting service and available to those who have purchased this option.

```
for inv in xc.investigations.search(source_reason="HUNTING"):  
    print(inv)
```

### 4.3.19 Snippet: Return devices with a specific investigative action support

Before starting an investigative action, it is sometimes helpful to look up the capabilities of your onboarded devices to make sure you have a device that supports a particular investigative action. This example will use Capabilities to look for *ENDPOINT* devices, such as EDR or antivirus devices, that support the Query Domain capability.

```
capabilities = xc.capabilities()  
supported = capabilities.get('ENDPOINT', {}).get('query_domain', {}).get('security_  
↪devices')  
if supported:  
    print("Devices supporting this capability: ", supported)  
else:  
    print("No devices support this capability")
```

### 4.3.20 Snippet: Close a remediation action as completed

Update a remediation action as completed, and close it in Expel Workbench.

```
with xc.remediation_actions.get(id='remediation_action_id') as action:  
    action.status = 'COMPLETED'  
    action.close_reason = 'We remediated this system.'
```

### 4.3.21 Script: Export Expel Alerts with Evidence Fields

See the example script [Export Expel Alert Evidence](#). This script will write a CSV containing timestamp of alert, expel alert name, vendor name, and associated evidence fields.

### 4.3.22 Script: Poll for new Incidents

See the example script [Poll For New Incidents](#). This script will poll Expel Workbench for any incidents created in the past five minutes.

### 4.3.23 Script: Sync to JIRA

See the example script [Jira Sync](#). This script will sync the following to JIRA from Expel Workbench:

- Investigative Actions details and outcome as sub tasks
- Investigation description, lead alert
- Investigative comments
- Incident findings
- Investigation status closed/opened

### 4.3.24 Script: Poll unhealthy devices

See the example script [Poll For Unhealthy Devices](#). This script will poll Expel Workbench for any devices marked unhealthy in the past five minutes.

### 4.3.25 Script: Poll for investigation / incident changes

See the example script [Poll For Investigaiton / Incident updates](#). This script will poll Expel Workbench for any updates to investigations or incidents in the past five minutes.

### 4.3.26 Script: Pretty Print Lead Expel Alert Evidence

See the example script [Pretty Print Lead Expel Alert Evidence](#). This script will pretty print the Expel Alert details along with all correlated vendor evidences.

### 4.3.27 Notebook: Metrics notebook

The example metrics notebook [Expel Metrics Example](#). Shows a few different ways you can interact with Expel data to draw some interesting insights.



---

## Workbench API Reference

---

**class** `pyexclient.workbench.ActivityMetrics` (*data, com, included=None*)

Bases: `pyexclient.workbench.ResourceInstance`

Defines/retrieves expel.io activity\_metric records

Resource type name is **activity\_metrics**.

Example JSON record:

```
{
    'activity': 'string',
    'created_at': '2019-01-15T15:35:00-05:00',
    'data': {},
    'ended_at': '2019-01-15T15:35:00-05:00',
    'referring_url': 'https://company.com/',
    'started_at': '2019-01-15T15:35:00-05:00',
    'updated_at': '2019-01-15T15:35:00-05:00',
    'url': 'https://company.com/'
}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Rela-tionship
Activity Allows: "", null	activity	string	Y	N
Created timestamp: readonly	created_at	string	Y	N
Additional data about the activity Allows: null: no-sort	data	object	Y	N
Date/Time of when the activity concluded	ended_at	string	Y	N
Referring url Allows: "", null	refer-ring_url	string	Y	N
Date/Time of when the activity started	started_at	string	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Url Allows: "", null	url	string	Y	N
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Expel alerts	expel_alert	<i>ExpelAlerts</i>	N	Y
Investigations	investiga-tion	<i>Investigation</i>	N	Y
Security devices	secu-rity_device	<i>SecurityDevices</i>	N	Y
Defines/retrieves expel.io actor records	up-dated_by	<i>Actors</i>	N	Y

**class** pyexclient.workbench.**Actors** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io actor records

Resource type name is **actors**.

Example JSON record:

```
{'actor_type': 'system', 'created_at': '2019-01-15T15:35:00-05:00', 'display_name
↳': 'string', 'is_expel': True, 'updated_at': '2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Actor type Restricted to: "system", "user", "organization", "api"	actor_type	any	Y	N
Created timestamp: readonly	created_at	string	Y	N
Display name Allows: "", null	display_name	string	Y	N
Meta: readonly, no-sort, no-filter	is_expel	boolean	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
investigative actions	analysis_assigned_investigative_actions	<i>InvestigativeActions</i>	N	Y
Expel alerts	assigned_expel_alerts	<i>ExpelAlerts</i>	N	Y
Investigations	assigned_investigations	<i>Investigations</i>	N	Y
investigative actions	assigned_investigative_actions	<i>InvestigativeActions</i>	N	Y
Organization to resilience actions	assigned_organization_resilience_actions	<i>OrganizationResilienceActions</i>	N	Y
Organization to resilience actions	assigned_organization_resilience_actions_list	<i>OrganizationResilienceActions</i>	N	Y
Remediation actions	assigned_remediation_actions	<i>RemediationActions</i>	N	Y
Defines/retrieves expel.io remediation_action_setting records	auto-mate_opt_in_for_remediation_action_setting	<i>RemediationActionSettings</i>	N	Y
Defines/retrieves expel.io actor records	child_actors	<i>Actors</i>	N	Y
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
User Notification Preferences	notification_preferences	<i>NotificationPreferences</i>	N	Y
Defines/retrieves expel.io organization records	organization	<i>Organizations</i>	N	Y
Defines/retrieves expel.io actor records	parent_actor	<i>Actors</i>	N	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y
User accounts	user_account	<i>UserAccounts</i>	N	Y

**class** pyexclient.workbench.**ApiKeys** (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io api\_key records. These can only be created by a user and require an OTP token.

Resource type name is **api\_keys**.

Example JSON record:

```
{
    'access_token': 'string',
    'active': True,
```

(continues on next page)

(continued from previous page)

```
'assignable': True,
'created_at': '2019-01-15T15:35:00-05:00',
'display_name': 'string',
'name': 'string',
'realm': 'public',
'role': 'expel_admin',
'updated_at': '2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Rela-tion-ship
Only upon initial api key creation (POST), contains the bearer api key token required for api access.: readonly, no-sort, no-filter	access_token	string	Y	N
Active Allows: null	active	boolean	Y	N
Can Api key be assigned items (e.g. investigations, etc)	assignable	boolean	Y	N
Created timestamp: readonly	created_at	string	Y	N
Display name Allows: null	display_name	string	Y	N
Missing Description	name	string	Y	N
Realm in which the api key can be used. Restricted to: “public”, “internal”	realm	any	Y	N
Role Restricted to: “expel_admin”, “expel_analyst”, “organization_admin”, “organization_analyst”, “system”, “anonymous”, “restricted”	role	any	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Defines/retrieves expel.io actor records	created_by	Actors	N	Y
Defines/retrieves expel.io organization records	organization	Organizations	N	Y
Defines/retrieves expel.io actor records	updated_by	Actors	N	Y

**class** pyexclient.workbench.AssemblerImages (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

Assembler Images

Resource type name is **assembler\_images**.

Example JSON record:

```
{
  'created_at': '2019-01-15T15:35:00-05:00',
  'hash_md5': 'string',
  'hash_sha1': 'string',
  'hash_sha256': 'string',
  'platform': 'VMWARE',
  'release_date': '2019-01-15T15:35:00-05:00',
```

(continues on next page)



(continued from previous page)

```
'size': 100,
'updated_at': '2019-01-15T15:35:00-05:00',
'version': 'string'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Rela-tionship
Created timestamp: readonly	cre-ated_at	string	Y	N
Assembler image md5 hash Allows: null	hash_md5	string	Y	N
Assembler image sh1 hash Allows: null	hash_sha1	string	Y	N
Assembler image sha256 hash Allows: null	hash_sha256	string	Y	N
Platform Restricted to: "VMWARE", "HYPERV", "AZURE", "AMAZON"	platform	any	Y	N
Assembler image release date Allows: null	re-lease_date	string	Y	N
Assembler image size Allows: null	size	number	Y	N
Last Updated timestamp: readonly	up-dated_at	string	Y	N
Assembler image version Allows: "", null	version	string	Y	N
Defines/retrieves expel.io actor records	cre-ated_by	<i>Actors</i>	N	Y
Defines/retrieves expel.io actor records	up-dated_by	<i>Actors</i>	N	Y

**class** pyexclient.workbench.**Assemblers** (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

Assemblers

Resource type name is **assemblers**.

Example JSON record:

```
{
  'connection_status': 'Never Connected',
  'connection_status_updated_at': '2019-01-15T15:35:00-05:00',
  'created_at': '2019-01-15T15:35:00-05:00',
  'deleted_at': '2019-01-15T15:35:00-05:00',
  'install_code': 'string',
  'lifecycle_status': 'New',
  'lifecycle_status_updated_at': '2019-01-15T15:35:00-05:00',
  'location': 'string',
  'name': 'string',
  'status': 'string',
  'status_updated_at': '2019-01-15T15:35:00-05:00',
  'updated_at': '2019-01-15T15:35:00-05:00',
  'vpn_ip': 'string'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Assembler connection status Restricted to: “Never Connected”, “Connection Lost”, “Connected to Provisioning”, “Connected to Service” Allows: null	connection_status	any	Y	N
Assembler connection status update timestamp: readonly	connection_status_updated_at	string	Y	N
Created timestamp: readonly	created_at	string	Y	N
Deleted At timestamp Allows: null	deleted_at	string	Y	N
Assembler install code Allows: null	install_code	string	Y	N
Assembler life cycle status Restricted to: “New”, “Authorized”, “Transitioning”, “Transitioned”, “Transition Failed”, “Configuring”, “Configuration Failed”, “Active”, “Inactive”, “Deleted” Allows: null	lifecycle_status	any	Y	N
Assembler lifecycle status update timestamp: readonly	lifecycle_status_updated_at	string	Y	N
Location of assembler Allows: “”, null	location	string	Y	N
Name of assembler Allows: “”, null	name	string	Y	N
Assembler status Allows: “”, null: readonly, no-sort, no-filter	status	string	Y	N
Assembler last status update timestamp: readonly	status_updated_at	string	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Assembler VPN ip address Allows: null	vpn_ip	string	Y	N
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Defines/retrieves expel.io organization records	organization	<i>Organizations</i>	N	Y
Security devices	security_devices	<i>SecurityDevices</i>	Y	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y
Vendor alerts	vendor_alerts	<i>VendorAlerts</i>	N	Y

```
class pyexclient.workbench.BaseResourceObject (cls, content=None, api_type=None, conn=None)
```

Bases: object

**count** ()

Return the number of records in a JSON API response. You can get the count for entries returned by filtering, or you can request the count of the total number of resource instances. The total number of resource instances does not require paginating overall entries.

**Returns** The number of records in a JSON API response

**Return type** int

**Examples:**

```
>>> xc = WorkbenchClient('https://workbench.expel.io', username=username,
↳ password=password, mfa_code=mfa_code)
>>> print("Investigation Count: ", xc.investigations.filter_by(customer_
↳ id='1').count())
>>> print("Investigation Count: ", xc.investigations.count())
```

**create** (*\*\*kwargs*)

Create a ResourceInstance object that represents some Json API resource.

**Parameters** *kwargs* (*dict*) – Attributes to set on the new JSON API resource.

**Returns** A ResourceInstance object that represents the JSON API resource type requested by the dev.

**Return type** *ResourceInstance*

**Examples:**

```
>>> xc = WorkbenchClient('https://workbench.expel.io', username=username,
↳ password=password, mfa_code=mfa_code)
>>> i = xc.investigations.create(title='Peter: new investigation 1',
↳ relationship_customer=CUSTOMER_GUID, relationship_assigned_to_
↳ actor=PETER_S)
>>> i.save()
```

**filter\_by** (*\*\*kwargs*)

Issue a JSON API call requesting a JSON API resource is filtered by some set of attributes, id, limit, etc.

**Parameters** *kwargs* (*dict*) – The base JSON API resource type

**Returns** A BaseResourceObject object

**Return type** *BaseResourceObject*

**Examples:**

```
>>> xc = WorkbenchClient('https://workbench.expel.io', username=username,
↳ password=password, mfa_code=mfa_code)
>>> for inv in xc.investigations.filter_by(customer_id='1'):
>>>     print(inv.title)
```

**get** (*\*\*kwargs*)

Request a JSON api resource by id.

**Parameters** *id* (*str*) – The GUID of the resource

**Returns** A BaseResourceObject object

**Return type** *BaseResourceObject*

**Examples:**

```
>>> xc = WorkbenchClient('https://workbench.expel.io', username=username,
↳ password=password, mfa_code=mfa_code)
>>> inv = xc.investigations.get(id=investigation_guid)
>>> print(inv.title)
```

**one\_or\_none()**

Return one record from a JSON API response or None if there were no records.

**Returns** A BaseResourceObject object

**Return type** *BaseResourceObject*

**Examples:**

```
>>> xc = WorkbenchClient('https://workbench.expel.io', username=username,
↳ password=password, mfa_code=mfa_code)
>>> inv = xc.investigations.filter_by(customer_id=CUSTOMER_GUID).one_or_
↳ none()
>>> print(inv.title)
```

**search(\*args, \*\*kwargs)**

Search based on a set of criteria made up of operators and attributes.

**Parameters**

- **args** (*tuple*) – Operators of relationship|limit|include|sort
- **kwargs** (*dict*) – Fields and values to search on

**Returns** A BaseResourceObject object

**Return type** *BaseResourceObject*

**Examples:**

```
>>> # field filter
>>> for inv in xc.investigations.search(customer_id=CUSTOMER_GUID):
>>>     print(inv.title)
```

```
>>> # operator field filter
>>> for inv in xc.investigations.search(customer_id=CUSTOMER_GUID,
↳ created_at=gt("2020-01-01")):
>>>     print(inv.title)
```

```
>>> # relationship field filter
>>> for inv in xc.investigations.search(customer_id=CUSTOMER_GUID,
↳ relationship("investigative_actions.created_at", gt("2020-01-01"))):
>>>     print(inv.title)
```

**class pyexclient.workbench.CommentHistories** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io comment\_history records

Resource type name is **comment\_histories**.

Example JSON record:

```
{'action': 'CREATED', 'created_at': '2019-01-15T15:35:00-05:00', 'value': {}}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Rela-tion-ship
Comment history action Restricted to: “CREATED”, “UPDATED”, “DELETED” Allows: null	action	any	Y	N
Created timestamp: readonly	cre-ated_at	string	Y	N
Comment history details Allows: null: no-sort	value	object	Y	N
Defines/retrieves expel.io comment records	com-ment	<i>Comments</i>	N	Y
Defines/retrieves expel.io actor records	cre-ated_by	<i>Actors</i>	N	Y
Investigations	inves-tiga-tion	<i>Investigations</i>	N	Y

**class** pyexclient.workbench.**Comments** (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io comment records

Resource type name is **comments**.

Example JSON record:

```
{'comment': 'string', 'created_at': '2019-01-15T15:35:00-05:00', 'updated_at':
→ '2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Rela-tion-ship
Comment	comment	string	Y	N
Created timestamp: readonly	created_at	string	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Defines/retrieves expel.io com-ment_history records	com-ment_histories	<i>CommentHistories</i>	N	Y
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Investigations	investigation	<i>Investigations</i>	N	Y
Defines/retrieves expel.io organiza-tion records	organization	<i>Organizations</i>	N	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y

**class** pyexclient.workbench.**Configurations** (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io configuration records

Resource type name is **configurations**.

Example JSON record:

```
{
  'created_at': '2019-01-15T15:35:00-05:00',
  'default_value': 'object',
  'description': 'string',
  'is_override': True,
  'key': 'string',
  'metadata': {},
  'title': 'string',
  'updated_at': '2019-01-15T15:35:00-05:00',
  'validation': {},
  'value': 'object',
  'visibility': 'EXPEL',
  'write_permission_level': 'EXPEL'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Rela-tion-ship
Created timestamp: readonly	created_at	string	Y	N
Default configuration value Allows: null: read-only, no-sort	de-fault_value	any	Y	N
Description of configuration value Allows: "", null: readonly	description	string	Y	N
Configuration value is an override: readonly	is_override	boolean	Y	N
Configuration key: readonly	key	string	Y	N
Configuration metadata Allows: null: readonly, no-sort	metadata	object	Y	N
Title of configuration value Allows: "", null: readonly	title	string	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Configuration value validation Allows: null: readonly, no-sort	validation	object	Y	N
Configuration value Allows: null: no-sort	value	any	Y	N
Configuration visibility Restricted to: "EXPEL", "ORGANIZATION", "SYSTEM"	visibility	any	Y	N
Write permission required Restricted to: "EXPEL", "ORGANIZATION", "SYSTEM"	write_permission_level	any	Y	N
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Defines/retrieves expel.io organization records	organization	<i>Organizations</i>	N	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y

**class** pyexclient.workbench.ContextLabelActionHistories (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io context\_label\_action\_history records

Resource type name is **context\_label\_action\_histories**.

Example JSON record:

```
{'action': 'CREATED', 'action_type': 'ALERT_ON', 'created_at': '2019-01-15T15:35:00-05:00', 'value': {}}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Context label action history Restricted to: "CREATED", "UPDATED", "DELETED" Allows: null	action	any	Y	N
Action type of source parent remediation action Restricted to: "ALERT_ON", "ADD_TO", "SUPPRESS"	action_type	any	Y	N
Created timestamp: readonly	created_at	string	Y	N
Context label action history details (contains important fields that changed) Allows: null: no-sort	value	object	Y	N
Defines/retrieves expel.io context_label records	context_label	<i>ContextLabels</i>	N	Y
Defines/retrieves expel.io context_label_action records	context_label_action	<i>ContextLabelActions</i>	N	Y
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Investigations	investigation	<i>Investigations</i>	N	Y

**class** pyexclient.workbench.**ContextLabelActions** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io context\_label\_action records

Resource type name is **context\_label\_actions**.

Example JSON record:

```
{'action_type': 'ALERT_ON', 'created_at': '2019-01-15T15:35:00-05:00', 'expel_
→severity_threshold': 'CRITICAL', 'expel_signature_id': 'string', 'updated_at':
→'2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
What action to take Restricted to: "ALERT_ON", "ADD_TO", "SUPPRESS"	action_type	any	Y	N
Created timestamp: readonly	created_at	string	Y	N
Restricted to: "CRITICAL", "HIGH", "MEDIUM", "LOW", "TESTING", "TUNING" Allows: null	expel_severity_threshold	any	Y	N
Expel alert signature Allows: "", null	expel_signature_id	string	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Defines/retrieves expel.io context_label records	context_label	<i>ContextLabels</i>	N	Y
Defines/retrieves expel.io context_label_action_history records	context_label_action_histories	<i>ContextLabelActionHistories</i>	N	Y
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Investigations	investigation	<i>Investigations</i>	N	Y
Timeline Entries	time-line_entries	<i>TimelineEntries</i>	N	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y

**class** pyexclient.workbench.ContextLabelHistories (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io context\_label\_history records

Resource type name is **context\_label\_histories**.

Example JSON record:

```
{'action': 'CREATED', 'created_at': '2019-01-15T15:35:00-05:00', 'value': {}}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Context label history action Restricted to: "CREATED", "UPDATED", "DELETED", "ASSIGNED_TAG_CREATED", "ASSIGNED_TAG_DELETED" Allows: null	action	any	Y	N
Created timestamp: readonly	created_at	string	Y	N
Context label history details (contains important fields that changed) Allows: null: no-sort	value	object	Y	N
Defines/retrieves expel.io context_label records	context_label	<i>ContextLabels</i>	N	Y
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y



**class** pyexclient.workbench.**ContextLabelTags** (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io context\_label\_tag records

Resource type name is **context\_label\_tags**.

Example JSON record:

```
{'created_at': '2019-01-15T15:35:00-05:00', 'description': 'string', 'metadata':
→ {}, 'tag': 'string', 'updated_at': '2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Rela-tion-ship
Created timestamp: readonly	created_at	string	Y	N
Description Allows: null, ""	description	string	Y	N
Metadata about the context label tag Allows: null: no-sort	metadata	object	Y	N
Tag	tag	string	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Defines/retrieves expel.io context_label records	context_labels	<i>ContextLabels</i>	N	Y
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Defines/retrieves expel.io organization records	organization	<i>Organizations</i>	N	Y
Remediation action assets	remediation_action_assets	<i>RemediationActionAssets</i>	N	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y

**class** pyexclient.workbench.**ContextLabels** (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io context\_label records

Resource type name is **context\_labels**.

Example JSON record:

```
{
  'created_at': '2019-01-15T15:35:00-05:00',
  'definition': {},
  'description': 'string',
  'ends_at': '2019-01-15T15:35:00-05:00',
  'initial_edit_at': '2019-01-15T15:35:00-05:00',
  'metadata': {},
  'review_status': 'APPROVED',
  'starts_at': '2019-01-15T15:35:00-05:00',
  'title': 'string',
  'updated_at': '2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tributela-relationship	Re-tribution-ship
Created timestamp: readonly	created_at	string	Y	N
Definition: no-sort	definition	object	Y	N
Description Allows: null, ""	description	string	Y	N
Date/Time of when the context_label should end being tested Allows: null	ends_at	string	Y	N
Date/Time of when the drawer was first opened to create the context label Allows: null	initial_edit_at	string	Y	N
Metadata about the context label Allows: null: no-sort	metadata	object	Y	N
Current status of the review process Restricted to: "APPROVED", "REVIEW_REQUESTED", "CHANGES_REQUESTED", "DECLINED" Allows: null	review_status	any	Y	N
Date/Time of when the context_label should start being tested	starts_at	string	Y	N
Title Allows: null, ""	title	string	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Defines/retrieves expel.io context_label_action records	add_to_actions	<i>ContextLabelActions</i>	N	Y
Defines/retrieves expel.io context_label_action records	alert_on_actions	<i>ContextLabelActions</i>	N	Y
Defines/retrieves expel.io context_label_history records	approval_histories	<i>ContextLabelHistories</i>	N	Y
Defines/retrieves expel.io context_label_action_history records	context_label_action_histories	<i>ContextLabelActionHistories</i>	N	Y
Defines/retrieves expel.io context_label_action records	context_label_actions	<i>ContextLabelActions</i>	N	Y
Defines/retrieves expel.io context_label_history records	context_label_histories	<i>ContextLabelHistories</i>	N	Y
Defines/retrieves expel.io context_label_tag records	context_label_tags	<i>ContextLabelTags</i>	N	Y
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Expel alerts	expel_alerts	<i>ExpelAlerts</i>	N	Y
Investigations	investigations	<i>Investigations</i>	N	Y
Defines/retrieves expel.io organization records	organization	<i>Organizations</i>	N	Y
Expel alerts	originating_expel_alerts	<i>ExpelAlerts</i>	N	Y
Defines/retrieves expel.io remediation_action_setting_list_source_history records	remediation_action_setting_list_source_histories	<i>RemediationActionSettingListSourceHistories</i>	N	Y
Defines/retrieves expel.io remediation_action_setting_list_source records	remediation_action_setting_list_sources	<i>RemediationActionSettingListSources</i>	N	Y
Defines/retrieves expel.io remediation_action_setting records	remediation_action_settings	<i>RemediationActionSettings</i>	N	Y
Defines/retrieves expel.io context_label_action records	suppress_actions	<i>ContextLabelActions</i>	N	Y
Timeline Entries	time-	<i>TimelineEntries</i>	N	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y

---

**class** pyexclient.workbench.**Detections** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

How we determine security-relevant data for analyst review

Resource type name is **detections**.

Example JSON record:

```
{
    'author_type': 'string',
    'configuration_slugs': [],
    'created_date': '2019-01-15T15:35:00-05:00',
    'customer_context_tags': [],
    'description': 'string',
    'logic_blob': 'string',
    'name': 'string',
    'priority': 100,
    'severity': 'string',
    'status': 'string',
    'supported_tech': [],
    'tags': [],
    'test_blob': 'string',
    'unique_on': [],
    'unique_within': 'string',
    'updated_at': '2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
The author of this detection Allows: "", null	author_type	string	Y	N
A list of configuration tags used in the logic of this detection: no-filter, no-sort	configuration_slugs	array	Y	N
The time this detection was created Allows: null	created_date	string	Y	N
A list of customer context tags used in the logic of this detection: no-filter, no-sort	customer_context_tags	array	Y	N
What this detection is looking for Allows: "", null	description	string	Y	N
A JSON formatted blob of Josie detection logic Allows: "", null	logic_blob	string	Y	N
The name of the detection Allows: "", null	name	string	Y	N
The order in which Josie applies this detection to security alerts, 0 being a higher priority Allows: null	priority	number	Y	N
The potential impact of detected incidents Allows: "", null	severity	string	Y	N
The production status of this detection Allows: "", null	status	string	Y	N
The technology that this detection supports	supported_tech	array	Y	N
A list of tags which may affect upstream behavior: no-filter, no-sort	tags	array	Y	N
A JSON formatted list of conditions (vendor alerts) that trigger this detection, in turn creating an Expel alert Allows: "", null	test_blob	string	Y	N
A list of Josie expressions that make this alert unique: no-filter, no-sort	unique_on	array	Y	N
The time period in which this alert is unique Allows: "", null	unique_within	string	Y	N
The last time this detection was updated Allows: null	updated_at	string	Y	N
Detection categories	expel_alert_categories	<i>ExpelDetectionCategories</i>	N	Y
Detection categories	expel_response_categories	<i>ExpelDetectionCategories</i>	N	Y
Detection categories	expel_triage_categories	<i>ExpelDetectionCategories</i>	N	Y
Mitre Tactics	mitre_tactics	<i>MitreTactics</i>	N	Y

**class** pyexclient.workbench.**EngagementManagers** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io engagement\_manager records

Resource type name is **engagement\_managers**.

Example JSON record:

```
{'created_at': '2019-01-15T15:35:00-05:00', 'display_name': 'string', 'email':
↳'name@company.com', 'pagerduty_id': 'string', 'phone_number': 'string',
↳'updated_at': '2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Relation-ship
Created timestamp: readonly	created_at	string	Y	N
Display name Allows: "", null	display_name	string	Y	N
Email Allows: null	email	string	Y	N
Pagerduty ID Allows: null	pagerduty_id	string	Y	N
Phone number Allows: null	phone_number	string	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Defines/retrieves expel.io actor records	created_by	Actors	N	Y
Defines/retrieves expel.io organization records	organizations	Organizations	N	Y
Defines/retrieves expel.io actor records	updated_by	Actors	N	Y

**class** pyexclient.workbench.**Entitlements** (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

path to entitlements service

Resource type name is **entitlements**.

Example JSON record:

```
{'entitlement': {'organization_id': 'string', 'service_offerings': [], 'service_
↳offerings_for_hunting': [], 'service_types': []}, 'skus': []}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Relation-ship
Allows: null: readonly	entitlement	object	Y	N
Missing Description	skus	array	Y	N
Defines/retrieves expel.io organization records	organization	Organizations	N	Y

**class** pyexclient.workbench.**ExpelAlertHistories** (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

Expel alert histories

Resource type name is **expel\_alert\_histories**.

Example JSON record:

```
{'action': 'CREATED', 'created_at': '2019-01-15T15:35:00-05:00', 'value': {}}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Expel alert history action Restricted to: “CREATED”, “AS-SIGNED”, “STATUS_CHANGED”, “INVESTIGATING”, “TUNING_CHANGED”, “DELETED”, “VIEWED” Allows: null	action	any	Y	N
Created timestamp: readonly	created_at	string	Y	N
Expel alert history details Allows: null: no-sort	value	object	Y	N
Defines/retrieves expel.io actor records	assigned_to_actor	Actors	N	Y
Defines/retrieves expel.io actor records	created_by	Actors	N	Y
Expel alerts	expel_alert	ExpelAlerts	N	Y
Investigations	investigation	Investigations	N	Y
Defines/retrieves expel.io organization records	organization	Organizations	N	Y

**class** pyexclient.workbench.**ExpelAlertThresholdHistories** (*data*, *conn*, *included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io expel\_alert\_threshold\_history records

Resource type name is **expel\_alert\_threshold\_histories**.

Example JSON record:

```
{'action': 'CREATED', 'created_at': '2019-01-15T15:35:00-05:00', 'value': {}}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Expel alert threshold history action Restricted to: "CREATED", "BREACHED", "ACKNOWLEDGED", "RECOVERED", "DELETED"	action	any	Y	N
Created timestamp: readonly	created_at	string	Y	N
Expel alert threshold history details Allows: null: no-sort	value	object	Y	N
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Defines/retrieves expel.io expel_alert_threshold records	expel_alert_threshold	<i>ExpelAlertThresholds</i>	N	Y

**class** pyexclient.workbench.**ExpelAlertThresholds** (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io expel\_alert\_threshold records

Resource type name is **expel\_alert\_thresholds**.

Example JSON record:

```
{'created_at': '2019-01-15T15:35:00-05:00', 'name': 'string', 'threshold': 100,
  →'updated_at': '2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Created timestamp: readonly	created_at	string	Y	N
Name	name	string	Y	N
Threshold value	threshold	number	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Defines/retrieves expel.io expel_alert_threshold_history records	expel_alert_threshold_histories	<i>ExpelAlertThresholdHistories</i>	N	Y
Defines/retrieves expel.io expel_alert_threshold records	suppressed_by	<i>ExpelAlertThresholds</i>	N	Y
Defines/retrieves expel.io expel_alert_threshold records	suppresses	<i>ExpelAlertThresholds</i>	N	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y

**class** pyexclient.workbench.**ExpelAlerts** (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

Expel alerts

Resource type name is **expel\_alerts**.

Example JSON record:

```
{
  'activity_first_at': '2019-01-15T15:35:00-05:00',
  'activity_last_at': '2019-01-15T15:35:00-05:00',
  'alert_type': 'ENDPOINT',
  'close_comment': 'string',
  'close_reason': 'FALSE_POSITIVE',
  'created_at': '2019-01-15T15:35:00-05:00',
  'cust_disp_alerts_in_critical_incidents_count': 100,
  'cust_disp_alerts_in_incidents_count': 100,
  'cust_disp_alerts_in_investigations_count': 100,
  'cust_disp_closed_alerts_count': 100,
  'cust_disp_disposed_alerts_count': 100,
  'disposition_alerts_in_critical_incidents_count': 100,
  'disposition_alerts_in_incidents_count': 100,
  'disposition_alerts_in_investigations_count': 100,
  'disposition_closed_alerts_count': 100,
  'disposition_disposed_alerts_count': 100,
  'expel_alert_time': '2019-01-15T15:35:00-05:00',
  'expel_alias_name': 'string',
  'expel_message': 'string',
  'expel_name': 'string',
  'expel_severity': 'CRITICAL',
  'expel_signature_id': 'string',
  'expel_version': 'string',
  'git_rule_url': 'https://company.com/',
  'investigative_action_count': 100,
  'ref_event_id': 'string',
  'status': 'string',
  'status_updated_at': '2019-01-15T15:35:00-05:00',
  'tuning_requested': True,
  'updated_at': '2019-01-15T15:35:00-05:00',
  'vendor_alert_count': 100}
```

Below are valid filter by parameters:

Field Description
Allows: null: readonly, no-sort, no-filter
Allows: null: readonly, no-sort, no-filter
Expel alert type Restricted to: "ENDPOINT", "NETWORK", "SIEM", "RULE_ENGINE", "EXTERNAL", "OTHER", "CLOUD", "
Expel alert close comment Allows: "", null
Expel alert close reason Restricted to: "FALSE_POSITIVE", "TRUE_POSITIVE", "OTHER", "ATTACK_FAILED", "POLICY_VIC
Created timestamp: readonly
Allows: null
Allows: null
Allows: null
Allows: null
Allows: null
Allows: null
Allows: null
Allows: null
Allows: null
Allows: null
Allows: null



Field Description
Expel Alert Time first seen time: immutable
Expel alert alias Allows: "", null
Expel alert message Allows: "", null
Expel alert name Allows: "", null
Expel alert severity Restricted to: "CRITICAL", "HIGH", "MEDIUM", "LOW", "TESTING", "TUNING" Allows: null
Expel alert signature Allows: "", null
Expel alert version Allows: "", null
URL to rule definition for alert Allows: "", null
Allows: null: readonly, no-sort, no-filter
Referring event id Allows: null
Expel alert status Restricted to: "OPEN", "IN_PROGRESS", "CLOSED" Allows: null
Status Updated At Allows: null: readonly
tuning requested
Last Updated timestamp: readonly
Allows: null: readonly, no-sort, no-filter
Defines/retrieves expel.io actor records
Defines/retrieves expel.io context_label records
Defines/retrieves expel.io actor records
IP addresses
Vendor alert evidences are extracted from a vendor alert's evidence summary
Expel alert histories
Investigations
Investigative action histories
investigative actions
Defines/retrieves expel.io organization records
Defines/retrieves expel.io context_label records
Phishing submissions
Investigations
Investigations
Expel alerts
IP addresses
Defines/retrieves expel.io actor records
Expel alert histories
Defines/retrieves expel.io actor records
Vendors
Vendor alerts

**class** pyexclient.workbench.**ExpelDetectionCategories** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Detection categories

Resource type name is **expel\_detection\_categories**.

Example JSON record:

```
{'category_type': 'string', 'customer_context_tags': [], 'description': 'string',
↪ 'investigative_questions': [], 'name': 'string', 'number_of_detections': 100}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Which part of the detection & response process this category takes effect in Allows: "", null: no-sort	category_type	string	Y	N
Customer context tags used in this category's detections	customer_context_tags	array	Y	N
The category description Allows: "", null	description	string	Y	N
The questions analysts ask when responding to Expel alerts	investigative_questions	array	Y	N
The name of the category Allows: "", null	name	string	Y	N
The number of detections in this category, based on the plugin_slugs filter	number_of_detections	number	Y	N

**class** pyexclient.workbench.**Files** (*data, conn, included=None*)

Bases: *pyexclient.workbench.FilesResourceInstance*

File

Resource type name is **files**.

Example JSON record:

```
{
    'created_at': '2019-01-15T15:35:00-05:00',
    'expel_file_type': 'string',
    'file_meta': {'investigative_action': {'file_type': 'string'}},
    'filename': 'string',
    'updated_at': '2019-01-15T15:35:00-05:00'
}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Created timestamp: read-only	created_at	string	Y	N
Expel file type Allows: null, ""	expel_file_type	string	Y	N
Metadata about the file Allows: null: no-sort	file_meta	object	Y	N
Filename	filename	string	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Investigations	investigations	<i>Investigations</i>	N	Y
investigative actions	investigative_actions	<i>InvestigativeActions</i>	N	Y
Defines/retrieves expel.io organization records	organization	<i>Organizations</i>	N	Y
Phishing submissions	phishing_submission	<i>PhishingSubmissions</i>	N	Y
Phishing submission attachments	phishing_submission_attachment	<i>PhishingSubmissionAttachments</i>	N	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y

**class** pyexclient.workbench.**FilesResourceInstance** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

**download** (*fd, fmt='json'*)

Download data from an investigative action. This can only be called on InvestigativeAction or Files objects.

#### Parameters

- **fd** (*File bytes object*) – Buffer to write response too.
- **fmt** (*str*) – The format to request the data be returned in.

#### Examples:

```
>>> import json
>>> import pprint
>>> import tempfile
>>> xc = WorkbenchClient('https://workbench.expel.io', username=username,
↳ password=password, mfa_code=mfa_code)
>>> with xc.investigative_actions.get(id=inv_act_id) as ia:
>>>     fd = tempfile.NamedTemporaryFile(delete=False)
>>>     ia.download(fd)
>>>     with open(fd.name, 'r') as fd:
>>>         pprint.pprint(json.loads(fd.read()))
```

**class** pyexclient.workbench.**Findings** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io finding records

Resource type name is **findings**.

Example JSON record:

```
{'created_at': '2019-01-15T15:35:00-05:00', 'rank': 100, 'title': 'string',
↪'updated_at': '2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Relation-ship
Created timestamp: readonly	created_at	string	Y	N
Seed Rank	rank	number	Y	N
Title Allows: "", null	title	string	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y

**class** pyexclient.workbench.**Integrations** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io integration records

Resource type name is **integrations**.

Example JSON record:

```
{
  'account': 'string',
  'created_at': '2019-01-15T15:35:00-05:00',
  'integration_meta': {},
  'integration_type': 'pagerduty',
  'last_tested_at': '2019-01-15T15:35:00-05:00',
  'service_name': 'string',
  'status': 'UNTESTED',
  'updated_at': '2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Service account identifier	account	string	Y	N
Created timestamp: readonly	created_at	string	Y	N
Needed information for integration type Allows: null: no-sort	integration_meta	object	Y	N
Type of integration Restricted to: “pagerduty”, “slack”, “ticketing”, “service_now”, “teams”, “ops_genie”: immutable	integration_type	any	Y	N
Last Successful Test Allows: null: readonly	last_tested_at	string	Y	N
Service display name	service_name	string	Y	N
Integration status Restricted to: “UNTESTED”, “TEST_SUCCESS”, “TEST_FAIL”: readonly	status	any	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Defines/retrieves expel.io organization records	organization	<i>Organizations</i>	N	Y
Organization secrets. Note - these requests must be in the format of <code>/secrets/security_device-&lt;guid&gt;</code>	secret	<i>Secrets</i>	N	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y

**class** pyexclient.workbench.**InvestigationFindingHistories** (*data*, *conn*, *included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io investigation\_finding\_history records

Resource type name is **investigation\_finding\_histories**.

Example JSON record:

```
{'action': 'CREATED', 'created_at': '2019-01-15T15:35:00-05:00', 'updated_at':
→'2019-01-15T15:35:00-05:00', 'value': {}}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Rela-tion-ship
Investigation finding history action Restricted to: "CREATED", "CHANGED", "DELETED" Allows: null	action	any	Y	N
Created timestamp: readonly	cre-ated_at	string	Y	N
Last Updated timestamp: readonly	up-dated_at	string	Y	N
Investigation finding history details Allows: null: no-sort	value	object	Y	N
Defines/retrieves expel.io actor records	cre-ated_by	<i>Actors</i>	N	Y
Investigations	investiga-tion	<i>Investigation</i> s	N	Y
Investigation findings	investiga-tion_finding	<i>Investigation</i> Findings	N	Y
Defines/retrieves expel.io actor records	up-dated_by	<i>Actors</i>	N	Y

**class** pyexclient.workbench.**InvestigationFindings** (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

Investigation findings

Resource type name is **investigation\_findings**.

Example JSON record:

```
{'created_at': '2019-01-15T15:35:00-05:00', 'deleted_at': '2019-01-15T15:35:00-05:00', 'finding': 'string', 'rank': 100, 'title': 'string', 'updated_at': '2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Created timestamp: readonly	created_at	string	Y	N
Deleted At timestamp Allows: null	deleted_at	string	Y	N
Finding Allows: "", null	finding	string	Y	N
Visualization Rank	rank	number	Y	N
Title Allows: "", null	title	string	Y	N
Last Updated timestamp: read-only	updated_at	string	Y	N
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Investigations	investigation	<i>Investigations</i>	N	Y
Defines/retrieves expel.io investigation_finding_history records	investigation_finding_histories	<i>InvestigationFindingHistories</i>	N	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y

**class** pyexclient.workbench.**InvestigationHistories** (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

Investigation histories

Resource type name is **investigation\_histories**.

Example JSON record:

```
{'action': 'CREATED', 'created_at': '2019-01-15T15:35:00-05:00', 'is_incident':
↳True, 'value': {}}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Investigation history action Restricted to: “CREATED”, “ASSIGNED”, “CHANGED”, “CLOSED”, “SUMMARY”, “REOPENED”, “PUBLISHED” Allows: null	action	any	Y	N
Created timestamp: readonly	created_at	string	Y	N
Is Incidence	is_incident	boolean	Y	N
Investigation history details Allows: null: no-sort	value	object	Y	N
Defines/retrieves expel.io actor records	assigned_to_actor	<i>Actors</i>	N	Y
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Investigations	investigation	<i>Investigations</i>	N	Y
Defines/retrieves expel.io organization records	organization	<i>Organizations</i>	N	Y

**class** pyexclient.workbench.**InvestigationResilienceActionHints** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io investigation\_organization\_resilience\_action\_hint records

Resource type name is **investigation\_resilience\_action\_hints**.

Example JSON record:

```
{}
```

Below are valid filter by parameters:

**class** pyexclient.workbench.**InvestigationResilienceActions** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Investigation to resilience actions

Resource type name is **investigation\_resilience\_actions**.

Example JSON record:

```
{'created_at': '2019-01-15T15:35:00-05:00', 'updated_at': '2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Created timestamp: readonly	created_at	string	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Investigations	investigation	<i>Investigations</i>	N	Y
Organization to resilience actions	organization_resilience_action	<i>OrganizationResilienceActions</i>	N	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y

**class** pyexclient.workbench.**Investigations** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Investigations

Resource type name is **investigations**.

Example JSON record:

```
{
    'analyst_severity': 'CRITICAL',
    'attack_lifecycle': 'INITIAL_RECON',
    'attack_timing': 'HISTORICAL',
    'attack_vector': 'DRIVE_BY',
    'close_comment': 'string',
}
```

(continues on next page)



(continued from previous page)

```
'created_at': '2019-01-15T15:35:00-05:00',
'critical_comment': 'string',
'decision': 'FALSE_POSITIVE',
'deleted_at': '2019-01-15T15:35:00-05:00',
'detection_type': 'UNKNOWN',
'has_hunting_status': True,
'initial_attack_vector': 'string',
'is_downgrade': True,
'is_incident': True,
'is_incident_status_updated_at': '2019-01-15T15:35:00-05:00',
'is_surge': True,
'last_published_at': '2019-01-15T15:35:00-05:00',
'last_published_value': 'string',
'lead_description': 'string',
'malware_family': 'string',
'next_steps': 'string',
'open_reason': 'ACCESS_KEYS',
'open_summary': 'string',
'review_requested_at': '2019-01-15T15:35:00-05:00',
'short_link': 'string',
'source_reason': 'HUNTING',
'status_updated_at': '2019-01-15T15:35:00-05:00',
'threat_type': 'TARGETED',
'title': 'string',
'updated_at': '2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description
Analyst Severity Restricted to: “CRITICAL”, “HIGH”, “MEDIUM”, “LOW”, “INFO” Allows: null
Attack Lifecycle Restricted to: “INITIAL_RECON”, “DELIVERY”, “EXPLOITATION”, “INSTALLATION”, “COMMAND_CONT
Attack Timing Restricted to: “HISTORICAL”, “PRESENT” Allows: null
Attack Vector Restricted to: “DRIVE_BY”, “PHISHING”, “PHISHING_LINK”, “PHISHING_ATTACHMENT”, “REV_MEDIA”, “
Close Comment Allows: “”, null
Created timestamp: readonly
Critical Comment Allows: “”, null
Decision Restricted to: “FALSE_POSITIVE”, “TRUE_POSITIVE”, “CLOSED”, “OTHER”, “ATTACK_FAILED”, “POLICY_VIOL
Deleted At timestamp Allows: null
Detection Type Restricted to: “UNKNOWN”, “ENDPOINT”, “SIEM”, “NETWORK”, “EXPEL”, “HUNTING”, “CLOUD”, “PHISH
Meta: readonly, no-sort, no-filter
Initial attack vector Allows: “”, null
Is downgrade
Is Incident
Incident Status timestamp Allows: null: readonly
Is surge
Last Published At Allows: null
Last Published Value Allows: “”, null
Lead Description Allows: null
Malware family Allows: “”, null
Recommended next steps for starting this investigation or handling this incident Allows: “”, null
Open Reason Restricted to: “ACCESS_KEYS”, “EC2”, “S3_BUCKETS”, “OTHER” Allows: null
Reason the investigation/incident was opened Allows: “”, null
Review Requested At Allows: null

Field Description
Investigation short link: readonly
Source Reason Restricted to: "HUNTING", "ORGANIZATION_REPORTED", "DISCOVERY", "PHISHING" Allows: null
Status Updated At Allows: null: readonly
Threat Type Restricted to: "TARGETED", "TARGETED_APT", "TARGETED_RANSOMWARE", "BUSINESS_EMAIL_COMPROMISE"
Title Allows: "", null
Last Updated timestamp: readonly
Defines/retrieves expel.io actor records
Defines/retrieves expel.io comment_history records
Defines/retrieves expel.io comment records
Defines/retrieves expel.io context_label_action_history records
Defines/retrieves expel.io context_label_action records
Defines/retrieves expel.io context_label records
Defines/retrieves expel.io actor records
IP addresses
Vendor alert evidences are extracted from a vendor alert's evidence summary
Expel alert histories
Expel alerts
File
Defines/retrieves expel.io finding records
Defines/retrieves expel.io investigation_finding_history records
Investigation histories
Investigation to resilience actions
Investigative action histories
investigative actions
IP addresses
Defines/retrieves expel.io actor records
Expel alerts
investigative actions
Defines/retrieves expel.io organization records
Organization to resilience actions
Organization to resilience actions
Investigations
Remediation action asset histories
Remediation action assets
Remediation action histories
Remediation actions
Defines/retrieves expel.io actor records
IP addresses
Defines/retrieves expel.io actor records
Timeline Entries
Defines/retrieves expel.io actor records

```
class pyexclient.workbench.InvestigativeActionHistories (data, conn, included=None) in-
```

```
    Bases: pyexclient.workbench.ResourceInstance
```

```
    Investigative action histories
```

```
    Resource type name is investigative_action_histories.
```

```
    Example JSON record:
```

```
{'action': 'CREATED', 'created_at': '2019-01-15T15:35:00-05:00', 'deleted_at':
→'2019-01-15T15:35:00-05:00', 'value': {}}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Re-lation-ship
Investigative action history action Restricted to: “CREATED”, “ASSIGNED”, “CLOSED”, “ACKNOWLEDGED” Allows: null	action	any	Y	N
Created timestamp: readonly	cre-ated_at	string	Y	N
Deleted At timestamp Allows: null	deleted_at	string	Y	N
Investigative action history details Allows: null: no-sort	value	object	Y	N
Defines/retrieves expel.io actor records	as-signed_to_actor	<i>Actors</i>	N	Y
Defines/retrieves expel.io actor records	cre-ated_by	<i>Actors</i>	N	Y
Expel alerts	ex-pel_alert	<i>ExpelAlerts</i>	N	Y
Investigations	investi-gation	<i>Investigations</i>	N	Y
investigative actions	in-vestiga-tive_action	<i>InvestigativeActions</i>	N	Y

**class** pyexclient.workbench.**InvestigativeActions** (data, conn, included=None)

Bases: *pyexclient.workbench.InvestigativeActionsResourceInstance*

investigative actions

Resource type name is **investigative\_actions**.

Example JSON record:

```
{
  'action_type': 'TASKABILITY',
  'activity_authorized': True,
  'activity_verified_by': 'string',
  'capability_name': 'string',
  'close_reason': 'string',
  'content_driven_results': {},
  'created_at': '2019-01-15T15:35:00-05:00',
  'deleted_at': '2019-01-15T15:35:00-05:00',
  'downgrade_reason': 'FALSE_POSITIVE',
  'files_count': 100,
  'input_args': {},
  'instructions': 'string',
  'reason': 'string',
  'result_byte_size': 100,
  'result_task_id': 'object',
  'results': 'string',
```

(continues on next page)

(continued from previous page)

```
'robot_action': True,
'status': 'RUNNING',
'status_updated_at': '2019-01-15T15:35:00-05:00',
'taskability_action_id': 'string',
'tasking_error': {},
'title': 'string',
'updated_at': '2019-01-15T15:35:00-05:00',
'workflow_job_id': 'string',
'workflow_name': 'string'}
```

Below are valid filter by parameters:

Field Description
Investigative Action Type Restricted to: "TASKABILITY", "HUNTING", "MANUAL", "RESEARCH", "PIVOT", "QUICK_UPLOAD"
Verify Investigative action is authorized Allows: null
Verify Investigative action verified by Allows: null
Capability name Allows: "", null
Close Reason Allows: null
Content driven results Allows: "", null: no-sort
Created timestamp: readonly
Deleted At timestamp Allows: null
Downgrade reason Restricted to: "FALSE_POSITIVE", "ATTACK_FAILED", "POLICY_VIOLATION", "ACTIVITY_BLOCKED"
Downgrade reason: readonly
Task input arguments Allows: null: no-sort
Instructions Allows: "", null
Reason
Result byte size: readonly
Result task id Allows: null: readonly
Results/Analysis Allows: "", null
Investigative action created by robot action: readonly
Status Restricted to: "RUNNING", "FAILED", "READY_FOR_ANALYSIS", "CLOSED", "COMPLETED"
Status Updated At Allows: null: readonly
Taskability action id Allows: "", null
Taskabilities error Allows: "", null: no-sort
Title
Last Updated timestamp: readonly
Workflow job id Allows: "", null
Workflow name Allows: "", null
Defines/retrieves expel.io actor records
Defines/retrieves expel.io actor records
Defines/retrieves expel.io actor records
investigative actions
investigative actions
Expel alerts
File
Investigations
Investigative action histories
Security devices
Defines/retrieves expel.io actor records

**class** pyexclient.workbench.**InvestigativeActionsResourceInstance** (*data, conn, included=None*)

Bases: *pyexclient.workbench.FilesResourceInstance*

**upload** (*filename, fbytes, expel\_file\_type=None, file\_meta=None*)

Upload data associated with an investigative action. Can only be called on InvestigativeAction objects.

**Parameters**

- **filename** (*str*) – Filename, this shows up in Workbench.
- **fbytes** (*bytes*) – A bytes string representing raw bytes to upload

**Examples:**

```
>>> xc = WorkbenchClient('https://workbench.expel.io', username=username,
↳ password=password, mfa_code=mfa_code)
>>> with xc.investigative_actions.get(id=inv_act_id) as ia:
>>>     ia.upload('test.txt', b'hello world')
```

**class** pyexclient.workbench.**IpAddresses** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

IP addresses

Resource type name is **ip\_addresses**.

Example JSON record:

```
{'address': 'string', 'created_at': '2019-01-15T15:35:00-05:00', 'updated_at':
↳ '2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Relation-ship
IP Address: readonly	address	string	Y	N
Created timestamp: readonly	created_at	string	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Expel alerts	destina-tion_expel_alerts	<i>ExpelAlerts</i>	N	Y
Investigations	destina-tion_investigations	<i>Investigation</i>	N	Y
Investigations	investigations	<i>Investigation</i>	N	Y
Expel alerts	source_expel_alerts	<i>ExpelAlerts</i>	N	Y
Investigations	source_investigations	<i>Investigation</i>	N	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y
Vendor alerts	vendor_alerts	<i>VendorAlerts</i>	N	Y

**class** pyexclient.workbench.**JsonApiRelationship** (*relationships: dict = None*)

Bases: object

The object acts a helper to handle JSON API relationships. The object is just a dummy that allows for setting / getting attributes that are extracted from the relationship part of the JSON API response. Additionally, the object will allow for conversion to a JSON API compliant relationship block to include in a request.

**to\_relationship()**

Generate a JSON API compliant relationship section.

**Returns** A dict that is JSON API compliant relationship section.

**Return type** dict

**class** pyexclient.workbench.MitreTactics (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

Mitre Tactics

Resource type name is **mitre\_tactics**.

Example JSON record:

```
{'description': 'string', 'name': 'string', 'number_of_expel_detections': 100,
↪'number_of_vendor_detections': 100}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Rela-tion-ship
The tactic description Allows: "", null	description	string	Y	N
The name of the tactic Allows: "", null	name	string	Y	N
The number of Expel-based detections in this category, based on the plugin_slugs filter	num-ber_of_expel_detections	num-ber_of_expel_detections	Y	N
The number of vendor-based detections in this category, based on the plugin_slugs filter	num-ber_of_vendor_detections	num-ber_of_vendor_detections	Y	N

**class** pyexclient.workbench.NistCategories (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io nist\_category records

Resource type name is **nist\_categories**.

Example JSON record:

```
{'created_at': '2019-01-15T15:35:00-05:00', 'function_type': 'IDENTIFY',
↪'identifier': 'string', 'name': 'string', 'updated_at': '2019-01-15T15:35:00-
↪05:00' }
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Rela-tion-ship
Created timestamp: readonly	cre-ated_at	string	Y	N
Actor type Restricted to: “IDENTIFY”, “PRO-TECT”, “DETECT”, “RECOVER”, “RESPOND”	func-tion_type	any	Y	N
Nist category abbreviated identifier	identifier	string	Y	N
Nist category name	name	string	Y	N
Last Updated timestamp: readonly	up-dated_at	string	Y	N
Defines/retrieves expel.io actor records	cre-ated_by	<i>Actors</i>	N	Y
Defines/retrieves expel.io nist_subcategory records	nist_subcategories	<i>NistCategories</i>	N	Y
Defines/retrieves expel.io actor records	up-dated_by	<i>Actors</i>	N	Y

**class** pyexclient.workbench.NistSubcategories (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io nist\_subcategory records

Resource type name is **nist\_subcategories**.

Example JSON record:

```
{'created_at': '2019-01-15T15:35:00-05:00', 'identifier': 'string', 'name':
→'string', 'updated_at': '2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Rela-tion-ship
Created timestamp: readonly	created_at	string	Y	N
Nist subcategory abbreviated identifier	identifier	string	Y	N
Nist subcategory title Allows: “”, null	name	string	Y	N
Last Updated timestamp: read-only	updated_at	string	Y	N
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Defines/retrieves expel.io nist_category records	nist_category	<i>NistCategories</i>	N	Y
Latest NIST subcategory scores	nist_subcategory_scores	<i>NistSubcategoryScores</i>	N	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y

**class** pyexclient.workbench.NistSubcategoryScoreHistories (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

NIST Subcategory Score History

Resource type name is **nist\_subcategory\_score\_histories**.

Example JSON record:

```
{'action': 'SCORE_UPDATED', 'actual_score': 100, 'assessment_date': '2019-01-15T15:35:00-05:00', 'created_at': '2019-01-15T15:35:00-05:00', 'target_score': 100}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
NIST subcategory score history action Restricted to: "SCORE_UPDATED", "COMMENT_UPDATED", "PRIORITY_UPDATED", "IMPORT"	action	any	Y	N
Organization actual score for this nist subcategory	actual_score	number	Y	N
Recorded date of the score assessment (Note: Dates with times will be truncated to the day. Warning: Dates times and timezones will be converted to UTC before they are truncated. Providing non-UTC timezones is not recommended.): immutable	assessment_date	string	Y	N
Created timestamp: readonly	created_at	string	Y	N
Organization target score for this nist subcategory	target_score	number	Y	N
Defines/retrieves expel.io actor records	created_by	Actors	N	Y
Latest NIST subcategory scores	nist_subcategory_scores	SubcategoryScores	Y	Y

**class** pyexclient.workbench.NistSubcategoryScores (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

Latest NIST subcategory scores

Resource type name is **nist\_subcategory\_scores**.

Example JSON record:

```
{
  'actual_score': 100,
  'assessment_date': '2019-01-15T15:35:00-05:00',
  'category_identifier': 'string',
  'category_name': 'string',
  'comment': 'string',
  'created_at': '2019-01-15T15:35:00-05:00',
  'function_type': 'string',
  'is_priority': True,
  'subcategory_identifier': 'string',
  'subcategory_name': 'string',
  'target_score': 100,
  'updated_at': '2019-01-15T15:35:00-05:00'
}
```

Below are valid filter by parameters:



Field Description	Field Name	Field Type	At-tributela-tion-ship	Re-tion-ship
Organization actual score for this nist subcategory Allows: null	ac-tual_score	number	Y	N
Recorded date of the score assessment (Note: Dates with times will be truncated to the day. Warning: Dates times and timezones will be converted to UTC before they are truncated. Providing non-UTC timezones is not recommended.) Allows: null: immutable	assess-ment_date	string	Y	N
Allows: "", null: readonly, csv_ignore, no-sort, no-filter	cate-gory_identifier	string	Y	N
Allows: "", null: readonly, csv_ignore, no-sort, no-filter	cate-gory_name	string	Y	N
Organization comment for this nist subcategory Allows: "", null	com-ment	string	Y	N
Created timestamp: readonly	cre-ated_at	string	Y	N
Allows: "", null: readonly, csv_ignore, no-sort, no-filter	func-tion_type	string	Y	N
Organization nist subcategory is a priority	is_priority	boolean	Y	N
Allows: "", null: immutable, no-sort, no-filter	sub-cate-gory_identifier	string	Y	N
Allows: "", null: readonly, csv_ignore, no-sort, no-filter	sub-cate-gory_name	string	Y	N
Organization target score for this nist subcategory Allows: null	tar-get_score	number	Y	N
Last Updated timestamp: readonly	up-dated_at	string	Y	N
Defines/retrieves expel.io actor records	cre-ated_by	<i>Actors</i>	N	Y
Defines/retrieves expel.io nist_subcategory records	nist_subcategory	<i>NISTSubcategories</i>	N	Y
NIST Subcategory Score History	nist_subcategory_score_history	<i>NISTSubcategoryScoreHistories</i>	N	Y
Defines/retrieves expel.io organization records	organi-zation	<i>Organizations</i>	N	Y
Defines/retrieves expel.io actor records	up-dated_by	<i>Actors</i>	N	Y

**class** pyexclient.workbench.**NotificationPreferences** (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

User Notification Preferences

Resource type name is **notification\_preferences**.

Example JSON record:

```
{'preferences': []}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Missing Description	preferences	array	Y	N
Defines/retrieves expel.io actor records	actor	<i>Actors</i>	N	Y

**class** pyexclient.workbench.**OrganizationResilienceActionGroups** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io organization\_resilience\_action\_group records

Resource type name is **organization\_resilience\_action\_groups**.

Example JSON record:

```
{'category': 'DISRUPT_ATTACKERS', 'created_at': '2019-01-15T15:35:00-05:00',
  →'title': 'string', 'updated_at': '2019-01-15T15:35:00-05:00', 'visible': True}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Organization Resilience Group Category Restricted to: “DISRUPT_ATTACKERS”, “ENABLE_DEFENDERS”	category	any	Y	N
Created timestamp: readonly	created_at	string	Y	N
Group title	title	string	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Visible	visible	boolean	Y	N
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Defines/retrieves expel.io organization records	organization	<i>Organizations</i>	N	Y
Organization to resilience actions	organization_resilience_action_group_actions	<i>OrganizationResilienceActions</i>	N	Y
Defines/retrieves expel.io resilience_action_group records	source_resilience_action_group	<i>ResilienceActionGroups</i>	N	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y

**class** pyexclient.workbench.**OrganizationResilienceActions** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Organization to resilience actions

Resource type name is **organization\_resilience\_actions**.

Example JSON record:

```
{
  'category': 'DISRUPT_ATTACKERS',
  'comment': 'string',
  'created_at': '2019-01-15T15:35:00-05:00',
```

(continues on next page)

(continued from previous page)

```
'details': 'string',
'impact': 'LOW',
'status': 'TOP_PRIORITY',
'title': 'string',
'updated_at': '2019-01-15T15:35:00-05:00',
'visible': True}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribut	Re-lation-ship
Category Restricted to: “DISRUPT_ATTACKERS”, “ENABLE_DEFENDERS” Allows: null	category	any	Y	N
Comment Allows: “”, null	comment	string	Y	N
Created timestamp: readonly	created_at	string	Y	N
Details	details	string	Y	N
Impact Restricted to: “LOW”, “MEDIUM”, “HIGH”	impact	any	Y	N
Status Restricted to: “TOP_PRIORITY”, “IN_PROGRESS”, “WONT_DO”, “COMPLETED”	status	any	Y	N
Title	title	string	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Visible	visible	boolean	Y	N
Defines/retrieves expel.io actor records	as-signed_to_actor	<i>Actors</i>	N	Y
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Investigations	investiga-tion_hints	<i>Investigations</i>	N	Y
Investigation to resilience actions	investiga-tion_resilience_actions	<i>InvestigationResilienceActions</i>	N	Y
Investigations	investigations	<i>Investigations</i>	N	Y
Defines/retrieves expel.io organization records	organization	<i>Organizations</i>	N	Y
Defines/retrieves expel.io organiza-tion_resilience_action_group records	organiza-tion_resilience_action_group	<i>OrganizationResilienceActionGroups</i>	N	Y
Resilience actions	source_resilience_action	<i>ResilienceActions</i>	N	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y

**class** pyexclient.workbench.**OrganizationStatuses** (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

Organization status

Resource type name is **organization\_statuses**.

Example JSON record:

```
{'created_at': '2019-01-15T15:35:00-05:00', 'enabled_login_types': [],
 →'restrictions': [], 'updated_at': '2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Relation-ship
Meta: readonly	created_at	string	Y	N
Missing Description	en-abled_login_types	array	Y	N
Missing Description	restrictions	array	Y	N
Meta: readonly	updated_at	string	Y	N
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Defines/retrieves expel.io organization records	organization	<i>Organization</i>	N	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y

**class** pyexclient.workbench.**Organizations** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io organization records

Resource type name is **organizations**.

Example JSON record:

```
{
  'address_1': 'string',
  'address_2': 'string',
  'city': 'string',
  'country_code': 'string',
  'created_at': '2019-01-15T15:35:00-05:00',
  'deleted_at': '2019-01-15T15:35:00-05:00',
  'hq_city': 'string',
  'hq_utc_offset': 'string',
  'industry': 'string',
  'is_surge': True,
  'name': 'string',
  'nodes_count': 100,
  'o365_tos_id': 'string',
  'postal_code': 'string',
  'region': 'string',
  'service_renewal_at': '2019-01-15T15:35:00-05:00',
  'service_start_at': '2019-01-15T15:35:00-05:00',
  'short_name': 'EXP',
  'updated_at': '2019-01-15T15:35:00-05:00',
  'users_count': 100}
```

Below are valid filter by parameters:

Field Description	Field Name
Address 1 Allows: "", null	address_1
Address 2 Allows: "", null	address_2

Table 4 – continued from previous page

Field Description	Field Name
City Allows: "", null	city
Country Code Allows: null	country_code
Created timestamp: readonly	created_at
Deleted At timestamp Allows: null	deleted_at
The city where the organization's headquarters is located Allows: "", null	hq_city
Allows: "", null	hq_utc_offset
The organization's primary industry Allows: "", null	industry
Is surge	is_surge
The organization's operating name	name
Number of nodes covered for this organization Allows: null	nodes_count
o365 Terms of Service identifier (e.g. hubspot id, etc.) Allows: null	o365_tos_id
Postal Code Allows: null	postal_code
State/Province/Region Allows: "", null	region
Organization service renewal date Allows: null	service_renewal_at
Organization service start date Allows: null	service_start_at
Organization short name Allows: null	short_name
Last Updated timestamp: readonly	updated_at
Number of users covered for this organization Allows: null	users_count
Defines/retrieves expel.io actor records	actor
Defines/retrieves expel.io actor records	alert_assignables
investigative actions	analysis_assigned_invest
Defines/retrieves expel.io api_key records. These can only be created by a user and require an OTP token.	api_keys
Assemblers	assemblers
Expel alerts	assigned_expel_alerts
Investigations	assigned_investigations
investigative actions	assigned_investigative_a
Organization to resilience actions	assigned_organization_re
Organization to resilience actions	assigned_organization_re
Remediation actions	assigned_remediation_ac
Defines/retrieves expel.io remediation_action_setting records	automate_opt_in_for_re
Defines/retrieves expel.io comment records	comments
Defines/retrieves expel.io configuration records	configurations
Defines/retrieves expel.io context_label_tag records	context_label_tags
Defines/retrieves expel.io context_label records	context_labels
Defines/retrieves expel.io actor records	created_by
Defines/retrieves expel.io engagement_manager records	engagement_manager
path to entitlements service	entitlement
Expel alert histories	expel_alert_histories
Expel alerts	expel_alerts
File	files
Defines/retrieves expel.io integration records	integrations
Defines/retrieves expel.io actor records	investigation_assignables
Investigation histories	investigation_histories
Investigations	investigations
Latest NIST subcategory scores	nist_subcategory_scores
User Notification Preferences	notification_preferences
Defines/retrieves expel.io organization_resilience_action_group records	organization_resilience_a
Organization to resilience actions	organization_resilience_a
Organization status	organization_status

Table 4 – continued from previous page

Field Description	Field Name
Defines/retrieves expel.io user_account_role records	organization_user_account_role
Defines/retrieves expel.io remediation_action_setting_history records	remediation_action_setting_history
Defines/retrieves expel.io remediation_action_setting records	remediation_action_setting
SAML Identity Providers	saml_identity_provider
Security devices	security_devices
Defines/retrieves expel.io actor records	updated_by
User accounts	user_accounts
User accounts	user_accounts_with_role
Vendor alerts	vendor_alerts

```
class pyexclient.workbench.PhishingSubmissionAttachments(data, conn, included=None)
```

Bases: `pyexclient.workbench.ResourceInstance`

Phishing submission attachments

Resource type name is **phishing\_submission\_attachments**.

Example JSON record:

```
{'created_at': '2019-01-15T15:35:00-05:00', 'file_md5': 'string', 'file_mime': 'string', 'file_name': 'string', 'file_sha256': 'string'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Created timestamp: readonly	created_at	string	Y	N
File md5 hash	file_md5	string	Y	N
File mime type	file_mime	string	Y	N
File name	file_name	string	Y	N
File sha256 hash	file_sha256	string	Y	N
File	attachment_file	<i>Files</i>	N	Y
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Phishing submissions	phishing_submission	<i>PhishingSubmissions</i>	N	Y

```
class pyexclient.workbench.PhishingSubmissionDomains(data, conn, included=None)
```

Bases: `pyexclient.workbench.ResourceInstance`

Phishing submission domains

Resource type name is **phishing\_submission\_domains**.

Example JSON record:

```
{'created_at': '2019-01-15T15:35:00-05:00', 'value': 'string'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Relation-ship
Created timestamp: readonly	created_at	string	Y	N
Value	value	string	Y	N
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Phishing submissions	phish-ing_submission	<i>PhishingSubmissions</i>	N	Y

**class** pyexclient.workbench.**PhishingSubmissionHeaders** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Phishing submission headers

Resource type name is **phishing\_submission\_headers**.

Example JSON record:

```
{'created_at': '2019-01-15T15:35:00-05:00', 'name': 'string', 'value': 'string'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Relation-ship
Created timestamp: readonly	created_at	string	Y	N
Name	name	string	Y	N
Value	value	string	Y	N
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Phishing submissions	phish-ing_submission	<i>PhishingSubmissions</i>	N	Y

**class** pyexclient.workbench.**PhishingSubmissionUrls** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Phishing submission URLs

Resource type name is **phishing\_submission\_urls**.

Example JSON record:

```
{'contains_sub_elements': True, 'created_at': '2019-01-15T15:35:00-05:00', 'link_↵text': 'string', 'tags': 'https://company.com/', 'url_type': 'https://company.↵com/', 'value': 'string'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Relation-ship
Was sonar prediction correct	con-tains_sub_elements	boolean	Y	N
Created timestamp: readonly	created_at	string	Y	N
Link text value	link_text	string	Y	N
Interesting URL tags: no-sort	tags	string	Y	N
URL type	url_type	string	Y	N
Value	value	string	Y	N
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Phishing submissions	phish-ing_submission	<i>PhishingSubmissions</i>	N	Y

**class** pyexclient.workbench.**PhishingSubmissions** (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

Phishing submissions

Resource type name is **phishing\_submissions**.

Example JSON record:

```
{
    'add_to_at': '2019-01-15T15:35:00-05:00',
    'arthurai_inference_id': 'string',
    'automated_action_type': 'string',
    'created_at': '2019-01-15T15:35:00-05:00',
    'email_type': 'name@company.com',
    'ingest_source': 'string',
    'msg_id': 'string',
    'received_at': '2019-01-15T15:35:00-05:00',
    'reported_at': '2019-01-15T15:35:00-05:00',
    'return_path': 'string',
    'sender': 'string',
    'sender_domain': 'string',
    'subject': 'string',
    'submitted_by': 'string',
    'triaged_at': '2019-01-15T15:35:00-05:00',
    'updated_at': '2019-01-15T15:35:00-05:00',
    'was_sonar_correct': True}
```

Below are valid filter by parameters:



Field Description	Field Name	Field Type	Attribute	Relationship
Added at Allows: null	add_to_at	string	Y	N
ArthurAI Inference ID Allows: null	arthu- rai_inference_id	string	Y	N
Automated action type Allows: "", null	auto- mated_action_type	string	Y	N
Created timestamp: readonly	created_at	string	Y	N
Email type Allows: "", null	email_type	string	Y	N
Ingest source Allows: ""	ingest_source	string	Y	N
Message ID	msg_id	string	Y	N
Received at	received_at	string	Y	N
Reported at	reported_at	string	Y	N
Return path Allows: ""	return_path	string	Y	N
Sender	sender	string	Y	N
Sender domain	sender_domain	string	Y	N
Subject Allows: ""	subject	string	Y	N
Submitted by	submitted_by	string	Y	N
Triaged at Allows: null	triaged_at	string	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Was sonar prediction correct	was_sonar_correct	boolean	Y	N
File	analysis_email_file	<i>Files</i>	N	Y
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Expel alerts	expel_alert	<i>ExpelAlerts</i>	N	Y
File	initial_email_file	<i>Files</i>	N	Y
Phishing submission attachments	phish- ing_submission_attachments	<i>PhishingSubmissionAttachments</i>	N	Y
Phishing submission domains	phish- ing_submission_domains	<i>PhishingSubmissionDomains</i>	N	Y
Phishing submission headers	phish- ing_submission_headers	<i>PhishingSubmissionHeaders</i>	N	Y
Phishing submission URLs	phish- ing_submission_urls	<i>PhishingSubmissionURLs</i>	N	Y
File	raw_body_file	<i>Files</i>	N	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y

```
class pyexclient.workbench.RemediationActionAssetHistories(data, conn, included=None)
```

Bases: *pyexclient.workbench.ResourceInstance*

Remediation action asset histories

Resource type name is **remediation\_action\_asset\_histories**.

Example JSON record:

```
{
    'action': 'CREATED',
    'action_type': 'BLOCK_COMMAND_AND_CONTROL_COMMUNICATIONS',
    'asset_type': 'ACCOUNT',
    'automate_revert': True,
    'automate_status': 'NEW',
    'created_at': '2019-01-15T15:35:00-05:00',
    'status': 'OPEN',
    'value': {}}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Remediation action asset history action Restricted to: "CREATED", "COMPLETED", "REOPENED", "UPDATED", "DELETED" Allows: null	action	any	Y	N
Action type of associated parent remediation action Restricted to: "BLOCK_COMMAND_AND_CONTROL_COMMUNICATIONS", "BLOCK_KNOWN_BAD_HASHES", "CONTAIN_HOSTS", "CONTAIN_INFECTED_REMOVABLE_MEDIA", "DELETE_MALICIOUS_FILES", "DISABLE_AND_MODIFY_AWS_ACCESS_KEYS", "MITIGATE_VULNERABILITY", "OTHER_REMEDIATION", "REMOVE_AND_BLOCK_EMAIL_FORWARDING_ADDRESS", "REMOVE_COMPROMISED_SYSTEMS_FROM_NETWORK_OTHER", "REMOVE_COMPROMISED_SYSTEMS_FROM_NETWORK_AWS", "REMOVE_INBOX_RULES_FOR_KNOWN_COMPROMISED_ACCOUNTS", "RESET_CREDENTIALS_OTHER", "RESET_CREDENTIALS_AWS", "RESET_CREDENTIALS_O365" Allows: null	action_type	any	Y	N
Remediation asset type history Restricted to: "ACCOUNT", "ACCESS_KEY", "DESCRIPTION", "DEVICE", "DOMAIN_NAME", "EMAIL", "FILE", "HASH", "HOST", "INBOX_RULE_NAME", "IP_ADDRESS" Allows: null	asset_type	any	Y	N
Automated remediation asset revert history Allows: null	automate_revert	boolean	Y	N
Automated remediation asset status history Restricted to: "NEW", "CREATED", "STARTING", "VERIFYING", "COMPLETED", "FAILED" Allows: null	automate_status	any	Y	N
Created timestamp: readonly	created_at	string	Y	N
Remediation asset status history Restricted to: "OPEN", "COMPLETED" Allows: null	status	any	Y	N
Remediation action asset history details Allows: null: no-sort	value	object	Y	N
Defines/retrieves expel.io actor records	created_by	Action	N	Y
Investigations	investigation	Investigations	N	Y
Remediation action assets	remediation_action_asset	RemediationActionAssets	N	Y

**class** pyexclient.workbench.**RemediationActionAssets** (data, conn, included=None)  
 Bases: *pyexclient.workbench.ResourceInstance*

Remediation action assets

Resource type name is **remediation\_action\_assets**.

Example JSON record:

```
{
    'asset_type': 'ACCOUNT',
    'automate_revert': True,
    'automate_status': 'NEW',
    'automate_status_updated_at': '2019-01-15T15:35:00-05:00',
    'automate_task_error': {},
    'automate_task_id': 'string',
    'automate_task_result_id': 'object',
    'category': 'AFFECTED_ACCOUNT',
    'created_at': '2019-01-15T15:35:00-05:00',
    'status': 'OPEN',
    'updated_at': '2019-01-15T15:35:00-05:00',
    'value': 'object'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tributela-tion-ship	Re-tributela-tion-ship
Remediation asset type Restricted to: "ACCOUNT", "ACCESS_KEY", "DESCRIPTION", "DEVICE", "DOMAIN_NAME", "EMAIL", "FILE", "HASH", "HOST", "INBOX_RULE_NAME", "IP_ADDRESS"	as-set_type	any	Y	N
Start automated remediation revert process	auto-mate_revert	boolean	Y	N
Automated remediation status for this asset Restricted to: "NEW", "CREATED", "STARTING", "VERIFYING", "COMPLETED", "FAILED" Allows: null: read-only	auto-mate_status	any	Y	N
Automated remediation status last updated at Allows: null: read-only	auto-mate_status_updated_at	string	Y	N
Automated remediation task error Allows: "", null: read-only, no-sort	auto-mate_task_error	object	Y	N
Automated remediation task id Allows: "", null: read-only	auto-mate_task_id	string	Y	N
Automated remediation task result id Allows: null: read-only	auto-mate_task_result_id	any	Y	N
Remediation asset category Restricted to: "AFFECTED_ACCOUNT", "COMPROMISED_ACCOUNT", "FORWARDING_ADDRESS" Allows: null	category	any	Y	N
Created timestamp: read-only	cre-ated_at	string	Y	N
Asset status Restricted to: "OPEN", "COMPLETED"	status	any	Y	N
Last Updated timestamp: read-only	up-dated_at	string	Y	N
Remediation asset value: allowStringOperators, no-sort	value	alternatives	Y	N
Defines/retrieves expel.io context_label_tag records	con-text_label_tags	<i>ContextLabelTags</i>	N	Y
Defines/retrieves expel.io actor records	cre-ated_by	<i>Actors</i>	N	Y
Remediation actions	remedia-tion_action	<i>RemediationActions</i>	N	Y
Remediation action asset histories	remedia-tion_action_asset_histories	<i>RemediationActionAssetHistories</i>	N	Y
Defines/retrieves expel.io actor records	up-dated_by	<i>Actors</i>	N	Y

**class** pyexclient.workbench.**RemediationActionHistories** (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

Remediation action histories

Resource type name is **remediation\_action\_histories**.

Example JSON record:

```
{'action': 'CREATED', 'action_type': 'BLOCK_COMMAND_AND_CONTROL_COMMUNICATIONS',
↪ 'can_automate': True, 'created_at': '2019-01-15T15:35:00-05:00', 'value': {}}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Remediation action history action Restricted to: "CREATED", "ASSIGNED", "COMPLETED", "CLOSED", "UPDATED", "DELETED" Allows: null	action	any	Y	N
Action type of source parent remediation action Restricted to: "BLOCK_COMMAND_AND_CONTROL_COMMUNICATIONS", "BLOCK_KNOWN_BAD_HASHES", "CONTAIN_HOSTS", "CONTAIN_INFECTED_REMOVABLE_MEDIA", "DELETE_MALICIOUS_FILES", "DISABLE_AND_MODIFY_AWS_ACCESS_KEYS", "MITIGATE_VULNERABILITY", "OTHER_REMEDIATION", "REMOVE_AND_BLOCK_EMAIL_FORWARDING_ADDRESS", "REMOVE_COMPROMISED_SYSTEMS_FROM_NETWORK_OTHER", "REMOVE_COMPROMISED_SYSTEMS_FROM_NETWORK_AWS", "REMOVE_INBOX_RULES_FOR_KNOWN_COMPROMISED_ACCOUNTS", "RESET_CREDENTIALS_OTHER", "RESET_CREDENTIALS_AWS", "RESET_CREDENTIALS_O365" Allows: null	action_type	any	Y	N
Remediation action can be automated history	can_automated	boolean	Y	N
Created timestamp: readonly	created_at	string	Y	N
Remediation action history details Allows: null: no-sort	value	object	Y	N
Defines/retrieves expel.io actor records	assigned_to_actor	Action	N	Y
Defines/retrieves expel.io actor records	created_by	Action	N	Y
Investigations	investigation	Investigations	N	Y
Remediation actions	remediation_action	RemediationActions	N	Y
Security devices	security_device	SecurityDevices	N	Y

**class** pyexclient.workbench.**RemediationActionSettingHistories** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io remediation\_action\_setting\_history records

Resource type name is **remediation\_action\_setting\_histories**.

Example JSON record:

```
{'action': 'CREATED', 'action_type': 'BLOCK_COMMAND_AND_CONTROL_COMMUNICATIONS',
 → 'created_at': '2019-01-15T15:35:00-05:00', 'value': {}}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Remediation action setting history action Restricted to: "CREATED", "UPDATED", "DELETED" Allows: null	action	any	Y	N
Action type of source parent remediation action setting Restricted to: "BLOCK_COMMAND_AND_CONTROL_COMMUNICATIONS", "BLOCK_KNOWN_BAD_HASHES", "CONTAIN_HOSTS", "CONTAIN_INFECTED_REMOVABLE_MEDIA", "DELETE_MALICIOUS_FILES", "DISABLE_AND_MODIFY_AWS_ACCESS_KEYS", "MITIGATE_VULNERABILITY", "OTHER_REMEDIATION", "REMOVE_AND_BLOCK_EMAIL_FORWARDING_ADDRESS", "REMOVE_COMPROMISED_SYSTEMS_FROM_NETWORK_OTHER", "REMOVE_COMPROMISED_SYSTEMS_FROM_NETWORK_AWS", "REMOVE_INBOX_RULES_FOR_KNOWN_COMPROMISED_ACCOUNTS", "RESET_CREDENTIALS_OTHER", "RESET_CREDENTIALS_AWS", "RESET_CREDENTIALS_O365" Allows: null	action_type	any	Y	N
Created timestamp: readonly	created_at	string	Y	N
Remediation action setting history details Allows: null: no-sort	value	object	Y	N
Defines/retrieves expel.io actor records	created_by	Actor	N	Y
Defines/retrieves expel.io organization records	organization	Organization	N	Y
Defines/retrieves expel.io remediation_action_setting records	remediation_action_setting	RemediationActionSettings	N	Y

**class** pyexclient.workbench.**RemediationActionSettingListSourceHistories** (*data,*  
*conn,*  
*included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io remediation\_action\_setting\_list\_source\_history records

Resource type name is **remediation\_action\_setting\_list\_source\_histories**.

Example JSON record:

```
{'action': 'CREATED', 'created_at': '2019-01-15T15:35:00-05:00', 'source_type':
↪ 'CONTEXT_LABEL', 'value': {}}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Re-lation-ship
Remediation action setting list source history action Restricted to: “CREATED”, “UPDATED”, “DELETED” Allows: null	action	any	Y	N
Created timestamp: readonly	created_at	string	Y	N
Source type Restricted to: “CONTEXT_LABEL”	source_type	any	Y	N
Remediation action setting list source history details Allows: null: no-sort	value	object	Y	N
Defines/retrieves expel.io context_label records	context_label	<i>ContextLabels</i>	N	Y
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Defines/retrieves expel.io remediation_action_setting records	remediation_action_setting	<i>RemediationActionSettings</i>	N	Y
Defines/retrieves expel.io remediation_action_setting_list_source records	remediation_action_setting_list_source	<i>RemediationActionSettingListSources</i>	N	Y

**class** pyexclient.workbench.**RemediationActionSettingListSources** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io remediation\_action\_setting\_list\_source records

Resource type name is **remediation\_action\_setting\_list\_sources**.

Example JSON record:

```
{'created_at': '2019-01-15T15:35:00-05:00', 'source_type': 'CONTEXT_LABEL',
↪ 'updated_at': '2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:



Field Description	Field Name	Field Type	At-tributela-relationship	Re-tribution-ship
Created timestamp: readonly	created_at	string	Y	N
Source type Restricted to: "CONTEXT_LABEL"	source_type	any	Y	N
Last Updated timestamp: read-only	updated_at	string	Y	N
Defines/retrieves expel.io context_label records	context_label	<i>ContextLabels</i>	N	Y
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Defines/retrieves expel.io remediation_action_setting records	remediation_action_setting	<i>RemediationActionSettings</i>	N	Y
Defines/retrieves expel.io remediation_action_setting_list_source_history records	remediation_action_setting_list_source_history	<i>RemediationActionSettingListSourceHistories</i>	N	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y

**class** pyexclient.workbench.**RemediationActionSettings** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io remediation\_action\_setting records

Resource type name is **remediation\_action\_settings**.

Example JSON record:

```
{
    'action_type': 'BLOCK_COMMAND_AND_CONTROL_COMMUNICATIONS',
    'automate_list_type': 'DENY',
    'automate_opt_in': True,
    'automate_opt_in_agreement_version': 'V1<br/>Allows: null',
    'automate_opt_in_at': '2019-01-15T15:35:00-05:00',
    'created_at': '2019-01-15T15:35:00-05:00',
    'updated_at': '2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Setting's remediation action type Restricted to: "BLOCK_COMMAND_AND_CONTROL_COMMUNICATIONS", "BLOCK_KNOWN_BAD_HASHES", "CONTAIN_HOSTS", "CONTAIN_INFECTED_REMOVABLE_MEDIA", "DELETE_MALICIOUS_FILES", "DISABLE_AND_MODIFY_AWS_ACCESS_KEYS", "MITIGATE_VULNERABILITY", "OTHER_REMEDIATION", "REMOVE_AND_BLOCK_EMAIL_FORWARDING_ADDRESS", "REMOVE_COMPROMISED_SYSTEMS_FROM_NETWORK_OTHER", "REMOVE_COMPROMISED_SYSTEMS_FROM_NETWORK_AWS", "REMOVE_INBOX_RULES_FOR_KNOWN_COMPROMISED_ACCOUNTS", "RESET_CREDENTIALS_OTHER", "RESET_CREDENTIALS_AWS", "RESET_CREDENTIALS_O365"	action_type	any	Y	N
Automated remediation setting list type Restricted to: "DENY", "ALLOW" Allows: null	automate_list_type	any	Y	N
Opted in to automated remediations	automate_opt_in	boolean	Y	N
Automated remediations opt in agreement version Restricted to: "V1" Allows: null	automate_opt_in_agreement_version	any	Y	N
Last opted in to automated remediations time Allows: null: readonly	automate_opt_in_at	string	Y	N
Created timestamp: readonly	created_at	string	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Defines/retrieves expel.io actor records	automate_opt_in_actor	Actor	N	Y
Defines/retrieves expel.io remediation_action_setting_list_source records	context_label_list_sources	RemediationActionSettingListSource	N	Y
Defines/retrieves expel.io context_label records	context_labels	ContextLabel	N	Y
Defines/retrieves expel.io actor records	created_by	Actor	N	Y
Defines/retrieves expel.io remediation_action_setting_history records	opt_in_histories	RemediationActionSettingHistory	N	Y
Defines/retrieves expel.io remediation_action_setting_history records	opt_out_histories	RemediationActionSettingHistory	N	Y
Defines/retrieves expel.io organization records	organization	Organization	N	Y
Defines/retrieves expel.io remediation_action_setting_history records	remediation_action_setting_histories	RemediationActionSettingHistory	N	Y
Defines/retrieves expel.io remediation_action_setting_list_source_history records	remediation_action_setting_list_source_histories	RemediationActionSettingListSourceHistory	N	Y

---

**class** pyexclient.workbench.**RemediationActions** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Remediation actions

Resource type name is **remediation\_actions**.

Example JSON record:

```
{
    'action': 'string',
    'action_type': 'BLOCK_COMMAND_AND_CONTROL_COMMUNICATIONS',
    'can_automate': True,
    'can_automate_completely': True,
    'cannot_automate_reason': 'DEVICE_ID_MISSING',
    'close_reason': 'string',
    'comment': 'string',
    'created_at': '2019-01-15T15:35:00-05:00',
    'deleted_at': '2019-01-15T15:35:00-05:00',
    'status': 'IN_PROGRESS',
    'status_updated_at': '2019-01-15T15:35:00-05:00',
    'template_name': 'string',
    'updated_at': '2019-01-15T15:35:00-05:00',
    'version': 'V1'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Action Allows: "", null	action	string	Y	N
Action type Restricted to: "BLOCK_COMMAND_AND_CONTROL", "BLOCK_KNOWN_BAD_HASHES", "CONTAIN_HOSTS", "CONTAIN_INFECTED_REMOVABLE_MEDIA", "DELETE_MALICIOUS_FILES", "DISABLE_AND_MODIFY_AWS_ACCESS_KEYS", "MITIGATE_VULNERABILITY", "OTHER_REMEDIATION", "REMOVE_AND_BLOCK_EMAIL_FORWARDING_ADDRESS", "REMOVE_COMPROMISED_SYSTEMS_FROM_NETWORK_OTHER", "REMOVE_COMPROMISED_SYSTEMS_FROM_NETWORK_AWS", "REMOVE_INBOX_RULES_FOR_KNOWN_COMPROMISED_ACCOUNTS", "RESET_CREDENTIALS_OTHER", "RESET_CREDENTIALS_AWS", "RESET_CREDENTIALS_O365" Allows: null	communication_type	string	Y	N
Remediation action can be automated: readonly	can_automate	boolean	Y	N
Remediation action can be completely automated: readonly	can_automate_completely	boolean	Y	N
Reason this action cannot be automated Restricted to: "DEVICE_ID_MISSING", "ACTION_NOT_SUPPORTED", "DEVICE_NOT_SUPPORTED", "FEATURE_FLAG_ACTION_DISABLED", "FEATURE_FLAG_VENDOR_NOT_ALLOWED", "CUSTOMER_CONFIG_DISABLED", "ALLOWED_ASSETS_EMPTY", "ASSETS_NOT_ALLOWED" Allows: null: readonly	cannot_automate_reason	string	Y	N
Close Reason Allows: null	close_reason	string	Y	N
Comment Allows: "", null	comment	string	Y	N
Created timestamp: readonly	created_at	string	Y	N
Deleted At timestamp Allows: null	deleted_at	string	Y	N
Status Restricted to: "IN_PROGRESS", "COMPLETED", "CLOSED"	status	any	Y	N
Status Updated At Allows: null: readonly	status_updated_at	string	Y	N
Remediation Action Template Name Allows: "", null	template_name	string	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Version Restricted to: "V1", "V2", "V3"	version	any	Y	N
Defines/retrieves expel.io actor records	assigned_to_actor	string	N	Y
Defines/retrieves expel.io actor records	created_by	string	N	Y
Investigations	investigation	string	Y	Y

**class** pyexclient.workbench.**ResilienceActionGroups** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io resilience\_action\_group records

Resource type name is **resilience\_action\_groups**.

Example JSON record:

```
{'category': 'DISRUPT_ATTACKERS', 'created_at': '2019-01-15T15:35:00-05:00',
↪ 'title': 'string', 'updated_at': '2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Global Resilience Group Category Restricted to: “DISRUPT_ATTACKERS”, “ENABLE_DEFENDERS”	category	any	Y	N
Created timestamp: readonly	created_at	string	Y	N
Group title	title	string	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Resilience actions	resilience_actions	<i>ResilienceActions</i>	Y	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y

**class** pyexclient.workbench.**ResilienceActions** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Resilience actions

Resource type name is **resilience\_actions**.

Example JSON record:

```
{'category': 'DISRUPT_ATTACKERS', 'created_at': '2019-01-15T15:35:00-05:00',
↪ 'details': 'string', 'impact': 'LOW', 'title': 'string', 'updated_at': '2019-01-
↪ 15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Category Restricted to: “DISRUPT_ATTACKERS”, “ENABLE_DEFENDERS” Allows: null	category	any	Y	N
Created timestamp: readonly	created_at	string	Y	N
Details	details	string	Y	N
Impact Restricted to: “LOW”, “MEDIUM”, “HIGH”	impact	any	Y	N
Title	title	string	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Defines/retrieves expel.io resilience_action_group records	resilience_action_group	<i>ResilienceActionGroups</i>	N	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y

**class** pyexclient.workbench.ResourceInstance (data, conn, included=None)

Bases: object

Represents an instance of a base resource.

**classmethod** create (conn, \*\*kwargs)

Create a new resource instance. Users need to call save() after create to write changes to the server.

**Returns** The updated resource instance

**Return type** ResourceInstance

**Examples:**

```
>>> i = xc.investigations.create(title='Peter: new investigation 1',
↳relationship_customer=ORGANIZATION_ID, relationship_assigned_to_
↳actor=ACTOR_ID)
>>> i.save()
```

**delete** (prompt\_on\_delete=True)

Delete a resource instance.

**Parameters** prompt\_on\_delete (bool, optional) – True if user wants to be prompted when delete is issued and False otherwise., defaults to True.

**Examples:**

```
>>> inv = xc.investigations.get(id='a8bf9750-6a79-4415-9558-a56253606b9f')
>>> inv.delete()
```

**id**

Retrieve the identifier for the resource instance.

**Returns** A GUID representing the unique instance

**Return type** str

**Examples:**

```
>>> for inv in xc.investigations.filter_by(status='OPEN'):
>>>     print("Investigation ID is %s" % inv.id)
```

**save()**

Write changes made to a resource instance back to the sever.

**Returns** The updated resource instance

**Return type** *ResourceInstance*

**Examples:**

```
>>> i = xc.investigations.create(title='Peter: new investigation 1',
↳ relationship_customer=ORGANIZATION_ID, relationship_assigned_to_
↳ actor=ACTOR_ID)
>>> i.save()
```

**class** pyexclient.workbench.**SamlIdentityProviders** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

SAML Identity Providers

Resource type name is **saml\_identity\_providers**.

Example JSON record:

```
{'callback_uri': 'string', 'cert': 'string', 'entity_id': 'string', 'status':
↳ 'string'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Relation-ship
Allows: ""	call-back_uri	string	Y	N
Allows: "", null	cert	string	Y	N
Allows: ""	entity_id	string	Y	N
Restricted to: "not_configured", "configured"	status	string	Y	N
Defines/retrieves expel.io organization records	organization	<i>Organizations</i>	N	Y

**class** pyexclient.workbench.**Secrets** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Organization secrets. Note - these requests must be in the format of */secrets/security\_device-<guid>*

Resource type name is **secrets**.

Example JSON record:

```
{
  'secret': {
    'device_info': {'access_id': '7b0a343c-860e-
↳ 442e-ab0b-d6f349d364d9', 'access_key': 'secret-access-key', 'source_category':
↳ 'alpha'},
    'device_secret': {'console_url': 'https://console-
↳ access-point.com', 'password': 'password', 'username': 'admin@company.com'},
```

(continues on next page)

(continued from previous page)

```

    'two_factor_secret': 'GNFXSU2OKNJXUPTGJVQUMNDHM4YVEKRJ' }
  }

```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Relation-ship
Allows: null	secret	object	Y	N
Defines/retrieves expel.io organization records	organization	<i>Organizations</i>	N	Y

**class** pyexclient.workbench.**SecurityDevices** (*data, comm, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Security devices

Resource type name is **security\_devices**.

Example JSON record:

```

{
  'created_at': '2019-01-15T15:35:00-05:00',
  'deleted_at': '2019-01-15T15:35:00-05:00',
  'device_spec': {},
  'device_type': 'ENDPOINT',
  'has_two_factor_secret': True,
  'location': 'string',
  'name': 'string',
  'plugin_slug': 'string',
  'properties': {},
  'status': 'healthy',
  'status_details': {},
  'status_updated_at': '2019-01-15T15:35:00-05:00',
  'task_source': 'CUSTOMER_PREMISE',
  'updated_at': '2019-01-15T15:35:00-05:00'
}

```

Below are valid filter by parameters:



Field Description	Field Name	Field Type	Attribute	Relationship
Created timestamp: readonly	created_at	string	Y	N
Deleted At timestamp Allows: null	deleted_at	string	Y	N
Device Spec Allows: null: no-sort	device_spec	object	Y	N
Device Type Restricted to: "ENDPOINT", "NETWORK", "SIEM", "OTHER", "CLOUD"	device_type	any	Y	N
Has 2fa secret stored in vault: readonly	has_two_factor_secret	boolean	Y	N
Location Allows: "", null	location	string	Y	N
Name	name	string	Y	N
Allows: "", null	plugin_slug	string	Y	N
Properties about the security device: no-sort	properties	object	Y	N
Status. Note: By default if the security device has an assembler, and that assembler is unhealthy, the status will return that information rather than the raw status of the security device. To disable this behavior, add the query parameter <i>flag[raw_status]=true</i> . Restricted to: "healthy", "unhealthy", "health_checks_not_supported" Allows: null	status	any	Y	N
Status Details. Note: By default if the security device has an assembler, and that assembler is unhealthy, the status details will return that information rather than the raw status of the security device. To disable this behavior, add the query parameter <i>flag[raw_status]=true</i> . Allows: null: no-sort	status_details	object	Y	N
Status Updated At Allows: null: readonly	status_updated_at	string	Y	N
Location where tasks are run Restricted to: "CUSTOMER_PREMISE", "EXPEL_TASKPOOL"	task_source	any	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Assemblers	assembler	<i>Assemblers</i>	N	Y
Security devices	child_security_device	<i>SecurityDevices</i>	N	Y
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
investigative actions	investigative_actions	<i>InvestigativeActions</i>	N	Y
Defines/retrieves expel.io organization records	organization	<i>Organizations</i>	N	Y
Security devices	parent_security_device	<i>SecurityDevices</i>	N	Y
Remediation action histories	remediation_action_histories	<i>RemediationActionHistories</i>	N	Y
Remediation actions	remediation_actions	<i>RemediationActions</i>	N	Y

**class** pyexclient.workbench.**TimelineEntries** (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

Timeline Entries

Resource type name is **timeline\_entries**.

Example JSON record:

```
{
    'attack_phase': 'string',
    'comment': 'string',
    'created_at': '2019-01-15T15:35:00-05:00',
    'deleted_at': '2019-01-15T15:35:00-05:00',
    'dest_host': 'string',
    'event': 'string',
    'event_date': '2019-01-15T15:35:00-05:00',
    'event_type': 'string',
    'is_selected': True,
    'src_host': 'string',
    'updated_at': '2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Rela-tion-ship
Attack phase of the Timeline Entry Allows: "", null	attack_phase	string	Y	N
Comment on this Timeline Entry Allows: "", null	comment	string	Y	N
Created timestamp: readonly	created_at	string	Y	N
Deleted At timestamp Allows: null	deleted_at	string	Y	N
Destination Host (IP or Hostname) Allows: "", null	dest_host	string	Y	N
The event, such as Powershell Attack Allows: "", null	event	string	Y	N
Date/Time of when the event occurred	event_date	string	Y	N
The type of the event, such as Carbon Black Alert Allows: "", null	event_type	string	Y	N
Has been selected for final report.	is_selected	boolean	Y	N
Source Host (IP or Hostname) Allows: "", null	src_host	string	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Defines/retrieves expel.io context_label_action records	con-text_label_actions	<i>ContextLabelActions</i>	N	Y
Defines/retrieves expel.io context_label records	con-text_labels	<i>ContextLabels</i>	N	Y
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Expel alerts	expel_alert	<i>ExpelAlerts</i>	N	Y
Investigations	investigation	<i>Investigations</i>	N	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y

**class** pyexclient.workbench.**UserAccountRoles** (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

Defines/retrieves expel.io user\_account\_role records

Resource type name is **user\_account\_roles**.

Example JSON record:

```
{'active': True, 'assignable': True, 'created_at': '2019-01-15T15:35:00-05:00',
  ↪'role': 'expel_admin', 'updated_at': '2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
If this role is active	active	boolean	Y	N
Can user be assigned items (e.g. investigations, etc)	assignable	boolean	Y	N
Created timestamp: readonly	created_at	string	Y	N
User account role for this organization Restricted to: “expel_admin”, “expel_analyst”, “organization_admin”, “organization_analyst”, “system”, “anonymous”, “restricted”	role	any	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Defines/retrieves expel.io actor records	created_by	Actors	N	Y
Defines/retrieves expel.io organization records	organization	Organizations	N	Y
Defines/retrieves expel.io actor records	updated_by	Actors	N	Y
User accounts	user_account	Accounts	N	Y

**class** pyexclient.workbench.**UserAccountStatuses** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

User account status

Resource type name is **user\_account\_statuses**.

Example JSON record:

```
{
  'active': True,
  'active_status': 'ACTIVE',
  'created_at': '2019-01-15T15:35:00-05:00',
  'invite_token_expires_at': '2019-01-15T15:35:00-05:00',
  'password_reset_token_expires_at': '2019-01-15T15:35:00-05:00',
  'restrictions': [],
  'updated_at': '2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Missing Description	active	boolean	Y	N
Restricted to: "ACTIVE", "LOCKED", "LOCKED_INVITED", "LOCKED_EXPIRED", "ACTIVE_INVITED", "ACTIVE_EXPIRED": read-only	active_status	any	Y	N
Meta: readonly	created_at	string	Y	N
Allows: null: readonly	invite_token_expires_at	string	Y	N
Allows: null: readonly	password_reset_token_expires_at	string	Y	N
Missing Description	restrictions	array	Y	N
Meta: readonly	updated_at	string	Y	N
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Defines/retrieves expel.io organization records	primary_organization	<i>Organizations</i>	N	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y
User accounts	user_account	<i>UserAccounts</i>	N	Y

**class** pyexclient.workbench.**UserAccounts** (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

User accounts

Resource type name is **user\_accounts**.

Example JSON record:

```
{
  'active': True,
  'active_status': 'ACTIVE',
  'assignable': True,
  'created_at': '2019-01-15T15:35:00-05:00',
  'default_filter': 'ALL',
  'display_name': 'string',
  'email': 'name@company.com',
  'engagement_manager': True,
  'first_name': 'string',
  'homepage_preferences': {},
  'language': 'string',
  'last_name': 'string',
  'locale': 'string',
  'pagerduty_id': 'string',
  'phone_number': 'string',
  'timezone': 'string',
  'updated_at': '2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description
Active Allows: null

Field Description
Restricted to: "ACTIVE", "LOCKED", "LOCKED_INVITED", "LOCKED_EXPIRED", "ACTIVE_INVITED", "ACTIVE_EXPIRE"
Can user be assigned items (e.g. investigations, etc)
Created timestamp: readonly
User account default filter Restricted to: "ALL", "MDR", "PHISHING"
Display name Allows: "", null
Email
Is an engagement manager
First Name
Homepage preferences Allows: null: no-sort
Language Allows: "", null
Last Name
Locale Allows: "", null
Pagerduty ID Allows: null
Phone number Allows: null
Timezone Allows: "", null
Last Updated timestamp: readonly
Defines/retrieves expel.io actor records
investigative actions
Expel alerts
Investigations
investigative actions
Organization to resilience actions
Organization to resilience actions
Remediation actions
Defines/retrieves expel.io remediation_action_setting records
Defines/retrieves expel.io actor records
User Notification Preferences
Defines/retrieves expel.io organization records
Defines/retrieves expel.io organization records
Defines/retrieves expel.io actor records
Defines/retrieves expel.io user_account_role records
User account status

**class** pyexclient.workbench.**VendorAlertEvidences** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Vendor alert evidences are extracted from a vendor alert's evidence summary

Resource type name is **vendor\_alert\_evidences**.

Example JSON record:

```
{'evidence': 'string', 'evidence_type': 'HOSTNAME'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Evidence	evi-dence	string	Y	N
Type Restricted to: "HOSTNAME", "URL", "PROCESS_ARGUMENTS", "PROCESS_PATH", "PROCESS_MD5", "USERNAME", "SRC_IP", "DST_IP", "PARENT_ARGUMENTS", "PARENT_PATH", "PARENT_MD5", "SRC_USERNAME", "DST_USERNAME", "ALERT_ACTION", "ALERT_DESCRIPTION", "ALERT_MESSAGE", "ALERT_NAME", "SRC_PORT", "DST_PORT", "USER_AGENT", "VENDOR_NAME", "DOMAIN", "FILE_HASH", "FILE_PATH"	evi-dence_type	any	Y	N
Expel alerts	ev-i-denced_expel_alerts	<i>Expel Alerts</i>	N	Ys
Vendor alerts	ven-dor_alert	<i>Vendor Alerts</i>	N	Ys

**class** pyexclient.workbench.**VendorAlerts** (*data, conn, included=None*)

Bases: *pyexclient.workbench.ResourceInstance*

Vendor alerts

Resource type name is **vendor\_alerts**.

Example JSON record:

```
{
  'created_at': '2019-01-15T15:35:00-05:00',
  'description': 'string',
  'evidence_activity_end_at': '2019-01-15T15:35:00-05:00',
  'evidence_activity_start_at': '2019-01-15T15:35:00-05:00',
  'evidence_summary': [],
  'first_seen': '2019-01-15T15:35:00-05:00',
  'original_alert_id': 'string',
  'original_source_id': 'string',
  'signature_id': 'string',
  'status': 'NORMAL',
  'updated_at': '2019-01-15T15:35:00-05:00',
  'vendor_message': 'string',
  'vendor_severity': 'CRITICAL',
  'vendor_sig_name': 'string'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	Attribute	Relationship
Created timestamp: readonly	created_at	string	Y	N
Description Allows: "", null	description	string	Y	N
Evidence activity end datetime Allows: null: immutable	evidence_activity_end_at	string	Y	N
Evidence activity start datetime Allows: null: immutable	evidence_activity_start_at	string	Y	N
Evidence summary Allows: null: no-sort	evidence_summary	array	Y	N
First Seen	first_seen	string	Y	N
Allows: null: immutable	original_alert_id	string	Y	N
Allows: null: immutable	original_source_id	string	Y	N
Signature ID Allows: "", null	signature_id	string	Y	N
Status Restricted to: "NORMAL", "PROVISIONAL" Allows: null: readonly	status	any	Y	N
Last Updated timestamp: readonly	updated_at	string	Y	N
Vendor Message Allows: "", null	vendor_message	string	Y	N
Vendor alert severity Restricted to: "CRITICAL", "HIGH", "MEDIUM", "LOW", "TESTING", "TUNING" Allows: null	vendor_severity	any	Y	N
Vendor Sig Name Allows: "", null	vendor_sig_name	string	Y	N
Assemblers	assembler	<i>Assemblers</i>	N	Y
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Vendor alert evidences are extracted from a vendor alert's evidence summary	evidences	<i>VendorAlertEvidences</i>	N	Y
Expel alerts	expel_alerts	<i>ExpelAlerts</i>	N	Y
IP addresses	ip_addresses	<i>IpAddresses</i>	N	Y
Defines/retrieves expel.io organization records	organization	<i>Organizations</i>	N	Y
Security devices	security_device	<i>SecurityDevices</i>	N	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y
Vendors	vendor	<i>Vendors</i>	N	Y

**class** pyexclient.workbench.Vendors (data, conn, included=None)

Bases: *pyexclient.workbench.ResourceInstance*

Vendors

Resource type name is **vendors**.

Example JSON record:

```
{'created_at': '2019-01-15T15:35:00-05:00', 'icon': 'string', 'name': 'string',
  ↪ 'updated_at': '2019-01-15T15:35:00-05:00'}
```

Below are valid filter by parameters:

Field Description	Field Name	Field Type	At-tribute	Relation-ship
Created timestamp: readonly	created_at	string	Y	N
Icon Allows: “”, null	icon	string	Y	N
Name Allows: “”, null	name	string	Y	N
Last Updated timestamp: read-only	updated_at	string	Y	N
Defines/retrieves expel.io actor records	created_by	<i>Actors</i>	N	Y
Expel alerts	expel_alerts	<i>ExpelAlerts</i>	N	Y
Security devices	secu-rity_devices	<i>SecurityDevices</i>	N	Y
Defines/retrieves expel.io actor records	updated_by	<i>Actors</i>	N	Y
Vendor alerts	vendor_alerts	<i>VendorAlerts</i>	N	Y

```
class pyexclient.workbench.WorkbenchClient (base_url, username=None, pass-
                                             word=None, mfa_code=None, token=None,
                                             prompt_on_delete=True)
```

Bases: *pyexclient.workbench.WorkbenchCoreClient*

Instantiate a client that interacts with Workbench’s API server.

If the developer specifies a username, then password and mfa\_code are required inputs. If the developer has a token then username, password and mfa\_code parameters are ignored.

**Parameters**

- **cls** (*WorkbenchClient*) – A Workbench class reference.
- **username** (*str or None*) – The username
- **password** (*str or None*) – The username’s password
- **mfa\_code** (*int or None*) – The multi factor authenticate code generated by google authenticator.
- **token** (*str or None*) – The bearer token of an authorized session. Can be used instead of username/password combo.

**Returns** An initialized, and authorized Workbench client.

**Return type** *WorkbenchClient*

```
capabilities (customer_id: str)
```

Get a list of capabilities for a given customer.

**Parameters** **customer\_id** (*str*) – The customer ID

**Examples:**

```
>>> xc.workbench.capabilities("my-customer-guid-123")
```



**create\_auto\_inv\_action** (*customer\_id: str, vendor\_device\_id: str, created\_by\_id: str, capability\_name: str, input\_args: dict, title: str, reason: str, investigation\_id: str = None, expel\_alert\_id: str = None*)

Create an automatic investigative action.

#### Parameters

- **customer\_id** (*str*) – The customer ID
- **investigation\_id** (*str*) – The investigation ID to associate the action with.
- **expel\_alert\_id** (*str*) – The expel alert id
- **vendor\_device\_id** (*str*) – The vendor device ID, to dispatch the task against.
- **created\_by\_id** (*str*) – The user ID that created the action
- **capability\_name** (*str*) – The name of the capability we are running. Defined in classes <https://github.com/expel-io/taskabilities/tree/master/py/taskabilities/cpe/capabilities>, look at name class variable.
- **input\_args** (*dict*) – The input arguments to the capability to run. Defined in classes <https://github.com/expel-io/taskabilities/tree/master/py/taskabilities/cpe/capabilities>, look at name class variable.
- **title** (*str*) – The title of the investigative action, shows up in Workbench.
- **reason** (*str*) – The reason for running the investigative action, shows up in Workbench.

**Returns** Investigative action response

**Return type** *InvestigativeActions*

#### Examples:

```
>>> xc = WorkbenchClient('https://workbench.expel.io', username=username,
↳ password=password, mfa_code=mfa_code)
>>> input_args = &#123;"user_name": 'willy.wonka@expel.io', 'time_range_
↳ start': '2019-01-30T14:00:40Z', 'time_range_end': '2019-01-30T14:45:40Z'&
↳ #125;
>>> o = xc.create_auto_inv_action(customer_guid, inv_guid, device_guid,
↳ user_guid, 'query_user', input_args, 'Query User', 'Getting user login_
↳ activity to determine if login is normal')
>>> print("Investigative Action ID: ", o.id)
```

**create\_manual\_inv\_action** (*title: str, reason: str, instructions: str, investigation\_id: str = None, expel\_alert\_id: str = None, security\_device\_id: str = None, action\_type: str = 'MANUAL'*)

Create a manual investigative action.

#### Parameters

- **title** (*str*) – The title of the investigative action, shows up in Workbench.
- **reason** (*str*) – The reason for running the investigative action, shows up in Workbench.
- **instructions** (*str*) – The instructions for running the investigative action.
- **investigation\_id** (*str*) – The investigation ID to associate the action with.
- **expel\_alert\_id** (*str*) – The expel alert id
- **security\_device\_id** (*str*) – The security device ID, to dispatch the task against.
- **action\_type** (*str*) – The type of action that will be run.

**Returns** Investigative action response

**Return type** *InvestigativeActions*

**Examples:**

```
>>> xc = WorkbenchClient('https://workbench.expel.io', username=username,
↳ password=password, mfa_code=mfa_code)
>>> o = xc.create_manual_inv_action('title foo', 'reason bar',
↳ 'instructions blah')
>>> print("Investigative Action ID: ", o.id)
```

**plugins ()**

Get a list of plugins.

**Examples:**

```
>>> xc.workbench.plugins ()
```

```
class pyexclient.workbench.WorkbenchCoreClient (base_url, username=None, password=None, mfa_code=None, token=None, retries=3, prompt_on_delete=True)
```

Bases: object

Instantiate a Workbench core client that provides just authentication and request capabilities to Workbench

If the developer specifies a username, then password and mfa\_code are required inputs. If the developer has a token then username, password and mfa\_code parameters are ignored.

**Parameters**

- **cls** (*WorkbenchClient*) – A Workbench class reference.
- **username** (*str or None*) – The username
- **password** (*str or None*) – The username's password
- **mfa\_code** (*int or None*) – The multi factor authenticate code generated by google authenticator.
- **token** (*str or None*) – The bearer token of an authorized session. Can be used instead of username/password combo.

**Returns** An initialized, and authorized Workbench client.

**Return type** *WorkbenchClient*

**login (username, password, code)**

Authenticate as a human, this requires providing the 2FA code.

**Parameters**

- **username** (*str*) – The user's e-mail address.
- **password** (*str*) – The user's password.
- **code** (*str*) – The 2FA code

**Returns** The bearer token that allows users to call Workbench APIs.

**Return type** *str*

**make\_session ()**

Create a session with Workbench

**class** pyexclient.workbench.**base\_filter** (*filter\_value*)

Bases: *pyexclient.workbench.operator*

Base class for operators which take the form filter[field]. Can be used to create a basic one field filter, or subclassed by special operators for more complicated logic

**class** pyexclient.workbench.**contains** (\*args)

Bases: *pyexclient.workbench.base\_filter*

The contains operator is used to search for fields that contain a sub string..

**Parameters** **value** (*str*) – A substring to be checked against the value of a field.

**Examples:**

```
>>> for ea in xc.expel_alerts.search(close_comment=contains("foo")):
>>>     print("%s contains foo in the close comment" % ea.expel_name)
```

**class** pyexclient.workbench.**flag** (*filter\_value*)

Bases: *pyexclient.workbench.operator*

Base class for operators which take the form flag[field]. Can be used to create a basic one field flag, or subclassed by special operators for more complicated logic

**class** pyexclient.workbench.**gt** (*value*)

Bases: *pyexclient.workbench.base\_filter*

The gt (greater than) operator is used to search a specific field for values greater than X.

**Parameters** **value** (*str*) – The greater than value to be used in comparison during a search.

**Examples:**

```
>>> for ea in xc.expel_alerts.search(created_at=gt("2020-01-01")):
>>>     print("%s was created after 2020-01-01" % ea.expel_name)
```

**class** pyexclient.workbench.**include** (*include*)

Bases: *pyexclient.workbench.operator*

The include operator requests base resource names in a search. Cannot be used with sort or filtering. Passed as arg to search TODO enforce this constraint with asserts

**Parameters** **include** (*str*) – Include specific base resource names in request

Examples: >>> for ea in xc.expel\_alerts.search(include='organization,created\_by,updated\_by'): >>> print(ea.organization)

pyexclient.workbench.**is\_operator** (*value*)

Determine if a value implements an operator.

**Parameters** **value** (*object*) – The value to check

**Returns** *True* if value is an operator *False* otherwise.

**Return type** bool

**class** pyexclient.workbench.**isnull** (*filter\_value=True*)

Bases: *pyexclient.workbench.base\_filter*

The isnull operator is used to search for fields that are null.

**Examples:**

```
>>> for ea in xc.expel_alerts.search(close_comment=isnull()):
>>>     print("%s has no close comment" % ea.expel_name)
```

**class** pyexclient.workbench.**limit** (*limit*)  
Bases: *pyexclient.workbench.operator*

The limit operator adds a limit to a search. Passed as arg to search

**Parameters** **limit** (*int*) – Limit the number of results returned.

**class** pyexclient.workbench.**lt** (*value*)  
Bases: *pyexclient.workbench.base\_filter*

The lt (less than) operator is used to search a specific field for values greater than X.

**Parameters** **value** (*str*) – The less than value to be used in comparison during a search.

**Examples:**

```
>>> for ea in xc.expel_alerts.search(created_at=lt("2020-01-01")):
>>>     print("%s was created before 2020-01-01" % ea.expel_name)
```

**class** pyexclient.workbench.**neq** (*\*args*)  
Bases: *pyexclient.workbench.base\_filter*

The neq operator is used to search for fields that are not equal to a specified value.

**Parameters** **value** (*str*) – The value to assert the field is not equal too

**Examples:**

```
>>> for ea in xc.expel_alerts.search(close_comment=neq("foo")):
>>>     print("%s has a close comment that is not equal to 'foo'" % ea.expel_
↪name)
```

**class** pyexclient.workbench.**notnull** (*filter\_value=True*)  
Bases: *pyexclient.workbench.base\_filter*

The notnull operator is used to search for fields that are not null.

**Examples:**

```
>>> for ea in xc.expel_alerts.search(close_comment=notnull()):
>>>     print("%s has a close comment of %s" % (ea.expel_name, ea.close_
↪comment))
```

**class** pyexclient.workbench.**operator** (*filter\_value*)  
Bases: *object*

Base class for all operators. This should not be used directly.

**class** pyexclient.workbench.**relationship** (*rel\_path, value*)  
Bases: *pyexclient.workbench.operator*

relationship operator allows for searching of resource objects based on their relationship to other resource objects. Passed as arg to search

**Parameters**

- **rel\_path** (*str*) – A dot notation of the relationship path to a resource object.

- **value** (*object*) – The value the `rel_path` be compared to. This can be an operator, or a primitive value.

**Examples:**

```
>>> for inv_action in xc.investigative_actions.search(relationship(
↳ "investigation.close_comment", notnull()):
>>>     print("Found investigative action associated with an investigation_
↳ that has no close comment.")
```

**class** `pyexclient.workbench.sort` (*sort, order='asc'*)

Bases: `pyexclient.workbench.operator`

The sort operator passes a sort request to a search. Can add multiple sort operators to a single search. If no sort is provided the default of sorting by `created_at` (`asc`) -> `id` (`asc`) will be used. Passed as arg to search TODO enforce this with asserts

**Parameters** **sort** (*str*) – The column to sort on. Expects `asc` or `desc`. The database will translate `asc`->+ and `desc`->-

**class** `pyexclient.workbench.startswith` (*swith*)

Bases: `pyexclient.workbench.base_filter`

The startswith operator is used to search for values that start with a specified string..

**Parameters** **value** (*str*) – The startswith string

**Examples:**

```
>>> for ea in xc.expel_alerts.search(close_comment=startswith("foo")):
>>>     print("%s starts with foo in the close comment" % ea.expel_name)
```

**class** `pyexclient.workbench.window` (*start, end*)

Bases: `pyexclient.workbench.base_filter`

The window operator is used to search a specific field that is within a window (range) of values

**Parameters**

- **start** (*Union[str, int, datetime.datetime]*) – The beginning of the window range
- **end** (*str*) – The end of the window range

**Examples:**

```
>>> for ea in xc.expel_alerts.search(created_at=window("2020-01-01", "2020-05-
↳ 01")):
>>>     print("%s was created after 2020-01-01 and before 2020-05-01" % ea.
↳ expel_name)
```



**p**

`pyexclient.workbench`, 25





**A**

ActivityMetrics (class in *pyexclient.workbench*), 25  
 Actors (class in *pyexclient.workbench*), 26  
 ApiKeys (class in *pyexclient.workbench*), 27  
 AssemblerImages (class in *pyexclient.workbench*), 28  
 Assemblers (class in *pyexclient.workbench*), 29

**B**

base\_filter (class in *pyexclient.workbench*), 95  
 BaseResourceObject (class in *pyexclient.workbench*), 30

**C**

capabilities() (*pyexclient.workbench.WorkbenchClient* method), 92  
 CommentHistories (class in *pyexclient.workbench*), 32  
 Comments (class in *pyexclient.workbench*), 33  
 Configurations (class in *pyexclient.workbench*), 33  
 contains (class in *pyexclient.workbench*), 95  
 ContextLabelActionHistories (class in *pyexclient.workbench*), 34  
 ContextLabelActions (class in *pyexclient.workbench*), 35  
 ContextLabelHistories (class in *pyexclient.workbench*), 36  
 ContextLabels (class in *pyexclient.workbench*), 37  
 ContextLabelTags (class in *pyexclient.workbench*), 36  
 count() (*pyexclient.workbench.BaseResourceObject* method), 30  
 create() (*pyexclient.workbench.BaseResourceObject* method), 31  
 create() (*pyexclient.workbench.ResourceInstance* class method), 82

create\_auto\_inv\_action() (*pyexclient.workbench.WorkbenchClient* method), 92

create\_manual\_inv\_action() (*pyexclient.workbench.WorkbenchClient* method), 93

**D**

delete() (*pyexclient.workbench.ResourceInstance* method), 82

Detections (class in *pyexclient.workbench*), 39

download() (*pyexclient.workbench.FilesResourceInstance* method), 47

**E**

EngagementManagers (class in *pyexclient.workbench*), 40

Entitlements (class in *pyexclient.workbench*), 41

ExpelAlertHistories (class in *pyexclient.workbench*), 41

ExpelAlerts (class in *pyexclient.workbench*), 43

ExpelAlertThresholdHistories (class in *pyexclient.workbench*), 42

ExpelAlertThresholds (class in *pyexclient.workbench*), 43

ExpelDetectionCategories (class in *pyexclient.workbench*), 45

**F**

Files (class in *pyexclient.workbench*), 46

FilesResourceInstance (class in *pyexclient.workbench*), 47

filter\_by() (*pyexclient.workbench.BaseResourceObject* method), 31

Findings (class in *pyexclient.workbench*), 47

flag (class in *pyexclient.workbench*), 95

**G**

get() (*pyexclient.workbench.BaseResourceObject*

method), 31  
 gt (class in pyexclient.workbench), 95

## I

id (pyexclient.workbench.ResourceInstance attribute), 82  
 include (class in pyexclient.workbench), 95  
 Integrations (class in pyexclient.workbench), 48  
 InvestigationFindingHistories (class in pyexclient.workbench), 49  
 InvestigationFindings (class in pyexclient.workbench), 50  
 InvestigationHistories (class in pyexclient.workbench), 51  
 InvestigationResilienceActionHints (class in pyexclient.workbench), 52  
 InvestigationResilienceActions (class in pyexclient.workbench), 52  
 Investigations (class in pyexclient.workbench), 52  
 InvestigativeActionHistories (class in pyexclient.workbench), 54  
 InvestigativeActions (class in pyexclient.workbench), 55  
 InvestigativeActionsResourceInstance (class in pyexclient.workbench), 56  
 IpAddresses (class in pyexclient.workbench), 57  
 is\_operator() (in module pyexclient.workbench), 95  
 isnull (class in pyexclient.workbench), 95

## J

JsonApiRelationship (class in pyexclient.workbench), 57

## L

limit (class in pyexclient.workbench), 96  
 login() (pyexclient.workbench.WorkbenchCoreClient method), 94  
 lt (class in pyexclient.workbench), 96

## M

make\_session() (pyexclient.workbench.WorkbenchCoreClient method), 94  
 MitreTactics (class in pyexclient.workbench), 58

## N

neq (class in pyexclient.workbench), 96  
 NistCategories (class in pyexclient.workbench), 58  
 NistSubcategories (class in pyexclient.workbench), 59  
 NistSubcategoryScoreHistories (class in pyexclient.workbench), 59  
 NistSubcategoryScores (class in pyexclient.workbench), 60

NotificationPreferences (class in pyexclient.workbench), 61  
 notnull (class in pyexclient.workbench), 96

## O

one\_or\_none() (pyexclient.workbench.BaseResourceObject method), 31  
 operator (class in pyexclient.workbench), 96  
 OrganizationResilienceActionGroups (class in pyexclient.workbench), 62  
 OrganizationResilienceActions (class in pyexclient.workbench), 62  
 Organizations (class in pyexclient.workbench), 64  
 OrganizationStatuses (class in pyexclient.workbench), 63

## P

PhishingSubmissionAttachments (class in pyexclient.workbench), 66  
 PhishingSubmissionDomains (class in pyexclient.workbench), 66  
 PhishingSubmissionHeaders (class in pyexclient.workbench), 67  
 PhishingSubmissions (class in pyexclient.workbench), 68  
 PhishingSubmissionUrls (class in pyexclient.workbench), 67  
 plugins() (pyexclient.workbench.WorkbenchClient method), 94  
 pyexclient.workbench (module), 25

## R

relationship (class in pyexclient.workbench), 96  
 RemediationActionAssetHistories (class in pyexclient.workbench), 69  
 RemediationActionAssets (class in pyexclient.workbench), 71  
 RemediationActionHistories (class in pyexclient.workbench), 73  
 RemediationActions (class in pyexclient.workbench), 79  
 RemediationActionSettingHistories (class in pyexclient.workbench), 74  
 RemediationActionSettingListSourceHistories (class in pyexclient.workbench), 75  
 RemediationActionSettingListSources (class in pyexclient.workbench), 76  
 RemediationActionSettings (class in pyexclient.workbench), 77  
 ResilienceActionGroups (class in pyexclient.workbench), 81  
 ResilienceActions (class in pyexclient.workbench), 81

---

ResourceInstance (class in *pyexclient.workbench*),  
82

## S

SamlIdentityProviders (class in *pyexclient.workbench*), 83

save() (*pyexclient.workbench.ResourceInstance*  
method), 83

search() (*pyexclient.workbench.BaseResourceObject*  
method), 32

Secrets (class in *pyexclient.workbench*), 83

SecurityDevices (class in *pyexclient.workbench*),  
84

sort (class in *pyexclient.workbench*), 97

startswith (class in *pyexclient.workbench*), 97

## T

TimelineEntries (class in *pyexclient.workbench*),  
86

to\_relationship() (*pyexclient.workbench.JsonApiRelationship*  
method), 58

## U

upload() (*pyexclient.workbench.InvestigativeActionsResourceInstance*  
method), 57

UserAccountRoles (class in *pyexclient.workbench*),  
86

UserAccounts (class in *pyexclient.workbench*), 88

UserAccountStatuses (class in *pyexclient.workbench*), 87

## V

VendorAlertEvidences (class in *pyexclient.workbench*), 89

VendorAlerts (class in *pyexclient.workbench*), 90

Vendors (class in *pyexclient.workbench*), 91

## W

window (class in *pyexclient.workbench*), 97

WorkbenchClient (class in *pyexclient.workbench*),  
92

WorkbenchCoreClient (class in *pyexclient.workbench*), 94