

---

# **ProKnow DS - Python SDK Documentation**

**Kyle Burnett**

**Jan 15, 2019**



---

## Contents:

---

<b>1</b>	<b>Usage</b>	<b>3</b>
1.1	Installation . . . . .	3
1.2	Basic Usage . . . . .	3
<b>2</b>	<b>API Reference</b>	<b>5</b>
2.1	<code>proknow.ProKnow</code> – <code>ProKnow</code> . . . . .	5
2.2	<code>proknow.Requestor</code> – <code>Requestor</code> . . . . .	6
2.3	<code>proknow.Workspaces</code> – <code>Workspaces</code> . . . . .	8
2.4	<code>proknow.Roles</code> – <code>Roles</code> . . . . .	9
2.5	<code>proknow.Users</code> – <code>Users</code> . . . . .	12
<b>3</b>	<b>Contributing Guide</b>	<b>15</b>
3.1	Requirements . . . . .	15
3.2	Accepted Contributions . . . . .	15
<b>4</b>	<b>Indices and tables</b>	<b>17</b>
	<b>Python Module Index</b>	<b>19</b>



Welcome to the ProKnow SDK documentation. Use the links below to access the usage page and the complete api documentation.



# CHAPTER 1

---

## Usage

---

Read this guide to learn how to install ProKnow SDK and get started with your first script.

### 1.1 Installation

Use pip to install:

```
pip install proknow
```

### 1.2 Basic Usage

To use the SDK, you'll need to generate an [API token](#) in the ProKnow DS user interface. Once you have a credentials file, use it in your script to create an instance of the main ProKnow class:

```
from proknow import ProKnow

pk = ProKnow('https://example.proknow.com', credentials_file="./credentials.json")
workspaces = pk.workspaces.query()
print("The workspaces are as follows:")
for workspace in workspaces:
    print(workspace.name)
```





This guide includes a complete API reference.

## 2.1 proknow.ProKnow – ProKnow

**class** proknow.ProKnow.**ProKnow**(*base\_url*, *credentials\_file*=None, *credentials\_id*=None, *credentials\_secret*=None)

This is the main class that should be instantiated at the beginning of your Python program with your base URL (which should include your account subdomain) and your API token credentials.

**requestor**

*proknow.Requestor.Requestor* – An instance of the Requestor class.

**workspaces**

*proknow.Workspaces.Workspaces* – An instance of the Workspaces class.

**\_\_init\_\_**(*base\_url*, *credentials\_file*=None, *credentials\_id*=None, *credentials\_secret*=None)

Initializes the ProKnow class.

The *base\_url* must be provided as should either the *credentials\_file* or both the *credentials\_id* and *credentials\_secret*.

**Parameters**

- **base\_url** (*str*) – The base URL to use when making request to the ProKnow API.
- **credentials\_file** (*str*) – The path to the credentials file obtained by creating an API key in your ProKnow account. The file should be a JSON file containing an object with the fields *id* and *secret*.
- **credentials\_id** (*str*) – An API key id.
- **credentials\_secret** (*str*) – An API key secret.

**Raises** `AssertionError` – If the input parameters are invalid.

## 2.2 proknow.Requestor – Requestor

**class** proknow.Requestor.**Requestor** (*base\_url, username, password*)

A class used for issuing requests for the ProKnow API

**\_\_init\_\_** (*base\_url, username, password*)

Initializes the Requestor class.

### Parameters

- **base\_url** (*str*) – The base URL to use when making request to the ProKnow API.
- **username** (*str*) – The string used in Basic Authentication as the user name.
- **password** (*str*) – The string used in Basic Authentication as the user password.

**delete** (*route, query=None, body=None*)

Issues an HTTP DELETE request.

### Parameters

- **route** (*str*) – The API route to use in the request.
- **query** (*dict, optional*) – An optional dictionary of query parameters to use in the request.
- **body** (*dict, optional*) – An optional dictionary to be used as a JSON request body for the request.

### Returns

A tuple (status\_code, msg).

1. status\_code (int): the response code
2. msg (str, dict): the text response or, if the response was JSON, the decoded JSON dictionary.

### Return type tuple

**get** (*route, query=None*)

Issues an HTTP GET request.

### Parameters

- **route** (*str*) – The API route to use in the request.
- **query** (*dict, optional*) – An optional dictionary of query parameters to use in the request.

### Returns

A tuple (status\_code, msg).

1. status\_code (int): the response code
2. msg (str, dict): the text response or, if the response was JSON, the decoded JSON dictionary.

### Return type tuple

**patch** (*route, query=None, body=None*)

Issues an HTTP PATCH request.

### Parameters

- **route** (*str*) – The API route to use in the request.

- **query** (*dict*, *optional*) – An optional dictionary of query parameters to use in the request.
- **body** (*dict*, *optional*) – An optional dictionary to be used as a JSON request body for the request.

**Returns**

A tuple (status\_code, msg).

1. status\_code (int): the response code
2. msg (str, dict): the text response or, if the response was JSON, the decoded JSON dictionary.

**Return type** tuple

**post** (*route*, *query=None*, *body=None*)

Issues an HTTP POST request.

**Parameters**

- **route** (*str*) – The API route to use in the request.
- **query** (*dict*, *optional*) – An optional dictionary of query parameters to use in the request.
- **body** (*dict*, *optional*) – An optional dictionary to be used as a JSON request body for the request.

**Returns**

A tuple (status\_code, msg).

1. status\_code (int): the response code
2. msg (str, dict): the text response or, if the response was JSON, the decoded JSON dictionary.

**Return type** tuple

**put** (*route*, *query=None*, *body=None*)

Issues an HTTP PUT request.

**Parameters**

- **route** (*str*) – The API route to use in the request.
- **query** (*dict*, *optional*) – An optional dictionary of query parameters to use in the request.
- **body** (*dict*, *optional*) – An optional dictionary to be used as a JSON request body for the request.

**Returns**

A tuple (status\_code, msg).

1. status\_code (int): the response code
2. msg (str, dict): the text response or, if the response was JSON, the decoded JSON dictionary.

**Return type** tuple

## 2.3 proknow.Workspaces – Workspaces

**class** `proknow.Workspaces.Workspaces` (*proknow, requestor*)

This class should be used to interact with the workspaces in a ProKnow organization. It is instantiated for you as an attribute of the `proknow.ProKnow.ProKnow` class.

**\_\_init\_\_** (*proknow, requestor*)

Initializes the Workspaces class.

### Parameters

- **proknow** (`proknow.ProKnow.ProKnow`) – The ProKnow instance that is instantiating the object.
- **requestor** (`proknow.Requestor.Requestor`) – An object used to make API requests.

**create** (*slug, name, protected=True*)

Creates a new workspace.

### Parameters

- **slug** (*str*) – The workspace slug.
- **name** (*str*) – The workspace name.
- **protected** (*bool, optional*) – Indicates whether the workspace should be protected from accidental deletion.

**Returns** A representation of the created workspace.

**Return type** `proknow.Workspaces.WorkspaceItem`

**Raises** `AssertionError` – If the input parameters are invalid.

**delete** (*identifier*)

Deletes a workspace.

**Parameters** **identifier** (*str*) – The id of the workspace to delete.

**Raises** `AssertionError` – If the input parameters are invalid.

**find** (*\*\*kwargs*)

Finds a workspace by id, slug, or name.

---

**Note:** This method utilizes a cache of workspaces. Once it has a cache of workspaces, it will use that cache until the `proknow.Workspaces.Workspaces.query()` method is called to refresh the cache. If you wish to make your code resilient to workspaces changes (i.e., new workspaces, renamed workspaces, deleted workspaces, etc.) while your script is running, you should call the `proknow.Workspaces.Workspaces.query()` method before this method. In most use cases, this is not necessary.

---

**Parameters** **\*\*kwargs** – A dictionary of keyword arguments that may include *identifier*, *slug*, and *name*. These arguments are considered—in that order—to find matching workspaces.

**Returns** A representation of the matching workspace.

**Return type** `proknow.Workspaces.WorkspaceItem`

**query** ()

Queries for workspaces.

---

**Note:** This method refreshes the workspaces cache.

---

**Returns** A list of `proknow.Workspaces.WorkspaceItem` objects, each representing a workspace in the organization.

**Return type** list

**class** `proknow.Workspaces.WorkspaceItem` (*workspaces, workspace*)

This class represents a workspace. It's instantiated by the `proknow.Workspaces.Workspaces` class to represent each of the workspaces in a query result and a created workspace.

**id**

*str* – The id of the workspace (readonly).

**data**

*dict* – The complete representation of the workspace as returned from the API (readonly).

**slug**

*str* – A string used in the URL that uniquely identifies the workspace.

**name**

*str* – The name of the workspace.

**protected**

*bool* – Indicates whether the workspace should be protected from accidental deletion.

**\_\_init\_\_** (*workspaces, workspace*)

Initializes the WorkspaceItem class.

**Parameters**

- **workspaces** (`proknow.Workspaces.Workspaces`) – The Workspaces instance that is instantiating the object.
- **workspace** (*dict*) – A dictionary of workspace attributes.

**delete** ()

Deletes the workspace.

**save** ()

Saves the changes made to a workspace.

### Example

The following example illustrates how to find a workspace by its slug, modify the name, and save it:

```
pk = ProKnow('https://example.proknow.com', credentials_file="./credentials.
↪json")
clinical = pk.workspaces.find(slug='clinical')
clinical.name = "Clinical Patients"
clinical.save()
```

## 2.4 proknow.Roles – Roles

**class** `proknow.Roles.Roles` (*proknow, requestor*)

This class should be used to interact with the roles in a ProKnow organization. It is instantiated for you as an

attribute of the `proknow.ProKnow.ProKnow` class.

**\_\_init\_\_** (*proknow, requestor*)  
Initializes the Roles class.

**Parameters**

- **proknow** (`proknow.ProKnow.ProKnow`) – The ProKnow instance that is instantiating the object.
- **requestor** (`proknow.Requestor.Requestor`) – An object used to make API requests.

**create** (*name, permissions*)  
Creates a new role.

**Parameters**

- **name** (*str*) – The name of the role.
- **permissions** (*dict*) – A dictionary of permissions.

**Returns** A representation of the created item.

**Return type** `proknow.Roles.RoleItem`

**Raises** `AssertionError` – If the input parameters are invalid.

**delete** (*identifier*)  
Deletes a role.

**Parameters** **identifier** (*str*) – The id of the role to delete.

**Raises** `AssertionError` – If the input parameters are invalid.

**find** (*\*\*kwargs*)  
Finds a role by id or name.

**Parameters** **\*\*kwargs** – A dictionary of keyword arguments that may include *identifier* and *name*. These arguments are considered—in that order—to find matching roles.

**Returns** A representation of the matching role.

**Return type** `proknow.Roles.RoleItem`

**get** (*identifier*)  
Gets a role.

**Parameters** **identifier** (*str*) – The id of the role to get.

**Returns** an object representing a role in the organization

**Return type** `proknow.Roles.RoleItem`

**query** ()  
Queries for roles.

**Returns** A list of `proknow.Roles.RoleSummary` objects, each representing a summarized role in the organization.

**Return type** list

**class** `proknow.Roles.RoleSummary` (*roles, role*)

This class represents a summary view of a role. It's instantiated by the `proknow.Roles.Roles.query()` method to represent each of the roles returned in a query result.

**id**  
*str* – The id of the role (readonly).

**name**  
*str* – The name of the role (readonly).

**data**  
*dict* – The summary representation of the role as returned from the API (readonly).

**\_\_init\_\_** (*roles, role*)  
 Initializes the RoleSummary class.

#### Parameters

- **roles** (`proknow.Roles.Roles`) – The Role instance that is instantiating the object.
- **role** (*dict*) – A dictionary of role attributes.

**get** ()  
 Gets the complete representation of the role.

**Returns** an object representing a role in the organization

**Return type** `proknow.Roles.RoleItem`

**class** `proknow.Roles.RoleItem` (*roles, role*)  
 This class represents a role. It's instantiated by the `proknow.Roles.Roles` class as a complete representation of a role.

**id**  
*str* – The id of the role (readonly).

**data**  
*dict* – The complete representation of the role as returned from the API (readonly).

**name**  
*str* – The name of the role.

**permissions**  
*dict* – The dictionary of role permissions.

**\_\_init\_\_** (*roles, role*)  
 Initializes the RoleItem class.

#### Parameters

- **roles** (`proknow.Roles.Roles`) – The Role instance that is instantiating the object.
- **role** (*dict*) – A dictionary of role attributes.

**delete** ()  
 Deletes the role.

**save** ()  
 Saves the changes made to a role.

### Example

The following example illustrates how to find a role by its slug, modify a permission, and save it:

```
pk = ProKnow('https://example.proknow.com', credentials_file="./credentials.
↪json")
researchers = pk.roles.find(name='researchers')
researchers.permissions["organization_read"] = True
researchers.save()
```

## 2.5 proknow.Users – Users

**class** `proknow.Users.Users` (*proknow, requestor*)

This class should be used to interact with the users in a ProKnow organization. It is instantiated for you as an attribute of the `proknow.ProKnow.ProKnow` class.

**\_\_init\_\_** (*proknow, requestor*)

Initializes the Users class.

### Parameters

- **proknow** (`proknow.ProKnow.ProKnow`) – The ProKnow instance that is instantiating the object.
- **requestor** (`proknow.Requestor.Requestor`) – An object used to make API requests.

**create** (*email, name, role\_id, password=None*)

Creates a new user.

### Parameters

- **email** (*str*) – The email of the user.
- **name** (*str*) – The name of the user.
- **role\_id** (*str*) – The id of the role for the user.
- **password** (*str, optional*) – The password of the user.

**Returns** A representation of the created item.

**Return type** `proknow.Users.UserItem`

**Raises** `AssertionError` – If the input parameters are invalid.

**delete** (*identifier*)

Deletes a user.

**Parameters** **identifier** (*str*) – The id of the user to delete.

**Raises** `AssertionError` – If the input parameters are invalid.

**find** (*\*\*kwargs*)

Finds a user by id, email, or name.

**Parameters** **\*\*kwargs** – A dictionary of keyword arguments that may include *identifier*, *email*, and *name*. These arguments are considered—in that order—to find matching roles.

**Returns** A representation of the matching user.

**Return type** `proknow.Users.UserItem`

**get** (*identifier*)

Gets a user.

**Parameters** **identifier** (*str*) – The id of the user to get.



**Returns** an object representing a user in the organization

**Return type** `proknow.Users.UserItem`

**query()**

Queries for users.

**Returns** A list of `proknow.Users.UserSummary` objects, each representing a summarized user in the organization.

**Return type** list

**class** `proknow.Users.UserSummary(users, user)`

This class represents a summary view of a user. It's instantiated by the `proknow.Users.Users.query()` method to represent each of the users returned in a query result.

**id**

*str* – The id of the user (readonly).

**email**

*str* – The email of the user (readonly).

**name**

*str* – The name of the user (readonly).

**data**

*dict* – The summary representation of the user as returned from the API (readonly).

**\_\_init\_\_**(*users, user*)

Initializes the UserSummary class.

**Parameters**

- **users** (`proknow.Users.Users`) – The User instance that is instantiating the object.
- **user** (*dict*) – A dictionary of user attributes.

**get()**

Gets the complete representation of the user.

**Returns** an object representing a user in the organization

**Return type** `proknow.Users.UserItem`

**class** `proknow.Users.UserItem(users, user)`

This class represents a user. It's instantiated by the `proknow.Users.Users` class as a complete representation of a user.

**id**

*str* – The id of the user (readonly).

**data**

*dict* – The complete representation of the user as returned from the API (readonly).

**name**

*str* – The name of the user.

**email**

*str* – The email of the user.

**active**

*bool* – Indicates whether the user is active.

**role\_id**

*str* – The id of the role for the user.

**\_\_init\_\_** (*users, user*)

Initializes the UserItem class.

### Parameters

- **users** (`proknow.Users.Users`) – The User instance that is instantiating the object.
- **user** (*dict*) – A dictionary of user attributes.

**delete** ()

Deletes the user.

**save** ()

Saves the changes made to a user.

### Example

The following example illustrates how to find a user by its email, set it to inactive, and save it:

```
pk = ProKnow('https://example.proknow.com', credentials_file='./credentials.  
↪json")  
jsmith = pk.users.find(email='jsmith@example.com')  
jsmith.active = False  
jsmith.save()
```

This is an open source repository, and we appreciate your contributions. Use this guide to learn how you can be involved.

### 3.1 Requirements

1. You should discuss your change with a ProKnow developer before you begin. We know your time is valuable, and submitting a proposal of your change on our [GitHub issues page](#) is a great way to ensure that you will be working on something that we will likely merge once you submit a pull request.
2. You should follow the existing coding styles. We try to follow the guidelines presented in **PEP 8**. Should you find code that not comply with these guidelines, please bring it to our attention so that we may address it.
3. You should have a ProKnow DS account to run basic tests on your changes. If you do not have an account but still wish to contribute, please contact us at [support@proknow.com](mailto:support@proknow.com).

### 3.2 Accepted Contributions

1. New features.
2. Bug fixes.
3. Documentation.
4. Tests.

We are unlikely to accept pull requests that modify the license or change the way we publish and deploy. For example, we would not accept a change that separated the source and build directories because it would change the way we published the documentation. Tests are the exception to this because we do not yet have a testing strategy in place.



## CHAPTER 4

---

### Indices and tables

---

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



### p

- `proknow.ProKnow`, 5
- `proknow.Requestor`, 6
- `proknow.Roles`, 9
- `proknow.Users`, 12
- `proknow.Workspaces`, 8





## Symbols

[\\_\\_init\\_\\_\(\) \(proknow.ProKnow.ProKnow method\), 5](#)  
[\\_\\_init\\_\\_\(\) \(proknow.Requestor.Requestor method\), 6](#)  
[\\_\\_init\\_\\_\(\) \(proknow.Roles.RoleItem method\), 11](#)  
[\\_\\_init\\_\\_\(\) \(proknow.Roles.RoleSummary method\), 11](#)  
[\\_\\_init\\_\\_\(\) \(proknow.Roles.Roles method\), 10](#)  
[\\_\\_init\\_\\_\(\) \(proknow.Users.UserItem method\), 13](#)  
[\\_\\_init\\_\\_\(\) \(proknow.Users.UserSummary method\), 13](#)  
[\\_\\_init\\_\\_\(\) \(proknow.Users.Users method\), 12](#)  
[\\_\\_init\\_\\_\(\) \(proknow.Workspaces.WorkspaceItem method\), 9](#)  
[\\_\\_init\\_\\_\(\) \(proknow.Workspaces.Workspaces method\), 8](#)

## A

[active \(proknow.Users.UserItem attribute\), 13](#)

## C

[create\(\) \(proknow.Roles.Roles method\), 10](#)  
[create\(\) \(proknow.Users.Users method\), 12](#)  
[create\(\) \(proknow.Workspaces.Workspaces method\), 8](#)

## D

[data \(proknow.Roles.RoleItem attribute\), 11](#)  
[data \(proknow.Roles.RoleSummary attribute\), 11](#)  
[data \(proknow.Users.UserItem attribute\), 13](#)  
[data \(proknow.Users.UserSummary attribute\), 13](#)  
[data \(proknow.Workspaces.WorkspaceItem attribute\), 9](#)  
[delete\(\) \(proknow.Requestor.Requestor method\), 6](#)  
[delete\(\) \(proknow.Roles.RoleItem method\), 11](#)  
[delete\(\) \(proknow.Roles.Roles method\), 10](#)  
[delete\(\) \(proknow.Users.UserItem method\), 14](#)  
[delete\(\) \(proknow.Users.Users method\), 12](#)  
[delete\(\) \(proknow.Workspaces.WorkspaceItem method\), 9](#)  
[delete\(\) \(proknow.Workspaces.Workspaces method\), 8](#)

## E

[email \(proknow.Users.UserItem attribute\), 13](#)  
[email \(proknow.Users.UserSummary attribute\), 13](#)

## F

[find\(\) \(proknow.Roles.Roles method\), 10](#)  
[find\(\) \(proknow.Users.Users method\), 12](#)  
[find\(\) \(proknow.Workspaces.Workspaces method\), 8](#)

## G

[get\(\) \(proknow.Requestor.Requestor method\), 6](#)  
[get\(\) \(proknow.Roles.Roles method\), 10](#)  
[get\(\) \(proknow.Roles.RoleSummary method\), 11](#)  
[get\(\) \(proknow.Users.Users method\), 12](#)  
[get\(\) \(proknow.Users.UserSummary method\), 13](#)

## I

[id \(proknow.Roles.RoleItem attribute\), 11](#)  
[id \(proknow.Roles.RoleSummary attribute\), 10](#)  
[id \(proknow.Users.UserItem attribute\), 13](#)  
[id \(proknow.Users.UserSummary attribute\), 13](#)  
[id \(proknow.Workspaces.WorkspaceItem attribute\), 9](#)

## N

[name \(proknow.Roles.RoleItem attribute\), 11](#)  
[name \(proknow.Roles.RoleSummary attribute\), 11](#)  
[name \(proknow.Users.UserItem attribute\), 13](#)  
[name \(proknow.Users.UserSummary attribute\), 13](#)  
[name \(proknow.Workspaces.WorkspaceItem attribute\), 9](#)

## P

[patch\(\) \(proknow.Requestor.Requestor method\), 6](#)  
[permissions \(proknow.Roles.RoleItem attribute\), 11](#)  
[post\(\) \(proknow.Requestor.Requestor method\), 7](#)  
[ProKnow \(class in proknow.ProKnow\), 5](#)  
[proknow.ProKnow \(module\), 5](#)  
[proknow.Requestor \(module\), 6](#)  
[proknow.Roles \(module\), 9](#)  
[proknow.Users \(module\), 12](#)  
[proknow.Workspaces \(module\), 8](#)  
[protected \(proknow.Workspaces.WorkspaceItem attribute\), 9](#)  
[put\(\) \(proknow.Requestor.Requestor method\), 7](#)

Python Enhancement Proposals

PEP 8, [15](#)

## Q

`query()` (`proknow.Roles.Roles` method), [10](#)

`query()` (`proknow.Users.Users` method), [13](#)

`query()` (`proknow.Workspaces.Workspaces` method), [8](#)

## R

`Requestor` (class in `proknow.Requestor`), [6](#)

`requestor` (`proknow.ProKnow.ProKnow` attribute), [5](#)

`role_id` (`proknow.Users.UserItem` attribute), [13](#)

`RoleItem` (class in `proknow.Roles`), [11](#)

`Roles` (class in `proknow.Roles`), [9](#)

`RoleSummary` (class in `proknow.Roles`), [10](#)

## S

`save()` (`proknow.Roles.RoleItem` method), [11](#)

`save()` (`proknow.Users.UserItem` method), [14](#)

`save()` (`proknow.Workspaces.WorkspaceItem` method), [9](#)

`slug` (`proknow.Workspaces.WorkspaceItem` attribute), [9](#)

## U

`UserItem` (class in `proknow.Users`), [13](#)

`Users` (class in `proknow.Users`), [12](#)

`UserSummary` (class in `proknow.Users`), [13](#)

## W

`WorkspaceItem` (class in `proknow.Workspaces`), [9](#)

`Workspaces` (class in `proknow.Workspaces`), [8](#)

`workspaces` (`proknow.ProKnow.ProKnow` attribute), [5](#)