
python-launch-library Documentation

Release 0.3

Plutoberth

Dec 10, 2020

Contents

1	Getting Started	3
1.1	Installation	3
1.2	Usage	3
1.3	Asynchronous Usage	4
2	Module Documentation	5
2.1	Api	5
2.2	Models	6
2.3	Asynchronous Models	8
2.4	Exceptions	9
3	Apache 2 License	11
4	About Launch Library	15
	Python Module Index	17
	Index	19

Welcome to python-launch-library's documentation! This is a fairly simple library built to easily interact with the launchlibrary.net API.

CHAPTER 1

Getting Started

1.1 Installation

The python-launch-library module only supports Python 3.6 and above. To install it, just use:

```
pip install python-launch-library
```

1.2 Usage

Usage of the wrapper is simple.

```
# First, import the library
import launchlibrary as ll

# Then, initialize an API object
api = ll.Api()

# And fetch whichever models you'd like.

# Either by an explicit API call
next_5_go_launches = api.fetch_launch(next=5, status=1)

# Or by one of the simpler methods available for some models.
next_5_go_launches = api.next_launches(5)

# Now, you can utilize whatever data you'd like. Data from the API is processed
↳recursively, so if a Launch object
# contains a Location object, you'll have models for both.
launch_loc = next_5_go_launches[0].location

# Some properties, like status, are only represented by ID. You can call the
↳appropriate methods to get a proper object from that ID
```

(continues on next page)

(continued from previous page)

```
launch_status = next_5_go_launches[0].get_status()

# It's now possible to also use the regular API names as well as pythonic names.
vid_urls = next_5_go_launches[0].vid_urls
vid_urls_2 = next_5_go_launches[0].vidURLs
```

1.3 Asynchronous Usage

```
import launchlibrary as ll

api = ll.Api()

async def foo():
    # Use the api as usual, but with an async prefix for api functions.
    next_5_go_launches = await api.async_next_launches(5)
    status = await next_5_go_launches[0].get_status()
```


2.1 Api

The Api class exposes the main interface for this library. A fetch method is provided for all models.

The fetch method will get the data, turn it into python objects recursively, and do a few other nice things like adding python date-time objects.

```
class launchlibrary.Api (api_url='https://ll.thespacedevs.com', version='2.0.0', unicode=True)
```

```
    __init__ (api_url='https://ll.thespacedevs.com', version='2.0.0', unicode=True)
```

The API class for the launchlibrary module.

Parameters

- **api_url** (*str*) – The URL of the launchlibrary website.
- **version** (*str*) – Version of the api
- **unicode** (*bool*) – Set to False to convert unicode characters to ASCII using `unicodecode`.

```
fetch_agency (**kwargs)
```

Fetch from the Agency endpoint

```
fetch_launch (**kwargs)
```

Fetch from the Launch endpoint

```
next_launches (num)
```

Get the next {num} launches.

Parameters **num** (*int*) – a number for the number of launches

Return type `List[UpcomingLaunch]`

```
fetch_pad (**kwargs)
```

Fetch from the Pad endpoint

fetch_location (***kwargs*)
Fetch from the Location endpoint

fetch_rocket (***kwargs*)
Fetch from the Rocket endpoint

async_fetch_agency (***kwargs*)
Fetch from the Agency endpoint

async_fetch_launch (***kwargs*)
Fetch from the Launch endpoint

async_fetch_location (***kwargs*)
Fetch from the Location endpoint

async_fetch_pad (***kwargs*)
Fetch from the Pad endpoint

async_fetch_rocket (***kwargs*)
Fetch from the Rocket endpoint

async_next_launches (*num*)
Get the next {num} launches.

Parameters *num* (int) – a number for the number of launches

Return type `List[AsyncUpcomingLaunch]`

2.2 Models

As this library is based on the **launchlibrary** API, you can find a lot of info on [their website](#) . Note that the wrapper only uses the *detailed* mode.

Additionally, the parameters of every model can be accessed post-creation by using *model.param_names*.

class `launchlibrary.models.BaseModel` (*network, param_translations, proper_name*)
Bases: `object`

The base model class all models should inherit from. Provides fetch and other utility functionalities.

_endpoint_name The endpoint to use in the api

_nested_name The name of that will appear in nested results. “Agencies” and the such.

Operation	Description
<code>x == y</code>	Checks if both objects are of the same type and have the same id.

classmethod **fetch** (*network, **kwargs*)
Initializes a class, or even a list of them from the api using the needed params.

Parameters

- **network** (`Network`) – An instance of the network class
- **kwargs** – Arguments to include in the GET request

Return type `list`

classmethod **init_from_json** (*network, json_object*)
Initializes a class from a json object. Only single classes.

Parameters

- **network** (`Network`) – `launchlibrary.Network`
- **json_object** (`dict`) – An object containing the “entry” we want to init.

Returns `cls`

class `launchlibrary.models.Agency` (*network*)

Bases: `launchlibrary.models.BaseModel`

A class representing an agency object.

class `launchlibrary.models.Launch` (*network*)

Bases: `launchlibrary.models.BaseModel`

A class representing a launch object.

You may use the ‘**windowstart**’, ‘**windowend**’, and ‘**net**’ params to access datetime objects of the times. They’ll be ‘None’ if the conversion fails.

The comparison magic methods that are implemented essentially compare the dates of the two objects.

Operation	Description
<code>x < y</code>	Checks if launch y occurs before launch x.
<code>x > y</code>	Checks if launch x occurs before launch y.

class `launchlibrary.models.UpcomingLaunch` (*network*)

Bases: `launchlibrary.models.Launch`

classmethod `next` (*network, num*)

A simple abstraction method to get the next {num} launches.

Parameters

- **network** (`Network`) – An instance of `launchlibrary.Api`
- **num** (`int`) – a number for the number of launches

Return type `List[UpcomingLaunch]`

class `launchlibrary.models.Pad` (*network*)

Bases: `launchlibrary.models.BaseModel`

A class representing a pad object.

class `launchlibrary.models.Location` (*network*)

Bases: `launchlibrary.models.BaseModel`

A class representing a location object.

class `launchlibrary.models.Rocket` (*network*)

Bases: `launchlibrary.models.BaseModel`

A class representing a rocket object.

get_pads ()

Returns Pad type objects of the pads the rocket uses.

Return type `List[Pad]`

2.3 Asynchronous Models

The library also supports asynchronous operation. To receive proper coroutines, just prepend Async to the name of the class.

class `launchlibrary.async_models.BaseAsync` (*network*, *param_translations*, *proper_name*)

Bases: `launchlibrary.models.BaseModel`

classmethod `fetch` (*network*, ***kwargs*)

The fetch method implements fetch with an async HTTP GET function.

Parameters

- **network** (`Network`) – A network instance
- **kwargs** – args for the api call

Returns objects based on BaseAsync

class `launchlibrary.async_models.AsyncAgency` (*network*)

Bases: `launchlibrary.models.Agency`, `launchlibrary.async_models.BaseAsync`

A class representing an async agency object.

class `launchlibrary.async_models.AsyncLaunch` (*network*)

Bases: `launchlibrary.models.Launch`, `launchlibrary.async_models.BaseAsync`

A class representing an async launch object.

class `launchlibrary.async_models.AsyncUpcomingLaunch` (*network*)

Bases: `launchlibrary.models.UpcomingLaunch`, `launchlibrary.async_models.BaseAsync`

A class representing an upcoming launch object.

classmethod `next` (*network*, *num*)

Get the next {num} launches.

Parameters

- **network** (`Network`) – A network instance
- **num** (`int`) – a number for the number of launches

class `launchlibrary.async_models.AsyncPad` (*network*)

Bases: `launchlibrary.models.Pad`, `launchlibrary.async_models.BaseAsync`

A class representing an async pad object.

class `launchlibrary.async_models.AsyncLocation` (*network*)

Bases: `launchlibrary.models.Location`, `launchlibrary.async_models.BaseAsync`

A class representing an async Location object.

class `launchlibrary.async_models.AsyncRocket` (*network*)

Bases: `launchlibrary.models.Rocket`, `launchlibrary.async_models.BaseAsync`

A class representing an async rocket.

get_pads ()

Returns Pad type objects of the pads the rocket uses.

Return type `List[AsyncPad]`

2.4 Exceptions

The library attempts not to leak any exceptions except the regular Python ones, like `ValueError` and `KeyError`.

exception `launchlibrary.exceptions.LlException` (*message='There was an unspecified exception regarding the LaunchLibrary wrapper'*)

Bases: `Exception`

The base class for all exceptions emitted by the library

exception `launchlibrary.exceptions.ApiException` (*message='There was an unknown issue with the API. Please reevaluate your call.'*)

Bases: `launchlibrary.exceptions.LlException`

An exception related to the API's response

exception `launchlibrary.exceptions.NetworkException` (*message*)

Bases: `launchlibrary.exceptions.LlException`

Some network failure that's unrelated to the request, like a dropped connection

exception `launchlibrary.exceptions.TimeoutException` (*message=""*)

Bases: `launchlibrary.exceptions.LlException`

All timeout failures, both during the initial connection and subsequent messages

Apache 2 License

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

“License” shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

“Licensor” shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

“Legal Entity” shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, “control” means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

“You” (or “Your”) shall mean an individual or Legal Entity exercising permissions granted by this License.

“Source” form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

“Object” form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

“Work” shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

“Derivative Works” shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other

modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

“Contribution” shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, “submitted” means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as “Not a Contribution.”

“Contributor” shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. **Grant of Copyright License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. **Grant of Patent License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. **Redistribution.** You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a “NOTICE” text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

First, visit the getting started page, and then feel free to proceed to the Modules page to learn about the rest of the wrapper.

CHAPTER 4

About Launch Library

Launch Library is a free to use website that exposes a rich API with a plethora of data about future (and past) launches. Check them out [here!](#)

I

`launchlibrary.async_models`, 8
`launchlibrary.exceptions`, 9
`launchlibrary.models`, 6

Symbols

`__init__()` (*launchlibrary.Api* method), 5

A

Agency (class in *launchlibrary.models*), 7

Api (class in *launchlibrary*), 5

ApiException, 9

`async_fetch_agency()` (*launchlibrary.Api* method), 6

`async_fetch_launch()` (*launchlibrary.Api* method), 6

`async_fetch_location()` (*launchlibrary.Api* method), 6

`async_fetch_pad()` (*launchlibrary.Api* method), 6

`async_fetch_rocket()` (*launchlibrary.Api* method), 6

`async_next_launches()` (*launchlibrary.Api* method), 6

AsyncAgency (class in *launchlibrary.async_models*), 8

AsyncLaunch (class in *launchlibrary.async_models*), 8

AsyncLocation (class in *launchlibrary.async_models*), 8

AsyncPad (class in *launchlibrary.async_models*), 8

AsyncRocket (class in *launchlibrary.async_models*), 8

AsyncUpcomingLaunch (class in *launchlibrary.async_models*), 8

B

BaseAsync (class in *launchlibrary.async_models*), 8

BaseModel (class in *launchlibrary.models*), 6

F

`fetch()` (*launchlibrary.async_models.BaseAsync* class method), 8

`fetch()` (*launchlibrary.models.BaseModel* class method), 6

`fetch_agency()` (*launchlibrary.Api* method), 5

`fetch_launch()` (*launchlibrary.Api* method), 5

`fetch_location()` (*launchlibrary.Api* method), 5

`fetch_pad()` (*launchlibrary.Api* method), 5

`fetch_rocket()` (*launchlibrary.Api* method), 6

G

`get_pads()` (*launchlibrary.async_models.AsyncRocket* method), 8

`get_pads()` (*launchlibrary.models.Rocket* method), 7

I

`init_from_json()` (*launchlibrary.models.BaseModel* class method), 6

L

Launch (class in *launchlibrary.models*), 7

launchlibrary.async_models (module), 8

launchlibrary.exceptions (module), 9

launchlibrary.models (module), 6

LlException, 9

Location (class in *launchlibrary.models*), 7

N

NetworkException, 9

`next()` (*launchlibrary.async_models.AsyncUpcomingLaunch* class method), 8

`next()` (*launchlibrary.models.UpcomingLaunch* class method), 7

`next_launches()` (*launchlibrary.Api* method), 5

P

Pad (class in *launchlibrary.models*), 7

R

Rocket (class in *launchlibrary.models*), 7

T

TimeoutException, 9

U

UpcomingLaunch (*class in launchlibrary.models*), 7