
Python MWS Client Documentation

Release 0.2.0

Fulfil.IO Inc.

Feb 23, 2022

Contents:

1	Python MWS Client	1
1.1	Installation	1
1.2	Usage	1
2	Installation	3
2.1	Stable release	3
2.2	From sources	3
3	Usage	5
3.1	Quick start	5
3.2	Design pattern	6
4	pymws package	9
4.1	pymws	9
4.2	orders	10
4.3	product	10
4.4	fulfillment	10
4.5	reports	11
4.6	exceptions	12
4.7	utils	13
5	Contributing	15
5.1	Types of Contributions	15
5.2	Get Started!	16
5.3	Pull Request Guidelines	17
5.4	Tips	17
5.5	Deploying	17
6	Credits	19
6.1	Development Lead	19
6.2	Contributors	19
7	History	21
7.1	0.1.0 (2020-08-08)	21
8	Indices and tables	23
	Python Module Index	25

CHAPTER 1

Python MWS Client

Python client for Amazon Marketplace Web Services (MWS)

- Free software: MIT license
- Documentation: <https://pymws.readthedocs.io>.

1.1 Installation

1.2 Usage

Import the package

```
from pymws import MWS
```

Create a client instance

```
client = MWS(  
    marketplace="US", merchant_id="1234",  
    access_key_id="key", secret_key="secret",  
    auth_token="token"  
)
```

Once you have a client with valid credentials, you can now call operations.

```
start_date = datetime(2020, 1, 20, 10, 30)  
client.orders.list_orders(CreatedAfter=start_date)
```


2.1 Stable release

To install Python MWS Client, run this command in your terminal:

```
$ pip install pymws
```

This is the preferred method to install Python MWS Client, as it will always install the most recent stable release.

If you don't have [pip](#) installed, this [Python installation guide](#) can guide you through the process.

2.2 From sources

The sources for Python MWS Client can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/fulfilio/pymws
```

Or download the [tarball](#):

```
$ curl -OJL https://github.com/fulfilio/pymws/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```


To use Python MWS Client in a project:

```
import pymws
```

Create a client instance by providing credentials:

```
client = MWS(  
    marketplace="US", merchant_id="1234",  
    access_key_id="key", secret_key="secret",  
    auth_token="token"  
)
```

Now you have a client instance, that can then be used for operations.

3.1 Quick start

Fetch a list of orders:

```
start_date = datetime(2020, 1, 20, 10, 30)  
response = client.orders.list_orders(CreatedAfter=start_date)
```

The returned response is the XML result from the API call as an lxml object. You can browse through the response:

```
for order in response.Orders.getchildren():  
    print(order.AmazonOrderId)
```

The returned lxml objectify object is a quick and easy way to navigate the XML response.

To fetch the next page of orders:

```
orders_page_2 = client.orders.list_orders_by_next_token(  
    response.NextToken  
)
```

3.2 Design pattern

The API client is a thin layer over the MWS api. Methods accept the same parameters as the MWS api and hence MWS API documentation for your marketplace should be your primary reference.

3.2.1 Supported parameters

All parameters provided are passed directly to the MWS API after type casting into the appropriate format. For example, dates and datetimes are converted into the ISO format.

List parameters are treated differently. Amazon MWS API requires you to enumerate parameters.

For example, if you want to get the report list for two report ids, Amazon expects it to be enumerated:

```
POST /Reports/2009-01-01 HTTP/1.1
...
...

&ReportRequestIdList.Id.1=2291326454
&ReportRequestIdList.Id.2=2294446454
...
```

The equivalent with the api would be:

```
client.reports.get_report_list(
    ReportRequestIdList=['2291326454', '2294446454']
)
```

3.2.2 Response objects

Amazon MWS API supports two primary response formats:

- XML
- Tab separated value (TSV) list

TSVs are more common for reports while XML is the default response type for most responses.

XML responses

XML responses are converted into lxml objectify objects. Knowing how to use the lxml API can make your code prettier and simpler.

- [Example parsing with lxml](#)
- [Objectify API docs](#)

TSV responses

Any tsv response is converted into a list of dictionaries.

Example using settlement reports:

```
available_reports = reports = client.reports.get_report_list(  
    ReportTypeList=['_GET_V2_SETTLEMENT_REPORT_DATA_FLAT_FILE_V2_']  
)  
tsv_report = client.reports.get_report(  
    available_reports.ReportInfo[0].ReportId  
)
```

Browse the list:

```
for row in tsv_report:  
    print(row['marketplace-name'])
```


4.1 pymws

class `pymws.pymws.MWS` (*marketplace*, *merchant_id=None*, *access_key_id=None*, *secret_key=None*, *auth_token=None*)

Primary client class that acts as a gateway to all of the functionality offered by this package.

Example usage:

```
client = MWS(
    marketplace="US", merchant_id="1234",
    access_key_id="key", secret_key="secret",
    auth_token="token"
)
client.reports.get_reports_list()
```

Parameters

- **marketplace** – marketplace to connect to.
- **merchant_id** – Amazon merchant id.
- **access_key_id** – Access key of your app
- **secret_key** – Secret key of your app
- **auth_token** – Token obtained by the merchant after installing your app.

orders

Fetch the order API client Returns an instance of `pymws.orders.Orders`

products

Fetch the products API client. Returns an instance of `pymws.products.Products`

reports

Fetch the reports API client Returns an instance of `pymws.reports.Reports`

4.2 orders

class pymws.orders.Orders(*client*)

Implements an order API client for Amazon MWS

```
response = client.orders.list_orders(CreatedAfter=start_date)
for order in response.Orders.getchildren():
    print(order.AmazonOrderId)
```

for other attributes of the order, refer to the Amazon MWS documentation.

To fetch the next page of orders:

```
response = client.orders.list_orders(CreatedAfter=start_date)
page2 = client.orders.list_orders_by_next_token(
    response.NextToken
)
```

get_order(*AmazonOrderId*)

Returns orders based on the AmazonOrderId values that you specify.

[Learn more](#)

get_service_status()

Returns the operational status of the Orders API section.

[Learn more](#)

list_order_items(*AmazonOrderId*)

Returns order items based on the AmazonOrderId that you specify.

[Learn more](#)

list_order_items_by_next_token(*NextToken*)

Returns the next page of order items using the NextToken parameter.

[Learn more](#)

list_orders(***kwargs*)

Returns orders created or updated during a time frame that you specify.

[Learn more](#)

list_orders_by_next_token(*NextToken*)

Returns the next page of orders using the NextToken parameter.

[Learn more](#)

The *NextToken* argument is not very pythonic, but keeps the package user side code consistent with Amazon's documentation and avoids a surprise for the user.

4.3 product

4.4 fulfillment

class pymws.fulfillment.outbound_shipment.OutboundShipment(*client*)

Implements outbound shipment client for Amazon MWS

```

client.fulfillment_outbound_shipment.create_fulfillment_order(
    MarketplaceId='A2Q3Y263D00KWC',
    SellerFulfillmentOrderId='SO3421',
    FulfillmentAction='Ship',
    DisplayableOrderId='SO3421',
    DisplayableOrderDateTime=datetime.utcnow(),
    DisplayableOrderComment='Some comment',
    ShippingSpeedCategory='Standard',
    DestinationAddress={
        'Name': 'John Doe',
        'Line1': 'Random street',
        'StateOrProvinceCode': 'CA',
        'CountryCode': 'US',
    },
    Items=[{
        'SellerSKU': 'SKU-1',
        'SellerFulfillmentOrderItemId': 'SO3421-1',
        'Quantity': 1,
    }],
)

```

for other attributes refer to the Amazon MWS documentation.

create_fulfillment_order (***kwargs*)

Requests that Amazon ship items from the seller's inventory in Amazon's fulfillment network to a destination address.

[Learn more](#)

get_fulfillment_order (*SellerFulfillmentOrderId*)

Returns a fulfillment order based on a specified SellerFulfillmentOrderId.

[Learn more](#)

4.5 reports

class pymws.reports.**Reports** (*client*)

Implements a report client for Amazon MWS.

The report API works differently from other APIs and it is important to have a good understanding of how this API works from [MWS reports overview](#)

cancel_report_request (***kwargs*)

Cancels one or more report requests.

[Learn more](#)

get_report (*ReportId*)

Returns the contents of a report and the Content-MD5 header for the returned report body.

[Learn more](#)

get_report_count (***kwargs*)

Returns a count of the reports, created in the previous 90 days, with a status of `_DONE_` and that are available for download.

[Learn more](#)

get_report_list (***kwargs*)

Returns a list of reports that were created in the previous 90 days.

[Learn more](#)

get_report_list_by_next_token (*NextToken*)

Returns a list of reports using the NextToken, which was supplied by a previous request to either GetReportListByNextToken or GetReportList, where the value of HasNext was true in the previous call.

[Learn more](#)

get_report_request_count (***kwargs*)

Returns a count of report requests that have been submitted to Amazon MWS for processing.

[Learn more](#)

get_report_request_list (***kwargs*)

Returns a list of report requests that you can use to get the ReportRequestId for a report.

[Learn more](#)

get_report_request_list_by_next_token (*NextToken*)

Returns a list of report requests using the NextToken, which was supplied by a previous request to either GetReportRequestListByNextToken or GetReportRequestList, where the value of HasNext was true in that previous request.

[Learn more](#)

request_report (***kwargs*)

Creates a report request and submits the request to Amazon MWS.

Amazon MWS processes the report request and when the report is completed, sets the status of the report request to `_DONE_`. Reports are retained for 90 days.

[Learn more](#)

4.6 exceptions

All errors and exceptions

exception `pymws.exceptions.AccessDenied`

Bases: `pymws.exceptions.MWSError`

Access was denied

exception `pymws.exceptions.MWSError`

Bases: `pymws.exceptions.MWSEException`

Parent class of all Amazon returned Errors.

exception `pymws.exceptions.MWSEException`

Bases: `Exception`

Parent class of all exceptions this package handles

exception `pymws.exceptions.QuotaExceeded`

Bases: `pymws.exceptions.MWSError`

The total number of requests in an hour was exceeded.

exception `pymws.exceptions.RequestThrottled`

Bases: `pymws.exceptions.MWSError`

The frequency of requests was greater than allowed..

exception `pymws.exceptions.SignatureDoesNotMatch`

Bases: `pymws.exceptions.MWSError`

The signature used does not match the server's calculated signature value.

4.7 utils

`pymws.utils.MARKETPLACES = [Marketplace(code='BR', currency='BRL', id='A2Q3Y263D00KWC', en`

A list of [Marketplaces](#)

class `pymws.utils.Marketplace` (*code, currency, id, endpoint, name*)

code

Alias for field number 0

currency

Alias for field number 1

endpoint

Alias for field number 3

id

Alias for field number 2

name

Alias for field number 4

`pymws.utils.flatten_dict` (*kwargs, key*)

Convert a dict into URL parameters the way amazon like it.

Example:

```
flatten_dict(
    {'DestinationAddress': {'Name': 'John', 'Country': 'US'}},
    'DestinationAddress',
)
```

Becomes:

```
DestinationAddress.Name=John&DestinationAddress.Country=US
```

`pymws.utils.flatten_list` (*kwargs, key, separator*)

Convert a list into URL parameters the way amazon like it.

Example:

```
flatten_list(
    {'ReportTypeList': ['_A_', '_B_']},
    'ReportTypeList',
    'Type'
)
```

Becomes:

```
ReportTypeList.Type.1=_A_&ReportTypeList.Type.2=_B_
```

`pymws.utils.get_marketplace(id_or_code)`

Given a code, get the marketplace

`pymws.utils.parse_xsv(text)`

Python 2 and 3 compatible (X)SV - CSV/TSV parser that returns a list of dictionary objects.

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at <https://github.com/fulfilio/pymws/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

5.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

5.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

5.1.4 Write Documentation

Python MWS Client could always use more documentation, whether as part of the official Python MWS Client docs, in docstrings, or even on the web in blog posts, articles, and such.

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/fulfilio/pymws/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

5.2 Get Started!

Ready to contribute? Here's how to set up *pymws* for local development.

1. Fork the *pymws* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/pymws.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv pymws
$ cd pymws/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 pymws tests
$ python setup.py test or pytest
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

5.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 3.5, 3.6, 3.7 and 3.8, and for PyPy. Check https://travis-ci.com/fulfilio/pymws/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

```
$ pytest tests.test_pymws
```

5.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bump2version patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

CHAPTER 6

Credits

6.1 Development Lead

- Fulfil.IO Inc. <help@fulfil.io>

6.2 Contributors

None yet. Why not be the first?

CHAPTER 7

History

7.1 0.1.0 (2020-08-08)

- First release on PyPI.

CHAPTER 8

Indices and tables

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

p

- `pymws.exceptions`, [12](#)
- `pymws.fulfillment.outbound_shipment`, [10](#)
- `pymws.orders`, [10](#)
- `pymws.products`, [10](#)
- `pymws.reports`, [11](#)
- `pymws.utils`, [13](#)

A

AccessDenied, 12

C

cancel_report_request() *(pymws.reports.Reports method)*, 11

code *(pymws.utils.Marketplace attribute)*, 13

create_fulfillment_order() *(pymws.fulfillment.outbound_shipment.OutboundShipment method)*, 11

currency *(pymws.utils.Marketplace attribute)*, 13

E

endpoint *(pymws.utils.Marketplace attribute)*, 13

F

flatten_dict() *(in module pymws.utils)*, 13

flatten_list() *(in module pymws.utils)*, 13

G

get_fulfillment_order() *(pymws.fulfillment.outbound_shipment.OutboundShipment method)*, 11

get_marketplace() *(in module pymws.utils)*, 13

get_order() *(pymws.orders.Orders method)*, 10

get_report() *(pymws.reports.Reports method)*, 11

get_report_count() *(pymws.reports.Reports method)*, 11

get_report_list() *(pymws.reports.Reports method)*, 11

get_report_list_by_next_token() *(pymws.reports.Reports method)*, 12

get_report_request_count() *(pymws.reports.Reports method)*, 12

get_report_request_list() *(pymws.reports.Reports method)*, 12

get_report_request_list_by_next_token() *(pymws.reports.Reports method)*, 12

get_service_status() *(pymws.orders.Orders method)*, 10

I

id *(pymws.utils.Marketplace attribute)*, 13

L

list_order_items() *(pymws.orders.Orders method)*, 10

list_order_items_by_next_token() *(pymws.orders.Orders method)*, 10

list_orders() *(pymws.orders.Orders method)*, 10

list_orders_by_next_token() *(pymws.orders.Orders method)*, 10

M

Marketplace *(class in pymws.utils)*, 13

MARKETPLACES *(in module pymws.utils)*, 13

MWS *(class in pymws.pymws)*, 9

MWSError, 12

MWSEnvironment, 12

N

name *(pymws.utils.Marketplace attribute)*, 13

O

Orders *(class in pymws.orders)*, 10

orders *(pymws.pymws.MWS attribute)*, 9

OutboundShipment *(class in pymws.fulfillment.outbound_shipment)*, 10

P

parse_xsv() *(in module pymws.utils)*, 14

products *(pymws.pymws.MWS attribute)*, 9

pymws.exceptions *(module)*, 12

pymws.fulfillment.outbound_shipment *(module)*, 10

pymws.orders *(module)*, 10

pymws.products *(module)*, 10

`pymws.reports` (*module*), [11](#)
`pymws.utils` (*module*), [13](#)

Q

`QuotaExceeded`, [12](#)

R

`Reports` (*class in pymws.reports*), [11](#)
`reports` (*pymws.pymws.MWS attribute*), [9](#)
`request_report()` (*pymws.reports.Reports*
method), [12](#)
`RequestThrottled`, [12](#)

S

`SignatureDoesNotMatch`, [13](#)