

---

# **aiosnow Documentation**

**Robert Wikman <[rbw@vault13.org](mailto:rbw@vault13.org)>**

**Jan 24, 2021**



---

## Contents

---

<b>1</b>	<b>Requirements</b>	<b>3</b>
<b>2</b>	<b>Installation</b>	<b>5</b>
<b>3</b>	<b>Dependencies</b>	<b>7</b>
<b>4</b>	<b>Table of Contents</b>	<b>9</b>
	<b>Python Module Index</b>	<b>27</b>
	<b>Index</b>	<b>29</b>



**aiosnow** is a Python [asyncio](#) library for interacting with ServiceNow programmatically. It hopes to be:

- Convenient: A good deal of work is put into making the library flexible and easy to use.
- Performant: Remote API calls uses non-blocking sockets tracked by an event loop, allowing large amounts of lightweight request tasks to run concurrently.
- Modular: Core functionality is componentized into modules that are built with composability and extensibility in mind.

For usage examples, visit [github.com/rbw/aiosnow/examples](https://github.com/rbw/aiosnow/examples).



# CHAPTER 1

---

## Requirements

---

- Python 3.7+





## CHAPTER 2

---

### Installation

---

The library is available on PyPI and can be installed using **pip**.

```
$ pip install aiosnow
```



## CHAPTER 3

---

### Dependencies

---

The `aiohttp` library depends on a small set of stable and permissively licensed libraries.

- `aiohttp`: Communication
- `marshmallow`: Schema system



### 4.1 Funding

The `aiosnow` code is permissively licensed, and can be incorporated into any type of application—commercial or otherwise—without costs or limitations. Its author believes it’s in the commercial best-interest for users of the project to invest in its ongoing development.

Consider leaving a [donation](#) if you like this software.

### 4.2 Client

The *Client* is typically invoked via *Models* and provides means to communicate with ServiceNow.

#### 4.2.1 API

```
class aiosnow.Client (address: Union[str, bytes], basic_auth: tuple = None, use_ssl:
    bool = True, verify_ssl: bool = None, pool_size: int = 100, re-
    sponse_cls: Type[aiosnow.request.response.Response] = None, session_cls:
    Type[aiohttp.client.ClientSession] = None)
```

Client for communicating with ServiceNow

Parses client config and provides a `ClientSession` factory.

#### Parameters

- **address** – Instance TCP address, example: `my-instance.service-now.com`
- **basic\_auth** – Tuple of (username, password)
- **use\_ssl** – Whether to use SSL
- **verify\_ssl** – Whether to verify SSL certificates
- **pool\_size** – Connection pool size

- **response\_cls** – Custom Response class

**config**

Client configuration object

## 4.3 Models

A model in **aiosnow** must be declared a *type* and one or more *fields*. Once instantiated, an interface for reading and altering data in ServiceNow is provided.

### 4.3.1 Types

#### Table

The *TableModel* defines items of, and provides means for working with, ServiceNow Table API resources.

*Reference*

#### API

**class** aiosnow.models.table.**TableModel** (*client: aiosnow.client.Client, attachment: bool = True, \*\*kwargs*)

**create** (*payload: dict*) → aiosnow.request.response.Response

Create a new record

**Parameters** **payload** – New record payload

**Returns** Response

**delete** (*selection: Union[aiosnow.query.condition.Condition, str]*) → aiosnow.request.response.Response

Delete matching record

**Parameters** **selection** – Condition or ID

**Returns** Response

**download\_file** (*selection: Union[aiosnow.query.selector.Selector, aiosnow.query.condition.Condition, str], dst\_dir: str = '.'*) → aiosnow.models.attachment.file.FileHandler

Download incident attachment

**Parameters**

- **selection** – Attachment selection
- **dst\_dir** – Destination directory

**Returns** FileHandler object

**get** (*selection: Union[aiosnow.query.selector.Selector, aiosnow.query.condition.Condition, str] = None, \*\*kwargs*) → aiosnow.request.response.Response

Buffered many

Fetch and store the entire result in memory.

Note: It's recommended to use the stream method when dealing with a large number of records.

**Keyword Arguments**

- **selection** – Aiosnow-compatible query
- **limit** (*int*) – Maximum number of records to return
- **offset** (*int*) – Starting record index

**Returns** Response

**get\_attachment** (*selection*: *Union[aiosnow.query.selector.Selector, aiosnow.query.condition.Condition, str]* = *None*, *\*\*kwargs*) → *aiosnow.request.response.Response*

Returns Response if the given condition yielded exactly one attachment

**Parameters**

- **selection** – Attachment selection
- **\*\*kwargs** – arguments to pass along to AttachmentModel

**Returns** Response object

**get\_attachments** (*selection*: *Union[aiosnow.query.selector.Selector, aiosnow.query.condition.Condition, str]* = *None*, *\*\*kwargs*) → *aiosnow.request.response.Response*

Returns list of attachments for this table

**Parameters**

- **selection** – Attachment selection
- **\*\*kwargs** – arguments to pass along to AttachmentModel

**Returns** Response object

**get\_object\_id** (*value*: *Union[aiosnow.query.selector.Selector, aiosnow.query.condition.Condition, str]*) → *str*

Get object id by str or Condition

Immediately return if value is of str type.

**Parameters** **value** – Condition or str**Returns** Object id

**get\_one** (*selection*: *Union[aiosnow.query.selector.Selector, aiosnow.query.condition.Condition, str]* = *None*, *\*\*kwargs*) → *aiosnow.request.response.Response*

Get one record

**Parameters** **selection** – aiosnow-compatible query**Returns** Response

**stream** (*selection*: *Union[aiosnow.query.selector.Selector, aiosnow.query.condition.Condition, str]* = *None*, *\*\*kwargs*) → *AsyncGenerator[T\_co, T\_contra]*

Stream-like async generator

Chunk size determines the number of records to fetch in one go. Setting a lower chunk size decreases memory usage, but increases the number of requests sent to the server.

**Keyword Arguments**

- **selection** – aiosnow-compatible query
- **limit** (*int*) – Maximum number of records to return
- **offset** (*int*) – Starting record index

- **page\_size** (*int*) – Number of records to fetch in one go

**Yields** Chunk of records

**update** (*selection: Union[aiosnow.query.condition.Condition, str], payload: dict*) → aiosnow.request.response.Response  
Update matching record

**Parameters**

- **selection** – Condition or ID of object to update
- **payload** – Update payload

**Returns** Response

**upload\_file** (*selection: Union[aiosnow.query.selector.Selector, aiosnow.query.condition.Condition, str], path: str*) → aiosnow.request.response.Response  
Upload incident attachment

**Parameters**

- **selection** – Attachment selection
- **path** – Source file path

**Returns** Response

## Schemas

A growing collection of declared *TableModel* classes is included with the library in `aiosnow.models.table.declared`, these classes are typically subclassed and adjusted to reflect the ServiceNow instance at hand, but can be instantiated and used directly as well.

### Incident

ServiceNow default Incident definition. Used by `aiosnow.models.table.declared.incident.IncidentModel`.

```
class aiosnow.models.table.declared.incident.IncidentModelSchema
```

```
    active = <Boolean [maps_to=Pluck.VALUE, primary=False]>
    activity_due = <DateTime [maps_to=Pluck.VALUE, primary=False]>
    additional_assignee_list = <String [maps_to=Pluck.VALUE, primary=False]>
    approval = <String [maps_to=Pluck.VALUE, primary=False]>
    approval_history = <String [maps_to=Pluck.VALUE, primary=False]>
    approval_set = <DateTime [maps_to=Pluck.VALUE, primary=False]>
    assigned_to = <String [maps_to=Pluck.VALUE, primary=False]>
    assignment_group = <String [maps_to=Pluck.VALUE, primary=False]>
    business_duration = <DateTime [maps_to=Pluck.VALUE, primary=False]>
    business_service = <String [maps_to=Pluck.VALUE, primary=False]>
    business_stc = <Integer [maps_to=Pluck.VALUE, primary=False]>
    calendar_duration = <DateTime [maps_to=Pluck.VALUE, primary=False]>
```



```
calendar_stc = <Integer [maps_to=Pluck.VALUE, primary=False]>
caller_id = <String [maps_to=Pluck.VALUE, primary=False]>
category = <String [maps_to=Pluck.VALUE, primary=False]>
caused_by = <String [maps_to=Pluck.VALUE, primary=False]>
child_incidents = <Integer [maps_to=Pluck.VALUE, primary=False]>
close_code = <String [maps_to=Pluck.VALUE, primary=False]>
close_notes = <String [maps_to=Pluck.VALUE, primary=False]>
closed_at = <DateTime [maps_to=Pluck.VALUE, primary=False]>
closed_by = <String [maps_to=Pluck.VALUE, primary=False]>
cmdb_ci = <String [maps_to=Pluck.VALUE, primary=False]>
comments = <String [maps_to=Pluck.VALUE, primary=False]>
comments_and_work_notes = <String [maps_to=Pluck.VALUE, primary=False]>
company = <String [maps_to=Pluck.VALUE, primary=False]>
contact_type = <String [maps_to=Pluck.VALUE, primary=False]>
correlation_display = <String [maps_to=Pluck.VALUE, primary=False]>
correlation_id = <String [maps_to=Pluck.VALUE, primary=False]>
delivery_plan = <String [maps_to=Pluck.VALUE, primary=False]>
delivery_task = <String [maps_to=Pluck.VALUE, primary=False]>
description = <String [maps_to=Pluck.VALUE, primary=False]>
due_date = <DateTime [maps_to=Pluck.VALUE, primary=False]>
escalation = <Integer [maps_to=Pluck.VALUE, primary=False]>
expected_start = <DateTime [maps_to=Pluck.VALUE, primary=False]>
follow_up = <DateTime [maps_to=Pluck.VALUE, primary=False]>
group_list = <String [maps_to=Pluck.VALUE, primary=False]>
hold_reason = <Integer [maps_to=Pluck.VALUE, primary=False]>
impact = <IntegerMap [maps_to=Pluck.VALUE, primary=False]>
incident_state = <IntegerMap [maps_to=Pluck.VALUE, primary=False]>
knowledge = <Boolean [maps_to=Pluck.VALUE, primary=False]>
location = <String [maps_to=Pluck.VALUE, primary=False]>
made_sla = <Boolean [maps_to=Pluck.VALUE, primary=False]>
notify = <Integer [maps_to=Pluck.VALUE, primary=False]>
number = <String [maps_to=Pluck.VALUE, primary=False]>
opened_at = <DateTime [maps_to=Pluck.VALUE, primary=False]>
opened_by = <String [maps_to=Pluck.VALUE, primary=False]>
order = <Integer [maps_to=Pluck.VALUE, primary=False]>
parent = <String [maps_to=Pluck.VALUE, primary=False]>
```

```
parent_incident = <String [maps_to=Pluck.VALUE, primary=False]>
priority = <IntegerMap [maps_to=Pluck.VALUE, primary=False]>
problem_id = <String [maps_to=Pluck.VALUE, primary=False]>
reassignment_count = <Integer [maps_to=Pluck.VALUE, primary=False]>
reopen_count = <Integer [maps_to=Pluck.VALUE, primary=False]>
reopened_by = <String [maps_to=Pluck.VALUE, primary=False]>
reopened_time = <DateTime [maps_to=Pluck.VALUE, primary=False]>
resolved_at = <DateTime [maps_to=Pluck.VALUE, primary=False]>
resolved_by = <String [maps_to=Pluck.VALUE, primary=False]>
rfc = <String [maps_to=Pluck.VALUE, primary=False]>
severity = <IntegerMap [maps_to=Pluck.VALUE, primary=False]>
short_description = <String [maps_to=Pluck.VALUE, primary=False]>
sla_due = <DateTime [maps_to=Pluck.VALUE, primary=False]>
state = <IntegerMap [maps_to=Pluck.VALUE, primary=False]>
subcategory = <String [maps_to=Pluck.VALUE, primary=False]>
sys_class_name = <String [maps_to=Pluck.VALUE, primary=False]>
sys_created_by = <String [maps_to=Pluck.VALUE, primary=False]>
sys_created_on = <DateTime [maps_to=Pluck.VALUE, primary=False]>
sys_domain = <String [maps_to=Pluck.VALUE, primary=False]>
sys_domain_path = <String [maps_to=Pluck.VALUE, primary=False]>
sys_id = <String [maps_to=Pluck.VALUE, primary=True]>
sys_mod_count = <Integer [maps_to=Pluck.VALUE, primary=False]>
sys_tags = <String [maps_to=Pluck.VALUE, primary=False]>
sys_updated_by = <String [maps_to=Pluck.VALUE, primary=False]>
sys_updated_on = <DateTime [maps_to=Pluck.VALUE, primary=False]>
time_worked = <Integer [maps_to=Pluck.VALUE, primary=False]>
upon_approval = <String [maps_to=Pluck.VALUE, primary=False]>
upon_reject = <String [maps_to=Pluck.VALUE, primary=False]>
urgency = <IntegerMap [maps_to=Pluck.VALUE, primary=False]>
user_input = <String [maps_to=Pluck.VALUE, primary=False]>
watch_list = <String [maps_to=Pluck.VALUE, primary=False]>
work_end = <DateTime [maps_to=Pluck.VALUE, primary=False]>
work_notes = <String [maps_to=Pluck.VALUE, primary=False]>
work_notes_list = <String [maps_to=Pluck.VALUE, primary=False]>
work_start = <DateTime [maps_to=Pluck.VALUE, primary=False]>
```

## Journal

ServiceNow default Journal definition, used for working with collections of work\_notes, comments, etc. Used by *aiosnow.models.table.declared.journal.JournalModel*.

```
class aiosnow.models.table.declared.JournalModelSchema
```

```

    element = <String [maps_to=Pluck.VALUE, primary=False]>
    element_id = <String [maps_to=Pluck.VALUE, primary=False]>
    name = <String [maps_to=Pluck.VALUE, primary=False]>
    sys_created_by = <String [maps_to=Pluck.VALUE, primary=False]>
    sys_created_on = <DateTime [maps_to=Pluck.VALUE, primary=False]>
    sys_id = <String [maps_to=Pluck.VALUE, primary=True]>
    sys_tags = <String [maps_to=Pluck.VALUE, primary=False]>
    value = <String [maps_to=Pluck.VALUE, primary=False]>

```

## Attachment

The *AttachmentModel* is typically used in context of another API, e.g. via a *TableModel* type, but can be used directly in a global context as well.

*Reference*

## API

```
class aiosnow.models.attachment.model.AttachmentModel(*args, **kwargs)
    Attachment API Model
```

```
delete (selection: Union[aiosnow.query.condition.Condition, str]) → aios-
    now.request.response.Response
    Delete matching record
```

**Parameters** **selection** – Condition or ID

**Returns** Response

```
download (selection: Union[aiosnow.query.selector.Selector, aiosnow.query.condition.Condition, str],
    dst_dir: str = '.') → aiosnow.models.attachment.file.FileHandler
    Download file
```

**Parameters**

- **selection** – Attachment selection
- **dst\_dir** – Destination directory

Returns: FileWriter

```
get (selection: Union[aiosnow.query.selector.Selector, aiosnow.query.condition.Condition, str] = None,
    **kwargs) → aiosnow.request.response.Response
    Buffered many
```

Fetch and store the entire result in memory.

Note: It's recommended to use the stream method when dealing with a large number of records.

**Keyword Arguments**

- **selection** – Aiosnow-compatible query
- **limit** (*int*) – Maximum number of records to return
- **offset** (*int*) – Starting record index

**Returns** Response

**get\_object\_id** (*value*: *Union[aiosnow.query.selector.Selector, aiosnow.query.condition.Condition, str]*) → *str*

Get object id by str or Condition

Immediately return if value is of str type.

**Parameters** **value** – Condition or str

**Returns** Object id

**get\_one** (*selection*: *Union[aiosnow.query.selector.Selector, aiosnow.query.condition.Condition, str]* = *None*, *\*\*kwargs*) → *aiosnow.request.response.Response*

Get one record

**Parameters** **selection** – aiosnow-compatible query

**Returns** Response

**stream** (*selection*: *Union[aiosnow.query.selector.Selector, aiosnow.query.condition.Condition, str]* = *None*, *\*\*kwargs*) → *AsyncGenerator[T\_co, T\_contra]*

Stream-like async generator

Chunk size determines the number of records to fetch in one go. Setting a lower chunk size decreases memory usage, but increases the number of requests sent to the server.

**Keyword Arguments**

- **selection** – aiosnow-compatible query
- **limit** (*int*) – Maximum number of records to return
- **offset** (*int*) – Starting record index
- **page\_size** (*int*) – Number of records to fetch in one go

**Yields** Chunk of records

**upload** (*table\_name*: *str*, *record\_sys\_id*: *str*, *file\_name*: *str*, *dir\_name*: *str*) → *aiosnow.request.response.Response*

Upload file

**Parameters**

- **table\_name** – Table name, e.g. incident
- **record\_sys\_id** – Sys id of the record to attach to
- **file\_name** – Source file name
- **dir\_name** – Source directory name

Returns: ClientResponse

## Schema

**class** `aiosnow.models.attachment.model.AttachmentModelSchema`  
Attachment API model schema

```

average_image_color = <String [maps_to=Pluck.VALUE, primary=False]>
chunk_size_bytes = <String [maps_to=Pluck.VALUE, primary=False]>
compressed = <Boolean [maps_to=Pluck.VALUE, primary=False]>
content_type = <String [maps_to=Pluck.VALUE, primary=False]>
download_link = <String [maps_to=Pluck.VALUE, primary=False]>
file_name = <String [maps_to=Pluck.VALUE, primary=False]>
hash = <String [maps_to=Pluck.VALUE, primary=False]>
image_height = <String [maps_to=Pluck.VALUE, primary=False]>
image_width = <String [maps_to=Pluck.VALUE, primary=False]>
size_bytes = <Integer [maps_to=Pluck.VALUE, primary=False]>
size_compressed = <String [maps_to=Pluck.VALUE, primary=False]>
state = <String [maps_to=Pluck.VALUE, primary=False]>
sys_created_by = <String [maps_to=Pluck.VALUE, primary=False]>
sys_created_on = <DateTime [maps_to=Pluck.VALUE, primary=False]>
sys_id = <String [maps_to=Pluck.VALUE, primary=True]>
sys_mod_count = <String [maps_to=Pluck.VALUE, primary=False]>
sys_tags = <String [maps_to=Pluck.VALUE, primary=False]>
sys_updated_by = <String [maps_to=Pluck.VALUE, primary=False]>
sys_updated_on = <DateTime [maps_to=Pluck.VALUE, primary=False]>
table_name = <String [maps_to=Pluck.VALUE, primary=False]>
table_sys_id = <String [maps_to=Pluck.VALUE, primary=False]>

```

## Aggregate

TBC

## Importset

TBC

## 4.4 Fields

The *aiosnow.fields* module is a collection of typed field classes for:

- Declaring Models
- Building queries
- Serializing data

*Classes*

### 4.4.1 String

**class** aiosnow.fields.string.**StringQueryable**

**between** (*value1: str, value2: str*) → aiosnow.query.condition.Condition

Example: short\_description.between("q", "t")

All records in which the first letter in the Short description field is "q," "r," "s," or "t."

**contains** (*value: str*) → aiosnow.query.condition.Condition

Example: short\_description.contains("SAP")

All records in which the characters "SAP" appear anywhere in the value for the Short description field.

**ends\_with** (*value: str*) → aiosnow.query.condition.Condition

Example: short\_description.ends\_with("outage")

All records in which the string "outage" appears at the end of the value for the Short description field.

**equals** (*value: str*) → aiosnow.query.condition.Condition

Example: short\_description.equals("Network storage is unavailable")

All records in which the Short description says nothing else but "Network storage is unavailable."

**greater\_or\_equals** (*value: str*) → aiosnow.query.condition.Condition

Example: short\_description.greater\_or\_equals("s")

**All records in which the string in the Short description field is one of the following:**

- the first letter is any letter between "s" and "z"
- the exact value is "s"

**less\_or\_equals** (*value: str*) → aiosnow.query.condition.Condition

Example: short\_description.less\_or\_equals("s")

**All records in which the string in the Short description field is one of the following:**

- the first letter is any letter between "a" and "s"
- the exact value is "s"

**not\_contains** (*value: str*) → aiosnow.query.condition.Condition

Example: short\_description.not\_contains("SAP")

All records in which the characters "SAP" do not appear anywhere in the value for the Short description field.

**not\_equals** (*value: str*) → aiosnow.query.condition.Condition

Example: short\_description.not\_equals("Network storage is unavailable")

All records in which the value for the Short description field says anything but "Network storage is unavailable."

**starts\_with** (*value: str*) → aiosnow.query.condition.Condition

Example: short\_description.starts\_with("SAP")

All records in which the characters "SAP" appear at the beginning of the value for the Short description field.

**class** aiosnow.fields.string.**String** (\*args, pluck: aiosnow.models.\_schema.helpers.pluck.Pluck  
= <Pluck.VALUE: 'value'>, is\_primary: bool = False,  
\*\*kwargs)

## 4.4.2 Integer

**class** aiosnow.fields.integer.IntegerQueryable

**between** (*value1: int, value2: int*) → aiosnow.query.condition.Condition

Example: impact.between([1, 3])

All records in which the Impact field has one of the following values: 1 - High 2 - Medium 3 - Low

**equals** (*value: int*) → aiosnow.query.condition.Condition

Example: reassignment\_count.equals(0)

All records in which the Reassignment count is nothing else but 0.

**greater\_or\_equals** (*value: int*) → aiosnow.query.condition.Condition

Example: impact.greater\_or\_equals(2)

All records in which the Impact field has a value of 2 - Medium or 3 - Low.

**greater\_than** (*value: int*) → aiosnow.query.condition.Condition

Example: impact.greater\_than(2)

All records in which the Impact field has a value of 3 - Low

**less\_or\_equals** (*value: int*) → aiosnow.query.condition.Condition

Example: impact.less\_or\_equals(2)

All records in which the Impact field has a value of 1 - High or 2 - Medium.

**less\_than** (*value: int*) → aiosnow.query.condition.Condition

Example: impact.less\_than(2)

All records in which the Impact field has a value of 1 - High.

**not\_equals** (*value: int*) → aiosnow.query.condition.Condition

Example: reassignment\_count.not\_equals(0)

All records in which the value for the Reassignment count is any number but 0.

**class** aiosnow.fields.integer.Integer (\*, *strict: bool = False*, \*\*kwargs)

## 4.4.3 DateTime

**class** aiosnow.fields.datetime.DateTimeQueryable

**after** (*value: str*) → aiosnow.query.condition.Condition

Example: sla\_due.after("2019-12-24 02:03:04")

All records in which the value for the SLA due field is any date after the given date.

**as\_of** (*value: str*) → aiosnow.query.condition.Condition

Example: sla\_due.as\_of("2019-12-24 02:03:04")

**All records in which the date value for the SLA due field is one of the following:**

- today
- any date after today

**before** (*value: str*) → aiosnow.query.condition.Condition

Example: sla\_due.before("2019-12-24 02:03:04")

All records in which the value for the SLA due field is any date previous to the given date.

**between** (*value1: str, value2: str*) → aiosnow.query.condition.Condition

Example:

All records in which the value for the SLA due field is between the given dates

**not\_on** (*value: str*) → aiosnow.query.condition.Condition

Example: sla\_due.not\_on("2019-12-24 02:03:04")

All records in which the value for the SLA due field is any other but the given date

**on** (*value: str*) → aiosnow.query.condition.Condition

Example: sla\_due.on("2019-12-24 02:03:04")

All records in which the value for the SLA due field matches the given date

**until** (*value: str*) → aiosnow.query.condition.Condition

Example: sla\_due.until("2019-12-24 02:03:04")

All records in which the value for the SLA due field is one of the following:

- any date previous to today
- today

**class** aiosnow.fields.datetime.**DateTime** (*format: Optional[str] = None, \*\*kwargs*)

#### 4.4.4 Boolean

**class** aiosnow.fields.boolean.**BooleanQueryable**

**is\_falsy** () → aiosnow.query.condition.Condition

Example: active.is\_falsy()

All records in which the Active field is False, empty, or NULL.

**is\_true** () → aiosnow.query.condition.Condition

Example: active.is\_true()

All records in which the Active field is True.

**class** aiosnow.fields.boolean.**Boolean** (\*, *truthy: Optional[Set[T]] = None, falsy: Optional[Set[T]] = None, \*\*kwargs*)

#### 4.4.5 IntegerMap

Converts nested mapping to a named tuple.

Response content of:

```
{
  "impact": {
    "value": 2,
    "display_value": "2 - Medium"
  }
}
```

Gets deserialized into:

```
Mapping(key=2, value="2 - Medium")
```



### 4.4.6 StringMap

Converts nested mapping to a named tuple.

Response content of:

```
{
  "impact": {
    "value": 2,
    "display_value": "2 - Medium"
  }
}
```

Gets deserialized into:

```
Mapping(key="2", value="2 - Medium")
```

## 4.5 Condition

The *Condition* class is mainly for creating queries, see [query usage](#) for examples.

**class** aiosnow.query.condition.**Condition** (*key: str, operator: str, value: Union[str, int, None]*)  
Bases: object

**\_\_and\_\_** (*next\_cond: aiosnow.query.condition.Condition*) → aiosnow.query.condition.Condition  
Appends ^ Condition to chain

**\_\_or\_\_** (*next\_cond: aiosnow.query.condition.Condition*) → aiosnow.query.condition.Condition  
Appends ^OR Condition to chain

**\_\_str\_\_** () → str  
Return str(self).

**\_\_xor\_\_** (*next\_cond: aiosnow.query.condition.Condition*) → aiosnow.query.condition.Condition  
Appends ^NQ Condition to chain

**serialize** (*cond: Optional[aiosnow.query.condition.Condition] = None*) → str  
Serializes this Condition into a sysparm query condition  
Returns: sysparm query string

**serialize\_registry** () → str  
Condition string representation  
Returns: sysparm query

## 4.6 Response

ServiceNow API operations performed by **aiosnow** returns an instance of *Response*.

### 4.6.1 API

**class** aiosnow.request.response.**Response** (*\*args, \*\*kwargs*)  
Aiosnow Response class

The Response object holds information about the ServiceNow HTTP response.

Subclass of `aiohttp.ClientResponse`, its base reference documentation can be found here: [https://docs.aiohttp.org/en/latest/client\\_reference.html#aiohttp.ClientResponse](https://docs.aiohttp.org/en/latest/client_reference.html#aiohttp.ClientResponse)

- **data**  
Deserialized (`ContentSchema`) response content
  - **status**  
HTTP status code of response (int), e.g. 200
  - **reason**  
HTTP status reason of response (str), e.g. "OK"
  - **url**  
Request URL
- load\_document** () → None  
Deserialize and set response content
- Raises** `RequestError` – If there was an error in the request-response content

## 4.7 Exceptions

- exception** `aiosnow.exceptions.AiosnowException`  
Bases: `Exception`
- exception** `aiosnow.exceptions.AmbiguousClientAuthentication`  
Bases: `aiosnow.exceptions.AiosnowException`  
Multiple authentication methods provided
- exception** `aiosnow.exceptions.ClientConnectionError`  
Bases: `aiosnow.exceptions.AiosnowException`  
Raised when there was a problem connecting to the server
- exception** `aiosnow.exceptions.ConfigurationError`  
Bases: `aiosnow.exceptions.AiosnowException`  
Configuration error
- exception** `aiosnow.exceptions.DeleteError`  
Bases: `aiosnow.exceptions.AiosnowException`  
Raised if there was an error deleting a record
- exception** `aiosnow.exceptions.DeserializationError`  
Bases: `aiosnow.exceptions.AiosnowException`  
Raised when there was an issue with deserialization
- exception** `aiosnow.exceptions.ErrorResponse` (*message: str, status: int*)  
Bases: `aiosnow.exceptions.RequestError`  
The server returned an error in the response
- exception** `aiosnow.exceptions.IncompatiblePayloadField`  
Bases: `aiosnow.exceptions.AiosnowException`  
An incompatible field was found in the payload
- exception** `aiosnow.exceptions.InvalidContentMethod`  
Bases: `aiosnow.exceptions.AiosnowException`

Raised if the response content was incorrectly accessed

**exception** `aiosnow.exceptions.InvalidFieldName`

Bases: `aiosnow.exceptions.AiosnowException`

Usually raised if an attempt is made to override a base member with a Field

**exception** `aiosnow.exceptions.MissingClientAuthentication`

Bases: `aiosnow.exceptions.AiosnowException`

No authentication method provided

**exception** `aiosnow.exceptions.NoItems`

Bases: `aiosnow.exceptions.AiosnowException`

The request yielded no results

**exception** `aiosnow.exceptions.PayloadValidationError`

Bases: `aiosnow.exceptions.AiosnowException`

Local payload validation against a Resource Schema failed

**exception** `aiosnow.exceptions.RequestError` (*message: str, status: int*)

Bases: `aiosnow.exceptions.AiosnowException`

The application returned a parsable error in the response

**exception** `aiosnow.exceptions.SchemaError`

Bases: `aiosnow.exceptions.AiosnowException`

Generic exception raised on schema issues, e.g. integrity errors

**exception** `aiosnow.exceptions.SelectError`

Bases: `aiosnow.exceptions.AiosnowException`

Raised on query builder issues

**exception** `aiosnow.exceptions.SerializationError`

Bases: `aiosnow.exceptions.AiosnowException`

Raised when there was an issue with serialization

**exception** `aiosnow.exceptions.TooManyItems`

Bases: `aiosnow.exceptions.AiosnowException`

The request yielded too many results

**exception** `aiosnow.exceptions.UnexpectedContentType`

Bases: `aiosnow.exceptions.AiosnowException`

Unexpected content type from server

**exception** `aiosnow.exceptions.UnexpectedPayloadType`

Bases: `aiosnow.exceptions.AiosnowException`

Raised when the request payload was of an unexpected type

**exception** `aiosnow.exceptions.UnexpectedResponseContent` (*message: str, status: int*)

Bases: `aiosnow.exceptions.RequestError`

Unexpected content in response from server

**exception** `aiosnow.exceptions.UnexpectedValue`

Bases: `aiosnow.exceptions.AiosnowException`

Typically raised when a method receives unexpected input

**exception** aiosnow.exceptions.UnknownPayloadField

Bases: *aiosnow.exceptions.AiosnowException*

A field unknown to the schema was found in the payload

## 4.8 Connect

To connect to ServiceNow, a *Client* must be created.

```
import aiosnow

client = aiosnow.Client(
    address="<instance-name>.service-now.com",
    basic_auth=("<username>", "<password>")
)
```

## 4.9 Model

The Model object provides an interface for reading and altering data in ServiceNow.

*Usage*

### 4.9.1 Declare

The Model Declaration is done by making one or more *aiosnow.fields* members of the Model.

#### Model

Assign fields directly on the *BaseModel* derived class.

```
from aiosnow import TableModel, fields

class IncidentModel(TableModel):
    sys_id = fields.String(is_primary=True)
    number = fields.String()
    impact = fields.IntegerMap()
    priority = fields.Integer()
    assignment_group = fields.StringMap()
    sys_created_on = fields.DateTime()
    made_sla = fields.Boolean()
```

#### ModelSchema

Assign fields in separate class, which is then subclassed.

```
from aiosnow import TableModel, fields

class IncidentModelSchema:
    sys_id = fields.String(is_primary=True)
    number = fields.String()
```

(continues on next page)

(continued from previous page)

```

    impact = fields.IntegerMap()
    priority = fields.Integer()
    assignment_group = fields.StringMap()
    sys_created_on = fields.DateTime()
    made_sla = fields.Boolean()

class IncidentModel(TableModel, IncidentModelSchema):
    """Incident API Model"""

```

## 4.9.2 Use

A declared Model can be instantiated for reading and altering data in ServiceNow.

*Example*

```

async with Incident(client, table_name="incident") as inc:
    response = await inc.get_one(Incident.number == "INC0000001")
    print(response.data)

```

## 4.10 Query

The *aiosnow.fields* classes provides a set of type-specific methods for creating queries. Calling such a method returns a *Condition* with the string representation of a ServiceNow sysparm.

Conditions can be chained using bitwise operators for AND, OR and XOR.

*Usage*

### 4.10.1 Single

Single conditions are typically used when updating, deleting or fetching a specific item by a unique identifier.

*Example*

```

async with Incident(client, table_name="incident") as inc:
    response = await inc.update(
        Incident.number == "INC0000001",
        dict(
            description="Hello!"
        )
    )
    print("Updated description: {}".format(response["description"]))

```

### 4.10.2 List

When querying for a list of items, `aiosnow.select()` with chained conditions can be used.

*Example*

```
query = aiosnow.select(
    Incident.number.starts_with("INC123")
    &
    Incident.impact.less_than(3)
    ^
    Incident.assignment_group.name.equals("Hardware")
).order_desc(Incident.number)

async with Incident(client, table_name="incident") as inc:
    response = await inc.get(query, limit=10)
    print("The query yielded {} items.".format(len(response.data)))
```

### 4.10.3 Raw

Shows how a sysparm query string can be passed directly to a query method. See the *Condition* equivalent [here](#).

```
async with Incident(client, table_name="incident") as inc:
    response = await inc.get(
        "numberSTARTSWITHINC123^impact<3^NQassignment_group.name=Hardware^
↪ORDERBYDESCnumber"
    )
    print("The query yielded {} items.".format(len(response.data)))
```

### a

- `aiosnow`, [9](#)
- `aiosnow.exceptions`, [22](#)
- `aiosnow.fields.boolean`, [20](#)
- `aiosnow.fields.datetime`, [19](#)
- `aiosnow.fields.integer`, [19](#)
- `aiosnow.fields.string`, [18](#)
- `aiosnow.request.response`, [21](#)





## Symbols

`__and__()` (*aiosnow.query.condition.Condition* method), 21

`__or__()` (*aiosnow.query.condition.Condition* method), 21

`__str__()` (*aiosnow.query.condition.Condition* method), 21

`__xor__()` (*aiosnow.query.condition.Condition* method), 21

## A

`active` (*aiosnow.models.table.declared.incident.IncidentModelSchema* attribute), 12

`activity_due` (*aiosnow.models.table.declared.incident.IncidentModelSchema* attribute), 12

`additional_assignee_list` (*aiosnow.models.table.declared.incident.IncidentModelSchema* attribute), 12

`after()` (*aiosnow.fields.datetime.DateTimeQueryable* method), 19

`aiosnow` (module), 9

`aiosnow.exceptions` (module), 22

`aiosnow.fields.boolean` (module), 20

`aiosnow.fields.datetime` (module), 19

`aiosnow.fields.integer` (module), 19

`aiosnow.fields.string` (module), 18

`aiosnow.request.response` (module), 21

`AiosnowException`, 22

`AmbiguousClientAuthentication`, 22

`approval` (*aiosnow.models.table.declared.incident.IncidentModelSchema* attribute), 12

`approval_history` (*aiosnow.models.table.declared.incident.IncidentModelSchema* attribute), 12

`approval_set` (*aiosnow.models.table.declared.incident.IncidentModelSchema* attribute), 12

`as_of()` (*aiosnow.fields.datetime.DateTimeQueryable*

method), 19

`assigned_to` (*aiosnow.models.table.declared.incident.IncidentModelSchema* attribute), 12

`assignment_group` (*aiosnow.models.table.declared.incident.IncidentModelSchema* attribute), 12

`AttachmentModel` (class in *aiosnow.models.attachment.model*), 15

`AttachmentModelSchema` (class in *aiosnow.models.attachment.model*), 16

`average_image_color` (*aiosnow.models.attachment.model.AttachmentModelSchema* attribute), 16

## B

`before()` (*aiosnow.fields.datetime.DateTimeQueryable* method), 19

`between()` (*aiosnow.fields.datetime.DateTimeQueryable* method), 20

`between()` (*aiosnow.fields.integer.IntegerQueryable* method), 19

`between()` (*aiosnow.fields.string.StringQueryable* method), 18

`Boolean` (class in *aiosnow.fields.boolean*), 20

`BooleanQueryable` (class in *aiosnow.fields.boolean*), 20

`business_duration` (*aiosnow.models.table.declared.incident.IncidentModelSchema* attribute), 12

`business_service` (*aiosnow.models.table.declared.incident.IncidentModelSchema* attribute), 12

`business_stc` (*aiosnow.models.table.declared.incident.IncidentModelSchema* attribute), 12

## C

`calendar_duration` (*aiosnow.models.table.declared.incident.IncidentModelSchema* attribute), 12

calendar\_stc (aios- 10  
 now.models.table.declared.incident.IncidentModelSchema  
 attribute), 12

caller\_id (aiosnow.models.table.declared.incident.IncidentModelSchema (class in aiosnow.fields.datetime), 20  
 attribute), 13

category (aiosnow.models.table.declared.incident.IncidentModelSchema (aiosnow.fields.datetime), 19  
 attribute), 13

caused\_by (aiosnow.models.table.declared.incident.IncidentModelSchema), 15  
 attribute), 13

child\_incidents (aios- 10  
 now.models.table.declared.incident.IncidentModelSchema  
 attribute), 13

chunk\_size\_bytes (aios- now.models.table.declared.incident.IncidentModelSchema  
 now.models.attachment.model.AttachmentModelSchema attribute), 13  
 attribute), 17

Client (class in aiosnow), 9

ClientConnectionError, 22

close\_code (aiosnow.models.table.declared.incident.IncidentModelSchema (aiosnow.models.table.declared.incident.IncidentModelSchema  
 attribute), 13

close\_notes (aiosnow.models.table.declared.incident.IncidentModelSchema  
 attribute), 13

closed\_at (aiosnow.models.table.declared.incident.IncidentModelSchema), 15  
 attribute), 13

closed\_by (aiosnow.models.table.declared.incident.IncidentModelSchema), 10  
 attribute), 13

cmdb\_ci (aiosnow.models.table.declared.incident.IncidentModelSchema now.models.attachment.model.AttachmentModelSchema  
 attribute), 13

comments (aiosnow.models.table.declared.incident.IncidentModelSchema (aiosnow.models.table.declared.incident.IncidentModelSchema  
 attribute), 13

comments\_and\_work\_notes (aios-  
 now.models.table.declared.incident.IncidentModelSchema  
 attribute), 13

company (aiosnow.models.table.declared.incident.IncidentModelSchema attribute), 15  
 attribute), 13

compressed (aiosnow.models.attachment.model.AttachmentModelSchema attribute), 15  
 attribute), 17

Condition (class in aiosnow.query.condition), 21

config (aiosnow.Client attribute), 10

ConfigurationError, 22

contact\_type (aios- equals () (aiosnow.fields.string.StringQueryable  
 now.models.table.declared.incident.IncidentModelSchema method), 18  
 attribute), 13

contains () (aiosnow.fields.string.StringQueryable  
 method), 18

content\_type (aios- expected\_start (aios-  
 now.models.attachment.model.AttachmentModelSchema now.models.table.declared.incident.IncidentModelSchema  
 attribute), 17 attribute), 13

correlation\_display (aios-  
 now.models.table.declared.incident.IncidentModelSchema  
 attribute), 13

correlation\_id (aios- file\_name (aiosnow.models.attachment.model.AttachmentModelSchema  
 now.models.table.declared.incident.IncidentModelSchema attribute), 17  
 attribute), 13

create () (aiosnow.models.table.TableModel method), follow\_up (aiosnow.models.table.declared.incident.IncidentModelSchema  
 attribute), 13

## G

`get()` (*aiosnow.models.attachment.model.AttachmentModel* method), 15

`get()` (*aiosnow.models.table.TableModel* method), 10

`get_attachment()` (*aiosnow.models.table.TableModel* method), 11

`get_attachments()` (*aiosnow.models.table.TableModel* method), 11

`get_object_id()` (*aiosnow.models.attachment.model.AttachmentModel* method), 16

`get_object_id()` (*aiosnow.models.table.TableModel* method), 11

`get_one()` (*aiosnow.models.attachment.model.AttachmentModel* method), 16

`get_one()` (*aiosnow.models.table.TableModel* method), 11

`greater_or_equals()` (*aiosnow.fields.integer.IntegerQueryable* method), 19

`greater_or_equals()` (*aiosnow.fields.string.StringQueryable* method), 18

`greater_than()` (*aiosnow.fields.integer.IntegerQueryable* method), 19

`group_list` (*aiosnow.models.table.declared.incident.IncidentModelSchema* attribute), 13

## H

`hash` (*aiosnow.models.attachment.model.AttachmentModelSchema* attribute), 17

`hold_reason` (*aiosnow.models.table.declared.incident.IncidentModelSchema* attribute), 13

## I

`image_height` (*aiosnow.models.attachment.model.AttachmentModelSchema* attribute), 17

`image_width` (*aiosnow.models.attachment.model.AttachmentModelSchema* attribute), 17

`impact` (*aiosnow.models.table.declared.incident.IncidentModelSchema* attribute), 13

`incident_state` (*aiosnow.models.table.declared.incident.IncidentModelSchema* attribute), 13

`IncidentModelSchema` (class in *aiosnow.models.table.declared.incident*), 12

`IncompatiblePayloadField`, 22

`Integer` (class in *aiosnow.fields.integer*), 19

`IntegerQueryable` (class in *aiosnow.fields.integer*), 19

`InvalidContentMethod`, 22

`InvalidFieldName`, 23

`is_falsy()` (*aiosnow.fields.boolean.BooleanQueryable* method), 20

`is_true()` (*aiosnow.fields.boolean.BooleanQueryable* method), 20

## J

`JournalModelSchema` (class in *aiosnow.models.table.declared*), 15

## K

`knowledge` (*aiosnow.models.table.declared.incident.IncidentModelSchema* attribute), 13

## L

`less_or_equals()` (*aiosnow.fields.integer.IntegerQueryable* method), 19

`less_or_equals()` (*aiosnow.fields.string.StringQueryable* method), 18

`less_than()` (*aiosnow.fields.integer.IntegerQueryable* method), 19

`load_document()` (*aiosnow.request.response.Response* method), 22

`location` (*aiosnow.models.table.declared.incident.IncidentModelSchema* attribute), 13

## M

`made_sla` (*aiosnow.models.table.declared.incident.IncidentModelSchema* attribute), 13

`MissingClientAuthentication`, 23

## N

`name` (*aiosnow.models.table.declared.JournalModelSchema* attribute), 15

`NoItems`, 23

`not_contains()` (*aiosnow.fields.string.StringQueryable* method), 18

`not_equals()` (*aiosnow.fields.integer.IntegerQueryable* method), 19

`not_equals()` (*aiosnow.fields.string.StringQueryable* method), 18

`not_on()` (*aiosnow.fields.datetime.DateTimeQueryable* method), 20

`notify` (*aiosnow.models.table.declared.incident.IncidentModelSchema* attribute), 13

`number` (*aiosnow.models.table.declared.incident.IncidentModelSchema* attribute), 13

## O

[on\(\)](#) ([aiosnow.fields.datetime.DateTimeQueryable](#) [method](#)), 20  
[opened\\_at](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 13  
[opened\\_by](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 13  
[order](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 13  
[short\\_description](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 14  
[size\\_bytes](#) ([aiosnow.models.attachment.model.AttachmentModelSchema](#) [attribute](#)), 17  
[size\\_compressed](#) ([aiosnow.models.attachment.model.AttachmentModelSchema](#) [attribute](#)), 17  
[sla\\_due](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 14

## P

[parent](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 13  
[parent\\_incident](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 13  
[PayloadValidationError](#), 23  
[priority](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 14  
[problem\\_id](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 14  
[starts\\_with\(\)](#) ([aiosnow.fields.string.StringQueryable](#) [method](#)), 18  
[state](#) ([aiosnow.models.attachment.model.AttachmentModelSchema](#) [attribute](#)), 17  
[state](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 14  
[stream\(\)](#) ([aiosnow.models.attachment.model.AttachmentModelSchema](#) [method](#)), 16  
[stream\(\)](#) ([aiosnow.models.table.TableModel](#) [method](#)), 11

## R

[reassignment\\_count](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 14  
[reopen\\_count](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 14  
[reopened\\_by](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 14  
[reopened\\_time](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 14  
[RequestError](#), 23  
[resolved\\_at](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 14  
[resolved\\_by](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 14  
[Response](#) ([class in aiosnow.request.response](#)), 21  
[rfc](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 14  
[subcategory](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 14  
[sys\\_class\\_name](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 14  
[sys\\_created\\_by](#) ([aiosnow.models.attachment.model.AttachmentModelSchema](#) [attribute](#)), 17  
[sys\\_created\\_by](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 14  
[sys\\_created\\_by](#) ([aiosnow.models.table.declared.JournalModelSchema](#) [attribute](#)), 15  
[sys\\_created\\_on](#) ([aiosnow.models.attachment.model.AttachmentModelSchema](#) [attribute](#)), 17  
[sys\\_created\\_on](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 14

## S

[SchemaError](#), 23  
[SelectError](#), 23  
[SerializationError](#), 23  
[serialize\(\)](#) ([aiosnow.query.condition.Condition](#) [method](#)), 21  
[serialize\\_registry\(\)](#) ([aiosnow.query.condition.Condition](#) [method](#)), 21  
[severity](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 14  
[sys\\_created\\_on](#) ([aiosnow.models.table.declared.JournalModelSchema](#) [attribute](#)), 15  
[sys\\_domain](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 14  
[sys\\_domain\\_path](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 14  
[sys\\_id](#) ([aiosnow.models.attachment.model.AttachmentModelSchema](#) [attribute](#)), 17  
[sys\\_id](#) ([aiosnow.models.table.declared.incident.IncidentModelSchema](#) [attribute](#)), 14

