

---

# **AggCat Documentation**

***Release 0.3***

**Glen Zangirolami**

September 28, 2013



# CONTENTS

<b>1</b>	<b>Before Using The API Client</b>	<b>3</b>
<b>2</b>	<b>Installation</b>	<b>5</b>
<b>3</b>	<b>Known Issues</b>	<b>7</b>
<b>4</b>	<b>Initializing the API Client</b>	<b>9</b>
<b>5</b>	<b>Querying the API</b>	<b>11</b>
5.1	Getting all institutions . . . . .	11
5.2	Searching for your institution . . . . .	11
5.3	Getting the institution details . . . . .	12
<b>6</b>	<b>User's Guide</b>	<b>13</b>
6.1	API . . . . .	13
<b>7</b>	<b>Release Notes</b>	<b>17</b>



Ready to get your financial data? This quickstart is intended to get you setup to start using [Intuit's Customer Account Data API](#).



# BEFORE USING THE API CLIENT

You must take a few steps on Intuit's website before using the API client:

1. [Create a development account](#) and login.
2. [Create a new application](#) in the Customer Account Data category. It's easiest to [follow the instructions in the help documentation](#).
3. [Create and upload self a generated x509 certificate to your application](#). It's easiest to use openssl to generate the certificate. *It's best to name your key the same as your application name. **Don't lose these certificates!***
4. [Gather the login details](#) and store them somewhere. You will need *OAuth Consumer Key*, *OAuth Consumer Secret*, *SAML Identity Provider ID*





# INSTALLATION

```
pip install python-aggrcat
```



## KNOWN ISSUES

The SSL library in Python 2.6 and below has a bug and will not parse the `AlternativeNames` out of the Intuit SSL cert causing a name mismatch during certificate validation. For now, please pass `verify_ssl = False` to the `AggcatClient` when initializing it. While less secure, I wanted the verification to be turned off explicitly so you are aware. If possible, upgrade to Python 2.7+.



# INITIALIZING THE API CLIENT

Assuming you have an *OAuth Consumer Key*, *OAuth Consumer Secret*, *SAML Identity Provider ID*, and a path to the x509 certificates you generated you are ready to start querying:

```
from aggcat import AggcatClient

client = AggcatClient(
    'oauth_consumer_key',
    'oauth_consumer_secret',
    'saml_identity_provider_id',
    'customer_id',
    '/path/to/x509/appname.key'
)
```

---

**Note:** `customer_id` (Integer) It can be any integer. You should try using the database primary key of a user in your system or some other unique identifier such as a guid. If you are just testing you can use whatever integer you want.

`objectify` (Boolean) This is a BETA functionality. It will objectify the XML returned from intuit into standard python objects so you don't have to mess with XML. Default: `True`

---



# QUERYING THE API

Here are a few sample queries that don't require you to add an account

## 5.1 Getting all institutions

```
institutions = client.get_institutions()
```

---

**Note:** This query will take a very long time depending on your internet connection. It returns 18000+ institutions in XML format. Sux :(

---

If you are using the `objectify = True` keyword argument on the client you can access the institutions in a pythonic way

```
>>> institutions = client.get_institutions()
>>> len(institutions)
18716
>>> institutions[0].institution_name
'Carolina Foothills FCU Credit Card'
```

## 5.2 Searching for your institution

Currently finding an institution is somewhat of a manual process. Soon, there will be a helper method on the client that will have a better search. Patches welcome ;). This example searches for an institution that contains “chase” in any of the XML elements:

```
from aggcat import AggcatClient
from lxml import etree
from aggcat.utils import remove_namespaces

client = AggcatClient(
    'oauth_consumer_key',
    'oauth_consumer_secret',
    'saml_identity_provider_id',
    'customer_id',
    '/path/to/x509/appname.key'
)

search_string = 'Chase'
institutions = client.get_institutions()
```

```
xml = etree.fromstring(institutions.to_xml())
xml = etree.fromstring(remove_namespaces(xml))

for element in xml.xpath('./institution[contains(., "chase")]'):
    id = element.xpath('./institutionId')[0].text
    name = element.xpath('./institutionName')[0].text
    print id, name

13278 JP Morgan Chase Bank
13640 Quicken Visa
14554 Chase Bank Credit Card (Amazon.com)
14910 Chase e-Funds Card
14777 Fox Chase Bank - Business Banking
13718 Fox Chase Bank
14484 Chevy Chase Bank - Web Cash Manager
...
```

---

**Note:** This query will take a very long time depending on your internet connection. It returns 18000+ institutions in XML format. Sux :(

---

## 5.3 Getting the institution details

From the previous search example, we can use 13278 to get the institution details

```
institution_details = client.get_institution_details(13278)
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><InstitutionDetail xmlns="http://schema.intuit
```

If you are using the `objectify = True` keyword argument on the client you can access the institution parameters in a Pythonic way

```
>>> institution = client.get_institution_details(13278)
>>> institution
<Institutiondetail object @ 0x110389990>
>>> institution.institution_name
'JP Morgan Chase Bank'
>>> institution.home_url
'https://www.chase.com/'
>>> institution.currency_code
'USD'
```



# USER'S GUIDE

## 6.1 API

### 6.1.1 Initializing the client

### 6.1.2 Authorization & updating credentials

#### Getting credential fields

At some point in time you will need to allow a user to login to an institution so you can get their account information. `get_credential_fields()` is not part of Intuit's API, but aids you in getting the fields you need to present in your UI layer.

To get credential fields use `get_credential_fields()`. It will give you the fields in a `list` format pre-ordered. It's a precursor to `discover_and_add_accounts()`

#### Authenticating and adding accounts

Once you have gathered the username and password for a specific institution you will need to `discover_and_add_accounts()`. This particular endpoint will either return a list of accounts if authentication works or return a list of challenges that need to be answered.

#### Responding to a challenge

If you attempted to authenticate and received back a `Challenge` response then you will need to answer that challenge by sending your responses to `confirm_challenge()`

#### Updating outdated login credentials

If you have changed your institution credentials you will need to let Intuit know that it needs to use new credentials. For example, let's assume you have a Bank of America or JP Morgan Chase account and changed your password from the banking web site. At this point Intuit will need to be notified of this update.

## Updating outdated challenge responses

### 6.1.3 Working with Institutions

#### Getting all Institutions

#### Searching for your institution

Currently finding an institution is somewhat of a manual process. Soon, there will be a helper method on the client that will have a better search. Patches welcome ;). This example searches for an institution that contains “chase” in any of the XML elements:

```
from aggcat import AggcatClient
from lxml import etree
from aggcat.utils import remove_namespaces

client = AggcatClient(
    'oauth_consumer_key',
    'oauth_consumer_secret',
    'saml_identity_provider_id',
    'customer_id',
    '/path/to/x509/appname.key'
)

search_string = 'Chase'
institutions = client.get_institutions()

xml = etree.fromstring(institutions.to_xml())
xml = etree.fromstring(remove_namespaces(xml))

for element in xml.xpath('./institution[contains(., "chase")]'):
    id = element.xpath('./institutionId')[0].text
    name = element.xpath('./institutionName')[0].text
    print id, name

13278 JP Morgan Chase Bank
13640 Quicken Visa
14554 Chase Bank Credit Card (Amazon.com)
14910 Chase e-Funds Card
14777 Fox Chase Bank - Business Banking
13718 Fox Chase Bank
14484 Chevy Chase Bank - Web Cash Manager
...
```

---

**Note:** This query will take a very long time depending on your internet connection. It returns 18000+ institutions in XML format. Sux :(

---

#### Getting Institution details

### 6.1.4 Accounts & Account transactions

Here is the meat of the API. This is what you came for, the ability to aggregate your financials. Here be dragons or not. :)

## **Customer accounts**

### **Login Accounts**

### **Single Account Details**

### **Transactions**

This might hurt or be candy to your eyes depending on how good of a budgeter you are. heh!

### **Investement Positions**

### **Updating Account Type**

Certain banks do not present account types in any of the API data or data that comes back from screen scraping banks accounts without APIs. In those cases the account types are noted as “Other”. Most likely, you will know what the account actually is so you can update the type here.

### **Deleting An Account**

### **Deleting a customer**



# RELEASE NOTES

## 0.3

- Switched oAuth backend from oAuth2 to `requests-oauthlib` because it is maintained and causing less issues. Plus, `requests` is awesome ;)
- Added `verify_ssl` keyword argument to `AggcatClient` so that the library work under Python 2.6 due to an SSL library bug parsing Intuits SSL Certificate. See [Known Issues](#)
- Added Counter backport for Python 2.6 <http://code.activestate.com/recipes/576611-counter-class/>

## 0.2

- Cleanup
- Made `end_date` an optional parameter in `get_account_transactions` to reflect intuit
- Added `requirements.pip` file do that docs build correctly on readthedocs.org

## 0.1

- Initial Release