pytugboat Documentation

Release 0.1

Tugboat Yards Inc

January 23, 2014

Contents

Before you can get started with the API you will need an API token. If you are a publisher you can get one by emailing api@tugboatyards.com.

CHAPTER 1

Methods

All methods require an instatiated Tugboat object with a valid API Authorization Token. For now you can get one by emailing api@tugboatyards.com.

`Tugboat.__init__`(*api_auth=None*, *api_endpoint=None*)
To instantiate a Tugboat() client you will need an API token. If you are a publisher you can tokens by emailing api@tugboatyards.com. You can optionally pass in a custom api_endpoint but it is not required.

`Tugboat.CheckKey`()
A test method for verifying a key is valid. This method returns True if the key is valid, False if it is invalid.

The corresponding API method is GET /key-check.

`Tugboat.GetPage`(*page_id=None*)
A method that returns a `tugboatyards.Page` object. The page_id parameter is optional. If a page_id is not specified the API defaults to the API application's default page.

The corresponding API method is GET /page[/page_id].

`Tugboat.GetOffer`(*offer_id=0*)
A method that returns a `tugboatyards.Offer` object. The offer_id is required.

The corresponding API method is GET /offer/(offer_id).

`Tugboat.GetSubscribers`(*offer_id=0*)
A method that returns a list of `tugboatyards.Subscriber` objects. The method can only be called for an offer_id that is recurring.

The corresponding API method is GET /offer/(offer_id)/subscribers.

`Tugboat.CreateOffer`(*page_id=0*, *offer_name=''*, *title=''*, *thank_you_message=''*, *amount=0.0*, *is_recurring=False*, *desc=''*, *period=''*, *offer_type=''*, *issues=[]*)
This method creates an offer. The the return value is a `tugboatyards.Offer`. Required fields are page_id, offer_name, title, thank_you_message, amount, and desc.

The corresponding API method is POST /offers.

`Tugboat.UpdateOffer`(*offer_id=None*, *amount=None*, *title=None*, *desc=None*, *period=None*, *issues=[]*)
This method updates an offer. The the return value is the `tugboatyards.Offer` with the updated attributes. The offer_id and one attribute are the only fields that are required.

The corresponding API method is POST /offer/(offer_id).

`Tugboat.DeleteOffer`(*offer_id=None*)
This is a convenience method that sets an offer's status to deleted. The the return value is the `tugboatyards.Offer` with the new status. The offer_id is required.

The corresponding API method is POST /offer/(offer_id).

Tugboat.**AppendIssueToOffer**(*offer_id=0*, *issue={}*)

This method accepts an offer_id and two item dict that should contain two keys: issue_name and title.

Example:

```
{"issue_name":  "issue-01","title":  "The issue title."}
```

The response is a list of tugboatyards.Issue objects.

The corresponding API method is POST /offer/(offer_id)/issues.

Tugboat.**GetCheckout**(*checkout_id=0*)

This method returns a tugboatyards.Checkout object. The checkout_id is required. This method is mainly used to verify an IPN message.

The corresponding API method is GET /checkout/(checkout_id).

CHAPTER 2

Models

Tugboat's models are light-weight classes which contain attributes of the objects on the server they represent. There are no instance methods and each class only has one static method (createFromDict) which accepts a dictionary and tries to instantiate an object from it.

class tugboatyards.**Page**(*page_id=0*, *offers=*[ ])
A Page object represents a publisher's Tugboat Page.

attribute	type
page_id	Integer
offers	List of tugboatyards.Offer objects.

class tugboatyards.**Issue**(*issue_name=''*, *title=''*)
A Issue contains information about an individual issue attached to an Offer. Only offers of type "magazine" can have issues attached.

attribute	type
issue_name	Alphanumeric, dashes, and underscores. (e.g. issue-01)
title	String, spaces and other characters allowed.

class tugboatyards.**Offer**(*offer_id=0*, *offer_name=''*, *user_id=0*, *page_id=0*, *amount=0.0*, *title=''*, *desc=''*, *status=''*, *is_recurring=False*, *offer_type=None*, *issues=*[ ])
The Offer object represents something that can be sold within Tugboat Pages.

attribute	type
offer_id	Integer
offer_name	Alphanumeric, dashes, and underscores. (e.g. buy-a-unicorn)
user_id	Integer
page_id	Integer
amount	Decimal
title	String, spaces and other characters allowed.
desc	String, spaces and other characters allowed.
status	One of three strings: 'active', 'deleted', or 'archived'.
is_recurring	Boolean
offer_type	None or the string 'magazine'.
issues	List of tugboatyards.Issue objects.

class tugboatyards.**Subscriber**(*subscriber_id=0*, *status=''*, *user_name=''*, *email=''*)
A Subscriber is a buyer that has purchased a recurring offer. Both active and inactive subscribers are represented by this class.

attribute	type
subscriber_id	Integer
status	String, one of 'active' or 'inactive'.
user_name	Alphanumeric, dashes, and underscores. (e.g. alincoln_1809)
email	String that matches standard email format.

class tugboatyards.**Checkout**(*checkout_id=0*, *created_on=None*, *project_id=0*, *offer_id=0*, *issue_name=''*, *amount=0.0*, *state=''*, *state_changed_on=None*, *buyer={}*)

A Checkout object contains information about a specific checkout. Only returned via the GetCheckout method.

attribute	type
checkout_id	Integer
created_on	Datetime
project_id	Integer
offer_id	Integer
issue_name	Alphanumeric, dashes, and underscores. (e.g. issue-01)
amount	Decimal
state	String, one of the following: new, authorized, reserved, captured, settled, cancelled, refunded, charged back, failed, expired.
state_changed_on	Datetime
buyer	A tugboatyards.Buyer object.

class tugboatyards.**Buyer**(*email=''*, *full_name=''*, *user_id=0*, *user_name=''*)

A Buyer can be found attached to Checkout objects.

attribute	type
email	String that matches standard email format.
full_name	String
user_id	Integer
user_name	Alphanumeric, dashes, and underscores. (e.g. alincoln_1809)

Example usage:

```
>>> import pprint
>>> from tugboatyards import Tugboat
>>> t = Tugboat("XXXXXXX%YYYYYYYYYYYYY")
>>> pprint.pprint(t.GetPage().__dict__)
{'offers': [<tugboatyards.Offer object at 0x108bb9ed0>,
            <tugboatyards.Offer object at 0x1088c8b50>,
            <tugboatyards.Offer object at 0x108e18a50>],
 'page_id': 212385}
>>> pprint.pprint(t.GetPage().offers[0].__dict__)
{
    'amount': u'10.00',
    'desc': u'This offer will make you an official supporter!',
    'is_recurring': False,
    'offer_id': 301238,
    'offer_name': u'givesupport',
    'offer_type': None,
    'page_id': 212385,
    'status': u'active',
    'title': u'Give Support',
    'user_id': 312481
}
```

CHAPTER 3

Indices and tables

- *genindex*
- *search*