

# API Docs

## Contents

- [Linkedin](#)

linkedin-api

```
class linkedin_api.Linkedin(username: str, password: str, *,  
authenticate=True, refresh_cookies=False, debug=False, proxies={},  
cookies=None, cookies_dir: str = '')
```

Class for accessing the LinkedIn API.

### Parameters:

- **username** (*str*) – Username of LinkedIn account.
- **password** (*str*) – Password of LinkedIn account.

```
add_connection(profile_public_id: str, message='', profile_urn=None)
```

Add a given profile id as a connection.

### Parameters:

- **profile\_public\_id** (*str*) – public ID of a LinkedIn profile
- **message** – message to send along with connection request
- **profile\_urn** (*str, optional*) – member URN for the given LinkedIn profile

### Returns:

Error state. True if error occurred

### Return type:

boolean

```
follow_company(following_state_urn, following=True)
```

Follow a company from its ID.

### Parameters:

 [latest](#)

- **following\_state\_urn** (*str*) – LinkedIn State URN to append to URL to follow the company
- **following** (*bool, optional*) – The following state to set. True by default for following the company

**Returns:**

Error state. If True, an error occurred.

**Return type:**

boolean

**get\_company**(*public\_id*)

Fetch data about a given LinkedIn company.

**Parameters:**

**public\_id** (*str*) – LinkedIn public ID for a company

**Returns:**

Company data

**Return type:**

dict

**get\_company\_updates**(*public\_id: str | None = None, urn\_id: str | None = None, max\_results: int | None = None, results: List | None = None*) → **List**

Fetch company updates (news activity) for a given LinkedIn company.

**Parameters:**

- **public\_id** (*str, optional*) – LinkedIn public ID for a company
- **urn\_id** (*str, optional*) – LinkedIn URN ID for a company

**Returns:**

List of company update objects

**Return type:**

list

**get\_conversation**(*conversation\_urn\_id: str*) [latest](#)

Fetch data about a given conversation.

**Parameters:**

**conversation\_urn\_id** (*str*) – LinkedIn URN ID for a conversation

**Returns:**

Conversation data

**Return type:**

dict

### **get\_conversation\_details(profile\_urn\_id)**

Fetch conversation (message thread) details for a given LinkedIn profile.

**Parameters:**

**profile\_urn\_id** (*str*) – LinkedIn URN ID for a profile

**Returns:**

Conversation data

**Return type:**

dict

### **get\_conversations()**

Fetch list of conversations the user is in.

**Returns:**

List of conversations

**Return type:**

list

### **get\_current\_profile\_views()**

Get profile view statistics, including chart data.

**Returns:**

Profile view data

**Return type:**

dict

 [latest](#)

### **get\_feed\_posts(limit=-1, offset=0, exclude\_promoted\_posts=True)**

Get a list of URNs from feed sorted by 'Recent'

**Parameters:**

- **limit** (*int, optional*) – Maximum length of the returned list, defaults to -1 (no limit)
- **offset** (*int, optional*) – Index to start searching from
- **exclude\_promoted\_posts** (*bool, optional*) – Exclude from the output promoted posts

**Returns:**

List of URNs

**Return type:**

list

### **get\_invitations(start=0, limit=3)**

Fetch connection invitations for the currently logged in user.

**Parameters:**

- **start** (*int*) – How much to offset results by
- **limit** (*int*) – Maximum amount of invitations to return

**Returns:**

List of invitation objects

**Return type:**

list

### **get\_job(job\_id: str) → Dict**

Fetch data about a given job. :param job\_id: LinkedIn job ID :type job\_id: str

**Returns:**

Job data

**Return type:**

dict

### **get\_job\_skills(job\_id: str) → Dict**

  latest

Fetch skills associated with a given job. :param job\_id: LinkedIn job ID :type job\_id: str

**Returns:**

Job skills

**Return type:**

dict

**get\_post\_comments**(*post\_urn: str, comment\_count=100*) → List

get\_post\_comments: Get post comments

**Parameters:**

- **post\_urn** (*str*) – Post URN
- **comment\_count** (*int, optional*) – Number of comments to fetch

**Returns:**

List of post comments

**Return type:**

list

**get\_post\_reactions**(*urn\_id, max\_results=None, results=None*)

Fetch social reactions for a given LinkedIn post.

**Parameters:**

- **urn\_id** (*str*) – LinkedIn URN ID for a post
- **max\_results** (*int, optional*) – Maximum results to return

**Returns:**

List of social reactions

**Return type:**

list

# Note: This may need to be updated to GraphQL in the future, see

[🔗 tomquirk/linkedin-api#309](https://tomquirk.com/linkedin-api#309)

**get\_profile**(*public\_id: str | None = None, urn\_id: str | None = None*)

→ Dict

Fetch data for a given LinkedIn profile.



[latest](#)

**Parameters:**

- **public\_id** (*str, optional*) – LinkedIn public ID for a profile
- **urn\_id** (*str, optional*) – LinkedIn URN ID for a profile

**Returns:**

Profile data

**Return type:**

dict

**get\_profile\_connections**(*urn\_id: str, \*\*kwargs*) → List

Fetch connections for a given LinkedIn profile.

See `LinkedIn.search_people()` for additional searching parameters.

**Parameters:**

**urn\_id** (*str*) – LinkedIn URN ID for a profile

**Returns:**

List of search results

**Return type:**

list

**get\_profile\_contact\_info**(*public\_id: str | None = None, urn\_id: str | None = None*) → Dict

Fetch contact information for a given LinkedIn profile. Pass a [public\_id] or a [urn\_id].

**Parameters:**

- **public\_id** (*str, optional*) – LinkedIn public ID for a profile
- **urn\_id** (*str, optional*) – LinkedIn URN ID for a profile

**Returns:**

Contact data

**Return type:**

dict

**get\_profile\_experiences**(*urn\_id: str*) → List

Fetch experiences for a given LinkedIn profile.

[latest](#)

NOTE: data structure differs slightly from `Linkedin.get_profile()` experiences.

**Parameters:**

**urn\_id** (*str*) – LinkedIn URN ID for a profile

**Returns:**

List of experiences

**Return type:**

list

### `get_profile_member_badges(public_profile_id: str)`

Fetch badges for a given LinkedIn profile.

**Parameters:**

**public\_profile\_id** (*str*) – public ID of a LinkedIn profile

**Returns:**

Badges data

**Return type:**

dict

### `get_profile_network_info(public_profile_id: str)`

Fetch network information for a given LinkedIn profile.

Network information includes the following: - number of connections - number of followers - if the account is followable - the network distance between the API session user and the profile - if the API session user is following the profile

**Parameters:**

**public\_profile\_id** (*str*) – public ID of a LinkedIn profile

**Returns:**

Network data

**Return type:**

dict

### `get_profile_posts(public_id: str | None = None, urn_id: latest None, post_count=10) → List`

get\_profile\_posts: Get profile posts

**Parameters:**

- **public\_id** (*str, optional*) – LinkedIn public ID for a profile
- **urn\_id** (*str, optional*) – LinkedIn URN ID for a profile
- **post\_count** (*int, optional*) – Number of posts to fetch

**Returns:**

List of posts

**Return type:**

list

**get\_profile\_privacy\_settings**(*public\_profile\_id: str*)

Fetch privacy settings for a given LinkedIn profile.

**Parameters:**

**public\_profile\_id** (*str*) – public ID of a LinkedIn profile

**Returns:**

Privacy settings data

**Return type:**

dict

**get\_profile\_skills**(*public\_id: str | None = None, urn\_id: str | None = None*) → List

Fetch the skills listed on a given LinkedIn profile.

**Parameters:**

- **public\_id** (*str, optional*) – LinkedIn public ID for a profile
- **urn\_id** (*str, optional*) – LinkedIn URN ID for a profile

**Returns:**

List of skill objects

**Return type:**

list



**get\_profile\_updates**(*public\_id=None, urn\_id=None, max\_results=None, results=None*)

Fetch profile updates (newsfeed activity) for a given LinkedIn profile.

**Parameters:**

- **public\_id** (*str, optional*) – LinkedIn public ID for a profile
- **urn\_id** (*str, optional*) – LinkedIn URN ID for a profile

**Returns:**

List of profile update objects

**Return type:**

list

**get\_school**(*public\_id*)

Fetch data about a given LinkedIn school.

**Parameters:**

**public\_id** (*str*) – LinkedIn public ID for a school

**Returns:**

School data

**Return type:**

dict

**get\_user\_profile**(*use\_cache=True*) → Dict

Get the current user profile. If not cached, a network request will be fired.

**Returns:**

Profile data for currently logged in user

**Return type:**

dict

**mark\_conversation\_as\_seen**(*conversation\_urn\_id: str*)

Send 'seen' to a given conversation.

 [latest](#)

**Parameters:**

**conversation\_urn\_id** (*str*) – LinkedIn URN ID for a conversation

**Returns:**

Error state. If True, an error occurred.

**Return type:**

boolean

**react\_to\_post**(*post\_urn\_id*, *reaction\_type*='LIKE')

React to a given post. :param post\_urn\_id: LinkedIn Post URN ID :type post\_urn\_id: str :param reactionType: LinkedIn reaction type, defaults to "LIKE", can be "LIKE", "PRAISE", "APPRECIATION", "EMPATHY", "INTEREST", "ENTERTAINMENT" :type reactionType: str

**Returns:**

Error state. If True, an error occurred.

**Return type:**

boolean

**remove\_connection**(*public\_profile\_id*: str)

Remove a given profile as a connection.

**Parameters:**

**public\_profile\_id** (str) – public ID of a LinkedIn profile

**Returns:**

Error state. True if error occurred


**Return type:**

boolean

**reply\_invitation**(*invitation\_entity\_urn*: str, *invitation\_shared\_secret*: str, *action*='accept')

Respond to a connection invitation. By default, accept the invitation.

**Parameters:**

- **invitation\_entity\_urn** (int) – URN ID of the invitation
- **invitation\_shared\_secret** (str) – Shared secret of invitation  [latest](#)
- **action** (str, optional) – "accept" or "reject". Defaults to "accept"

**Returns:**

Success state. True if successful

### Return type:

boolean

**search**(*params: Dict, limit=-1, offset=0*) → List

Perform a LinkedIn search.

### Parameters:

- **params** (*dict*) – Search parameters (see code)
- **limit** (*int, optional*) – Maximum length of the returned list, defaults to -1 (no limit)
- **offset** (*int, optional*) – Index to start searching from

### Returns:

List of search results

### Return type:

list

**search\_companies**(*keywords: List[str] | None = None, \*\*kwargs*) → List

Perform a LinkedIn search for companies.

### Parameters:


**keywords** (*list, optional*) – A list of search keywords (str)

### Returns:

List of companies

### Return type:

list

**search\_jobs**(*keywords: str | None = None, companies: List[str] | None = None, experience: List[Literal['1', '2', '3', '4', '5', '6']] | None = None, job\_type: List[Literal['F', 'C', 'P', 'T', 'I', 'V', 'O']] | None = None, job\_title: List[str] | None = None, industries: List[str] | None = None, location\_name: str | None = None, remote: List[Literal['1', '2', '3']] | None = None, listed\_at=86400,  latest int | None = None, limit=-1, offset=0, \*\*kwargs*) → List[Dict]

Perform a LinkedIn search for jobs.

**Parameters:**

- **keywords** (*str, optional*) – Search keywords (str)
- **companies** (*list, optional*) – A list of company URN IDs (str)
- **experience** (*list, optional*) – A list of experience levels, one or many of "1", "2", "3", "4", "5" and "6" (internship, entry level, associate, mid-senior level, director and executive, respectively)
- **job\_type** (*list, optional*) – A list of job types , one or many of "F", "C", "P", "T", "I", "V", "O" (full-time, contract, part-time, temporary, internship, volunteer and "other", respectively)
- **job\_title** (*list, optional*) – A list of title URN IDs (str)
- **industries** (*list, optional*) – A list of industry URN IDs (str)
- **location\_name** (*str, optional*) – Name of the location to search within.  
Example: "Kyiv City, Ukraine"
- **remote** (*list, optional*) – Filter for remote jobs, onsite or hybrid. onsite:"1", remote:"2", hybrid:"3"
- **listed\_at** (*int/str, optional. Default value is equal to 24 hours.*) – maximum number of seconds passed since job posting. 86400 will filter job postings posted in last 24 hours.
- **distance** (*int/str, optional. If not specified, None or 0, the default value of 25 miles applied.*) – maximum distance from location in miles
- **limit** (*int, optional, default -1*) – maximum number of results obtained from API queries. -1 means maximum which is defined by constants and is equal to 1000 now.
- **offset** (*int, optional*) – indicates how many search results shall be skipped

**Returns:**

List of jobs

**Return type:**

list

```
search_people(keywords: str | None = None, connection_of: str | None = None, network_depths: List[Literal['F', 'S', 'O']] | None = None, current_company: List[str] | None = None, past_companies: List[str] | None = None, nonprofit_interests: List[str] | None = None, profile_languages: List[str] | None = None, regions: List[str] | None = None, industries: List[str] | None = None, schools: List[str] | None = None, contact_interests: List[str] | None = None,
```

```

service_categories: List[str] | None = None,
include_private_profiles=False, keyword_first_name: str | None = None,
keyword_last_name: str | None = None, keyword_title: str | None =
None, keyword_company: str | None = None, keyword_school: str | None =
None, network_depth: Literal['F'] | Literal['S'] | Literal['O'] | None
= None, title: str | None = None, **kwargs) → List[Dict]

```

Perform a LinkedIn search for people.

### Parameters:

- **keywords** (*str, optional*) – Keywords to search on
- **current\_company** (*list, optional*) – A list of company URN IDs (*str*)
- **past\_companies** (*list, optional*) – A list of company URN IDs (*str*)
- **regions** (*list, optional*) – A list of geo URN IDs (*str*)
- **industries** (*list, optional*) – A list of industry URN IDs (*str*)
- **schools** (*list, optional*) – A list of school URN IDs (*str*)
- **profile\_languages** (*list, optional*) – A list of 2-letter language codes (*str*)
- **contact\_interests** (*list, optional*) – A list containing one or both of "proBono" and "boardMember"
- **service\_categories** (*list, optional*) – A list of service category URN IDs (*str*)
- **network\_depth** (*str, optional*) – Deprecated, use *network\_depths*. One of "F", "S" and "O" (first, second and third+ respectively)
- **network\_depths** (*list, optional*) – A list containing one or many of "F", "S" and "O" (first, second and third+ respectively)
- **include\_private\_profiles** (*boolean, optional*) – Include private profiles in search results. If False, only public profiles are included. Defaults to False
- **keyword\_first\_name** (*str, optional*) – First name
- **keyword\_last\_name** (*str, optional*) – Last name
- **keyword\_title** (*str, optional*) – Job title
- **keyword\_company** (*str, optional*) – Company name
- **keyword\_school** (*str, optional*) – School name
- **connection\_of** (*str, optional*) – Connection of LinkedIn user, given by profile URN ID
- **limit** (*int, optional*) – Maximum length of the returned list, defaults to -1 (no limit)



### Returns:

List of profiles (minimal data only)

**Return type:**

list

**send\_message**(*message\_body: str, conversation\_urn\_id: str | None = None, recipients: List[str] | None = None*)

Send a message to a given conversation.

**Parameters:**

- **message\_body** (*str*) – Message text to send
- **conversation\_urn\_id** (*str, optional*) – LinkedIn URN ID for a conversation
- **recipients** (*list, optional*) – List of profile urn id's

**Returns:**

Error state. If True, an error occurred.

**Return type:**

boolean

**unfollow\_entity**(*urn\_id: str*)

Unfollow a given entity.

**Parameters:**

**urn\_id** (*str*) – URN ID of entity to unfollow

**Returns:**

Error state. Returns True if error occurred

**Return type:**

boolean