Sharpy Documentation

Release 0.6.4

Saaspire LLC

Sep 27, 2017

Contents

Sharpy is a client for the Cheddar Getter (https://cheddargetter.com/) API. Cheddar Getter is a great service for handling recurring and usage based billing.

There are some existing python Cheddar Getter clients but they have significant licensing problems, packaging problems, bugs, and are only partial implementations of the Cheddar Getter API.

Sharpy offers a number of advantages:

- Clear and simple BSD license.
- Both a high and low level API - Work with cheddar the way you want to.
- 100% test coverage.
- Proper packaging - Sharpy can be installed via easy_install and PIP.
- Implements almost all of the Cheddar Getter API (See TODOs below).
- Will have complete documentation soon.

That all being said, sharpy is still very new and is likely to undergo some significant API changes in the near future. The code should be fairly safe to use as long as you understand that future releases may not be backwards compatible.

CHAPTER 1

Getting Started

To get started with Sharpy, simply install it like you would any other python package

```
pip install sharpy
```

Optionally, you can also install lxml on your system for faster XML parsing.

Once you have sharpy installed, checkout our docs on how to use the library.

CHAPTER 2

Get the documentation

Sharpy's documentation is available at ReadTheDocs or in the `docs` directory of the project.

CHAPTER 3

Code

You can checkout and download Sharpy's latest code at Github.

CHAPTER 4

TODOs

- Flesh out the documentation to cover the full API.
- Add support for the various filtering options in the *get_customers* call.

CHAPTER 5

Documentation

Examples

Below are a series of examples of how to do common tasks using sharpy.

Customers

Creation

Customer creation is generally just a simple call on a product instance. The `CheddarProduct.create_customer()` method supports everything that the Cheddar Getter API add customer call supports.

Free Customers

Free customers require a minimal amount of information. Accordingly, the call to create them is pretty simple.

```
from sharpy.product import CheddarProduct

# Get a product instance to work with
product = CheddarProduct(
    username = CHEDDAR_USERNAME,
    password = CHEDDAR_PASSWORD,
    product_code = CHEDDAR_PRODUCT,
)

# Create the customer
customer = product.create_customer(
    code = 'cust-id-1', # An unique identifier for the customer
    first_name = 'Philip',
    last_name = 'Fry',
    email = 'pfry@planetexpress.com',
```

```
    plan_code = 'FREE_MONTHLY', # The code for plan to subscribe the customer to
)
```

Paypal Customers

Paypal customers require only a little bit more information. You ask CheddarGetter to create the account, then redirect the user to the Paypal site to log in and authorize the subscription.

```
from sharpy.product import CheddarProduct

# Get a product instance to work with
product = CheddarProduct(
    username = CHEDDAR_USERNAME,
    password = CHEDDAR_PASSWORD,
    product_code = CHEDDAR_PRODUCT,
)

# Create the customer
customer = product.create_customer(
    code = 'cust-id-2',
    first_name = 'Turanga',
    last_name = 'Leela',
    email = 'tleela@planetexpress.com',
    plan_code = 'PAID_MONTHLY',

    method = 'paypal',

    cc_first_name = 'Hubert',
    cc_last_name = 'Farnsworth',

    return_url = 'https://www.planetexpress.com/thanks/cust-id-2/',
    cancel_url = 'https://www.planetexpress.com/sorry/cust-id-2/',
)
# redirect the user to the Paypal site to authorize the subscription
redirect(
    customer.subscriptions[0].redirect_url,
)
```

When the user has accepted or cancelled the subscription they will be redirected to the URLs passed into the creation call. You should always verify the status of the account with CheddarGetter directly, rather than relying on the user visiting those URLs.

Paid Customers

Paid customers require a bit more information, but still are fairly easy to create.

```
from sharpy.product import CheddarProduct

# Get a product instance to work with
product = CheddarProduct(
    username = CHEDDAR_USERNAME,
    password = CHEDDAR_PASSWORD,
    product_code = CHEDDAR_PRODUCT,
)
```

```

# Create the customer
customer = product.create_customer(
    code = 'cust-id-2',
    first_name = 'Turanga',
    last_name = 'Leela',
    email = 'tleela@planetexpress.com',
    plan_code = 'PAID_MONTHLY',
    cc_number = '4111111111111111',
    cc_expiration = '03/3012',
    cc_card_code = '123',
    cc_first_name = 'Hubert',
    cc_last_name = 'Farnsworth',
    cc_address = '1 πth Ave.',
    cc_city = 'New New York',
    cc_state = 'New York',
    cc_zip = '10001',
)
```

Customers With Optional Info

Cheddar has a number of optional fields which they accept when creating a customer. Most of these fields are simply provided to help you keep track of customer info, but some of the fields (e.g. `initial_bill_date`) can affect Cheddar Getter's behavior. Check out Cheddar Getter's API docs for more details on specific optional fields.

```
from datetime import datetime, timedelta

from sharpy.product import CheddarProduct

# Get a product instance to work with
product = CheddarProduct(
    username = CHEDDAR_USERNAME,
    password = CHEDDAR_PASSWORD,
    product_code = CHEDDAR_PRODUCT,
)

# Set a intitial bill date in the future to provide a free trial
bill_date = datetime.now() + timedelta(days=60)

# Create the customer
customer = product.create_customer(
    # Required fields
    code = 'cust-id-3',
    first_name = 'Hermes',
    last_name = 'Conrad',
    email = 'hconrad@planetexpress.com',
    plan_code = 'PAID_MONTHLY',
    cc_number = '4111111111111111',
    cc_expiration = '03/3012',
    cc_card_code = '123',
    cc_first_name = 'Hubert',
    cc_last_name = 'Farnsworth',
    cc_address = '1 πth Ave.',
    cc_city = 'New New York',
    cc_state = 'New York',
    cc_zip = '10001',
```

```
    # Optional Fields
    initial_bill_date = bill_date,
    referer = 'http://www.momsfriendlyrobots.com/',
)
```

Note: Notice that the initial_bill_date is provided as a `datetime` object. Sharpy always expects and returns the best python native data structure for a given piece of information. That being said, sharpy is generally pretty flexible about what forms of data it will take in.

Customers With Tracked Items

Depending on your pricing model, you may want to create customers as having already consumed tracked items. Fortunately, Cheddar Getter's API allows us to provide tracked item information as part of the create customer call. While you can always report on tracked items with their own API calls, reporting tracked item usage as part of customer creation saves a potentially slow request/response cycle and avoids possible error states where a create customer call succeeds but subsequent API calls fail.

To create a customer with tracked item information, all you need to do is pass in a collection of dictionaries to the `items` parameter of the `CheddarProduct.create_customer()` which describe the desired item info.

```
from sharpy.product import CheddarProduct

# Get a product instance to work with
product = CheddarProduct(
    username = CHEDDAR_USERNAME,
    password = CHEDDAR_PASSWORD,
    product_code = CHEDDAR_PRODUCT,
)

# Describe item use
items = [
    {'code': 'poplers', 'quantity': '1000'},
    {'code': 'dark matter balls'},
]

# Create the customer
customer = product.create_customer(
    code = 'cust-id-3',
    first_name = 'Turanga',
    last_name = 'Leela',
    email = 'tleela@planetexpress.com',
    plan_code = 'PAID_MONTHLY',
    cc_number = '4111111111111111',
    cc_expiration = '03/3012',
    cc_card_code = '123',
    cc_first_name = 'Hubert',
    cc_last_name = 'Farnsworth',
    cc_address = '1 πth Ave.',
    cc_city = 'New New York',
    cc_state = 'New York',
    cc_zip = '10001',
    items = items,
)
```

Notice here that the `quantity` value of the item dictionary is optional. If you don't include a `quantity` value, sharpy assumes a quantity of `1`.

Customers With Custom Charges

Similar to creating customers with tracked items, the Cheddar Getter API allows you to provide custom charge information as part of the create customer call. Doing so saves a request to the API and eliminates the risk of customers getting created without the desired charges. Just like with tracked items, you provide charge information by building a collection of dictionaries and passing that collection to the `charges` parameter of `CheddarProduct.create_customer()`.

```
from decimal import Decimal
from sharpy.product import CheddarProduct

# Get a product instance to work with
product = CheddarProduct(
    username = CHEDDAR_USERNAME,
    password = CHEDDAR_PASSWORD,
    product_code = CHEDDAR_PRODUCT,
)

# Describe item use
charges = [
    {
        'code': 'Senior Discount',
        'quantity': '2',
        'each_amount': Decimal('-10.21'),
        'description': "A discount for every half-centry of the customer's age"
    },
    {
        'code': 'Mom Tax',
        'each_amount': 200,
    },
]

# Create the customer
customer = product.create_customer(
    code = 'cust-id-3',
    first_name = 'Turanga',
    last_name = 'Leela',
    email = 'tleela@planetexpress.com',
    plan_code = 'PAID_MONTHLY',
    cc_number = '4111111111111111',
    cc_expiration = '03/3012',
    cc_card_code = '123',
    cc_first_name = 'Hubert',
    cc_last_name = 'Farnsworth',
    cc_address = '1 πth Ave.',
    cc_city = 'New New York',
    cc_state = 'New York',
    cc_zip = '10001',
    charges = charges,
)
```

A few things to notice here:

- Quantity is optional and will be assumed to be `1` if not provided.

- Description is optional and will be assumed to be `""`.
- Each amount can be of any type which can be cast to Decimal.
- Negative values for `each_amount` are credits to the customer's account, positive values are charges.

Fetching

Get an individual customer

To fetch the information for an individual customer, simply call `CheddarProduct.get_customer()` with your customer's code. The customer's code is the unique identifier you provided in your customer creation call.

```
from sharpy.product import CheddarProduct
from sharpy import exceptions

# Get a product instance to work with
product = CheddarProduct(
    username = CHEDDAR_USERNAME,
    password = CHEDDAR_PASSWORD,
    product_code = CHEDDAR_PRODUCT,
)

try:
    # Get the customer from Cheddar Getter
    customer = product.get_customer(code='1BDI')
except exceptions.NotFound, err:
    print 'You do not appear to be a customer yet'
else:
    # Test if the customer's subscription is canceled
    if customer.subscription.canceled:
        if customer.subscription.cancel_type == 'paypal-pending':
            print 'Waiting for Paypal authorization'
        else:
            print 'Your subscription appears to have been cancelled'
    else:
        print 'Your subscription appears to be active'
```

If a customer with the given code cannot be found, a `sharpy.exceptions.NotFound` exception will be raised.

Get all customers

```
from sharpy.product import CheddarProduct

# Get a product instance to work with
product = CheddarProduct(
    username = CHEDDAR_USERNAME,
    password = CHEDDAR_PASSWORD,
    product_code = CHEDDAR_PRODUCT,
)

# Get the customer from Cheddar Getter
customers = product.get_customers()
```

Hosted Updates

CheddarGetter has a "hosted" solution that allows you to get up and running quickly. This solution allows you to direct the customer to a CheddarGetter-hosted site to sign up. Once signed up, you need to be able to send the customer back to the correct account to modify their subscriptions. To do this, you need to calculate a CheddarGetter key for the account.

```
import hashlib, urllib
key = hashlib.md5( '%s|%s'%( customer_code, CHEDDAR_PRODUCT_KEY, )).hexdigest()[:10]
query = urllib.urlencode({
    'code': customer_code,
    'key': key,
})
redirect(
    CHEDDAR_HOSTED_DOMAIN + '/update' + '?' + query
)
```

High Level API

Sharpy's high level API provides a nice pythonic interface for working with the Cheddar Getter API. While there are some variations to make things a bit nicer to work with in Python, most of the methods below map pretty directly. If you are looking for more detail on what exactly an API call does or what exactly a particular value means in a given context, you should look at the Cheddar Getter API Docs.

For the most part, sharpy's high level API presents itself as a collection of classes/objects which map to logical entities in the Cheddar Getter API. Below are details on those classes and some examples of their use.

CheddarProduct

Developing Sharpy

Due to Sharpy's nature as an API client and due to some limitations in Cheddar Getter's API, there is a bit of a setup one needs to do to work on sharpy.

Cheddar Getter Account

To run sharpy's test suite, you will need to create an account and product on cheddar getter. As part of that setup, you will need to manually create certain plans and set certain options as they are not available to change via Cheddar's API.

Create Account

Go to the Cheddar Getter website and click on the "sign up" link at the top right corner of the page to create account. Their free account should give you everything you need to work on sharpy. Once you have completed cheddar's sign-up form you will need to wait for a confirmation email.

> **Warning:** If you already have an account, you may use that account and simply create a new product, but creating a separate account will protect you from accidentally running the test suite against and modifying the wrong product.

Add a Product

Once you receive your confirmation email, click on the link it contains. You should be sent to a page asking you to add a product. What you name your product and what you select as your product code does not matter, but be sure to record what you set the product code to be. You will need this information later to configure Sharpy. When you have completed the form, click on "Add product" and move on to the next step.

Setup Pricing Plans

Once your product is setup, you will need to setup a few pricing plans. These plans will be used by the test suite in various ways.

> **Warning:** If you do not setup your plans exactly as described here, you may get false errors from the Sharpy test suite.

To get started, click on "Create Pricing Plans" on your dashboard page. From there click on the "Add Plan" button at the top right. Once on the "add plan" page, fill out the form according to the information in the first row below, and click on the "Add Plan" button at the bottom of the page.

Once your new plan is created click on "Add Plan" at the top of the page and repeat the process until all of the plans described below have been entered.

Free Monthly

Name Free Monthly

Code FREE_MONTHLY

Frequency Monthly

Description A free monthly plan

Monthly Amount 0.00

Charge Code FREE_MONTHLY_RECURRING

Setup Fee No

Setup Amount 0.00

Setup Code FREE_MONTHLY_SETUP

Tracked Items *None*

First Bill *Leave defaults*

Paid Monthly

Name Paid Monthly

Code PAID_MONTHLY

Frequency Monthly

Description *None*

Monthly Amount 20.00

Charge Code PAID_MONTHLY_RECURRING

Setup Fee No

Setup Amount 0.00

Setup Code PAID_MONTHLY_SETUP

Tracked Items *None*

First Bill *Leave defaults*

Tracked Monthly

Name Tracked Monthly

Code TRACKED_MONTHLY

Frequency Monthly

Description *None*

Monthly Amount 10.00

Charge Code TRACKED_MONTHLY_AMOUNT

Setup Fee No

Setup Amount 0.00

Setup Code TRACKED_MONTHLY_SETUP

Tracked Items **Monthly Item**

Name Monthly Item

Code MONTHLY_ITEM

Quantity Inc 2

Overage Amount 10.00

Monthly Yes

Once Item

Name Once Item

Code ONCE_ITEM

Quantity Inc 0

Overage Amount 10.00

Monthly No

Setup Payment Gateway

After you have entered all of the pricing plans, click on the overview tab. If you created a new cheddar account, then you don't need to do anything to setup your payment gateway. If you are using an existing cheddar account, make sure that you are setup to use Cheddar's simulator gateway. You can click on "Payment Gateway" to view your gateway settings.

Cheddar Plan

On your overview page, cheddar will prompt you to setup a paid plan. With your production account you will almost certainly want to do this, but for the purposes of testing Sharpy, the free plan is fine.

Configuration

Again, if you have setup a new account, you should be fine running the sharpy tests with the default cheddar settings. If you are using an existing account you will want to disable all email sending. If you do not, you will send out a ton of garbage emails every time you run the test suite. Additionally, you should do everything you can to make sure that your setting match cheddar's default settings. If you do not, you may get false failure while running sharpy's test suite.

Maintenance

Congratulations! You are done setting up your cheddar account for testing sharpy. Generally speaking, you shouldn't need to touch this account any further. The test suite should be pretty good about cleaning up after itself and leaving the account in the state it was in when the test run started. That said, if things really break you may occasionally need to log in and clean out old/bad test data. Similarly, future releases of sharpy should generally work with the data setup here, but they may occasionally require adjustments to your cheddar account. Should that happen, there will be a notice in the notes for the given release.

Now you just need to setup your local environment and you'll be all set.

Local Environment

There is a little bit of setup you need to do to get the sharpy test suite running on your local machine. This setup mostly involves installing some development/testing tools as well as configuring the test suite so that it knows what it needs to know about your Cheddar Getter account.

Getting the code

Sharpy's main repo is hosted on Github. The easiest way to work with the sharpy repo will be to login to Github and make your own fork of sharpy. Once logged in to Github, go to the Sharpy repo page and click on the "Fork" button at the top of the page. This will give you your own repo which you can push code up to. When you have any changes that you'd like to contribute back, you can make a pull request from your repo page and we'll check out your change. To get the code on to your local machine, go to your repo page, copy the SSH or HTTP url provided at the top of the page, and then run the command `git clone <your repo url>`. Git will run for a little bit and you will have a full copy of sharpy downloaded and ready to work on.

Setting up an environment

We recommend that you work within a virtualenv while working on sharpy, but it is not required. Working in the virtualenv makes adding and removing packages a bit easier and it reduces possible problems caused by conflicting packages. See the virtualenv docs for details on how to use virtualenv.

Add sharpy to your python path

To run the tests, sharpy must be along your python path. There are a few ways to possibly accomplish this but the easiest is a .pth file. Simply create a file called `sharpy.pth` in your site-packages directory (with virtualenv this will be something like `/path/to/your/env/lib/python2.#/site-packages/`) and put the path to your local clone of sharpy as the contents of the file.

Install Dependencies

Sharpy has a few dependencies which are normally handled by setup.py and there are a few additional packages which the test suite depends on. The easiest way to install these packages is with pip. Install pip on your system and then from the root of your sharpy directory, run the command `pip install -r dev-requirements.txt`. This will install everything you need.

> **Warning:** Be sure that you have activated your virtualenv before running `pip install`. If you have not, you will install all of the dependency packages to your global site-packages instead of your virtualenv.

Create Config File

Sharpy's test suite uses a simply ini style config file to handle your cheddar credentials. In the `tests` directory, there is a file called `config.ini.template`. Copy this file to a new file called `config.ini` in your `tests` directory. Once copied, open your `config.ini` and enter the proper values for your cheddar account.

> **Warning:** The sharpy test suite modifies and deletes data in the cheddar account/product which it is configured to work against. Be sure that you enter the credentials for your **testing** account. If you enter the credentials for your real cheddar account you will end up **DELETING CUSTOMERS**.
>
> **DO NOT RUN THE SHARPY TEST SUITE AGAINST ANYTHING BUT A TEST ACCOUNT!**

Running the Tests

We're finally ready to run some tests! Go into the root of your clone of Sharpy and run the command `nosetests`. You should see the output of the tests as they run and a coverage report at the end. Sharpy's goal is to maintain complete test coverage and any patches without appropriate, *passing* tests will not be accepted.

Be aware that the full test suite may take a while to run as many of the test are making actual calls to cheddar. Relatedly, if you don't have a working internet connection, your run of the test suite will fail.

Indices and tables

- genindex
- modindex
- search