
annict Documentation

Release 0.7.0

Author

Apr 07, 2021

Contents

1	User Guide	3
1.1	User Guide	3
1.1.1	Installation	3
1.1.2	Quickstart	3
2	API Reference	5
2.1	annict package	5
2.1.1	Submodules	5
2.1.2	annict.api module	5
2.1.3	annict.auth module	11
2.1.4	annict.cursors module	12
2.1.5	annict.models module	12
2.1.6	annict.parsers module	16
2.1.7	annict.services module	16
2.1.8	annict.utils module	17
2.1.9	Module contents	17
3	Indices and tables	19
	Python Module Index	21
	Index	23

Annict API wrapper for Python.

```
>>> from annict.api import API
>>> annict = API('your-access-token')
>>> results = annict.works(filter_title="Re:")
>>> print(results[0].title)
Re:
```

annict officially supports Python 3.6 or higher.

1.1 User Guide

1.1.1 Installation

From PyPI with the Python package manager:

```
pip install annict
```

1.1.2 Quickstart

Authentication

Acquire the URL for acquiring the authentication code.

```
>>> from annict.auth import OAuthHandler
>>> handler = OAuthHandler(client_id='your-client-id', client_secret='your-client-
↳secret')
>>> url = handler.get_authorization_url(scope='read write')
>>> print(url)
```

Open the browser and access the URL you obtained, the authentication code will be displayed. It will be passed to the `handler.authenticate()` 's argument to get the access token.

```
>>> handler.authenticate(code='your-authentication-code')
>>> print(handler.get_access_token())
```

Note that this authentication flow is unnecessary when issuing a personal access token on Annict and using it.

API

```
>>> from annict.api import API
>>> annict = API('your-access-token')
>>> results = annict.works(filter_title="Re:")
>>> print(results[0].title)
Re:
```

The API class provides access to the entire Annict RESTful API methods. Each method can accept various parameters and return responses. For more information about these methods please refer to API Reference.

Models

When we invoke an API method most of the time returned back to us will be a Annict model class instance. This will contain the data returned from Annict which we can then use inside our application. For example the following code returns to us an User model:

```
>>> user = api.me()
```

Models contain the data and some helper methods which we can then use:

```
>>> print(user.name)
>>> print(user.records_count)
>>> for follower in user.followers():
...     print(follower.username)
```

Cursors

```
>>> from annict.cursors import SimpleCursor
>>> for work in SimpleCursor(api.works, per_page=50, sort_id='desc').cursor():
...     print(work)
...
<Work:6417:>
<Work:6416:>
<Work:6415:>
```


2.1 annict package

2.1.1 Submodules

2.1.2 annict.api module

class `annict.api.API` (*token*, *base_url*='https://api.annict.com', *api_version*='v1', *parser*=<class 'annict.parsers.ModelParser'>)

Bases: `object`

API wrapper for Annict.

Basic Usage:

```
>>> from annict.api import API
>>> api = API('your-access-token')
>>> api.me()
<User:1229::@kk6>
```

activities (*fields*=None, *filter_user_id*=None, *filter_username*=None, *page*=None, *per_page*=None, *sort_id*=None)

Get activities

Reference <https://docs.annict.com/ja/api/v1/activities.html>

Parameters

- **fields** (*list of str*) – (optional) Narrow down the fields of data contained in the response body.
- **filter_user_id** (*int*) – (optional) Filter results by User's ID.
- **filter_username** (*str*) – (optional) Filter results by username.
- **page** (*int*) – (optional) Specify the number of pages.

- **per_page** (*int*) – (optional) Specify how many items to acquire per page.
- **sort_id** (*str*) – (optional) Sort the results by their ID. You can specify *asc* or *desc*.

Returns list of *Activity* objects.

Return type *annict.models.ResultSet*

create_record (*episode_id*, *comment=None*, *rating=None*, *share_twitter=False*,
share_facebook=False)

Create a record to the episode.

Reference <https://docs.annict.com/ja/api/v1/me-records.html>

Parameters

- **episode_id** (*int*) – Episode's ID
- **comment** (*str*) – (optional) Comment.
- **rating** (*float*) – (optional) Rating.
- **share_twitter** (*bool*) – (optional) Whether to share the record on Twitter. You can enter *True* or *False*.
- **share_facebook** (*bool*) – (optional) Whether to share the record on Facebook. You can enter *True* or *False*.

Returns *Record* object.

delete_record (*id_*)

Delete the created record.

Reference <https://docs.annict.com/ja/api/v1/me-records.html>

Parameters **id** (*int*) – Record's ID

Returns Returns *True* if deletion succeeded.

edit_record (*id_*, *comment=None*, *rating=None*, *share_twitter=False*, *share_facebook=False*)

Edit the created record.

Reference <https://docs.annict.com/ja/api/v1/me-records.html>

Parameters

- **id** (*int*) – Record's ID.
- **comment** (*str*) – (optional) Comment.
- **rating** (*float*) – (optional) Rating.
- **share_twitter** (*bool*) – (optional) Whether to share the record on Twitter. You can enter *True* or *False*.
- **share_facebook** (*bool*) – (optional) Whether to share the record on Facebook. You can enter *True* or *False*.

Returns *Record* object after update.

episodes (*fields=None*, *filter_ids=None*, *filter_work_id=None*, *page=None*, *per_page=None*,
sort_id=None, *sort_sort_number=None*)

Get episodes information

Reference <https://docs.annict.com/ja/api/v1/episodes.html>

Parameters

- **fields** (*list of str*) – (optional) Narrow down the fields of data contained in the response body.
- **filter_ids** (*list of int*) – (optional) Filter results by IDs.
- **filter_work_id** (*int*) – (optional) Filter results by Work’s ID.
- **page** (*int*) – (optional) Specify the number of pages.
- **per_page** (*int*) – (optional) Specify how many items to acquire per page.
- **sort_id** (*str*) – (optional) Sort the results by their ID. You can specify *asc* or *desc*.
- **sort_sort_number** (*str*) – (optional) Sort by number for sorting. You can specify *asc* or *desc*.

Returns list of *Episode* objects.

Return type *annict.models.ResultSet*

followers (*fields=None, filter_user_id=None, filter_username=None, page=None, per_page=None, sort_id=None*)
Get followers information

Reference <https://docs.annict.com/ja/api/v1/followers.html>

Parameters

- **fields** (*list of str*) – (optional) Narrow down the fields of data contained in the response body.
- **filter_user_id** (*int*) – (optional) Filter results by User’s ID.
- **filter_username** (*str*) – (optional) Filter results by username.
- **page** (*int*) – (optional) Specify the number of pages.
- **per_page** (*int*) – (optional) Specify how many items to acquire per page.
- **sort_id** (*str*) – (optional) Sort the results by their ID. You can specify *asc* or *desc*.

Returns list of *User* objects.

Return type *annict.models.ResultSet*

following (*fields=None, filter_user_id=None, filter_username=None, page=None, per_page=None, sort_id=None*)
Get following information

Reference <https://docs.annict.com/ja/api/v1/following.html>

Parameters

- **fields** (*list of str*) – (optional) Narrow down the fields of data contained in the response body.
- **filter_user_id** (*int*) – (optional) Filter results by User’s ID.
- **filter_username** (*str*) – (optional) Filter results by username.
- **page** (*int*) – (optional) Specify the number of pages.
- **per_page** (*int*) – (optional) Specify how many items to acquire per page.
- **sort_id** (*str*) – (optional) Sort the results by their ID. You can specify *asc* or *desc*.

Returns list of *User* objects.

Return type *annict.models.ResultSet*

following_activities (*fields=None, filter_actions=None, filter_muted=None, page=None, per_page=None, sort_id=None*)

Get the activity of the user you are following.

Reference <https://docs.annict.com/ja/api/v1/me-following-activities.html>

Parameters

- **fields** (*list of str*) – (optional) Narrow down the fields of data contained in the response body.
- **filter_actions** (*str*) – (optional) Filter results by action (create_record|create_multiple_records|create_status).
- **filter_muted** (*bool*) – (optional) Specify whether to exclude muted users with the mute function. You can exclude with *True* and not exclude with *False*. The default is *True* (exclude).
- **page** (*int*) – (optional) Specify the number of pages.
- **per_page** (*int*) – (optional) Specify how many items to acquire per page.
- **sort_id** (*str*) – (optional) Sort the results by their ID. You can specify *asc* or *desc*.

Returns list of *Activity* objects.

Return type *annict.models.ResultSet*

me (*fields=None*)

Get your profile information

Reference <https://docs.annict.com/ja/api/v1/me.html>

Parameters **fields** (*list of str*) – (optional) Narrow down the fields of data contained in the response body.

Returns *User* object of your user information.

my_programs (*fields=None, filter_ids=None, filter_channel_ids=None, filter_work_ids=None, filter_started_at_gt=None, filter_started_at_lt=None, filter_unwatched=None, filter_rebroadcast=None, page=None, per_page=None, sort_id=None, sort_started_at=None*)

Get the broadcast schedule.

Reference <https://docs.annict.com/ja/api/v1/me-programs.html>

Parameters

- **fields** (*list of str*) – (optional) Narrow down the fields of data contained in the response body.
- **filter_ids** (*list of int*) – (optional) Filter results by IDs.
- **filter_channel_ids** (*list of int*) – (optional) Filter results by Channel IDs.
- **filter_work_ids** (*list of int*) – (optional) Filter results by Work IDs.
- **filter_started_at_gt** (*datetime*) – (optional) Filter results results to those with the broadcast start date and time after the specified date and time.
- **filter_started_at_lt** (*datetime*) – (optional) Filter results results to those with the broadcast start date and time before the specified date and time.
- **filter_unwatched** (*bool*) – (optional) Only get unwatched broadcast schedules.

- **filter_rebroadcast** (*bool*) – (optional) Filter the broadcast schedule based on the rebroadcast flag. If you pass *True*, only rebroadcasting, passing *False* will get broadcast schedules other than rebroadcast.
- **page** (*int*) – (optional) Specify the number of pages.
- **per_page** (*int*) – (optional) Specify how many items to acquire per page.
- **sort_id** (*str*) – (optional) Sort the results by their ID. You can specify *asc* or *desc*.
- **sort_started_at** (*str*) – (optional) Sort the results by started_at.

Returns list of *Program* objects.

Return type *annict.models.ResultSet*

my_works (*fields=None, filter_ids=None, filter_season=None, filter_title=None, filter_status=None, page=None, per_page=None, sort_id=None, sort_season=None, sort_watchers_count=None*)

Get the information of the work you are setting status.

Reference <https://docs.annict.com/ja/api/v1/me-works.html>

Parameters

- **fields** (*list of str*) – (optional) Narrow down the fields of data contained in the response body.
- **filter_ids** (*list of int*) – (optional) Filter results by IDs.
- **filter_season** (*str*) – (optional) Filter results by release time of season.
- **filter_title** (*str*) – (optional) Filter results by title.
- **filter_status** (*str*) – (optional) Filter results by status. You can specify *wanna_watch*, *watching*, *watched*, *on_hold*, *stop_watching*.
- **page** (*int*) – (optional) Specify the number of pages.
- **per_page** (*int*) – (optional) Specify how many items to acquire per page.
- **sort_id** (*str*) – (optional) Sort the results by their ID. You can specify *asc* or *desc*.
- **sort_season** (*str*) – (optional) Sort the results by their release time of season. You can specify *asc* or *desc*.
- **sort_watchers_count** (*str*) – (optional) Sort the results by their watchers count. You can specify *asc* or *desc*.

Returns list of *Work* objects.

Return type *annict.models.ResultSet*

people (*fields=None, filter_ids=None, filter_name=None, page=None, per_page=None, sort_id=None*)

Parameters

- **fields** (*list of str*) – (optional) Narrow down the fields of data contained in the response body.
- **filter_ids** (*list of int*) – (optional) Filter results by IDs.
- **filter_name** (*str*) – Filter results by name.
- **page** (*int*) – (optional) Specify the number of pages.
- **per_page** (*int*) – (optional) Specify how many items to acquire per page.

- **sort_id** (*str*) – (optional) Sort the results by their ID. You can specify *asc* or *desc*.

Returns

records (*fields=None, filter_ids=None, filter_episode_id=None, filter_has_record_comment=None, page=None, per_page=None, sort_id=None, sort_likes_count=None*)
Get records to episodes

Reference <https://docs.annict.com/ja/api/v1/records.html>

Parameters

- **fields** (*list of str*) – (optional) Narrow down the fields of data contained in the response body.
- **filter_ids** (*list of int*) – (optional) Filter results by IDs.
- **filter_episode_id** (*int*) – (optional) Filter results by Episode's ID.
- **filter_has_record_comment** (*bool*) – (optional) Filter the results by the presence or absence of comments. If you specify *True*, only records with comments will be filtered. Specifying *False* Filter records without comments.
- **page** (*int*) – (optional) Specify the number of pages.
- **per_page** (*int*) – (optional) Specify how many items to acquire per page.
- **sort_id** (*str*) – (optional) Sort the results by their ID. You can specify *asc* or *desc*.
- **sort_likes_count** (*str*) – (optional) Sort the results by their number of likes. You can specify *asc* or *desc*.

Returns list of *Record* objects.

Return type *annict.models.ResultSet*

search_users (*fields=None, filter_ids=None, filter_usernames=None, page=None, per_page=None, sort_id=None*)
Get users information

Reference <https://docs.annict.com/ja/api/v1/users.html>

Parameters

- **fields** (*list of str*) – (optional) Narrow down the fields of data contained in the response body.
- **filter_ids** (*list of int*) – (optional) Filter results by IDs.
- **filter_usernames** (*list of str*) – (optional) Filter results by usernames.
- **page** (*int*) – (optional) Specify the number of pages.
- **per_page** (*int*) – (optional) Specify how many items to acquire per page.
- **sort_id** (*str*) – (optional) Sort the results by their ID. You can specify *asc* or *desc*.

Returns list of *User* objects.

Return type *annict.models.ResultSet*

set_status (*work_id, kind*)
Set the status of the work.

Reference <https://docs.annict.com/ja/api/v1/me-statuses.html>

Parameters

- **work_id** (*int*) – Work's ID

- **kind** (*str*) – Types of status. You can specify *wanna_watch*, *watching*, *watched*, *on_hold*, *stop_watching*, or *no_select*.

Returns Returns *True* if deletion succeeded.

works (*fields=None, filter_ids=None, filter_season=None, filter_title=None, page=None, per_page=None, sort_id=None, sort_season=None, sort_watchers_count=None*)
Get works information

Reference <https://docs.annict.com/ja/api/v1/works.html>

Parameters

- **fields** (*list of str*) – (optional) Narrow down the fields of data contained in the response body.
- **filter_ids** (*list of int*) – (optional) Filter results by IDs.
- **filter_season** (*str*) – (optional) Filter results by release time of season.
- **filter_title** (*str*) – (optional) Filter results by title.
- **page** (*int*) – (optional) Specify the number of pages.
- **per_page** (*int*) – (optional) Specify how many items to acquire per page.
- **sort_id** (*str*) – (optional) Sort the results by their ID. You can specify *asc* or *desc*.
- **sort_season** (*str*) – (optional) Sort the results by their release time of season. You can specify *asc* or *desc*.
- **sort_watchers_count** (*str*) – (optional) Sort the results by their watchers count. You can specify *asc* or *desc*.

Returns list of *Work* objects.

Return type *annict.models.ResultSet*

2.1.3 annict.auth module

```
class annict.auth.OAuthHandler (client_id, client_secret, name='annict',
                                base_url='https://api.annict.com',
                                rect_uri='urn:ietf:wg:oauth:2.0:oob')
                                redi-
```

Bases: object

OAuth authentication handler

authenticate (*code, decoder=<function OAuthHandler.<lambda>>*)

Acquire the access token using the authorization code acquired after approval.

Parameters

- **code** – Authorization code obtained after approval.
- **decoder** – (optional) A function used to parse the Response content. Should return a dictionary.

get_access_token ()

Returns the access token when authenticated.

get_authorization_url (*scope='read'*)

Returns an authorization url

Parameters **scope** – (optional) Specify authority to access resources. Readonly defaults.

Returns URL of page requesting permission

Return type str

2.1.4 annict.cursors module

class annict.cursors.SimpleCursor (*method, **kwargs*)

Bases: object

Simple cursor class

cursor ()

annict.cursors.cursor_support (*api_method*)

Cursor support decorator

Parameters *api_method* – API method that wan to correspond to the cursor.

Returns wrapped method

2.1.5 annict.models module

class annict.models.Activity (*api=None*)

Bases: *annict.models.Model*

Activity information model

classmethod parse (*api, json*)

Parse a JSON object into a model instance.

Parameters

- **api** (*annict.api.API*) – instance of *API* .
- **json** (*dict*) – JSON from Annict API.

Returns *Activity* object

Return type *Activity*

class annict.models.Episode (*api=None*)

Bases: *annict.models.Model*

Episode information model

create_record (*comment=None, rating=None, share_twitter=False, share_facebook=False*)

Create a record for this episode.

Parameters

- **comment** (*str*) – (optional) Comment.
- **rating** (*float*) – (optional) Rating.
- **share_twitter** (*bool*) – (optional) Whether to share the record on Twitter. You can enter *True* or *False*.
- **share_facebook** (*bool*) – (optional) Whether to share the record on Facebook. You can enter *True* or *False*.

Returns *Record* object.

classmethod parse (*api, json*)

Parse a JSON object into a model instance.

Parameters

- **api** (`annict.api.API`) – instance of *API*.
- **json** (*dict*) – JSON from Annict API.

Returns *Episode* object

Return type *Episode*

class `annict.models.Model` (*api=None*)

Bases: `object`

Abstract class of each models.

classmethod `parse` (*api, json*)

Parse a JSON object into a model instance.

classmethod `parse_list` (*api, json, payload_type*)

Parse JSON objects into list of model instances.

Parameters

- **api** (`annict.api.API`) – instance of *API*.
- **json** (*dict*) – JSON from Annict API.
- **payload_type** (*str*) – Type of payload.

Returns list of Model objects.

Return type *ResultSet*

class `annict.models.Person` (*api=None*)

Bases: `annict.models.Model`

Person information model such as cast and staff

classmethod `parse` (*api, json*)

Parse a JSON object into a model instance.

Parameters

- **api** (`annict.api.API`) – instance of *API*.
- **json** (*dict*) – JSON from Annict API.

Returns *Person* object

Return type *Person*

class `annict.models.Program` (*api=None*)

Bases: `annict.models.Model`

Program information model

classmethod `parse` (*api, json*)

Parse a JSON object into a model instance.

Parameters

- **api** (`annict.api.API`) – instance of *API*.
- **json** (*dict*) – JSON from Annict API.

Returns *Program* object

Return type *Program*

class annict.models.**Record**(*api=None*)

Bases: [annict.models.Model](#)

Record information model

delete()

Delete the created record.

Returns Returns *True* if deletion succeeded.

edit(*comment=None, rating=None, share_twitter=False, share_facebook=False*)

Edit the created record.

Parameters

- **comment** (*str*) – (optional) Comment.
- **rating** (*float*) – (optional) Rating.
- **share_twitter** (*bool*) – (optional) Whether to share the record on Twitter. You can enter *True* or *False*.
- **share_facebook** (*bool*) – (optional) Whether to share the record on Facebook. You can enter *True* or *False*.

Returns [Record](#) object after edit.

classmethod **parse**(*api, json*)

Parse a JSON object into a model instance.

Parameters

- **api** ([annict.api.API](#)) – instance of [API](#).
- **json** (*dict*) – JSON from Annict API.

Returns [Record](#) object

Return type [Record](#)

class annict.models.**ResultSet**(*total_count, prev_page=None, next_page=None*)

Bases: [list](#)

A list like object that holds results from an Annict API query.

class annict.models.**User**(*api=None*)

Bases: [annict.models.Model](#)

User information model

followers(*fields=None, page=None, per_page=None, sort_id=None*)

Get following information of this user.

Parameters

- **fields** (*list of str*) – (optional) Narrow down the fields of data contained in the response body.
- **page** (*int*) – (optional) Specify the number of pages.
- **per_page** (*int*) – (optional) Specify how many items to acquire per page.
- **sort_id** (*str*) – (optional) Sort the results by their ID. You can specify *asc* or *desc*.

Returns list of [User](#) objects.

Return type [annict.models.ResultSet](#)

following (*fields=None, page=None, per_page=None, sort_id=None*)

Get following information of this user.

Parameters

- **fields** (*list of str*) – (optional) Narrow down the fields of data contained in the response body.
- **page** (*int*) – (optional) Specify the number of pages.
- **per_page** (*int*) – (optional) Specify how many items to acquire per page.
- **sort_id** (*str*) – (optional) Sort the results by their ID. You can specify *asc* or *desc*.

Returns list of *User* objects.

Return type *annict.models.ResultSet*

classmethod parse (*api, json*)

Parse a JSON object into a model instance.

Parameters

- **api** (*annict.api.API*) – instance of *API*.
- **json** (*dict*) – JSON from Annict API.

Returns *User* object

Return type *User*

class *annict.models.Work* (**args, **kwargs*)

Bases: *annict.models.Model*

Work information model

episodes (*fields=None, filter_ids=None, page=None, per_page=None, sort_id=None, sort_sort_number=None*)

Get episodes information

Reference <https://docs.annict.com/ja/api/v1/episodes.html>

Parameters

- **fields** (*list of str*) – (optional) Narrow down the fields of data contained in the response body.
- **filter_ids** (*list of int*) – (optional) Filter results by IDs.
- **page** (*int*) – (optional) Specify the number of pages.
- **per_page** (*int*) – (optional) Specify how many items to acquire per page.
- **sort_id** (*str*) – (optional) Sort the results by their ID. You can specify *asc* or *desc*.
- **sort_sort_number** (*str*) – (optional) Sort by number for sorting. You can specify *asc* or *desc*.

Returns list of *Episode* objects.

Return type *annict.models.ResultSet*

get_episode (*number*)

Get Episode object

Parameters **number** – Episode number

Returns *Episode* object

Return type *Episode*

classmethod `parse(api, json)`

Parse a JSON object into a model instance.

Parameters

- **api** (`annict.api.API`) – instance of *API*.
- **json** (*dict*) – JSON from Annict API.

Returns *Work* object

Return type *Work*

select_episodes (**numbers*)

Select multiple episodes

Parameters **numbers** – Episode number.

Returns list of *Episode*

Return type list of *Episode*

set_status (*kind*)

Set the status of the work.

Parameters **kind** (*str*) – Types of status. You can specify *wanna_watch*, *watching*, *watched*, *on_hold*, *stop_watching*, or *no_select*.

Returns Returns *True* if deletion succeeded.

2.1.6 annict.parsers module

class `annict.parsers.ModelParser(api, model_mapping=None)`

Bases: `object`

parse (*json*, *payload_type*, *payload_is_list=False*)

2.1.7 annict.services module

class `annict.services.APIMethod(api, path, method, allowed_params=None, payload_type=None, payload_is_list=False)`

Bases: `object`

A class abstracting each method of AnnictAPI

Parameters

- **api** (`annict.api.API`) – instance of *API*.
- **path** (*str*) – Endpoint path
- **method** (*str*) – HTTP Method
- **allowed_params** (*tuple*) – (optional) List of request parameter names that can be sent.
- **payload_type** (*str*) – Type of payload
- **payload_list** (*bool*) – Specifies whether the payload is a list or not.

build_parameters (*dic*)

Build a suitable parameters for request.

It filters the given dictionary based on *self.allowed_params* and returns a dictionary with an additional access token.

Parameters **dic** (*dict*) – dict of arguments given to annict.API's method.

Returns dict for request parameter

Return type dict

build_path (*id_=None*)

Build an suitable path

If *id_* is given, it is embedded into path.

Parameters **id** (*int*) – Target resource ID

build_url ()

Build request url

Returns request url

Return type str

2.1.8 annict.utils module

`annict.utils.do_not_stringify` (*arg*)

`annict.utils.stringify` (*arg*)

`annict.utils.stringify_boolean` (*arg*)

`annict.utils.stringify_list` (*arg*)

2.1.9 Module contents

python-annict

Annict API for Python.

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`

a

- `annict`, [17](#)
- `annict.api`, [5](#)
- `annict.auth`, [11](#)
- `annict.cursors`, [12](#)
- `annict.models`, [12](#)
- `annict.parsers`, [16](#)
- `annict.services`, [16](#)
- `annict.utils`, [17](#)

A

`activities()` (*annict.api.API method*), 5
`Activity` (*class in annict.models*), 12
`annict` (*module*), 17
`annict.api` (*module*), 5
`annict.auth` (*module*), 11
`annict.cursors` (*module*), 12
`annict.models` (*module*), 12
`annict.parsers` (*module*), 16
`annict.services` (*module*), 16
`annict.utils` (*module*), 17
`API` (*class in annict.api*), 5
`APIMethod` (*class in annict.services*), 16
`authenticate()` (*annict.auth.OAuthHandler method*), 11

B

`build_parameters()` (*annict.services.APIMethod method*), 16
`build_path()` (*annict.services.APIMethod method*), 17
`build_url()` (*annict.services.APIMethod method*), 17

C

`create_record()` (*annict.api.API method*), 6
`create_record()` (*annict.models.Episode method*), 12
`cursor()` (*annict.cursors.SimpleCursor method*), 12
`cursor_support()` (*in module annict.cursors*), 12

D

`delete()` (*annict.models.Record method*), 14
`delete_record()` (*annict.api.API method*), 6
`do_not_stringify()` (*in module annict.utils*), 17

E

`edit()` (*annict.models.Record method*), 14
`edit_record()` (*annict.api.API method*), 6
`Episode` (*class in annict.models*), 12

`episodes()` (*annict.api.API method*), 6
`episodes()` (*annict.models.Work method*), 15

F

`followers()` (*annict.api.API method*), 7
`followers()` (*annict.models.User method*), 14
`following()` (*annict.api.API method*), 7
`following()` (*annict.models.User method*), 14
`following_activities()` (*annict.api.API method*), 7

G

`get_access_token()` (*annict.auth.OAuthHandler method*), 11
`get_authorization_url()` (*annict.auth.OAuthHandler method*), 11
`get_episode()` (*annict.models.Work method*), 15

M

`me()` (*annict.api.API method*), 8
`Model` (*class in annict.models*), 13
`ModelParser` (*class in annict.parsers*), 16
`my_programs()` (*annict.api.API method*), 8
`my_works()` (*annict.api.API method*), 9

O

`OAuthHandler` (*class in annict.auth*), 11

P

`parse()` (*annict.models.Activity class method*), 12
`parse()` (*annict.models.Episode class method*), 12
`parse()` (*annict.models.Model class method*), 13
`parse()` (*annict.models.Person class method*), 13
`parse()` (*annict.models.Program class method*), 13
`parse()` (*annict.models.Record class method*), 14
`parse()` (*annict.models.User class method*), 15
`parse()` (*annict.models.Work class method*), 16
`parse()` (*annict.parsers.ModelParser method*), 16

`parse_list()` (*annict.models.Model* class method),
13
`people()` (*annict.api.API* method), 9
`Person` (class in *annict.models*), 13
`Program` (class in *annict.models*), 13

R

`Record` (class in *annict.models*), 13
`records()` (*annict.api.API* method), 10
`ResultSet` (class in *annict.models*), 14

S

`search_users()` (*annict.api.API* method), 10
`select_episodes()` (*annict.models.Work* method),
16
`set_status()` (*annict.api.API* method), 10
`set_status()` (*annict.models.Work* method), 16
`SimpleCursor` (class in *annict.cursors*), 12
`stringify()` (in module *annict.utils*), 17
`stringify_boolean()` (in module *annict.utils*), 17
`stringify_list()` (in module *annict.utils*), 17

U

`User` (class in *annict.models*), 14

W

`Work` (class in *annict.models*), 15
`works()` (*annict.api.API* method), 11