

---

# **swagger-py**

***Release 0.9.6***

**Oct 28, 2018**



---

## Contents

---

<b>1</b>	<b>Quickstart</b>	<b>3</b>
<b>2</b>	<b>Advanced options</b>	<b>5</b>
<b>3</b>	<b>Changelog</b>	<b>9</b>
<b>4</b>	<b>Configuring swagger-py</b>	<b>11</b>
<b>5</b>	<b>swaggerpy Package</b>	<b>13</b>
<b>6</b>	<b>Indices and tables</b>	<b>33</b>
	<b>Python Module Index</b>	<b>35</b>



Source code is available on [Github](#).

This project acts as a generic client library for services which follow *Swagger* schema.

More information on Swagger can be found [on the Swagger website](#)

It aims to be a complete replacement to [swagger codegen](#).

Features include:

- Synchronous and Asynchronous clients out of the box.
- Caching of api-docs with regular staleness check.
- Strict validations to check swagger spec is [v1.2](#) compatible.
- Validations on the parameter and response types.
- Request and Response values are handled with Python types (no need to deal with JSON).
- Doc strings are provided for Operations and Models to give more information about the API.
- Local file path to api-docs is also acceptable.

Contents:



### 1.1 Usage

Install directly from github as:

```
$ pip install --upgrade git+git://github.com/Yelp/bravado@swaggerpy
```

### 1.2 Your first Hello World! (or Hello Pet)

Here is a simple one to try from REPL (like IPython):

```
from swaggerpy import client
swagger_client = client.get_client(
    "http://petstore.swagger.wordnik.com/api/api-docs")
client.pet.getPetById(petId=42).result()
```

If you were lucky, and pet Id with 42 was present, you will get back a result. It will be an instance of `swaggerpy.swagger_model.Pet` with attributes `category`, etc. You can even try `result.category.id` or `result.tags[0]`.

Sample Response:

```
Pet(category=Category(id=0L, name=u''), status=u'', name=u'', tags=[Tag(id=0L, name=u'↪'), photoUrls=[u''], id=2)
```

If you got a 404, try some other petId.

## 1.3 Lets try a POST call

Here we will demonstrate how `swagger-py` hides all the JSON handling from the user, and makes the code more Pythonic.

```
Pet = swagger_client.pet.models.Pet
Category = swagger_client.pet.addPet['_models']['Category']
pet = Pet(id=42, name="tommy", category=Category(id=24))
swagger_client.pet.addPet(body=pet).result()
```

It should give a 200 response like: `{u'code': 200, u'message': u'SUCCESS'}`

## 1.4 Time to get Twisted! (Asynchronous client)

`swagger-py` gives an out of the box Asynchronous client to the user, with an optional timeout parameter.

*Your first Hello World! (or Hello Pet)* above can be rewritten to use Asynchronous client like so:

```
from swaggerpy import client
from swaggerpy.async_http_client import AsynchronousHttpClient
swagger_client = client.get_client(
    "http://petstore.swagger.wordnik.com/api/api-docs",
    AsynchronousHttpClient())
client.pet.getPetById(petId=42).result(timeout=4)
```

---

**Note:** `timeout` parameter here is the timeout (in seconds) the call will block waiting for complete response. The default time is 5 seconds.

---

## 1.5 This is too fancy for me! I want simple dict response!

`swagger-py` has taken care of that as well. `result._flat_dict()` results in complete dict response.

Hello Pet response would look like:

```
{'category': {'id': 0L, 'name': u''},
 'id': 2,
 'name': u'',
 'photoUrls': [u''],
 'status': u'',
 'tags': [{'id': 0L, 'name': u''}]}
```

---

**Note:** `result.__dict__` returns only one level dict conversion, hence should be avoided.

---



### 2.1 Validations

swagger-py validates the schema as per v1.2 swagger spec. Validations are also done on the requests and the responses.

Validation example:

```
pet = Pet(id="I should be integer :", name="tommy")
swagger_client.pet.addPet(body=pet).result()
```

will result in error like so:

```
TypeError: id's value: 'I should be integer :' should be in types (<type 'long'>,
↳ <type 'int'>)
```

---

**Note:** If you think it is acceptable for fields in your response to be null, and want the validator to ignore the type check you can add `allow_null=True` as a parameter to `result()`.

If response validations and type conversions are totally needed to be skipped, you can pass `raw_response=True` as a parameter to `result()` to get back raw API response.

---

### 2.2 Caching

swagger-py exposes a factory method `get_client` to give back the swagger client. It caches the api-docs responses so that they are not made on each API call. The default timeout is 300 seconds, which can be altered by passing “timeout” to `get_client`.

---

**Note:** Caching can totally be bypassed by using `swaggerpy.client.SwaggerClient()` directly.

---

```
client = SwaggerClient.from_url(  
    "http://petstore.swagger.wordnik.com/api/api-docs")
```

---

## 2.3 Adding Request Headers

swagger-py allows you to pass request headers along with any request.

```
Pet = swagger_client.pet.models.Pet  
Category = swagger_client.pet.models.Category  
pet = Pet(id=42, name="tommy", category=Category(id=24))  
swagger_client.pet.addPet(  
    body=pet,  
    _request_options={"headers": {"foo": "bar"}},  
).result()
```

## 2.4 Wrapping HTTP response error with custom class

swagger-py provided an option `raise_with` for wrapping HTTP errors with your custom Exception class. This is helpful for catching particular exception in your code or logging with particular exception class name.

```
class MyAwesomeException(Exception):  
    pass  
swagger_client = client.get_client(  
    "http://petstore.swagger.wordnik.com/api/api-docs",  
    raise_with=MyAwesomeException)
```

## 2.5 Passing Headers to the api-docs requests

swagger-py provides an option to pass custom headers with requests to api-docs

```
swagger_client = client.get_client(  
    "http://petstore.swagger.wordnik.com/api/api-docs",  
    api_doc_request_headers={'foo': 'bar'})
```

## 2.6 Docstrings

swagger-py provides docstrings to operations and models to quickly get the parameter and response types. A sample operation `getPetById` docstring looks like:

```
Docstring:  
[GET] Find pet by ID  
Returns a pet based on ID  
Args:  
    petId (int64) : ID of pet that needs to be fetched  
Returns:  
    Pet
```

(continues on next page)

(continued from previous page)

```

Raises:
    400: Invalid ID supplied
    404: Pet not found
Class Docstring:Operation object.
Call def:    c.pet.getPetById(self, kwargs)

```

Even the Pet model description can be found in the docstring:

```

Docstring:
Attributes:
    category (Category)
    status (str) : pet status in the store
    name (str)
    tags (list(Tag))
    photoUrls (list(str))
    id (long) : unique identifier for the pet
Constructor information:
    Definition:Pet(self, kwargs)

```

## 2.7 Default Values

swagger-py uses the default values from the spec if the value is not provided in the request.

In the [Pet Store](#) example, operation `findPetByStatus` has a `defaultValue` of `available`. That means, swagger-py will plug that value if no value is provided for the parameter. Example:

```
swagger_client.pet.findPetByStatus()
```

## 2.8 Api-docs from file path

swagger-py also accepts api-docs from file path. Like so:

```
client = client.get_client('file:///path/to/api-docs')
```

**Note:** This needs a nested level file structure. Resources should be present under `api-docs/`. File path should not have `.json` with the api-docs. It will be added by swagger-py. This feature is still in beta phase.

Other alternative way is by using helper method `load_file`. This doesn't need the resources to be nested.

```

from swaggerpy.swagger_model import load_file
client = client.get_client(load_file('/path/to/api-docs'))

```

**Note:** Both of the above methods also take an optional parameter `api_base_path` which can define the base path for the API call if `basePath` in schema is defined as `'/'`. It can be used like: `client.get_client('file:///path/to/api-docs', api_base_path='http://foo')`



### 3.1 0.7.10 (2015-07-07)

- Make request and response available as attrs on HTTPError

### 3.2 0.7.5 (2015-01-21)

- Handle request path parameters with spaces correctly
- Performance improvements for loading large api docs
- Misc bug fixes

### 3.3 0.7.4 (2014-12-11)

- Requests urlencode params as utf8
- Docs related to 0.7.2
- Declare utf-8 encoding for all files

### 3.4 0.7.3 (2014-12-11)

- request logging is now done on the debug level instead of info level.

### 3.5 0.7.2 (2014-12-11)

- Allow headers to be passed in the api\_docs request

### 3.6 0.7.1 (2014-12-11)

- Requests no longer mutate clients

### 3.7 0.7.0 (2014-11-26)

- headers are no longer cached and required as part of async and http client setup.

### 3.8 0.6.0 (2014-10-30)

- format='date' params are now represented and passed as datetime.date objects instead of datetime.datetimes.

### 3.9 0.5.0 (2014-08-08)

- Allow form request parameters. (Uploading files is supported)
- Default Values are taken if parameter not provided.
- Detailed exception error is raised (containing server response)
- New Optional parameters to `result(): allow_null` and `raw_response`.
- Headers passed to HttpClient will be passed to `/api-docs` call as well.

### 3.10 0.4.0 (2014-07-15)

- Allow MultiDict params. (for query parameters with `allowMultiple: True`)
- Query Parameters with type `array` are not further allowed.

## CHAPTER 4

---

### Configuring swagger-py

---

There are some configurations which can be handy.

```
# Default time in seconds api-docs is cached
swaggerpy.client.SWAGGER_SPEC_TIMEOUT_S = 300

# Default timeout in seconds for client to get complete response
swaggerpy.response.DEFAULT_TIMEOUT_S = 5.0
```





## 5.1 swaggerpy Package

## 5.2 http\_client Module

HTTP client abstractions.

```
class swaggerpy.http_client.ApiKeyAuthenticator (host, api_key,  
                                              param_name=u'api_key')
```

Bases: *swaggerpy.http\_client.Authenticator*

?api\_key authenticator.

This authenticator adds a query parameter to specify an API key.

**Parameters**

- **host** – Host to authenticate for.
- **api\_key** – API key.
- **param\_name** – Query parameter specifying the API key.

**apply** (*request*)

Apply authentication to a request.

**Parameters request** – Request to add authentication information to.

```
class swaggerpy.http_client.Authenticator (host)
```

Bases: *object*

Authenticates requests.

**Parameters host** – Host to authenticate for.

**apply** (*request*)

Apply authentication to a request.

**Parameters request** – Request to add authentication information to.

**matches** (*url*)

Returns true if this authenticator applies to the given url.

**Parameters** *url* – URL to check.

**Returns** True if matches host, port and scheme, False otherwise.

**class** `swaggerpy.http_client.BasicAuthenticator` (*host, username, password*)

Bases: `swaggerpy.http_client.Authenticator`

HTTP Basic authenticator.

**Parameters**

- **host** – Host to authenticate for.
- **username** – Username.
- **password** – Password

**apply** (*request*)

Apply authentication to a request.

**Parameters** *request* – Request to add authentication information to.

**class** `swaggerpy.http_client.HttpClient`

Bases: `object`

Interface for a minimal HTTP client.

**request** (*method, url, params=None, data=None*)

Issue an HTTP request.

**Parameters**

- **method** (*str*) – HTTP method (GET, POST, DELETE, etc.)
- **url** (*str*) – URL to request
- **params** (*dict*) – Query parameters (?key=value)
- **data** (*Dictionary, bytes, or file-like object*) – Request body

**Returns** Implementation specific response object

**set\_api\_key** (*host, api\_key, param\_name=u'api\_key'*)

Configures client to use api\_key authentication.

The api\_key is added to every query parameter sent.

**Parameters**

- **host** – Hostname to limit authentication to.
- **api\_key** – Value for api\_key.
- **param\_name** – Parameter name to use in query string.

**set\_basic\_auth** (*host, username, password*)

Configures client to use HTTP Basic authentication.

**Parameters**

- **host** – Hostname to limit authentication to.
- **username** – Username
- **password** – Password

**start\_request** (*request\_params*)

**Parameters** **request\_params** (*dict*) – Complete request data.

**Returns** The client's request object

**class** `swaggerpy.http_client.SynchronousEventual` (*session, request*)

Bases: `object`

An adapter which supports the `crochet.EventualResult` interface for the `SynchronousHttpClient` class.

**cancel** ()

**wait** (*timeout=None*)

Perform the request.

**Parameters** **timeout** – timeout for the request, in seconds

**class** `swaggerpy.http_client.SynchronousHttpClient`

Bases: `swaggerpy.http_client.HttpClient`

Synchronous HTTP client implementation.

**apply\_authentication** (*request*)

**authenticated\_request** (*request\_params*)

**set\_api\_key** (*host, api\_key, param\_name=u'api\_key'*)

Configures client to use `api_key` authentication.

The `api_key` is added to every query parameter sent.

**Parameters**

- **host** – Hostname to limit authentication to.
- **api\_key** – Value for `api_key`.
- **param\_name** – Parameter name to use in query string.

**set\_basic\_auth** (*host, username, password*)

Configures client to use HTTP Basic authentication.

**Parameters**

- **host** – Hostname to limit authentication to.
- **username** – Username
- **password** – Password

**start\_request** (*request\_params*)

**Returns** request

**Return type** `requests.Request`

## 5.3 `async_http_client` Module

Asynchronous HTTP client abstractions.

**class** `swaggerpy.async_http_client.AsyncResponse` (*req, resp, data*)

Bases: `object`

Remove the property text and content and make them as overridable attrs

```
json (**kwargs)

raise_for_status ()
    Raises stored HTTPError, if one occurred.

class swaggerpy.async_http_client.AsynchronousHttpClient
    Bases: swaggerpy.http_client.HttpClient

    Asynchronous HTTP client implementation.

    fetch_deferred (request_params)
        The main core to start the reactor and run the API in the background. Also the callbacks are registered here

        Returns crochet EventualResult

    start_request (request_params)
        Sets up the request params as per Twisted Agent needs. Sets up crochet and triggers the API request in
        background

        Parameters request_params (dict) – request parameters for API call

        Returns crochet EventualResult

swaggerpy.async_http_client.listify_headers (headers)
    Twisted agent requires header values as lists
```

## 5.4 swagger\_type Module

Code to check the validity of swagger types and conversion to python types

```
class swaggerpy.swagger_type.SwaggerTypeCheck (name, value, type_, models=None, al-
                                                low_null=False, to_wire=False)
    Bases: object

    Initialization of the class checks for the validity of the value to the type.

    Raises TypeError/AssertionError if validation fails

swaggerpy.swagger_type.extract_format (_type_format)
    returns the Format extracted from Type:Format Type:Format is the convention followed for type conversion to
    string

    Parameters _type_format (str or unicode) – converted internal type format eg. “inte-
    ger:int64”

    Returns extracted format eg. “int64”

swaggerpy.swagger_type.get_array_item_type (type_)
    returns the Array Type extracted from ‘Array:ArrayType’ ‘Array:ArrayType’ is the convention followed for
    converting swagger array type into a string

    Parameters type (str or unicode) – converted internal type format eg. “array:integer:int64”

    Returns extracted array type eg. “integer:int64”

swaggerpy.swagger_type.get_primitive_mapping (type_)
    Returns the Python type from the swagger internal type string

    Parameters type (str or unicode) – swagger type, eg. integer, number:float

    Return type type eg. int, string
```

`swaggerpy.swagger_type.get_swagger_type(json_)`

Converts swagger type from json to swagger internal type

Example:

```
{
  ...
  "type": "array",
  "items": {
    "type": "integer",
    "format": "int64"
  }
  ...
}
```

Returns:

```
"array:integer:int64"
```

**Parameters** `json` (*dict*) – dict containing type and rest of the data

**Return type** `str` or unicode

`swaggerpy.swagger_type.get_swagger_types(props)`

Converts dict of swagger types to dict of swagger internal types

**Parameters** `props` (*dict*) – dict of json properties

**Return type** `dict`

`swaggerpy.swagger_type.is_array(type_)`

checks whether the swagger type is array :rtype: boolean

`swaggerpy.swagger_type.is_complex(type_)`

checks whether the swagger type is neither primitive nor array :rtype: boolean

`swaggerpy.swagger_type.is_file(type_)`

checks whether the swagger type is file :rtype: boolean

`swaggerpy.swagger_type.is_primitive(type_)`

checks whether the swagger type is primitive :rtype: boolean

`swaggerpy.swagger_type.primitive_formats()`

returns Swagger primitive formats allowed after internal conversion.

**Returns** a list of typed formats eg. ['integer:int64', 'string:str']

**Return type** `list`

`swaggerpy.swagger_type.primitive_types()`

returns all allowed Swagger primitive types

**Returns** a list of only types

**Return type** `list`

`swaggerpy.swagger_type.swagger_to_py_type(type_)`

returns the python type from swagger type

**Parameters** `type` (*str or unicode*) – swagger internal type

**Return type** Python type

`swaggerpy.swagger_type.swagger_to_py_type_string(type_)`  
returns the string repr of Python type. Used during docstring display of a Model

**Parameters** `type` (*str* or *unicode*) – swagger internal type

**Return type** *str* or *unicode*

## 5.5 response Module

Code for checking the response from API. If correct, it proceeds to convert it into Python class types

**class** `swaggerpy.response.HTTPFuture` (*http\_client, request\_params, post\_receive*)

Bases: *object*

A future which inputs HTTP params

**cancel** ()

Try to cancel the API (meaningful for Asynchronous client)

**cancelled** ()

Checks if API is cancelled Once cancelled, it can't be resumed

**result** (*\*\*kwargs*)

Blocking call to wait for API response If API was cancelled earlier, `CancelledError` is raised If everything goes fine, callback registered is triggered with response

**Parameters**

- **timeout** (*integer*) – timeout in seconds to wait for response
- **allow\_null** (*boolean*) – if True, allow null fields in response
- **raw\_response** (*boolean*) – if True, return raw response w/o any validations

**class** `swaggerpy.response.SwaggerResponseConstruct` (*response, type\_, models*)

Bases: *object*

**create\_object** ()

Only public method in the class

Creates the object assuming the response is checked and valid

**Returns** instance of complex Py object or simple primitive object

`swaggerpy.response.handle_response_errors` (*e*)

**Parameters** *e* – Exception object

**Raises** *HTTPError* –

**class** `swaggerpy.exception.HTTPError`

`swaggerpy.response.post_receive` (*response, type\_, models, \*\*kwargs*)

Convert the response body to swagger models.

Example API Response

```
{
  "id": 1,
  "category": {
    "name": "chihuahua"
  },
  "name": "tommy",
```

(continues on next page)

(continued from previous page)

```

    "photoUrls": [
        ""
    ],
    "tags": [
        {
            "name": "cute"
        }
    ],
    "status": "available"
}

```

SwaggerResponse:

..code-block:: python

```

Pet(category=Category(id=0L, name=u'chihuahua'), status=u'available', name=u'tommy',
    tags=[Tag(id=0L, name=u'cute')], photoUrls=[u''], id=1)

```

#### Parameters

- **response** (*dict*) – response body
- **type** (*str* or *unicode*) – expected swagger type
- **models** (*namedtuple*) – namedtuple which maps complex type string to py type

## 5.6 exception Module

**exception** `swaggerpy.exception.CancelledError`

Bases: `exceptions.Exception`

Error raised when `result()` is called from `HTTPFuture` and call was actually cancelled

**exception** `swaggerpy.exception.HTTPError` (*\*args*, *\*\*kwargs*)

Bases: `exceptions.IOError`

Initialize `HTTPError` with 'response' and 'request' object

**exception** `swaggerpy.exception.SwaggerError` (*msg*, *context*, *cause=None*)

Bases: `exceptions.Exception`

Raised when an error is encountered mapping a response objects into a model.

#### Parameters

- **msg** – String message for the error.
- **context** – `ParsingContext` object
- **cause** – Optional exception that caused this one.

## 5.7 processors Module

Swagger processors enrich and validate the Swagger data model.

This can be to make templating easier, or ensure values required for a particular use case (such as ensuring that description and summary fields exist)

**class** swaggerpy.processors.**ParsingContext**

Bases: `object`

Context information for parsing.

This object is immutable. To change contexts (like adding an item to the stack), use the `next()` and `next_stack()` functions to build a new one.

**is\_empty** ()

Tests whether context is empty.

**Returns** True if empty, False otherwise.

**pop** ()

Pops the most recent object out of the context

**push** (*obj\_type*, *json*, *id\_field*)

Pushes a new self-identifying object into the context.

**Parameters**

- **json** (*dict*) – Specifies type of object json represents
- **json** – Current Jsonified object.
- **id\_field** (*str*) – Field name in json that identifies it.

**push\_str** (*obj\_type*, *json*, *id\_string*)

Pushes a new object into the context.

**Parameters**

- **obj\_type** (*str*) – Specifies type of object json represents
- **json** (*dict*) – Current Jsonified object.
- **id\_string** (*str*) – Identifier of the given json.

**class** swaggerpy.processors.**SwaggerProcessor**

Bases: `object`

Post processing interface for Swagger API's.

This processor can add fields to model objects for additional information to use in the templates.

**apply** (*resources*)

Apply this processor to a loaded Swagger definition.

It assumes Swagger resource listing is valid and verified.

**Parameters** **resources** (*dict*) – Top level Swagger definition.

**pre\_apply** (*resources*)

Apply this processor to a Swagger definition before loading resources.

It fails if resource listing is not valid.

**Parameters** **resources** (*dict*) – Top level Swagger definition.

**process\_api\_declaration** (*resources*, *resource*, *context*)

Post process a resource object.

This is parsed from a .json file reference by a resource listing's 'api' array.

**Parameters**

- **resources** – Resource listing object



- **resource** – resource object.
- **context** ([ParsingContext](#)) – Current context in the API.

**process\_model** (*resources, resource, model, context*)

Post process a model from a resources model dictionary.

#### Parameters

- **resources** – Resource listing object
- **resource** – resource object.
- **model** – Model object.
- **context** ([ParsingContext](#)) – Current context in the API.

**process\_operation** (*resources, resource, api, operation, context, model\_ids*)

Post process an operation on an api.

#### Parameters

- **resources** – Resource listing object
- **resource** – resource object.
- **api** – API object
- **operation** – Operation object.
- **context** ([ParsingContext](#)) – Current context in the API.

**process\_parameter** (*resources, resource, api, operation, parameter, context, model\_ids*)

Post process a parameter on an operation.

#### Parameters

- **resources** – Resource listing object
- **resource** – resource object.
- **api** – API object
- **operation** – Operation object.
- **parameter** – Parameter object.
- **context** ([ParsingContext](#)) – Current context in the API.

**process\_property** (*resources, resource, model, prop, context, model\_ids*)

Post process a property from a model.

#### Parameters

- **resources** – Resource listing object
- **resource** – resource object.
- **model** – Model object.
- **prop** – Property object.
- **context** ([ParsingContext](#)) – Current context in the API.

**process\_resource\_api** (*resources, resource, api, context*)

Post process entries in a resource's api array

#### Parameters

- **resources** – Resource listing object

- **resource** – resource object.
- **api** – API object
- **context** (*ParsingContext*) – Current context in the API.

**process\_resource\_listing** (*resources, context*)

Post process a resources.json object.

**Parameters**

- **resources** – ResourceApi object.
- **context** (*ParsingContext*) – Current context in the API.

**process\_resource\_listing\_api** (*resources, listing\_api, context*)

Post process entries in a resource.json's api array.

**Parameters**

- **resources** – Resource listing object
- **listing\_api** – ResourceApi object.
- **context** (*ParsingContext*) – Current context in the API.

**process\_response\_message** (*resources, resource, api, operation, response\_message, context, model\_ids*)

Post process an Response on an operation.

**Parameters**

- **resources** – Resource listing object
- **resource** – resource object.
- **api** – API object
- **operation** – Operation object.
- **response** – Response object.
- **context** (*ParsingContext*) – Current context in the API.

## 5.8 swagger\_model Module

**class** `swaggerpy.swagger_model.FileEventual` (*path*)

Bases: *object*

Adaptor which supports the `crochet.EventualResult` interface for retrieving api docs from a local file.

**class** `FileResponse` (*data*)

Bases: *object*

`json()`

`cancel()`

`get_path()`

`wait` (*timeout=None*)

**class** `swaggerpy.swagger_model.ValidationProcessor`

Bases: *swaggerpy.processors.SwaggerProcessor*

A processor that validates the Swagger model.

**process\_api\_declaration** (*resources, resource, context*)

Post process a resource object.

This is parsed from a .json file reference by a resource listing's 'api' array.

#### Parameters

- **resources** – Resource listing object
- **resource** – resource object.
- **context** (*ParsingContext*) – Current context in the API.

**process\_model** (*resources, resource, model, context*)

Post process a model from a resources model dictionary.

#### Parameters

- **resources** – Resource listing object
- **resource** – resource object.
- **model** – Model object.
- **context** (*ParsingContext*) – Current context in the API.

**process\_operation** (*resources, resource, api, operation, context, model\_ids*)

Post process an operation on an api.

#### Parameters

- **resources** – Resource listing object
- **resource** – resource object.
- **api** – API object
- **operation** – Operation object.
- **context** (*ParsingContext*) – Current context in the API.

**process\_parameter** (*resources, resource, api, operation, parameter, context, model\_ids*)

Post process a parameter on an operation.

#### Parameters

- **resources** – Resource listing object
- **resource** – resource object.
- **api** – API object
- **operation** – Operation object.
- **parameter** – Parameter object.
- **context** (*ParsingContext*) – Current context in the API.

**process\_property** (*resources, resource, model, prop, context, model\_ids*)

Post process a property from a model.

#### Parameters

- **resources** – Resource listing object
- **resource** – resource object.
- **model** – Model object.
- **prop** – Property object.

- **context** (`ParsingContext`) – Current context in the API.

**process\_resource\_api** (*resources, resource, api, context*)

Post process entries in a resource's api array

**Parameters**

- **resources** – Resource listing object
- **resource** – resource object.
- **api** – API object
- **context** (`ParsingContext`) – Current context in the API.

**process\_resource\_listing** (*resources, context*)

Post process a resources.json object.

**Parameters**

- **resources** – ResourceApi object.
- **context** (`ParsingContext`) – Current context in the API.

**process\_resource\_listing\_api** (*resources, listing\_api, context*)

Post process entries in a resource.json's api array.

**Parameters**

- **resources** – Resource listing object
- **listing\_api** – ResourceApi object.
- **context** (`ParsingContext`) – Current context in the API.

**process\_response\_message** (*resources, resource, api, operation, response\_message, context, model\_ids*)

Post process an Response on an operation.

**Parameters**

- **resources** – Resource listing object
- **resource** – resource object.
- **api** – API object
- **operation** – Operation object.
- **response** – Response object.
- **context** (`ParsingContext`) – Current context in the API.

`swaggerpy.swagger_model.compare` (*first, second*)

Compares the two types for equivalence.

If a type composes another model types, `__dict__` recurse on those and compares again on those dict values

`swaggerpy.swagger_model.create_flat_dict` (*model*)

Generates `__dict__` of the model traversing recursively each of the list item of an array and calling it again. While `__dict__` only converts it on one level.

**param model** generated model type reference

**type model** type

**returns** flat dict repr of the model

Example:

```
Pet(id=3, name="Name", photoUrls=["7"], tags=[Tag(id=2, name='T')])
```

converts to:

```
{'id': 3,
 'name': 'Name',
 'photoUrls': ['7'],
 'tags': [{'id': 2,
             'name': 'T'}
          ]
}
```

swaggerpy.swagger\_model.**create\_model\_docstring**(props)

Generates a docstring for the type from the props

**param props** dict containing properties of the type

**type props** dict

**returns** Generated string

Example:

```
"Pet": {
  "id": "Pet",
  "properties": {
    "id": {
      "type": "integer",
      "format": "int64",
      "description": "unique identifier for the pet",
    },
    "category": {
      "$ref": "Category"
    },
    "name": {
      "type": "string"
    },
    "status": {
      "type": "string",
      "description": "pet status in the store",
    }
  }
}
```

Result:

```
Attributes:

category (Category)
status (str) : pet status in the store
name (str)
id (long) : unique identifier for the pet
```

swaggerpy.swagger\_model.**create\_model\_repr**(model)

Generates the repr string for the model

**Parameters** **model** (*type*) – generated model type reference

**Returns** repr string for the model

`swaggerpy.swagger_model.create_model_type(model)`

Create a dynamic class from the model data defined in the swagger spec.

The docstring for this class is dynamically generated because generating the docstring is relatively expensive, and would only be used in rare cases for interactive debugging in a REPL.

**Parameters** `model` – Resource model `dict` with keys `id` and `properties`

**Returns** dynamic type created with attributes, docstrings attached

**Return type** `type`

**class** `swaggerpy.swagger_model.docstring_property(func)`

Bases: `object`

`swaggerpy.swagger_model.is_file_scheme_uri(url)`

`swaggerpy.swagger_model.load_file(resource_listing_file, http_client=None)`

Loads a resource listing file.

**Parameters**

- **http\_client** – HTTP client interface.
- **resource\_listing\_file** – File name for a resource listing.

**Returns** Processed object model from

**Raise** `IOError`: On error reading api-docs.

`swaggerpy.swagger_model.load_resource_listing(url, http_client, base_url=None, request_options=None)`

Load a complete swagger api spec and return all schemas compiled into a single dict.

**Parameters**

- **url** – url to the swagger spec (file or http)
- **http\_client** – a `swaggerpy.http_client.HttpClient` for performing the requests to fetch api documents.
- **base\_url** – optional url to use as the base url for api doc paths
- **request\_options** – mapping of additional fields to specify in the http request to fetch resources.

`swaggerpy.swagger_model.load_url(url, http_client=None, **kwargs)`

Loads a resource listing.

**Parameters**

- **resource\_listing\_url** – URL for a resource listing.
- **http\_client** – HTTP client interface.
- **base\_url** – Optional URL to be the base URL for finding API declarations. If not specified, 'basePath' from the resource listing is used.

**Returns** Processed object model from

**Raise** `IOError`, `URLError`: On error reading api-docs.

`swaggerpy.swagger_model.set_props(model, **kwargs)`

Constructor for the generated type - assigns given or default values

**Parameters**

- **model** (`type`) – generated model type reference

- **kwargs** (*dict*) – attributes to override default values of constructor

`swaggerpy.swagger_model.start_request` (*http\_client, url, request\_options*)  
Download and parse JSON from a URL.

#### Parameters

- **http\_client** – a `swaggerpy.http_client.HttpClient`
- **url** – url for api docs
- **request\_options** – additional fields to send with the request

**Returns** an object with a `:func'wait'` method which returns the api docs

`swaggerpy.swagger_model.validate_params_body_or_form` (*json*)

Validates that form request parameters are present or body request params but not both

`swaggerpy.swagger_model.validate_required_fields` (*json, required\_fields, context*)

Checks a JSON object for a set of required fields.

If any required field is missing, a `SwaggerError` is raised.

#### Parameters

- **json** – JSON object to check.
- **required\_fields** – List of required fields.
- **context** – Current context in the API.

`swaggerpy.swagger_model.validate_type_or_ref` (*json, model\_ids, allowed\_types, allowed\_refs, context*)

Validates that either type OR ref is present in the json

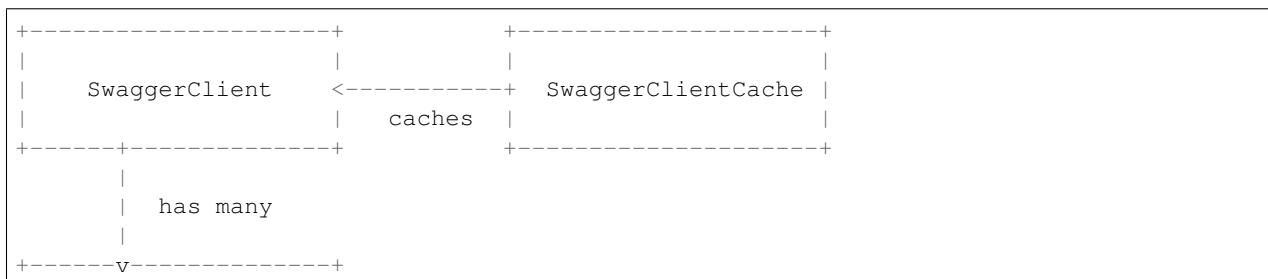
#### Parameters

- **json** – dict to check whether type or ref is present
- **model\_ids** – list of allowed \$ref ids (all models)
- **allowed\_types** – list of all kind of types allowed
- **allowed\_refs** – list of all kind of refs allowed
- **context** – only used for Request Operation and Paramter

## 5.9 client Module

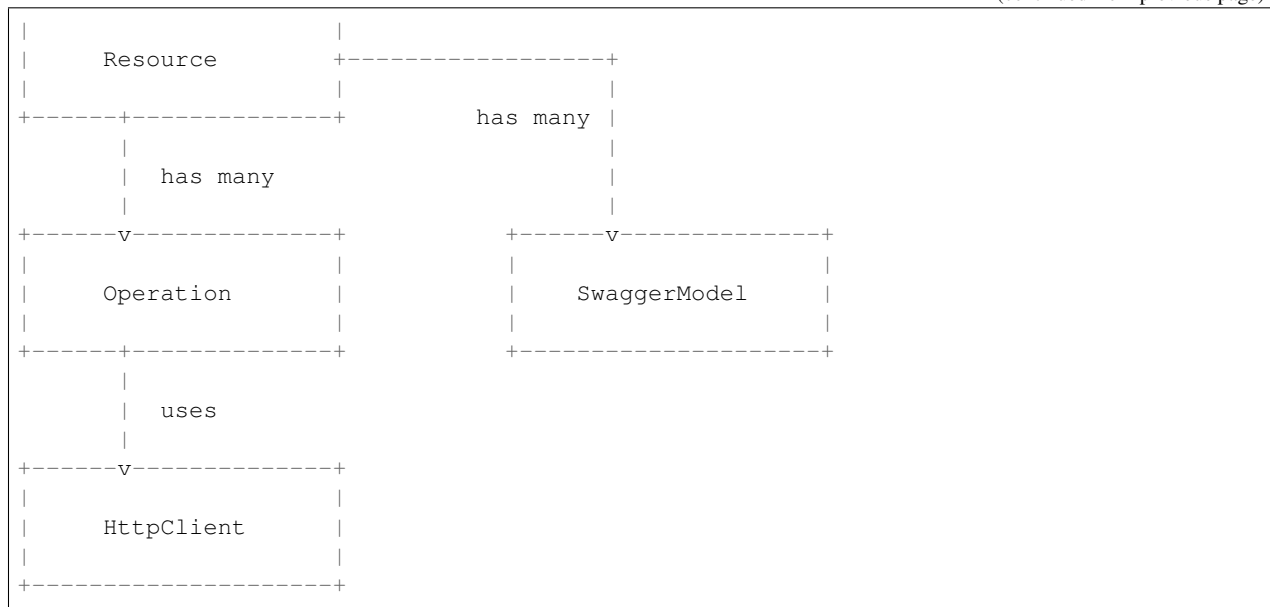
The `SwaggerClient` provides an interface for making API calls based on a swagger spec, and returns responses of python objects which build from the API response.

Structure Diagram:



(continues on next page)

(continued from previous page)



To get a client with caching

```
client = swaggerpy.client.get_client(api_docs_url)
```

without caching

```
client = swaggerpy.client.SwaggerClient.from_url(api_docs_url)
```

**class** `swaggerpy.client.CacheEntry` (*item, ttl, timestamp=None*)

Bases: `object`

An entry in the cache. Each item has it's own ttl.

#### Parameters

- **item** – the item to cache
- **ttl** (*int*) – time-to-live in seconds after which the client expires

**is\_stale** (*timestamp=None*)

Checks if the instance has become stale :return: True if the cache item is stale, False otherwise

**class** `swaggerpy.client.Operation` (*uri, operation, http\_client, models*)

Bases: `object`

Perform a request by taking the kwargs passed to the call and constructing an HTTP request.

**class** `swaggerpy.client.Resource` (*name, operations*)

Bases: `object`

Swagger resource, described in an API declaration.

**classmethod** `from_api_doc` (*api\_doc, http\_client, base\_path, url\_base=None*)

#### Parameters

- **api\_doc** (*dict*) – api doc which defines this resource
- **http\_client** – a `swaggerpy.http_client.HttpClient`



- **base\_path** – base url to perform api requests. Used to override the path provided in the api spec
- **url\_base** – a url used as the base for resource definitions that include a relative basePath

**class** `swaggerpy.client.SwaggerClient` (*api\_url, resources*)

Bases: `object`

A client for accessing a Swagger-documented RESTful service.

#### Parameters

- **api\_url** – the url for the swagger api docs, only used for the repr.
- **resources** – a list of `:Resource:` objects used to perform requests

**classmethod** `from_resource_listing` (*resource\_listing, http\_client=None, api\_base\_path=None, url=None*)

Build a `SwaggerClient` from swagger api docs

#### Parameters

- **resource\_listing** – a dict with a list of api definitions
- **http\_client** (`swaggerpy.http_client.HttpClient`) – an HTTP client used to perform requests
- **api\_base\_path** (*str*) – a url, override the path used to make api requests
- **url** (*str*) – the url used to retrieve the resource listing

**classmethod** `from_url` (*url, http\_client=None, api\_base\_path=None, request\_options=None*)

Build a `SwaggerClient` from a url to api docs describing the api.

#### Parameters

- **url** (*str*) – url pointing at the swagger api docs
- **http\_client** (`swaggerpy.http_client.HttpClient`) – an HTTP client used to perform requests
- **api\_base\_path** (*str*) – a url, override the path used to make api requests
- **request\_options** (*dict*) – extra values to pass with api docs requests

**class** `swaggerpy.client.SwaggerClientCache`

Bases: `object`

Cache to store swagger clients and refetch the api-docs if the client becomes stale

**build\_client** (*api\_docs, \*args, \*\*kwargs*)

`swaggerpy.client.add_param_to_req` (*param, value, request*)

Populates request object with the request parameters

#### Parameters

- **param** (*dict*) – swagger spec details of a param
- **value** – value for the param given in the API call
- **request** – request object to be populated

`swaggerpy.client.append_name_to_api` (*api\_entry*)

`swaggerpy.client.build_models` (*model\_dicts*)

`swaggerpy.client.build_resources_from_spec` (*http\_client, apis, api\_base\_path, url\_base*)

`swaggerpy.client.create_operation_docstring(json_)`

Builds Operation docstring from the json dict

**Parameters** `json` (*dict*) – data to create docstring from

**Returns** string giving meta info

Example:

```
client.pet.findPetsByStatus?
```

Outputs:

```
[GET] Finds Pets by status

Multiple status values can be provided with comma seperated strings
Args:
    status (string) : Statuses to be considered for filter
    from_date (string) : Start date filter
Returns:
    array
Raises:
    400: Invalid status value
```

`swaggerpy.client.get_client(*args, **kwargs)`

Factory method to generate SwaggerClient instance.

---

**Note:** This factory method uses a global which maintains the state of swagger client. Use [\*SwaggerClientCache\*](#) if you want more control.

---

To change the freshness timeout, simply pass an argument: `ttl=<seconds>`

To remove the caching functionality, pass: `ttl=0`

---

**Note:** It is OKAY to call `get_swagger_client(...)` again and again. Do not keep a reference to the generated client and make it long lived as it strips out the refetching functionality.

---

#### Parameters

- `api_docs_url` (*str*) – url for swagger api docs used to build the client
- `ttl` – (optional) Timeout in secs. after which api-docs is stale

**Returns** [\*SwaggerClient\*](#)

`swaggerpy.client.get_resource_url(base_path, url_base, resource_base_path)`

`swaggerpy.client.handle_form_param(name, value, type_, request)`

`swaggerpy.client.stringify_body(value)`

Json dump the value to string if not already in string

`swaggerpy.client.validate_and_add_params_to_request(param, value, request, models)`

Validates if a required param is given And wraps 'add\_param\_to\_req' to populate a valid request

#### Parameters

- `param` (*dict*) – swagger spec details of a param

- **value** – value for the param given in the API call
- **request** – request object to be populated
- **models** (*namedtuple*) – models tuple containing all complex model types



## CHAPTER 6

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### S

- `swaggerpy.async_http_client`, [15](#)
- `swaggerpy.client`, [27](#)
- `swaggerpy.exception`, [19](#)
- `swaggerpy.http_client`, [13](#)
- `swaggerpy.processors`, [19](#)
- `swaggerpy.response`, [18](#)
- `swaggerpy.swagger_model`, [22](#)
- `swaggerpy.swagger_type`, [16](#)





## A

`add_param_to_req()` (in module `swaggerpy.client`), 29  
`ApiKeyAuthenticator` (class in `swaggerpy.http_client`), 13  
`append_name_to_api()` (in module `swaggerpy.client`), 29  
`apply()` (`swaggerpy.http_client.ApiKeyAuthenticator` method), 13  
`apply()` (`swaggerpy.http_client.Authenticator` method), 13  
`apply()` (`swaggerpy.http_client.BasicAuthenticator` method), 14  
`apply()` (`swaggerpy.processors.SwaggerProcessor` method), 20  
`apply_authentication()` (`swaggerpy.http_client.SynchronousHttpClient` method), 15  
`AsynchronousHttpClient` (class in `swaggerpy.async_http_client`), 16  
`AsyncResponse` (class in `swaggerpy.async_http_client`), 15  
`authenticated_request()` (`swaggerpy.http_client.SynchronousHttpClient` method), 15  
`Authenticator` (class in `swaggerpy.http_client`), 13

## B

`BasicAuthenticator` (class in `swaggerpy.http_client`), 14  
`build_client()` (`swaggerpy.client.SwaggerClientCache` method), 29  
`build_models()` (in module `swaggerpy.client`), 29  
`build_resources_from_spec()` (in module `swaggerpy.client`), 29

## C

`CacheEntry` (class in `swaggerpy.client`), 28  
`cancel()` (`swaggerpy.http_client.SynchronousEventual` method), 15  
`cancel()` (`swaggerpy.response.HTTPFuture` method), 18  
`cancel()` (`swaggerpy.swagger_model.FileEventual` method), 22

`cancelled()` (`swaggerpy.response.HTTPFuture` method), 18  
`CancelledError`, 19  
`compare()` (in module `swaggerpy.swagger_model`), 24  
`create_flat_dict()` (in module `swaggerpy.swagger_model`), 24  
`create_model_docstring()` (in module `swaggerpy.swagger_model`), 25  
`create_model_repr()` (in module `swaggerpy.swagger_model`), 25  
`create_model_type()` (in module `swaggerpy.swagger_model`), 25  
`create_object()` (`swaggerpy.response.SwaggerResponseConstruct` method), 18  
`create_operation_docstring()` (in module `swaggerpy.client`), 29

## D

`docstring_property` (class in `swaggerpy.swagger_model`), 26

## E

`extract_format()` (in module `swaggerpy.swagger_type`), 16

## F

`fetch_deferred()` (`swaggerpy.async_http_client.AsynchronousHttpClient` method), 16  
`FileEventual` (class in `swaggerpy.swagger_model`), 22  
`FileEventual.FileResponse` (class in `swaggerpy.swagger_model`), 22  
`from_api_doc()` (`swaggerpy.client.Resource` class method), 28  
`from_resource_listing()` (`swaggerpy.client.SwaggerClient` class method), 29  
`from_url()` (`swaggerpy.client.SwaggerClient` class method), 29

## G

`get_array_item_type()` (in module `swaggerpy.swagger_type`), 16  
`get_client()` (in module `swaggerpy.client`), 30  
`get_path()` (`swaggerpy.swagger_model.FileEventual` method), 22  
`get_primitive_mapping()` (in module `swaggerpy.swagger_type`), 16  
`get_resource_url()` (in module `swaggerpy.client`), 30  
`get_swagger_type()` (in module `swaggerpy.swagger_type`), 16  
`get_swagger_types()` (in module `swaggerpy.swagger_type`), 17

## H

`handle_form_param()` (in module `swaggerpy.client`), 30  
`handle_response_errors()` (in module `swaggerpy.response`), 18  
`HttpClient` (class in `swaggerpy.http_client`), 14  
`HTTPError`, 19  
`HTTPFuture` (class in `swaggerpy.response`), 18

## I

`is_array()` (in module `swaggerpy.swagger_type`), 17  
`is_complex()` (in module `swaggerpy.swagger_type`), 17  
`is_empty()` (`swaggerpy.processors.ParsingContext` method), 20  
`is_file()` (in module `swaggerpy.swagger_type`), 17  
`is_file_scheme_uri()` (in module `swaggerpy.swagger_model`), 26  
`is_primitive()` (in module `swaggerpy.swagger_type`), 17  
`is_stale()` (`swaggerpy.client.CacheEntry` method), 28

## J

`json()` (`swaggerpy.async_http_client.AsyncResponse` method), 15  
`json()` (`swaggerpy.swagger_model.FileEventual.FileResponse` method), 22

## L

`listify_headers()` (in module `swaggerpy.async_http_client`), 16  
`load_file()` (in module `swaggerpy.swagger_model`), 26  
`load_resource_listing()` (in module `swaggerpy.swagger_model`), 26  
`load_url()` (in module `swaggerpy.swagger_model`), 26

## M

`matches()` (`swaggerpy.http_client.Authenticator` method), 14

## O

`Operation` (class in `swaggerpy.client`), 28

## P

`ParsingContext` (class in `swaggerpy.processors`), 19  
`pop()` (`swaggerpy.processors.ParsingContext` method), 20  
`post_receive()` (in module `swaggerpy.response`), 18  
`pre_apply()` (`swaggerpy.processors.SwaggerProcessor` method), 20  
`primitive_formats()` (in module `swaggerpy.swagger_type`), 17  
`primitive_types()` (in module `swaggerpy.swagger_type`), 17  
`process_api_declaration()` (`swaggerpy.processors.SwaggerProcessor` method), 20  
`process_api_declaration()` (`swaggerpy.swagger_model.ValidationProcessor` method), 22  
`process_model()` (`swaggerpy.processors.SwaggerProcessor` method), 21  
`process_model()` (`swaggerpy.swagger_model.ValidationProcessor` method), 23  
`process_operation()` (`swaggerpy.processors.SwaggerProcessor` method), 21  
`process_operation()` (`swaggerpy.swagger_model.ValidationProcessor` method), 23  
`process_parameter()` (`swaggerpy.processors.SwaggerProcessor` method), 21  
`process_parameter()` (`swaggerpy.swagger_model.ValidationProcessor` method), 23  
`process_property()` (`swaggerpy.processors.SwaggerProcessor` method), 21  
`process_property()` (`swaggerpy.swagger_model.ValidationProcessor` method), 23  
`process_resource_api()` (`swaggerpy.processors.SwaggerProcessor` method), 21  
`process_resource_api()` (`swaggerpy.swagger_model.ValidationProcessor` method), 24  
`process_resource_listing()` (`swaggerpy.processors.SwaggerProcessor` method), 22  
`process_resource_listing()` (`swaggerpy.swagger_model.ValidationProcessor` method), 24  
`process_resource_listing_api()` (`swaggerpy.processors.SwaggerProcessor` method),

[22](#)  
[process\\_resource\\_listing\\_api\(\)](#) (swaggerpy.swagger\_model.ValidationProcessor method), [24](#)  
[process\\_response\\_message\(\)](#) (swaggerpy.processors.SwaggerProcessor method), [22](#)  
[process\\_response\\_message\(\)](#) (swaggerpy.swagger\_model.ValidationProcessor method), [24](#)  
[push\(\)](#) (swaggerpy.processors.ParsingContext method), [20](#)  
[push\\_str\(\)](#) (swaggerpy.processors.ParsingContext method), [20](#)

## R

[raise\\_for\\_status\(\)](#) (swaggerpy.async\_http\_client.AsyncResponse method), [16](#)  
[request\(\)](#) (swaggerpy.http\_client.HttpClient method), [14](#)  
[Resource](#) (class in swaggerpy.client), [28](#)  
[result\(\)](#) (swaggerpy.response.HTTPFuture method), [18](#)

## S

[set\\_api\\_key\(\)](#) (swaggerpy.http\_client.HttpClient method), [14](#)  
[set\\_api\\_key\(\)](#) (swaggerpy.http\_client.SynchronousHttpClient method), [15](#)  
[set\\_basic\\_auth\(\)](#) (swaggerpy.http\_client.HttpClient method), [14](#)  
[set\\_basic\\_auth\(\)](#) (swaggerpy.http\_client.SynchronousHttpClient method), [15](#)  
[set\\_props\(\)](#) (in module swaggerpy.swagger\_model), [26](#)  
[start\\_request\(\)](#) (in module swaggerpy.swagger\_model), [27](#)  
[start\\_request\(\)](#) (swaggerpy.async\_http\_client.AsynchronousHttpClient method), [16](#)  
[start\\_request\(\)](#) (swaggerpy.http\_client.HttpClient method), [14](#)  
[start\\_request\(\)](#) (swaggerpy.http\_client.SynchronousHttpClient method), [15](#)  
[stringify\\_body\(\)](#) (in module swaggerpy.client), [30](#)  
[swagger\\_to\\_py\\_type\(\)](#) (in module swaggerpy.swagger\_type), [17](#)  
[swagger\\_to\\_py\\_type\\_string\(\)](#) (in module swaggerpy.swagger\_type), [17](#)  
[SwaggerClient](#) (class in swaggerpy.client), [29](#)  
[SwaggerClientCache](#) (class in swaggerpy.client), [29](#)  
[SwaggerError](#), [19](#)  
[SwaggerProcessor](#) (class in swaggerpy.processors), [20](#)  
[swaggerpy.async\\_http\\_client](#) (module), [15](#)  
[swaggerpy.client](#) (module), [27](#)  
[swaggerpy.exception](#) (module), [19](#)  
[swaggerpy.http\\_client](#) (module), [13](#)  
[swaggerpy.processors](#) (module), [19](#)  
[swaggerpy.response](#) (module), [18](#)  
[swaggerpy.swagger\\_model](#) (module), [22](#)  
[swaggerpy.swagger\\_type](#) (module), [16](#)  
[SwaggerResponseConstruct](#) (class in swaggerpy.response), [18](#)  
[SwaggerTypeCheck](#) (class in swaggerpy.swagger\_type), [16](#)  
[SynchronousEventual](#) (class in swaggerpy.http\_client), [15](#)  
[SynchronousHttpClient](#) (class in swaggerpy.http\_client), [15](#)

## V

[validate\\_and\\_add\\_params\\_to\\_request\(\)](#) (in module swaggerpy.client), [30](#)  
[validate\\_params\\_body\\_or\\_form\(\)](#) (in module swaggerpy.swagger\_model), [27](#)  
[validate\\_required\\_fields\(\)](#) (in module swaggerpy.swagger\_model), [27](#)  
[validate\\_type\\_or\\_ref\(\)](#) (in module swaggerpy.swagger\_model), [27](#)  
[ValidationProcessor](#) (class in swaggerpy.swagger\_model), [22](#)

## W

[wait\(\)](#) (swaggerpy.http\_client.SynchronousEventual method), [15](#)  
[wait\(\)](#) (swaggerpy.swagger\_model.FileEventual method), [22](#)