ticketpy Documentation

Release 1.1.1

Edward Wells

Apr 28, 2018

Contents:

1 ticketpy **1**
- 1.1 Requirements . . . 1
- 1.2 Installation . . . 1
- 1.3 Example searches . . . 1
  - 1.3.1 Events . . . 1
  - 1.3.2 Venues . . . 3
  - 1.3.3 Attractions . . . 3
  - 1.3.4 Classifications . . . 4

2 ticketpy package **7**
- 2.1 ticketpy.client module . . . 7
- 2.2 ticketpy.query module . . . 7
- 2.3 ticketpy.model module . . . 7

3 Indices and tables **9**

CHAPTER 1

ticketpy

Python wrapper/SDK for Ticketmaster's Discovery API

More info: http://developer.ticketmaster.com/products-and-docs/apis/discovery-api/v2/

1.1 Requirements

- Python >= 3.5.2 (anything >= 3 is probably OK)
- Requests >= 2.13.0

1.2 Installation

To install via *pip*:

```
$ pip install ticketpy
```

Or, locally from the same directory as `setup.py`:

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

1.3 Example searches

1.3.1 Events

To pull all Hip-Hop events in Georgia between May 19th, 2017 and May 21st, 2017:

```
import ticketpy

tm_client = ticketpy.ApiClient('your_api_key')

pages = tm_client.events.find(
    classification_name='Hip-Hop',
    state_code='GA',
    start_date_time='2017-05-19T20:00:00Z',
    end_date_time='2017-05-21T20:00:00Z'
)

for page in pages:
    for event in page:
        print(event)
```

Output:

```
Event:        Atlanta Funk Fest 2017 3 Day Ticket
Venue(s):     'Wolf Creek Amphitheater' at 3025 Merk Road in Atlanta GA
Start date:   2017-05-19
Start time:   19:00:00
Price ranges: 128.01-424.0
Status:       onsale
Genres:       R&B

Event:        Atlanta Funk Fest 2017
Venue(s):     'Wolf Creek Amphitheater' at 3025 Merk Road in Atlanta GA
Start date:   2017-05-19
Start time:   19:00:00
Price ranges: 63.0-158.0
Status:       onsale
Genres:       R&B

Event:        Atlanta Funk Fest 2017
Venue(s):     'Wolf Creek Amphitheater' at 3025 Merk Road in Atlanta GA
Start date:   2017-05-20
Start time:   17:00:00
Price ranges: 63.0-158.0
Status:       onsale
Genres:       Hip-Hop/Rap

Event:        NF
Venue(s):     'Center Stage Theater' at 1374 W Peachtree St. NW in Atlanta GA
Start date:   2017-05-20
Start time:   20:00:00
Price ranges: 22.0-83.0
Status:       onsale
Genres:       Hip-Hop/Rap
```

Calling `ApiClient.find()` returns a `ticketpy.PagedResponse` object, which iterates through API response pages (as `ticketpy.Page`).

By default, pages have 20 elements. If there are >20 total elements, calling `PagedResponse.next()` will request the next page from the API.

You can simplify that/do away with the nested loop by using `PagedResponse.limit()`. By default, this requests a maximum of 5 pages, and returns the elements of each in a flat list.

Use `PagedResponse.one()` to return just the list from the first page.

For example, the previous example could also be written as:

```
import ticketpy

tm_client = ticketpy.ApiClient('your_api_key')

pages = tm_client.events.find(
    classification_name='Hip-Hop',
    state_code='GA',
    start_date_time='2017-05-19T20:00:00Z',
    end_date_time='2017-05-21T20:00:00Z'
).limit()

for event in pages:
    print(event)
```

The output here would be the same as there was <1 page available, however, this can save you some wasted API calls for large result sets. If you really want *every page*, though, use `all()` to request every available page.

1.3.2 Venues

To search for all venues based on the string "*Tabernacle*":

```
import ticketpy

tm_client = ticketpy.ApiClient("your_api_key")
venues = tm_client.venues.find(keyword="Tabernacle").all()
for v in venues:
    print("Name: {} / City: {}".format(v.name, v.city))
```

Output:

```
Name: Tabernacle / City: London
Name: The Tabernacle / City: Atlanta
Name: Tabernacle, Notting Hill / City: London
Name: Bethel Tabernacle / City: Penticton
Name: Revivaltime Tabernacle / City: Toronto
Name: Auckland Baptist Tabernacle / City: Auckland
Name: Pentecostal Tabernacle / City: Nashville
Name: The Tabernacle / City: Oak Bluffs
Name: Tabernacle, Shoreditch / City: London
Name: Revivaltime Tabernacle / City: Toronto
Name: Tabernacle, Notting Hill / City: London
Name: The Tabernacle / City: London
Name: Tabernacle Junction / City: Yeovil
Name: New Tabernacle 4th Baptist Church / City: Charleston
```

1.3.3 Attractions

Searching for attractions works similarly to the above:

```
import ticketpy

tm_client = ticketpy.ApiClient("your_api_key")
attractions = tm_client.attractions.find(keyword="Yankees").one()
```

(continues on next page)

(continued from previous page)

```
for attr in attractions:
    print(attr.name)
```

Output:

```
New York Yankees
Scranton Wilkes-Barre RailRiders
Staten Island Yankees
Yankee Stadium Tours
Tampa Yankees
New York Yankees  Bomber Bucks
Hands On History At Yankee Stadium
Damn Yankees
Damn Yankees
Battle Creek Yankees
New York Yankees Parking
Offsite Parking at Yankee Stadium
Quikpark at Yankee Stadium- NYCFC
New York Yankees Fan Fest
New York Yankees 3 (Do Not Use)
New York Yankees 1 (Do Not Use)
New York Yankees 2 (Do Not Use)
Behind the Scenes At Yankee Stadium
```

1.3.4 Classifications

Searching for classifications works similarly to the above:

```
import ticketpy

tm_client = ticketpy.ApiClient("your_api_key")
classifications = tm_client.classifications.find(keyword="Drama").one()

for cl in classifications:
    print("Segment: {}".format(cl.segment.name))
    for genre in cl.segment.genres:
        print("--Genre: {}".format(genre.name))
```

Output:

```
Segment: Film
--Genre: Drama
Segment: Arts & Theatre
--Genre: Theatre
```

Querying details for classifications by ID will return either a `Segment`, `Genre`, or `SubGenre`, whichever matches the given ID.

For example,

```
import ticketpy

tm_client = ticketpy.ApiClient("your_api_key")
x = tm_client.classifications.by_id('KZFzniwnSyZfZ7v7nJ')
y = tm_client.classifications.by_id('KnvZfZ7vAvE')
```

(continues on next page)

(continued from previous page)

```
z = tm_client.classifications.by_id('KZazBEonSMnZfZ7vkdl')

s = "Name: {} / Type: {}"
print(s.format(x.name, type(x)))
print(s.format(y.name, type(y)))
print(s.format(z.name, type(z)))
```

Output:

```
Name: Music / Type: <class 'ticketpy.model.Segment'>
Name: Jazz / Type: <class 'ticketpy.model.Genre'>
Name: Bebop / Type: <class 'ticketpy.model.SubGenre'>
```

CHAPTER 2

ticketpy package

2.1 ticketpy.client module

2.2 ticketpy.query module

2.3 ticketpy.model module

CHAPTER 3

Indices and tables

- genindex
- modindex
- search