
Basiq Rapid Documentation

Release 1.0.0

Michael Gee

Mar 25, 2019

Contents

1	About	1
2	Install	3
3	Getting started	5
4	Contents	7
4.1	Quickstart Guide	7
4.2	Sessions	8
4.3	Users	9
4.4	Connections	12
4.5	Transactions	13
4.6	Filters	14
4.7	Utilities	15
4.8	Class Object Details	15

CHAPTER 1

About

Basiq Rapid is a Python wrapper for the [basiq.io](#) API.

Perfect for the developer and analyst:

- API Filters

- Quick Integration with Pandas

- Object-oriented wrapper allows for simultaneous users and operations

Refer to [basiq API](#) for API (and thus Basiq Rapid) capabilities.

Basiq Rapid is not affiliated with basiq.io. See disclaimer.

CHAPTER 2

Install

Installation:

```
pip install basiqrapi
```

Dependencies: json, requests, pandas

CHAPTER 3

Getting started

Get started with our [Quickstart Guide](#).

Read about the API at [basiq.io](#).

4.1 Quickstart Guide

Import Basiq Rapid into Python:

```
import basiqrapid as bq
```

Create a session:

```
API_KEY = ""  
  
session = bq.Session(API_KEY)
```

Get institutions:

```
institution_dict = session.getInstitutions()  
wbc_code = institution_dict['Westpac Banking Corporation']
```

Create a User:

```
usr = session.createUser(email="gavin@hooli.com", mobile="+61888000888")
```

Create a connection:

```
job = usr.createConnection(loginID="gavinBelson", password="hooli2016", institutionID=  
    ↪ "AU00000")  
print(job.getStatus())
```

Some User Capabilities:

```
transactions = usr.getTransactions() # Returns list of Transaction objects  
connections = usr.getConnections() # Returns list of Connection objects  
income = usr.getIncome()  
income_year = income.regularIncomeYear
```

Some Connection Capabilities:

```
first_connection = connections[0]
first_connection.institution_id # Returns institution ID
accounts = first_connection.getAccounts()
```

Some Account Capabilities:

```
first_account = accounts[0]
transactions = first_account.getTransactions()
```

Some Transaction Capabilities:

```
first_transaction = transactions[0]
print(transaction.balance)
print(transaction.description)
```

Convert to pandas DataFrame:

```
import pandas as pd

transactions = usr.getTransactions()
trans_df = bq.to_df(transactions)

accounts = usr.getAccounts()
acc_df = bq.to_df(accounts)
```

Filters:

```
f = bq.newFilter()
f.gt("transaction.postDate", "2018-01-28")
transactions = usr.getTransactions(f)
```

4.2 Sessions

4.2.1 Creating a session

Creating a basiq.io Session:

```
import basiqrapid as bq

API_KEY = "YOUR_API_KEY"

session = bq.Session(API_KEY)
```

4.2.2 Capabilities

Detailed object methods and subroutines available [here](#).

Create a User

Example Usage:

```
usr = session.createUser(email="gavinBelson",mobile="+61888000888")
```

Get a User by ID

Example Usage:

```
user_id = "888888888"
usr = session.getUser(user_id)
```

Get dict of institutions

Example Usage:

```
inst_dict = session.getInstitutions()
wbc_ID = inst_dict["Westpac Banking Corporation"]
```

Get institutions JSON

Example Usage:

```
inst_json = session.getInstitutionsJSON()
```

Search for an institution code

Example Usage:

```
results = session.searchInstitutions("Westpac")
for result in results:
    print(result)
```

Delete User

Example Usage:

```
user_id = "888888888"
session.deleteUser(user_id)
```

Or:

```
del usr
```

4.3 Users

4.3.1 Creating a user

Creating a user:

```
usr = session.createUser(email="gavinBelson",mobile="+61888000888")
```

Or:

```
user_id = "88888888"
usr = session.getUser(user_id)
```

4.3.2 Capabilities

Detailed object methods and subroutines available [here](#).

Basics

Example usage:

```
# Return email
usr.email
# Return mobile
usr.mobile
# Return ID
usr.id
```

Update user details

Example Usage:

```
usr.update(email="new@email.com", mobile="+61409838099")
# Returns True if success
```

Get transactions

Returns a list of Transaction objects.

Example Usage:

```
transactions = usr.getTransactions()
```

Get transaction

Returns a Transaction object.

Example Usage:

```
transaction = usr.getTransaction("transaction-id823904")
```

Get income

Returns an Income object.

Example Usage:

```
income = usr.getIncome()  
print(income.regularIncomeYear)
```

Get Expenses

Returns an Expense object.

Example Usage:

```
expense = usr.getExpenses()  
print(expense.totalMonthly)
```

Add Connection

Returns a Job object

Example Usage:

```
job = usr.addConnection(loginID="gavinBelson", password="hooli2016", institutionID=  
↪ "AU00000")
```

Get Accounts

Returns list of Account objects.

Example Usage:

```
accounts = usr.getAccounts()
```

Get Account

Returns account object.

Example Usage:

```
account = usr.getAccount("account-id786239420")
```

Get Connections

Returns list of Connection objects.

Example Usage:

```
connections = usr.getConnections()
```

Get Connection

Returns connection object.

Example Usage:

```
connection = usr.getConnection("conn-id2138967192837")
```

4.4 Connections

4.4.1 Creating a connection

Example:

```
job = usr.addConnection(loginID="gavinBelson", password="hooli2016", institutionID=
↳ "AU00000")

if job.isComplete():
    connections = usr.getConnections()

conn = connections[0] # Choose connection from list
```

4.4.2 Capabilities

Detailed object methods and subroutines available [here](#).

Basics

Example usage:

```
# Return ID
conn.id
# Return institution ID
conn.institution_id
# Return when last used
conn.lastUsed
# Return status (whether pending a job)
conn.status
```

Refresh connection

Refresh connection to get latest transactions.

Example Usage:

```
job = conn.refresh()
```

Get accounts

Returns a list of Account objects linked with connection.

Example Usage:

```
accounts = conn.getAccounts()
```


Get transactions

Returns a list of Transaction objects linked with connection.

Example Usage:

```
transactions = conn.getTransactions()
```

4.5 Transactions

4.5.1 Creating a transaction

Use `.getTransaction()` method on User, Connection, or Account objects.

Example:

```
accounts = usr.getAccounts()
acc = accounts[1]
```

4.5.2 Capabilities

Detailed object methods and subroutines available [here](#).

Basics

Example usage:

```
# Return ANZSIC code
trans.ANZSIC_code

# Return ANZSIC title
trans.ANZSIC_title

# Return parent account ID
trans.account

# Return transaction amount
trans.amount

# Return balance after transaction
trans.balance

# Return parent connection ID
trans.connection

# Return transaction description
trans.description

# Return direction of transaction (eg. credit)
trans.direction

# Return transaction ID
trans.id
```

(continues on next page)

(continued from previous page)

```
# Return parent institution ID
trans.institution

# Return transaction post date
trans.postDate

# Return transaction status (eg. posted)
trans.status

# Return transaction date (empty if not complete)
trans.transactionDate
```

4.6 Filters

4.6.1 Creating a filter

Example:

```
f = bq.newFilter()
```

4.6.2 Options

Filters are used in accordance with the Basiq API's capabilities.

Filter objects can have any amount of conditionals: - equal to (eq) - between (bt) - greater than (gt) - greater than or equal to (gteq) - less than (lt) - less than or equal to (lteq)

Equal to (eq)

Example:

```
f.eq("account.id", "s55bf3")
```

Between (bt)

Values are inclusive

Example:

```
f.bt("transaction.postDate", "2017-09-28", "2018-01-30")
```

Greater than (gt)

Example:

```
f.gt("transaction.postDate", "2017-09-28")
```

Greater than or equal to (gteq)

Example:

```
f.gteq("transaction.postDate", "2017-09-28")
```

Less than (lt)

Example:

```
f.lt("transaction.postDate", "2017-09-28")
```

Less than or equal to (lteq)

Example:

```
f.lteq("transaction.postDate", "2017-09-28")
```

4.6.3 Usage

Add to any get methods.

Example usage:

```
# Passed as parameter to getAccounts method
accounts = usr.getAccounts(f)
```

4.7 Utilities

4.7.1 To Pandas Dataframe

to_df method

Example usage:: `accounts = usr.getAccounts()` `accounts_df = bq.to_df(accounts)`

`transactions = usr.getTransactions()` `trans_df = bq.to_df(transactions)`

4.8 Class Object Details

class `main.Session(API_KEY)`

createUser (*email*="", *mobile*="")

Creates User.

Creates user and returns User object.

Parameters

- **email** (*str*) – User's email. Optional if mobile entered.

- **mobile** (*str*) – User’s mobile number. Optional if email entered.

Returns Returns the created User class object.

Return type *User* (obj)

deleteUser (*userID*)

Delete user.

Parameters **userID** (*str*) – User’s ID.

Returns Indicates success

Return type True

getInstitutions ()

Fetches institutions

Fetches user and returns dict of institutions.

Parameters **None** –

Returns Returns dict of institutions.

Return type dict

getInstitutionsJSON ()

Fetches institutions JSON

Fetches user and returns raw json of institutions.

Parameters **None** –

Returns Returns raw json of institutions.

Return type json

getUser (*userID*)

Fetches user

Fetches user and returns User object.

Parameters **userID** (*str*) – User’s ID.

Returns Returns the created User class object.

Return type *User* (obj)

searchInstitutions (*searchTerm*)

Finds institutions based on search

Parameters **searchTerm** (*str*) – Search term of bank (eg. ‘Westpac’)

Returns Returns dict of search results.

Return type dict

class main.**User** (*user_json*, *header*)

addConnection (*loginID*=”, *password*=”, *institutionID*=”)

Add’s bank account connection to user.

Returns a job object that monitor’s API progress.

Parameters

- **loginID** (*str*) – User’s login username for banking account

- **password** (*str*) – User’s password for banking account
- **institutionID** (*str*) – Bank’s ID.

Returns Returns Job object that monitor’s connection progress

Return type *Job* (obj)

getAccounts (*input_filter=None*)

Fetches all user accounts

Parameters **None** –

Returns Returns list of Account objects linked with user.

Return type *list*

getConnections (*input_filter=None*)

Fetches list of user’s connections.

Parameters **None** –

Returns Returns list of Connection objects

Return type *list*

getExpenses ()

Fetches user’s expenses summary.

Fetches user’s expenses summary and returns an Expense object.

Parameters **None** –

Returns Returns the user’s expenses in form of Expense class object.

Return type *Expense* (obj)

getIncome ()

Fetches user’s income summary.

Fetches user’s income summary and returns an Income object.

Parameters **None** –

Returns Returns the user’s income in form of Income class object.

Return type *Income* (obj)

getTransaction (*transactionID*)

Fetches transaction.

Fetches transactions and returns list of Transaction objects.

Parameters **transactionID** – transactionID string

Returns Returns list of Transaction objects. (max: 500)

Return type *list*

getTransactions (*input_filter=None*)

Fetches transactions.

Fetches transactions and returns list of Transaction objects.

Parameters **None** –

Returns Returns list of Transaction objects. (max: 500)

Return type *list*

refreshAllConnections ()

Refreshes all of user's connections.

Ensure's most up-to-date information.

Parameters **None** –

Returns returns list of Job objects that monitor connections' update

Return type `list`

refreshConnection (*connectionID*)

Refreshes specific connection, ensuring most up-to-date information.

Parameters **connectionID** (*str*) – ID of connection to be refreshed.

Returns Returns Job object to monitor progress

Return type *Job* (obj)

to_dict ()

Convert object details to dict.

Hidden method for converting object to pandas dataframe.

Parameters **None** –

Returns Returns dict of object.

Return type `dict`

update (*email=""*, *mobile=""*)

Updates user's details.

Parameters

- **email** (*str*) – Email to be updated. Optional if mobile entered.
- **mobile** (*str*) – Mobile to be updated. Optional if email entered.

Returns Indicates success.

Return type `True`

class `main.Connection` (*connect_json*, *header*)

getAccounts (*input_filter=None*)

Fetches accounts linked with connection.

Returns a list of Account objects.

Parameters **None** –

Returns Returns a list of Account objects linked with the connection.

Return type `list`

getTransactions (*input_filter=None*)

Fetches all transactions linked with connection.

Parameters **None** –

Returns Returns a list of Transaction objects linked with connection.

Return type `list`

refresh ()

Refreshes specific connection, ensuring most up-to-date information.

Parameters `connectionID` (*str*) – ID of connection to be refreshed.

Returns Returns Job object to monitor progress

Return type *Job* (obj)

`to_dict()`

Convert object details to dict.

Hidden method for converting object to pandas dataframe.

Parameters `None` –

Returns Returns dict of object.

Return type *dict*

`class main.Account` (*account_json, header*)

`getTransactions` (*input_filter=None*)

Fetches transactions linked with account.

Parameters `None` –

Returns Returns list of Transaction objects linked with account.

Return type *list*

`to_dict()`

Convert object details to dict.

Hidden method for converting object to pandas dataframe.

Parameters `None` –

Returns Returns dict of object.

Return type *dict*

`class main.Transaction` (*transact_json, header*)

`to_dict()`

Convert object details to dict.

Hidden method for converting object to pandas dataframe.

Parameters `None` –

Returns Returns dict of object.

Return type *dict*

`class Job.Job` (*job_json, header*)

`getStatus()`

Gets status of job

Parameters `None` –

Returns Returns dict of job status

Return type *dict*

`isComplete()`

Checks if job is fully complete.

Parameters **None** –

Returns Returns if job is complete.

Return type `bool`

refresh()

Refreshes job status

Parameters **None** –

Returns Indicate success.

Return type `True`

class `Income.Income` (*income_json, header*)

class `Expense.Expense` (*expense_json, header*)

A

Account (*class in main*), 19
addConnection() (*main.User method*), 16

C

Connection (*class in main*), 18
createUser() (*main.Session method*), 15

D

deleteUser() (*main.Session method*), 16

E

Expense (*class in Expense*), 20

G

getAccounts() (*main.Connection method*), 18
getAccounts() (*main.User method*), 17
getConnections() (*main.User method*), 17
getExpenses() (*main.User method*), 17
getIncome() (*main.User method*), 17
getInstitutions() (*main.Session method*), 16
getInstitutionsJSON() (*main.Session method*),
16
getStatus() (*Job.Job method*), 19
getTransaction() (*main.User method*), 17
getTransactions() (*main.Account method*), 19
getTransactions() (*main.Connection method*), 18
getTransactions() (*main.User method*), 17
getUser() (*main.Session method*), 16

I

Income (*class in Income*), 20
isComplete() (*Job.Job method*), 19

J

Job (*class in Job*), 19

R

refresh() (*Job.Job method*), 20

refresh() (*main.Connection method*), 18
refreshAllConnections() (*main.User method*),
17
refreshConnection() (*main.User method*), 18

S

searchInstitutions() (*main.Session method*), 16
Session (*class in main*), 15

T

to_dict() (*main.Account method*), 19
to_dict() (*main.Connection method*), 19
to_dict() (*main.Transaction method*), 19
to_dict() (*main.User method*), 18
Transaction (*class in main*), 19

U

update() (*main.User method*), 18
User (*class in main*), 16