

---

# **JikanPy**

***Release 4.0.0***

**Apr 19, 2021**



---

## Contents

---

<b>1</b>	<b>Jikan</b>	<b>1</b>
<b>2</b>	<b>Aio,Jikan</b>	<b>9</b>
<b>3</b>	<b>Indices and tables</b>	<b>17</b>
	<b>Index</b>	<b>19</b>



```
class jikanpy.Jikan (selected_base: Optional[str] = None, session: Optional[requests.sessions.Session] = None)
```

Synchronous Jikan wrapper for the jikan.moe unofficial MyAnimeList API.

Note that the API has a limit of 30 requests/minute and 2 requests/second; this module does not make any effort to prevent abuse of that limit, so use it responsibly.

**Usage Example:**

```
from jikanpy import Jikan
jikan = Jikan()
```

**base**

The base URL of the Jikan API being accessed.

**session**

The Requests session.

```
__init__ (selected_base: Optional[str] = None, session: Optional[requests.sessions.Session] = None)  
    → None  
Constructs the Jikan object.
```

**Parameters**

- **selected\_base** (*str, optional*) – Base url of Jikan API. Defaults to the official Jikan API URL.
- **session** (*requests.Session, optional*) – to a new Requests session object.

**Returns** Instance of Jikan.

**Return type** *Jikan*

## Examples

```
>>> jikan_1 = Jikan()
>>> jikan_2 = Jikan(selected_base='http://localhost:8000/v3')
>>> jikan_3 = jikan = Jikan(session=requests.Session())
```

**anime** (*id: int, extension: Optional[str] = None, page: Optional[int] = None*) → Dict[str, Any]  
Gets information on an anime.

### Parameters

- **id** (int) – ID of the anime to get the information of.
- **extension** (str, optional) – Special information to get of the anime. Possible values are in the Jikan API documentation. Defaults to None.
- **page** (int, optional) – to None.

**Returns** Dictionary containing information about the anime.

**Return type** Dict

## Examples

```
>>> jikan.anime(14719)
>>> jikan.anime(14719, extension='episodes')
>>> jikan.anime(14719, extension='episodes', page=2)
>>> jikan.anime(14719, extension='news')
```

**character** (*id: int, extension: Optional[str] = None*) → Dict[str, Any]  
Gets information on a character.

### Parameters

- **id** (int) – ID of the character to get the information of.
- **extension** (str, optional) – Special information to get of the character. Possible values are in the Jikan API documentation. Defaults to None.

**Returns** Dictionary containing information about the character.

**Return type** Dict

## Examples

```
>>> jikan.character(6356)
```

**club** (*id: int, extension: Optional[str] = None, page: Optional[int] = None*) → Dict[str, Any]  
Gets information on a club.

### Parameters

- **id** (int) – ID of the club to get the information of.
- **extension** (str, optional) – Special information to get of the club. Possible values are in the Jikan API documentation. Defaults to None.
- **page** (int, optional) – to None.

**Returns** Dictionary containing information about the club.

**Return type** Dict

### Examples

```
>>> jikan.club(379)
```

**genre** (*type: str, genre\_id: int, page: Optional[int] = None*) → Dict[str, Any]

Gets anime or manga by genre.

#### Parameters

- **type** (str) – Type to get items from. Possible values are anime and manga.
- **genre\_id** (int) – Genre ID from MyAnimeList.
- **page** (int, optional) – Page number of the results. Defaults to None.

**Returns** Dictionary containing anime or manga by genre.

**Return type** Dict

### Examples

```
>>> jikan.genre(type='anime', genre_id=1)
>>> jikan.genre(type='manga', genre_id=2)
```

**magazine** (*magazine\_id: int, page: Optional[int] = None*) → Dict[str, Any]

Gets manga by the magazine/serializer/publisher.

#### Parameters

- **magazine\_id** (int) – Magazine ID from MyAnimeList.
- **page** (int, optional) – Page number of the results. Defaults to None.

#### Returns

**Dictionary containing manga by the** magazine/serializer/publisher.

**Return type** Dict

### Examples

```
>>> jikan.magazine(magazine_id=83)
>>> jikan.magazine(magazine_id=83, page=2)
```

**manga** (*id: int, extension: Optional[str] = None, page: Optional[int] = None*) → Dict[str, Any]

Gets information on a manga.

#### Parameters

- **id** (int) – ID of the manga to get the information of.
- **extension** (str, optional) – Special information to get of the manga. Possible values are in the Jikan API documentation. Defaults to None.
- **page** (int, optional) – to None.

**Returns** Dictionary containing information about the manga.

**Return type** Dict

## Examples

```
>>> jikan.manga(1630)
```

**meta** (*request: str, type: Optional[str] = None, period: Optional[str] = None, offset: Optional[int] = None*) → Dict[str, Any]

Gets meta information regarding the Jikan API.

### Parameters

- **request** (str) – Type of request. Possible values are requests and status.
- **type** (str, optional) – Type of information to get for requests. Possible values are in the Jikan API documentation. Defaults to None.
- **period** (str, optional) – Time period to get for requests. Possible values are today, weekly, and monthly. Defaults to None.
- **offset** (int, optional) – 1,000 requests are shown per page. Offset is used to show more. Defaults to None.

**Returns** Dictionary containing meta information.

**Return type** Dict

## Examples

```
>>> jikan.meta('requests')
>>> jikan.meta('requests', type='anime', period='today')
>>> jikan.meta('status')
```

**person** (*id: int, extension: Optional[str] = None*) → Dict[str, Any]

Gets information on a person.

### Parameters

- **id** (int) – ID of the person to get the information of.
- **extension** (str, optional) – Special information to get of the person. Possible values are in the Jikan API documentation. Defaults to None.

**Returns** Dictionary containing information about the person.

**Return type** Dict

## Examples

```
>>> jikan.person(2)
```

**producer** (*producer\_id: int, page: Optional[int] = None*) → Dict[str, Any]

Gets anime by the producer/studio/licensor.

### Parameters

- **producer\_id** (int) – Producer ID from MyAnimeList.
- **page** (int, optional) – Page number of the results. Defaults to None.

**Returns** Dictionary containing anime by the producer/studio/licensor.

**Return type** Dict



## Examples

```
>>> jikan.producer(producer_id=4)
>>> jikan.producer(producer_id=4, page=2)
```

**schedule** (*day: Optional[str] = None*) → Dict[str, Any]

Gets anime scheduled.

**Parameters** **day** (str, optional) – Day of the week to get the scheduled anime. Defaults to None.

**Returns** Dictionary containing anime scheduled.

**Return type** Dict

## Examples

```
>>> jikan.schedule()
>>> jikan.schedule(day='monday')
```

**search** (*search\_type: str, query: str, page: Optional[int] = None, parameters: Optional[Mapping[str, Union[int, str, float, None]]] = None*) → Dict[str, Any]  
Searches for a query on MyAnimeList.

### Parameters

- **search\_type** (str) – Where to search. Possible values are anime, manga, person, and character.
- **query** (str) – Query to search for.
- **page** (int, optional) – Page number of the results. Defaults to None.
- **parameters** (dict, optional) – Dictionary containing key,value pairs for ?key=value in url query. Defaults to None.

**Returns** Dictionary containing search results.

**Return type** Dict

## Examples

```
>>> jikan.search('anime', 'Jojo')
>>> jikan.search('anime', 'Jojo', page=2)
>>> jikan.search('anime', 'Jojo', parameters={'type': 'tv'})
>>> jikan.search(
    'anime', 'Jojo', page=2, parameters={'genre': 37, 'type': 'tv'}
)
```

**season** (*year: Optional[int] = None, season: Optional[str] = None*) → Dict[str, Any]

**Gets information on anime of the specific season or the current season if no parameters are specified.**

### Parameters

- **year** (int, optional) – Year to get anime of. Defaults to None.

- **season** (`str`, optional) – Season to get anime of. Possible values are winter, spring, summer, and fall. Defaults to None.

**Returns** Dictionary containing information on anime of the season.

**Return type** Dict

### Examples

```
>>> jikan.season()
>>> jikan.season(year=2018, season='winter')
>>> jikan.season(year=2016, season='spring')
```

**season\_archive()** → Dict[str, Any]

Gets all the years and their respective seasons from MyAnimeList.

**Returns** Dictionary containing all the years and seasons.

**Return type** Dict

### Examples

```
>>> jikan.season_archive()
```

**season\_later()** → Dict[str, Any]

Gets anime that have been announced for upcoming seasons.

**Returns** Dictionary containing anime in upcoming seasons.

**Return type** Dict

### Examples

```
>>> jikan.season_later()
```

**top** (*type: str, page: Optional[int] = None, subtype: Optional[str] = None*) → Dict[str, Any]

Gets top items on MyAnimeList.

#### Parameters

- **type** (`str`) – Type to get top items from. Possible values are anime and manga.
- **page** (`int`, optional) – Page number of the results. Defaults to None.
- **subtype** (`str`, optional) – Subtype to get filtered top items. Possible values are in the Jikan API documentation. Defaults to None.

**Returns** Dictionary containing top items on MyAnimeList.

**Return type** Dict

### Examples

```
>>> jikan.top(type='manga')
>>> jikan.top(type='anime', page=2, subtype='upcoming')
```

**user** (*username: str, request: Optional[str] = None, argument: Union[int, str, None] = None, page: Optional[int] = None, parameters: Optional[Mapping[str, Any]] = None*) → Dict[str, Any]  
 Gets information about the user.

#### Parameters

- **username** (str) – MyAnimeList username.
- **request** (str, optional) – Type of data to get. Possible values are profile, history, friends, animelist, and mangalist. Defaults to None.
- **argument** (str or int, optional) – For history, possible values are anime and manga. For animelist and mangalist, possible values are in the Jikan API documentation. Defaults to None.
- **page** (int, optional) – Page number for friends. Defaults to None.
- **parameters** (dict, optional) – Dictionary containing key,value pairs for ?key=value in url query. Defaults to None.

**Returns** Dictionary containing information about the user.

**Return type** Dict

#### Examples

```
>>> jikan.user(username='Xinil')
>>> jikan.user(username='Xinil', request='profile')
>>> jikan.user(username='Xinil', request='friends', page=2)
>>> jikan.user(username='Xinil', request='history')
>>> jikan.user(username='Xinil', request='animelist', argument='ptw')
>>> jikan.user(
    username='Xinil', request='animelist', parameters={'page': 2}
)
>>> jikan.user(
    username='Xinil',
    request='animelist',
    argument='ptw',
    parameters={'page': 2}
)
```

**static user\_list** (*id: int, extension: Optional[str] = None*) → Dict[str, Any]  
 Deprecated: Gets user list information.



## CHAPTER 2

---

### AioJikan

---

**class** jikanpy.**AioJikan**(*selected\_base: Optional[str] = None, session: Optional[aiohttp.client.ClientSession] = None*)

Asynchronous Jikan wrapper for the jikan.moe unofficial MyAnimeList API.

Note that the API has a limit of 30 requests/minute and 2 requests/second; this module does not make any effort to prevent abuse of that limit, so use it responsibly.

#### Usage Example:

```
import asyncio
from jikanpy import AioJikan

async def main():
    async with AioJikan() as aio_jikan:
        pass

    # You can also construct AioJikan like below, but make sure to close
    # the object
    aio_jikan_2 = AioJikan()
    await aio_jikan_2.close()

asyncio.run(main())
```

#### **base**

The base URL of the Jikan API being accessed.

#### **session**

The aiohttp session.

**\_\_init\_\_**(*selected\_base: Optional[str] = None, session: Optional[aiohttp.client.ClientSession] = None*) → None  
Constructs the AioJikan object.

#### **Parameters**

- **selected\_base** (*str, optional*) – Base url of Jikan API. Defaults to the official Jikan API URL.

- **session** (*aiohttp.ClientSession*, *optional*) – Defaults to a new aiohttp session object.

**Returns** Instance of AioJikan.

**Return type** *AioJikan*

### Examples

```
>>> aio_jikan_1 = AioJikan()
>>> aio_jikan_2 = AioJikan(selected_base='http://localhost:8000/v3')
>>> aio_jikan_3 = AioJikan(
    session=aiohttp.ClientSession(headers={'x-test': 'true'})
)
```

**anime** (*id: int, extension: Optional[str] = None, page: Optional[int] = None*) → Dict[str, Any]

Gets information on an anime.

#### Parameters

- **id** (*int*) – ID of the anime to get the information of.
- **extension** (*str*, *optional*) – Special information to get of the anime. Possible values are in the Jikan API documentation. Defaults to None.
- **page** (*int*, *optional*) – to None.

**Returns** Dictionary containing information about the anime.

**Return type** Dict[str, Any]

### Examples

```
>>> await jikan.anime(14719)
>>> await jikan.anime(14719, extension='episodes')
>>> await jikan.anime(14719, extension='episodes', page=2)
>>> await jikan.anime(14719, extension='news')
```

**character** (*id: int, extension: Optional[str] = None*) → Dict[str, Any]

Gets information on a character.

#### Parameters

- **id** (*int*) – ID of the character to get the information of.
- **extension** (*str*, *optional*) – Special information to get of the character. Possible values are in the Jikan API documentation. Defaults to None.

**Returns** Dictionary containing information about the character.

**Return type** Dict

### Examples

```
>>> await jikan.character(6356)
```

**close** () → None

Close AioHTTP session

**club** (*id: int, extension: Optional[str] = None, page: Optional[int] = None*) → Dict[str, Any]  
Gets information on a club.

#### Parameters

- **id** (int) – ID of the club to get the information of.
- **extension** (str, optional) – Special information to get of the club. Possible values are in the Jikan API documentation. Defaults to None.
- **page** (int, optional) – to None.

**Returns** Dictionary containing information about the club.

**Return type** Dict

#### Examples

```
>>> await jikan.club(379)
```

**genre** (*type: str, genre\_id: int, page: Optional[int] = None*) → Dict[str, Any]  
Gets anime or manga by genre.

#### Parameters

- **type** (str) – Type to get items from. Possible values are anime and manga.
- **genre\_id** (int) – Genre ID from MyAnimeList.
- **page** (int, optional) – Page number of the results. Defaults to None.

**Returns** Dictionary containing anime or manga by genre.

**Return type** Dict

#### Examples

```
>>> await jikan.genre(type='anime', genre_id=1)
>>> await jikan.genre(type='manga', genre_id=2)
```

**magazine** (*magazine\_id: int, page: Optional[int] = None*) → Dict[str, Any]  
Gets manga by the magazine/serializer/publisher.

#### Parameters

- **magazine\_id** (int) – Magazine ID from MyAnimeList.
- **page** (int, optional) – Page number of the results. Defaults to None.

#### Returns

Dictionary containing manga by the magazine/serializer/publisher.

**Return type** Dict

#### Examples

```
>>> await jikan.magazine(magazine_id=83)
>>> await jikan.magazine(magazine_id=83, page=2)
```

**manga** (*id: int, extension: Optional[str] = None, page: Optional[int] = None*) → Dict[str, Any]  
Gets information on a manga.

**Parameters**

- **id** (int) – ID of the manga to get the information of.
- **extension** (str, optional) – Special information to get of the manga. Possible values are in the Jikan API documentation. Defaults to None.
- **page** (int, optional) – to None.

**Returns** Dictionary containing information about the manga.

**Return type** Dict

**Examples**

```
>>> await jikan.manga(1630)
```

**meta** (*request: str, type: Optional[str] = None, period: Optional[str] = None, offset: Optional[int] = None*) → Dict[str, Any]  
Gets meta information regarding the Jikan API.

**Parameters**

- **request** (str) – Type of request. Possible values are requests and status.
- **type** (str, optional) – Type of information to get for requests. Possible values are in the Jikan API documentation. Defaults to None.
- **period** (str, optional) – Time period to get for requests. Possible values are today, weekly, and monthly. Defaults to None.
- **offset** (int, optional) – 1,000 requests are shown per page. Offset is used to show more. Defaults to None.

**Returns** Dictionary containing meta information.

**Return type** Dict

**Examples**

```
>>> await jikan.meta('requests')
>>> await jikan.meta('requests', type='anime', period='today')
>>> await jikan.meta('status')
```

**person** (*id: int, extension: Optional[str] = None*) → Dict[str, Any]  
Gets information on a person.

**Parameters**

- **id** (int) – ID of the person to get the information of.
- **extension** (str, optional) – Special information to get of the person. Possible values are in the Jikan API documentation. Defaults to None.

**Returns** Dictionary containing information about the person.

**Return type** Dict



## Examples

```
>>> await jikan.person(2)
```

**producer** (*producer\_id: int, page: Optional[int] = None*) → Dict[str, Any]

Gets anime by the producer/studio/licensor.

### Parameters

- **producer\_id** (int) – Producer ID from MyAnimeList.
- **page** (int, optional) – Page number of the results. Defaults to None.

**Returns** Dictionary containing anime by the producer/studio/licensor.

**Return type** Dict

## Examples

```
>>> await jikan.producer(producer_id=4)
>>> await jikan.producer(producer_id=4, page=2)
```

**schedule** (*day: Optional[str] = None*) → Dict[str, Any]

Gets anime scheduled.

**Parameters** **day** (str, optional) – Day of the week to get the scheduled anime. Defaults to None.

**Returns** Dictionary containing anime scheduled.

**Return type** Dict

## Examples

```
>>> await jikan.schedule()
>>> await jikan.schedule(day='monday')
```

**search** (*search\_type: str, query: str, page: Optional[int] = None, parameters: Optional[Mapping[str, Union[int, str, float, None]]] = None*) → Dict[str, Any]

Searches for a query on MyAnimeList.

### Parameters

- **search\_type** (str) – Where to search. Possible values are anime, manga, person, and character.
- **query** (str) – Query to search for.
- **page** (int, optional) – Page number of the results. Defaults to None.
- **parameters** (dict, optional) – Dictionary containing key,value pairs for ?key=value in url query. Defaults to None.

**Returns** Dictionary containing search results.

**Return type** Dict

## Examples

```
>>> await jikan.search('anime', 'Jojo')
>>> await jikan.search('anime', 'Jojo', page=2)
>>> await jikan.search('anime', 'Jojo', parameters={'type': 'tv'})
>>> await jikan.search(
    'anime', 'Jojo', page=2, parameters={'genre': 37, 'type': 'tv'}
)
```

**season** (*year: Optional[int] = None, season: Optional[str] = None*) → Dict[str, Any]

**Gets information on anime of the specific season or the current season if** no parameters are specified.

### Parameters

- **year** (int, optional) – Year to get anime of. Defaults to None.
- **season** (str, optional) – Season to get anime of. Possible values are winter, spring, summer, and fall. Defaults to None.

**Returns** Dictionary containing information on anime of the season.

**Return type** Dict

## Examples

```
>>> await jikan.season()
>>> await jikan.season(year=2018, season='winter')
>>> await jikan.season(year=2016, season='spring')
```

**season\_archive** () → Dict[str, Any]

**Gets all the years and their respective seasons from MyAnimeList.**

**Returns** Dictionary containing all the years and seasons.

**Return type** Dict

## Examples

```
>>> await jikan.season_archive()
```

**season\_later** () → Dict[str, Any]

**Gets anime that have been announced for upcoming seasons.**

**Returns** Dictionary containing anime in upcoming seasons.

**Return type** Dict

## Examples

```
>>> await jikan.season_later()
```

**top** (*type: str, page: Optional[int] = None, subtype: Optional[str] = None*) → Dict[str, Any]

**Gets top items on MyAnimeList.**

### Parameters

- **type** (`str`) – Type to get top items from. Possible values are anime and manga.
- **page** (`int`, optional) – Page number of the results. Defaults to None.
- **subtype** (`str`, optional) – Subtype to get filtered top items. Possible values are in the Jikan API documentation. Defaults to None.

**Returns** Dictionary containing top items on MyAnimeList.

**Return type** Dict

## Examples

```
>>> await jikan.top(type='manga')
>>> await jikan.top(type='anime', page=2, subtype='upcoming')
```

**user** (`username: str`, `request: Optional[str] = None`, `argument: Union[int, str, None] = None`, `page: Optional[int] = None`, `parameters: Optional[Mapping[str, Any]] = None`) → Dict[str, Any]  
Gets information about the user.

### Parameters

- **username** (`str`) – MyAnimeList username.
- **request** (`str`, optional) – Type of data to get. Possible values are profile, history, friends, animelist, and mangalist. Defaults to None.
- **argument** (`str` or `int`, optional) – For history, possible values are anime and manga. For animelist and mangalist, possible values are in the Jikan API documentation. Defaults to None.
- **page** (`int`, optional) – Page number for friends. Defaults to None.
- **parameters** (`dict`, optional) – Dictionary containing key,value pairs for ?key=value in url query. Defaults to None.

**Returns** Dictionary containing information about the user.

**Return type** Dict

## Examples

```
>>> await jikan.user(username='Xinil')
>>> await jikan.user(username='Xinil', request='profile')
>>> await jikan.user(username='Xinil', request='friends', page=2)
>>> await jikan.user(username='Xinil', request='history')
>>> await jikan.user(username='Xinil', request='animelist', argument='ptw')
>>> await jikan.user(
    username='Xinil', request='animelist', parameters={'page': 2}
)
>>> await jikan.user(
    username='Xinil',
    request='animelist',
    argument='ptw',
    parameters={'page': 2}
)
```

**user\_list** (`id: int`, `extension: Optional[str] = None`) → Dict[str, Any]  
Deprecated: Gets user list information.



## CHAPTER 3

---

### Indices and tables

---

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



## Symbols

`__init__()` (*jikanpy.AioJikan method*), 9  
`__init__()` (*jikanpy.Jikan method*), 1

## A

`AioJikan` (*class in jikanpy*), 9  
`anime()` (*jikanpy.AioJikan method*), 10  
`anime()` (*jikanpy.Jikan method*), 2

## B

`base` (*jikanpy.AioJikan attribute*), 9  
`base` (*jikanpy.Jikan attribute*), 1

## C

`character()` (*jikanpy.AioJikan method*), 10  
`character()` (*jikanpy.Jikan method*), 2  
`close()` (*jikanpy.AioJikan method*), 10  
`club()` (*jikanpy.AioJikan method*), 10  
`club()` (*jikanpy.Jikan method*), 2

## G

`genre()` (*jikanpy.AioJikan method*), 11  
`genre()` (*jikanpy.Jikan method*), 3

## J

`Jikan` (*class in jikanpy*), 1

## M

`magazine()` (*jikanpy.AioJikan method*), 11  
`magazine()` (*jikanpy.Jikan method*), 3  
`manga()` (*jikanpy.AioJikan method*), 11  
`manga()` (*jikanpy.Jikan method*), 3  
`meta()` (*jikanpy.AioJikan method*), 12  
`meta()` (*jikanpy.Jikan method*), 4

## P

`person()` (*jikanpy.AioJikan method*), 12  
`person()` (*jikanpy.Jikan method*), 4

`producer()` (*jikanpy.AioJikan method*), 13  
`producer()` (*jikanpy.Jikan method*), 4

## S

`schedule()` (*jikanpy.AioJikan method*), 13  
`schedule()` (*jikanpy.Jikan method*), 5  
`search()` (*jikanpy.AioJikan method*), 13  
`search()` (*jikanpy.Jikan method*), 5  
`season()` (*jikanpy.AioJikan method*), 14  
`season()` (*jikanpy.Jikan method*), 5  
`season_archive()` (*jikanpy.AioJikan method*), 14  
`season_archive()` (*jikanpy.Jikan method*), 6  
`season_later()` (*jikanpy.AioJikan method*), 14  
`season_later()` (*jikanpy.Jikan method*), 6  
`session` (*jikanpy.AioJikan attribute*), 9  
`session` (*jikanpy.Jikan attribute*), 1

## T

`top()` (*jikanpy.AioJikan method*), 14  
`top()` (*jikanpy.Jikan method*), 6

## U

`user()` (*jikanpy.AioJikan method*), 15  
`user()` (*jikanpy.Jikan method*), 6  
`user_list()` (*jikanpy.AioJikan method*), 15  
`user_list()` (*jikanpy.Jikan static method*), 7