

---

# **Khoros JX Python Library Documentation**

***Release 3.2.0***

**Jeff Shurtliff**

**Jun 22, 2022**



CONTENTS

1 Table of Contents 3

1.1 Introduction . . . . . 3

1.2 Core Functionality . . . . . 6

1.3 Using the Helper Utility . . . . . 11

1.4 Managing Users . . . . . 15

1.5 Primary Modules . . . . . 17

1.6 Supporting Modules . . . . . 61

1.7 Change Log . . . . . 73

1.8 Roadmap . . . . . 107

2 Module Index and Search 109

Python Module Index 111

Index 113



Welcome to the official documentation for the Khoros JX Python Library.



## TABLE OF CONTENTS

### 1.1 Introduction

The **khorsjx** library acts as a Python software development kit (SDK) to administer and manage [Khoros JX](#) (formerly [Jive-x](#)) and [Jive-n](#) online community platforms.

#### 1.1.1 Installation

The package can be installed via pip using the syntax below.

```
# pip install khorsjx
```

You may also clone the repository and install from source using the syntax below.

```
# git clone git://github.com/jeffshurtliff/khorsjx.git
# cd khorsjx/
# python3 setup.py install
```

#### 1.1.2 Change Log

Changes for each release can be found on the [Change Log](#) page.

### 1.1.3 Basic Usage

This section provides basic usage instructions for the package.

#### Importing the package

The package can be imported into a Python script using the syntax below.

```
import khorosjx
```

#### Initializing the modules

While it is certainly possible to import modules directly (e.g. `from khorosjx import users`), it is recommended that you instead leverage the `init_module()` function as shown below.

```
khorosjx.init_module('content', 'users')
```

In the example above, both the `khorosjx.content` and the `khoros.users` modules have been initiated.

---

**Note:** It is not necessary to import the `khorosjx.core` module as it is imported by default.

---

#### Establishing the API connection

Before leveraging the API in function calls, you must first establish your connection by providing the base URL for the environment (e.g. `https://community.example.com`) and the username and password for the unfederated service account through which the API calls will be made. This is demonstrated below.

```
base_url = 'https://community.example.com'
credentials = ('adminuser', 'password123!')
khorosjx.core.connect(base_url, credentials)
```

---

**Note:** At this time the library only allow connections using [basic authentication](#), but there are plans to include the ability to leverage [OAuth 2.0](#) in a future release.

---



Once the connection has been established, you can proceed to leverage the library of functions in the various modules as needed.

### 1.1.4 Requirements

The following packages are leveraged within the khorosjx package:

- numpy 1.17.4
- pandas-0.25.3
- python-dateutil 2.8.1
- pytz 2019.3
- requests 2.22.0
- urllib3 1.25.7

The full requirements list can be found in the [requirements.txt](#) file.

### 1.1.5 Modules

The KhorosJX Python Library consists of the following *primary modules*:

#### *Core Module (khorosjx.core)*

This module contains core functions such as initializing the connection to the API, getting API version information, performing GET and PUT requests, etc.

#### *Admin Module (khorosjx.admin)*

This module contains administrative functions that would only be performed by a platform administrator or a community manager.

#### *Content Module (khorosjx.content)*

This module contains functions relating to content within the platform which allows for creating, editing and managing content such as documents, ideas, videos, etc.

#### *Groups Module (khorosjx.groups)*

This module contains functions for working with security groups (and eventually social groups) such as obtaining and managing group membership.

#### *Spaces Module (khorosjx.spaces)*

This module contains functions for working with spaces, such as identifying content within spaces, etc.

#### *Users Module (khorosjx.users)*

This module contains functions for working with users, such as obtaining their account/profile information, getting a count of their created content, etc.

The library also includes some *supporting modules* to support the overall functionality of the *primary modules*, as well as modules containing global *Classes and Exceptions* for the library, which are listed below.

### *Core Utilities Module (khorosjx.utils.core\_utils)*

This module includes various utilities to assist in converting dictionaries to JSON, formatting timestamps, etc.

### *Classes Module (khorosjx.utils.classes)*

This module contains nearly all classes utilized by other modules within the library.

### *Exceptions Module (khorosjx.errors.exceptions)*

This module contains all of the exception classes leveraged in functions throughout the library.

## 1.1.6 License

This package falls under the [MIT License](#).

## 1.1.7 Reporting Issues

Issues can be reported within the [GitHub repository](#).

## 1.1.8 Disclaimer

This package is in no way endorsed or supported by the [Khoros](#) or [Aurea Software, Inc.](#) companies.

## 1.2 Core Functionality

**Warning:** This page is currently in development.

This page provides instructions and examples around the core functionality of the library. This includes a review of the *Basic Usage* as well as base functionality such as performing generic API requests.

- *Getting Started*
  - *Importing the package*
  - *Initializing the modules*
  - *Establishing the API connection*
- *Performing GET Requests*
  - *Using the `get_data()` function*

- \* *Optional arguments in the `get_data()` function*
  - *Ignoring exceptions in the `get_data()` function*
  - *Returning the `get_data()` response in JSON format*
- *Using the `get_request_with_retries()` function*
  - \* *Returning the `get_request_with_retries()` response in JSON format*
- *Performing POST Requests*
- *Performing PUT Requests*

## 1.2.1 Getting Started

The topics in this section are necessary to leverage the `khorsjx` library.

### Importing the package

The package can be imported into a Python script using the syntax below.

```
import khorsjx
```

### Initializing the modules

While it is certainly possible to import modules directly (e.g. `from khorsjx import users`), it is recommended that you instead leverage the `init_module()` function as shown below.

```
khorsjx.init_module('content', 'users')
```

In the example above, both the `khorsjx.content` and the `khorsjx.users` modules have been initiated.

---

**Note:** It is not necessary to import the `khorsjx.core` module as it is imported by default.

---

### Establishing the API connection

Before leveraging the API in function calls, you must first establish your connection by providing the base URL for the environment (e.g. `https://community.example.com`) and the username and password for the unfederated service account through which the API calls will be made. This is demonstrated below.

```
base_url = 'https://community.example.com'
credentials = ('adminuser', 'password123!')
khorosjx.core.connect(base_url, credentials)
```

---

**Note:** At this time the library only allow connections using [basic authentication](#), but there are plans to include the ability to leverage [OAuth 2.0](#) in a future release.

---

Once the connection has been established, you can proceed to leverage the library of functions in the various modules as needed.

**See also:**

To facilitate the getting started process, you may want to consider *Using the Helper Utility*.

### 1.2.2 Performing GET Requests

There are two primary ways to perform GET requests against the Core API. The first is to use the `get_data()` function, which avoids the necessity to construct a [query URI](#) but is somewhat restrictive in what can be returned. The second is more flexible but does require the construction of API query URIs.

Both of these methods are addressed in the sections below.

#### Using the `get_data()` function

The `get_data()` function performs GET requests by requiring only two arguments: the API **endpoint** and a **lookup value**.

The API **endpoints**, also known as the API *services* in the [Jive REST API documentation](#), are the different avenues within the API through which data can be retrieved and, in some cases, created and/or modified. The endpoints supported within the khorosjx library are listed below.

abuseReports	acclaim	actions	activities	addOns	announcements
attachments	calendar	checkpoints	collaborations	comments	contents
deletedObjects	dms	events	eventTypes	executeBatch	extprops
extstreamDefs	extstreams	ideaVotes	images	inbox	invites
members	mentions	messages	moderation	oembed	outcomes
pages	people	places	placeTemplateCategories	placeTemplates	placeTopics
profileImages	publications	questions	rsvp	search	sections
securityGroups	shares	slides	stages	statics	streamEntries
streams	tags	tileDefs	tiles	urls	versions
videos	vitals	votes	webhooks		

By default, the respective ID is used as the lookup value for each endpoint. For example, when querying the *people* endpoint the default lookup value is the **User ID**, whereas when querying the *contents* endpoint the default lookup value is the **Content ID**.

The example below shows how you would use the function to retrieve data for content with 12345 as its *Content ID*.

```
api_response = khorosjx.core.get_data('contents', 12345)
```

**Note:** Because *f-strings* are leveraged to construct the query URIs in the function, the lookup value can be supplied as an *integer* or a *string*.

When querying against specific endpoints, other identifiers may be permitted as well. For example, if querying for user data via the *people* endpoint it is possible to supply a *User ID*, *username* or *email* address as demonstrated in the examples below.

```
response_from_id = khorosjx.core.get_data('people', 1234)
response_from_email = khorosjx.core.get_data('people', 'john.doe@example.com', 'email')
response_from_username = khorosjx.core.get_data('people', 'john_doe', 'username')
```

### Optional arguments in the `get_data()` function

There are two optional arguments that may be supplied within the `get_data()` function which can be useful in certain circumstances. Both are addressed below.

### Ignoring exceptions in the `get_data()` function

When utilizing the `get_data()` function multiple times, such as when looping through a list of users, it may be useful to ignore exceptions so that the entire script doesn't fail if data for a single user cannot be returned successfully. This can be done by setting the `ignore_exceptions` argument to `True` in the function arguments, as demonstrated below.

```
api_response = khorosjx.core.get_data('people', 1234, ignore_exceptions=True)
```

Leveraging this option will result in failed responses printing an error message and then either returning the API response (which will include a status code other than 200) or an empty JSON string. (See the next section)

## Returning the `get_data()` response in JSON format

Rather than having to convert the API response to JSON in your script, you can do so within the `get_data()` function by setting the `return_json` argument to `True` as shown in the example below.

```
user_json = khorosjx.core.get_data('people', 1234, return_json=True)
```

---

**Note:** As stated in the previous section, if the `ignore_exceptions` argument is also set to `true` then JSON data will still be returned but will simply be empty.

---

## Using the `get_request_with_retries()` function

If you have a need to perform a less generic GET request then it will likely be better to leverage the `get_request_with_retries()` function instead of `get_data()` so that the specific query URI can be supplied.

The `get_request_with_retries()` function performs the GET query and will retry the operation up to five times to account for any unexpected connection aborts or timeouts, which is known to happen on occasion with the Jive APIs.

This function is demonstrated in the example below.

```
query_uri = f"{base_url}/people/1234/contents?count=100&startIndex=200&fields=@all"
api_response = khorosjx.core.get_request_with_retries(query_uri)
```

---

**Note:** Notice that the `base_url` global variable is being utilized above, which was defined when *establishing the API connection* when *getting started* above.

---

## Returning the `get_request_with_retries()` response in JSON format

Similar to the `get_data()` function, you have the option of returning the response in JSON format by setting the `return_json` argument to `True` as shown in the example below.

```
get_request_json = khorosjx.core.get_request_with_retries(query_uri, return_json=True)
```

### 1.2.3 Performing POST Requests

---

**Todo:** This section is not yet written but will be at a future date.

---

### 1.2.4 Performing PUT Requests

---

**Todo:** This section is not yet written but will be at a future date.

---

## 1.3 Using the Helper Utility

This page provides instructions and examples around leveraging the *Helper Module* (*khorsjx.utils.helper*) which allows a “helper” configuration file to be imported and parsed to facilitate the use of the library (e.g. defining the base URL and API credentials) and defining additional settings.

- *What is a helper?*
- *The configuration file*
  - *API Connection*
    - \* *Supplying credentials in the config file*
    - \* *Retrieving credentials using a script function*
  - *Script Styling*
- *Initializing the Helper*
  - *Global Variables*

### 1.3.1 What is a helper?

When talking about the KhorosJX Python Library, a **helper** is a configuration file which defines various settings that can be leveraged by the library to simplify the *Getting Started* process. It also allows additional features to be enabled in certain circumstances that may be deemed helpful.

### 1.3.2 The configuration file

The Helper configuration file is a [YAML](#) file that follows the formatting shown below.

```
# Helper configuration file for the khorosjx package

# Define how to obtain the connection information
connection:
  base_url: https://community.example.com
  credentials:
    # Uncomment the lines below to provide plaintext credentials
    #
    use_script: no
    username: exampleuser
    password: examplePWD123!

    # Uncomment the lines below to provide the credentials via module
    # for a module and function
    #
    # use_script: yes
    # module_name: jxhelper
    # function_name: get_credentials
    # function_kwargs: username='adminuser'

# Define whether or not to color-code the function output
styling:
  use_console_colors: no
```

---

**Note:** In a future release, you will have the ability to utilize a JSON file rather than YAML if preferred.

---

#### API Connection

The primary functionality of the Helper configuration file is to allow you to define the **Base URL** and the **API credentials** which are both necessary to leverage the library. (See [Basic Usage](#))

This can be done in two different ways:

- Supplying the credentials within the configuration file
- Leveraging another Python module and function to retrieve the credentials.



## Supplying credentials in the config file

If your configuration will be stored in a secure location then you have the option of storing the credentials directly within the configuration file, as demonstrated in the example above. This method requires the fields listed below to be defined, which all reside within the **credentials** subsection under the **connection** section.

### **use\_script**

This field specifies whether or not a script (i.e. module/function) will be used to retrieve the credentials and should be set to **no** in this situation.

### **username**

This field is where you will store the username for the account you will utilize with the Core API.

### **password**

This field is where you will store the corresponding password for the username above.

## Retrieving credentials using a script function

As an alternative to providing credentials directly within the configuration file, you have the option to create a module and function (or leverage an existing module and function) to return the credentials needed for the *khrosjx* library.

This can be done by commenting out (or removing) the section mentioned above in the configuration file for storing credentials within the file itself and instead uncomment the other section which includes these fields:

### **use\_script**

This field specifies whether or not a script (i.e. module/function) will be used to retrieve the credentials and should be set to **yes** in this situation.

### **module\_name**

This field defines the name of the module that must be imported.

### **function\_name**

This field defines the name of the function within the module above that will provide the credentials.

### **function\_kwargs**

This is an optional field where you can define any keyword arguments that must be defined in order to return the appropriate data for the **khrosjx** library.

---

**Note:** The configuration file only supports basic keyword arguments (e.g. *username="apiuser"*) with integer, string or Boolean values. Non-keyword arguments and values that are lists, tuples, dictionaries or other data types are not permitted at this time.

---

For example, let's assume that you have a package entitled *myutils* with an underlying module entitled *jxhelper* which lets you return credentials for a specified user.

```
# Define function to return credentials for a specific user
get_credentials(username):
    # Retrieve the credentials from our SQL server database
    un, pw = get_credentials_from_sql(username)
    return un, pw
```

In this example, if you wanted this function to return credentials for a user with the username *adminuser*, then you would define the values below in the configuration file.

- **use\_script:** no
- **module\_name:** myutils.jxhelper
- **function\_name:** get\_credentials
- **function\_kwargs:** username='adminuser'

### Script Styling

A secondary section in the configuration file address **script styling**, with a single option to enable **console colors**, which is not currently an enabled feature and will be available in a future release. As such, the setting in the configuration file is present for future preparation purposes and current has no effect.

### 1.3.3 Initializing the Helper

Once the configuration file is created, you can initialize it from the primary module with the syntax below.

```
khorosjx.init_helper('/path/to/khorosjx_helper.yml')
```

This function call will not only initialize the Helper settings as global variables to use throughout the library, but will also perform the API connection process as well so this will not need to be performed in a separate function call.

### Global Variables

Once the Helper utility has been initialized, the following global variables become accessible:

#### **helper\_settings**

A dictionary that includes all settings defined by the utility

#### **use\_console\_colors**

A Boolean value that defines whether console colors should be utilized throughout the library. *(Currently has no effect)*

## 1.4 Managing Users

**Warning:** This page is currently in development.

This page has instructions relating to managing users within a KhorosJX (or Jive-n) community, including tasks such as obtaining user information and statistics and other similar operations.

- *Leveraging the Users Module*
- *Obtaining User Information*
  - *Obtain the User ID of a user*
  - *Obtain the username, email address and profile URL of a user*
    - \* *Obtain the username*
    - \* *Obtain the primary email address*
    - \* *Obtain the user profile URL*

### 1.4.1 Leveraging the Users Module

In order to leverage the Users Module (and any other module within the khorosjx library) you must first perform the following steps:

1. Import the high-level **khorosjx** package.
2. Initialize the **users** module.
3. Establish the connection to the environment.

This is demonstrated in the example below.

```
# Import the package
import khorosjx

# Initialize the users module
khorosjx.init_module('users')

# Establish a connection to the environment
environment_url = 'https://community.example.com'
api_username = 'adminuser'
api_password = 'password123$'
khorosjx.core.connect(environment_url, (api_username, api_password))

# Global variables established from the connection above
base_url          # Global variable (str) derived from the environment_url above
api_credentials    # Global variable (tuple) derived from api_username and api_password_
↪ above
```

**Warning:** The API credentials were defined in the example above for the purpose of demonstrating the process. However, it is strongly recommended that you **never** include API credentials in plaintext within scripts or other locations that can potentially be accessed by unauthorized parties.

### See also:

To simplify the initialization process, you should consider leveraging the *Helper Module* ([khorosjx.utils.helper](#)).

## 1.4.2 Obtaining User Information

This section addresses how you can obtain information about a user to use in other administrative functions and/or to obtain for statistical purposes.

### Obtain the User ID of a user

The User ID value is a unique identifier within the KhorosJX / Jive environment which identifies a user and allows the platform—as well as third-party integrations via the API—to take actions against the user in various capacities.

However, sometimes the User ID for a user is not readily available and you only have their email address or username. In these situations, the `get_user_id()` function can be used to retrieve the needed User ID, as shown in the examples below.

```
user_id_from_email = khorosjx.users.get_user_id('john.doe@example.com')
user_id_from_username = khorosjx.users.get_user_id('john_doe', 'username')
```

---

**Note:** You will notice that, because `email` is the default identifier, you must supply `username` as a second argument if you are providing a username.

---

### Obtain the username, email address and profile URL of a user

Similar to how you obtain the User ID of a user, you can also retrieve a user's username, email address and profile URL using simple functions within the *Users Module* ([khorosjx.users](#)). These operations are explained in the following subsections.

#### Obtain the username

Assuming you have the User ID or email address for a user, you can quickly obtain their username via the [khorosjx.users.get\\_username\(\)](#) function, as demonstrated in the examples below.

```
username_from_id = khorosjx.users.get_username(1234)
username_from_email = khorosjx.users.get_username('john.doe@example.com', 'email')
```

---

**Note:** Similar to the [khorosjx.users.get\\_user\\_id\(\)](#) function, you must provide the *lookup type* (e.g. `email`) as a second argument if not querying with the User ID.

You may also notice that when leveraging the User ID, you can provide it as either an integer or a string value and both will be acceptable.

---

## Obtain the primary email address

You can quickly retrieve the primary email address for a user by leveraging the `khorojx.users.get_primary_email()` function and supplying the User ID or username of the user. This is demonstrated below.

```
email_address_from_id = khorojx.users.get_primary_email(1234)
email_address_from_username = khorojx.users.get_primary_email('john_doe', 'username')
```

---

**Note:** As with the other functions above, you must provide the *lookup type* (e.g. `username`) as a second argument if not querying with the User ID. Also in similar fashion, the User ID can be provided as either an integer or a string value and both will be acceptable.

---

## Obtain the user profile URL

When you have the username for a user, it is easy to determine the URL of their respective profile as the URL structure is simply your base URL, the `people` endpoint and then the username. (e.g. `https://community.example.com/people/john_doe`)

However, the `khorojx.users.get_profile_url()` function makes the process even easier by constructing the URL for you, and can even do so when supplied a User ID or email address rather than a username.

All three methods are demonstrated below.

```
profile_url_from_id = khorojx.users.get_profile_url(1234)
profile_url_from_email = khorojx.users.get_profile_url('john.doe@example.com', 'email')
profile_url_from_username = khorojxusers.get_profile_url('john_doe', 'username')
```

---

**Note:** Despite the profile URL being constructed using the username, functions within this module will generally utilize the **User ID** as the primary lookup type as it is the main unique identifier leveraged within the Khoros JX / Jive platform.

---

---

**Todo:** The remainder of this document is still in progress. Please check back later for updates.

---

## 1.5 Primary Modules

This section provides details around the primary modules used in the **khorojx** package, which are listed below.

- *Init Module* (`khorojx`)
- *Core Module* (`khorojx.core`)
- *Admin Module* (`khorojx.admin`)
- *Content Module* (`khorojx.content`)
  - *Base Content Module* (`khorojx.content.base`)
  - *Documents Module* (`khorojx.content.docs`)

- *Events Module* (*khorosjx.content.events*)
  - *Ideas Module* (*khorosjx.content.ideas*)
  - *Threads Module* (*khorosjx.content.threads*)
  - *Videos Module* (*khorosjx.content.videos*)
- *Groups Module* (*khorosjx.groups*)
- *News Module* (*khorosjx.news*)
- ***Places Module*** (*khorosjx.places*)
  - *Base Places Module* (*khorosjx.places.base*)
  - *Blogs Module* (*khorosjx.places.blogs*)
  - *Spaces Module* (*khorosjx.places.spaces*)
- *Spaces Module* (*khorosjx.spaces*)
- *Users Module* (*khorosjx.users*)

### 1.5.1 Init Module (khorosjx)

This module (being the primary `__init__.py` file for the library) contains the functions to initialize the modules and the *Helper Utility*.

**Package**

khorosjx

**Synopsis**

This package includes custom exceptions and a function to call them with specific error messages

**Usage**

```
import khorosjx
```

**Example**

```
khorosjx.init_helper('/home/user/jxhelper.yml')
```

**Created By**

Jeff Shurtliff

**Last Modified**

Jeff Shurtliff

**Modified Date**

24 Mar 2020

`khorosjx.init_helper(file_path, file_type='yaml')`

This function initializes a helper configuration file to define package settings including the API connection.

**Parameters**

- **file\_path** (*str*) – Path to the helper configuration file
- **file\_type** (*str*) – The type of file utilized as the configuration file (Default: `yaml`)

**Returns**

None (Defines global variables and establishes API connection)

**Raises**

`FileNotFoundError`, `khorsjx.errors.exceptions.CredentialsUnpackingError`,  
`khorsjx.errors.exceptions.InvalidHelperArgumentsError`, `khorsjx.errors.exceptions.HelperFunctionNotFoundError`

`khorsjx.init_module(*args)`

This function imports select modules from the library.

**Parameters**

**args** (*str*, *tuple*) – One or more module names to import

**Returns**

None

**Raises**

`ModuleNotFoundError`, `khorsjx.errors.exceptions.KhorosJXError`, `khorsjx.errors.exceptions.InvalidKhorosJXModuleError`

*Return to Top*

## 1.5.2 Core Module (khorsjx.core)

This module contains core functions such as initializing the connection to the API, getting API version information, performing GET and PUT requests, etc.

**Module**

`khorsjx.core`

**Synopsis**

Collection of core functions and tools to work with the Jive Core API v3

**Usage**

`import khorsjx.core` (Imported by default in primary package)

**Example**

```
user_info = khorsjx.core.get_data('people', 'john.doe@example.com',  
    'email')
```

**Created By**

Jeff Shurtliff

**Last Modified**

Jeff Shurtliff

**Modified Date**

23 Sep 2021

`khorsjx.core.connect(base_api_url, credentials)`

This function establishes the connection information for performing Core API queries.

**Parameters**

- **base\_api\_url** (*str*) – The base URL (e.g. <https://community.example.com>) for for environment
- **credentials** (*tuple*) – The username and password of the account to perform the API queries

#### Returns

None

`khorosjx.core.delete(uri, return_json=False, verify_ssl=True)`

This function performs a DELETE request against the Core API.

Changed in version 3.2.0: The query URL is now made into an absolute URL as necessary before performing the API request.

Changed in version 2.6.0: Added the `verify_ssl` argument.

#### Parameters

- **uri** (*str*) – The URI against which the DELETE request will be issued
- **return\_json** (*bool*) – Determines whether or not the response should be returned in JSON format (Default: `False`)
- **verify\_ssl** (*bool*) – Determines if API calls should verify SSL certificates (True by default)

#### Returns

The API response from the DELETE request (optionally in JSON format)

`khorosjx.core.ensure_absolute_url(query_url)`

This function adds the base URL to the beginning of a query URL if not already present.

New in version 3.2.0.

#### Parameters

**query\_url** (*str*) – The query URL that will be utilized in an API request

#### Returns

The query URL that includes a top-level domain

#### Raises

`TypeError`

`khorosjx.core.get_api_info(api_filter='none', verify_ssl=True)`

This function obtains the API version information for a Jive environment.

Changed in version 2.6.0: Added the `verify_ssl` argument.

#### Parameters

- **api\_filter** (*str*) – A filter to return a subset of API data (e.g. `v3`, `platform`, `sso`, etc.)
- **verify\_ssl** (*bool*) – Determines if API calls should verify SSL certificates (True by default)

#### Returns

API information in JSON, string or list format depending on the filter

`khorosjx.core.get_api_version(api_name='v3', verify_ssl=True)`

This function obtains, parses and returns the current version of one of the Jive Core APIs.

Changed in version 3.1.0: Refactored the function to be more efficient.

Changed in version 2.6.0: Added the `verify_ssl` argument.

#### Parameters

- **api\_name** (*str*) – The name of the API for which the version should be returned (Default: `v3`)



- **verify\_ssl** (*bool*) – Determines if API calls should verify SSL certificates (True by default)

**Returns**

The API version in major.minor notation (e.g. 3.15) in string format

`khorosjx.core.get_base_url(api_base=True)`

This function returns the base URL of the environment with or without the `/api/core/v3/` path appended.

Changed in version 3.1.0: Refactored the function to properly utilize the `base_url` global variable.

**Parameters**

- **api\_base** (*bool*) – Determines if the `/api/core/v3/` path should be appended (True by default)

**Returns**

The base URL for the Khoros JX or Jive-n environment

`khorosjx.core.get_connection_info()`

This function returns the connection information (Base URL and API credentials) to use in other modules.

**Returns**

Base URL in string format and API credentials within a tuple

`khorosjx.core.get_data(endpoint, lookup_value, identifier='id', return_json=False, ignore_exceptions=False, all_fields=False, verify_ssl=True)`

This function returns data for a specific API endpoint.

Changed in version 3.1.0: Fixed how the `query_url` variable is defined to proactively avoid raising any `NameError` exceptions.

Changed in version 2.6.0: Added the `verify_ssl` argument.

**Parameters**

- **endpoint** (*str*) – The API endpoint against which to request data (e.g. `people`, `contents`, etc.)
- **lookup\_value** (*int*, *str*) – The value to use to look up the endpoint data
- **identifier** (*str*) – The type of lookup value used to look up the endpoint data (Default: `id`)
- **return\_json** (*bool*) – Determines if the data should be returned in default or JSON format (Default: `False`)
- **ignore\_exceptions** (*bool*) – Determines whether or not exceptions should be ignored (Default: `False`)
- **all\_fields** (*bool*) – Determines whether or not the `fields=@all` query should be included (Default: `False`)
- **verify\_ssl** (*bool*) – Determines if API calls should verify SSL certificates (True by default)

**Returns**

The API response either as a requests response or in JSON format depending on the `return_json` value

**Raises**

`khorosjx.errors.exceptions.GETRequestError`

`khorojx.core.get_fields_from_api_response(json_data, dataset, return_fields=None, quiet=False)`

This function parses and retrieves fields from an API response from a specific dataset.

Changed in version 3.1.0: Changed the default `return_fields` value to `None` and adjusted the function accordingly.

Changed in version 2.6.0: Added conditional to ensure `quiet` is `False` before calling the `stderr` print statement.

Changed in version 2.5.3: Fixed the `email.value` filter and added the optional `quiet` argument.

#### Parameters

- **json\_data** (*dict*) – The JSON data from an API response
- **dataset** (*str*) – The nickname of a dataset from which fields should be retrieved (e.g. `people`, `group_admins`)
- **return\_fields** (*list*, *None*) – The fields that should be returned from the API response (Default: all fields in dataset)
- **quiet** (*bool*) – Silences any errors about being unable to locate API fields (`False` by default)

#### Returns

A dictionary with the field names and corresponding values

#### Raises

`khorojx.errors.exceptions.InvalidDatasetError`

`khorojx.core.get_paginated_results(query, response_data_type, start_index=0, filter_info=(), query_all=True, return_fields=None, ignore_exceptions=False, quiet=False, verify_ssl=True)`

This function performs a GET request for a single paginated response up to 100 records.

Changed in version 3.1.0: Changed the default `return_fields` value to `None` and adjusted the function accordingly.

Changed in version 2.6.0: Added the `verify_ssl` argument.

#### Parameters

- **query** (*str*) – The API query without the query string
- **response\_data\_type** (*str*) – The dataset of fields that will be in the API response (e.g. `group_members`)
- **start\_index** (*int*, *str*) – The `startIndex` value in the API query string (`0` by default)
- **filter\_info** (*tuple*, *list*) – A tuple of list of tuples containing the filter element and criteria (Optional)
- **query\_all** (*bool*) – Determines if `fields=@all` filter should be included in the query string (Default: `True`)
- **return\_fields** (*list*, *None*) – The fields that should be returned from the API response (Default: all fields in dataset)
- **ignore\_exceptions** (*bool*) – Determines whether or not exceptions should be ignored (Default: `False`)
- **quiet** (*bool*) – Silences any errors about being unable to locate API fields (`False` by default)
- **verify\_ssl** (*bool*) – Determines if API calls should verify SSL certificates (`True` by default)

**Returns**

The queried data as a list comprised of dictionaries

**Raises**

*[khorosjx.errors.exceptions.GETRequestError](#)*

`khorosjx.core.get_platform_version(verify_ssl=True)`

This function obtains the current Khoros JX (or Jive) version for an environment.

Changed in version 2.6.0: Added the `verify_ssl` argument.

**Parameters**

**`verify_ssl`** (*bool*) – Determines if API calls should verify SSL certificates (True by default)

**Returns**

The full platform version in string format (e.g. 2018.22.0.0\_jx)

`khorosjx.core.get_query_url(pre_endpoint, asset_id="", post_endpoint="")`

This function constructs an API query URL excluding any query strings.

Changed in version 3.1.0: Refactored the function to be more efficient.

**Parameters**

- **`pre_endpoint`** (*str*) – The endpoint portion of the URL preceding any ID numbers (e.g. places)
- **`asset_id`** (*str*, *int*) – The ID for an asset (e.g. User ID, Browse ID for a space/blog, etc.)
- **`post_endpoint`** (*str*) – Any remaining endpoints following the ID number (e.g. contents)

**Returns**

The fully structured query URL

`khorosjx.core.get_request_with_retries(query_url, return_json=False, verify_ssl=True)`

This function performs a GET request with a total of 5 retries in case of timeouts or connection issues.

Changed in version 3.2.0: The query URL is now made into an absolute URL as necessary before performing the API request.

Changed in version 3.1.0: Refactored the function to be more efficient.

Changed in version 2.6.0: Added the `verify_ssl` argument.

**Parameters**

- **`query_url`** (*str*) – The URI to be queried
- **`return_json`** (*bool*) – Determines whether or not the response should be returned in JSON format (Default: False)
- **`verify_ssl`** (*bool*) – Determines if API calls should verify SSL certificates (True by default)

**Returns**

The API response from the GET request (optionally in JSON format)

**Raises**

*[ValueError](#)*, *[TypeError](#)*, *[khorosjx.errors.exceptions.APIConnectionError](#)*

`khorojx.core.post_request_with_retries(url, json_payload, verify_ssl=True)`

This function performs a POST request with a total of 5 retries in case of timeouts or connection issues.

Changed in version 3.2.0: The query URL is now made into an absolute URL as necessary before performing the API request.

Changed in version 2.6.0: Added the `verify_ssl` argument.

#### Parameters

- **url** (*str*) – The URI to be queried
- **json\_payload** (*dict*) – The payload for the POST request in JSON format
- **verify\_ssl** (*bool*) – Determines if API calls should verify SSL certificates (True by default)

#### Returns

The API response from the POST request

#### Raises

`ValueError`, `khorojx.errors.exceptions.APIConnectionError`, `khorojx.errors.exceptions.POSTRequestError`

`khorojx.core.put_request_with_retries(url, json_payload, verify_ssl=True)`

This function performs a PUT request with a total of 5 retries in case of timeouts or connection issues.

Changed in version 3.2.0: The query URL is now made into an absolute URL as necessary before performing the API request.

Changed in version 2.6.0: Added the `verify_ssl` argument.

#### Parameters

- **url** (*str*) – The URI to be queried
- **json\_payload** (*dict*) – The payload for the PUT request in JSON format
- **verify\_ssl** (*bool*) – Determines if API calls should verify SSL certificates (True by default)

#### Returns

The API response from the PUT request

#### Raises

`ValueError`, `khorojx.errors.exceptions.APIConnectionError`, `khorojx.errors.exceptions.PUTRequestError`

`khorojx.core.set_base_url(domain_url, version=3, protocol='https', return_url=True)`

This function gets the base URL for API calls when supplied with a domain URL. (e.g. `community.example.com`)

Changed in version 3.2.0: Added the `return_url` parameter to determine if the base URL should be returned by the function.

#### Parameters

- **domain\_url** (*str*) – The domain URL of the environment, with or without the http/https prefix
- **version** (*int*) – The version of the REST API to utilize (Default: 3)
- **protocol** (*str*) – The protocol to leverage for the domain prefix if not already supplied (Default: https)

- **return\_url** (*bool*) – Determines if the base URL should be returned by the function (True by default)

**Returns**

The base URL for API calls in string format (e.g. `https://community.example.com/api/core/v3`)

**Raises**

`TypeError`, `ValueError`

`khorosjx.core.set_credentials(credentials)`

This function defines the Core API credentials as global variables and validates them.

Changed in version 3.1.0: Parenthesis were added to the exception classes and utilized the `isinstance()` builtin.

**Parameters**

**credentials** (*tuple*) – The username and password for the account that will be utilizing the Core API

**Returns**

None

**Raises**

`khorosjx.errors.exceptions.IncompleteCredentialsError`, `khorosjx.errors.exceptions.CredentialsUnpackingError`, `khorosjx.errors.exceptions.WrongCredentialTypeError`

`khorosjx.core.verify_connection()`

This function verifies that the base URL and API credentials have been defined.

Changed in version 3.1.0: Refactored the function to be more pythonic and to avoid depending on a try/except block.

**Returns**

None

**Raises**

`khorosjx.errors.exceptions.KhorosJXError`, `khorosjx.errors.exceptions.NoCredentialsError`

*[Return to Top](#)*

## 1.5.3 Admin Module (`khorosjx.admin`)

This module contains administrative functions that would only be performed by a platform administrator or a community manager.

**Module**

`khorosjx.admin`

**Synopsis**

Collection of functions relating to security groups

**Usage**

```
from khorosjx import admin
```

**Example**

Coming Soon

**Created By**

Jeff Shurtliff

**Last Modified**

Jeff Shurtliff

**Modified Date**

22 Sep 2021

**khrosjx.admin.retrieve\_connection\_info()**

This function initializes and defines the global variables for the connection information.

Changed in version 3.1.0: Refactored the function to be more efficient.

**Returns**

None

**Raises**

*khrosjx.errors.exceptions.KhorosJXError*, *khrosjx.errors.exceptions.NoCredentialsError*

**khrosjx.admin.verify\_core\_connection()**

This function verifies that the core connection information (Base URL and API credentials) has been defined.

Changed in version 3.1.0: Refactored the function to be more pythonic and to avoid depending on a try/except block.

**Returns**

None

**Raises**

*khrosjx.errors.exceptions.KhorosJXError*, *khrosjx.errors.exceptions.NoCredentialsError*

*[Return to Top](#)*

## 1.5.4 Content Module (khrosjx.content)

This module contains functions relating to content within the platform which allows for creating, editing and managing content such as documents, ideas, videos, etc.

**Module**

khrosjx.content

**Synopsis**

Collection of functions relating to content

**Usage**

```
from khrosjx import content
```

**Example**

```
content_id = content.base.get_content_id(url, 'document')
```

**Created By**

Jeff Shurtliff

**Last Modified**

Jeff Shurtliff

**Modified Date**

24 Mar 2020

`khrosjx.content.get_content_id(url, content_type='document')`

This function obtains the Content ID for a particular content asset. (Supports all but blog posts)

Deprecated since version 2.0.0: The `khrosjx.content.base.get_content_id()` function should be used.

**Parameters**

- **url** (*str*) – The URL to the content
- **content\_type** (*str*) – The content type for the URL for which to obtain the Content ID (Default: document)

**Returns**

The Content ID for the content URL

**Raises**

`ValueError`

`khrosjx.content.get_document_attachments(lookup_value, lookup_type='doc_id',  
 return_dataframe=False)`

This function retrieves information on any attachments associated with a document.

Deprecated since version 2.0.0: The `khrosjx.content.docs.get_document_attachments()` function should be used.

**Parameters**

- **lookup\_value** (*str*, *int*) – The value with which to look up the document
- **lookup\_type** (*str*) – Identifies the type of lookup value that has been provided (Default: doc\_id)
- **return\_dataframe** (*bool*) – Determines whether or not a pandas dataframe should be returned

**Returns**

A list, dictionary or pandas dataframe depending on the number of attachments and/or function arguments

**Raises**

`khrosjx.errors.exceptions.GETRequestError`, `khrosjx.errors.exceptions.InvalidDatasetError`, `khrosjx.errors.exceptions.InvalidLookupTypeError`, `khrosjx.errors.exceptions.LookupMismatchError`

`khrosjx.content.get_document_info(lookup_value, lookup_type='doc_id', return_fields=[],  
 ignore_exceptions=False)`

This function obtains the group information for a given document.

Deprecated since version 2.0.0: The `khrosjx.content.docs.get_document_info()` function should be used.

**Parameters**

- **lookup\_value** (*int*, *str*) – The value with which to look up the document
- **lookup\_type** (*str*) – Identifies the type of lookup value that has been provided (Default: doc\_id)

- **return\_fields** (*list*) – Specific fields to return if not all of the default fields are needed (Optional)
- **ignore\_exceptions** (*bool*) – Determines whether or not exceptions should be ignored (Default: False)

#### Returns

A dictionary with the group information

#### Raises

*khrosjx.errors.exceptions.GETRequestError*, *khrosjx.errors.exceptions.InvalidDatasetError*, *khrosjx.errors.exceptions.InvalidLookupTypeError*, *khrosjx.errors.exceptions.LookupMismatchError*

`khrosjx.content.override_doc_body(url, body_html, minor_edit=True, ignore_exceptions=False)`

This function overwrites the body of a document with new HTML content.

Deprecated since version 2.0.0: The *khrosjx.content.docs.override\_doc\_body()* function should be used.

#### Parameters

- **url** (*str*) – The URL of the document to be updated
- **body\_html** – The new HTML body to replace the existing document body
- **minor\_edit** (*bool*) – Determines whether the *Minor Edit* flag should be set (Default: True)
- **ignore\_exceptions** (*bool*) – Determines whether or not exceptions should be ignored (Default: False)

#### Returns

The response of the PUT request used to update the document

#### Raises

*khrosjx.errors.exceptions.ContentPublishError*

*[Return to Top](#)*

## Base Content Module (*khrosjx.content.base*)

This module contains the core functions relating to content within the platform which allows for creating, editing and managing content.

#### Module

`khrosjx.content.base`

#### Synopsis

Collection of core functions relating to content

#### Usage

```
import khrosjx.content.base as content_core
```

#### Example

```
content_id = content_core.get_content_id(url, 'document')
```

#### Created By

Jeff Shurtliff



**Last Modified**

Jeff Shurtliff

**Modified Date**

22 Sep 2021

`khorosjx.content.base.get_content_id(url, content_type='document', verify_ssl=True)`

This function obtains the Content ID for a particular content asset. (Supports all but blog posts)

Changed in version 3.1.0: Made some minor syntax improvements.

Changed in version 2.6.0: Added the `verify_ssl` argument.

**Parameters**

- **url** (*str*) – The URL to the content
- **content\_type** (*str*) – The content type for the URL for which to obtain the Content ID (Default: `document`)
- **verify\_ssl** (*bool*) – Determines if API calls should verify SSL certificates (True by default)

**Returns**

The Content ID for the content URL

**Raises**

`ValueError`, `khorosjx.errors.exceptions.ContentNotFoundError`

`khorosjx.content.base.get_paginated_content(endpoint, query_string="", start_index=0, dataset="", all_fields=False, return_fields=None, ignore_exceptions=False)`

This function returns paginated content information. (Up to 100 records at a time)

Changed in version 3.1.0: Changed the default `return_fields` value to `None` and adjusted the function accordingly.

**Parameters**

- **endpoint** (*str*) – The full endpoint without preceding slash (e.g. `securityGroups`, `people/email/user_email`)
- **query\_string** (*str*) – Any query strings to apply (without preceding `?`) excluding count and `startIndex`
- **start\_index** (*int*, *str*) – The `startIndex` API value
- **dataset** (*str*) – Defines the type of data returned in the API response (e.g. `security_group`, `people`, etc.)
- **all\_fields** (*bool*) – Determines if the `fields=@all` parameter should be passed in the query
- **return\_fields** (*list*, *None*) – Specific fields to return if not all of the default fields are needed (Optional)
- **ignore\_exceptions** (*bool*) – Determines whether or not exceptions should be ignored (Default: `False`)

**Returns**

A list of dictionaries containing information for each group in the paginated query

`khorosjx.content.base.retrieve_connection_info()`

This function initializes and defines the global variables for the connection information.

Changed in version 3.1.0: Refactored the function to be more efficient.

**Returns**

None

**Raises**

`khorosjx.errors.exceptions.KhorosJXError`, `khorosjx.errors.exceptions.NoCredentialsError`

`khorosjx.content.base.verify_core_connection()`

This function verifies that the core connection information (Base URL and API credentials) has been defined.

Changed in version 3.1.0: Refactored the function to be more pythonic and to avoid depending on a try/except block.

**Returns**

None

**Raises**

`khorosjx.errors.exceptions.KhorosJXError`, `khorosjx.errors.exceptions.NoCredentialsError`

*[Return to Top](#)*

## Documents Module (`khorosjx.content.docs`)

This module contains functions specific to handling documents.

**Module**

`khorosjx.content.docs`

**Synopsis**

Collection of functions relating to documents (e.g. <https://community.example.com/docs/DOC-1234>)

**Usage**

```
from khorosjx.content import docs
```

**Example**

```
content_id = docs.get_content_id(url)
```

**Created By**

Jeff Shurtliff

**Last Modified**

Jeff Shurtliff

**Modified Date**

22 Sep 2021

`khorosjx.content.docs.create_document(subject, body, place_id, categories=None, tags=None, verify_ssl=True)`

This function creates a new document.

Changed in version 3.1.0: Changed the default `categories` and `tags` values to `None` and adjusted the function accordingly.

Changed in version 2.6.0: Added the `verify_ssl` argument.

#### Parameters

- **subject** (*str*) – The title/subject of the document
- **body** (*str*) – The raw HTML making up the document body
- **place\_id** (*int*, *str*) – The Place ID (aka Browse ID) of the space where the document should reside
- **categories** (*list*, *None*) – Any categories associated with the document (none by default)
- **tags** (*list*, *None*) – Any tags associated with the document (none by default)
- **verify\_ssl** (*bool*) – Determines if API calls should verify SSL certificates (True by default)

#### Returns

The API response from the POST request for the document creation

#### Raises

*TypeError*, *khorosjx.errors.exceptions.POSTRequestError*

`khorosjx.content.docs.delete_document(lookup_value, lookup_type='content_id', return_json=False)`

This function deletes a document.

Changed in version 3.1.0: Parenthesis were added to the exception classes and the function was refactored to be more efficient.

#### Parameters

- **lookup\_value** (*str*, *int*) – The value with which to identify the document.
- **lookup\_type** (*str*) – Identifies the value as a `content_id` (default), `doc_id` or `url`
- **return\_json** (*bool*) – Determines if the API response should be returned in JSON format (False by default)

#### Returns

The API response for the DELETE request

#### Raises

*khorosjx.errors.exceptions.InvalidLookupTypeError*

`khorosjx.content.docs.get_content_id(lookup_value, lookup_type='url', verify_ssl=True)`

This function obtains the Content ID for a particular document.

Changed in version 2.6.0: Added the `verify_ssl` argument.

#### Parameters

- **lookup\_value** (*str*) – The URL of the document
- **lookup\_type** (*str*) – The type of value is being used for lookup (`url` by default)
- **verify\_ssl** (*bool*) – Determines if API calls should verify SSL certificates (True by default)

#### Returns

The Content ID for the document

**Raises**

`ValueError`, `khorsjx.errors.exceptions.ContentNotFoundError`, `khorsjx.errors.exceptions.InvalidLookupTypeError`

```
khorsjx.content.docs.get_document_attachments(lookup_value, lookup_type='doc_id',
                                              return_dataframe=False)
```

This function retrieves information on any attachments associated with a document.

**Parameters**

- **lookup\_value** (*str*, *int*) – The value with which to look up the document
- **lookup\_type** (*str*) – Identifies the type of lookup value that has been provided (Default: `doc_id`)
- **return\_dataframe** (*bool*) – Determines whether or not a pandas dataframe should be returned

**Returns**

A list, dictionary or pandas dataframe depending on the number of attachments and/or function arguments

**Raises**

`khorsjx.errors.exceptions.GETRequestError`, `khorsjx.errors.exceptions.InvalidDatasetError`, `khorsjx.errors.exceptions.InvalidLookupTypeError`, `khorsjx.errors.exceptions.LookupMismatchError`

```
khorsjx.content.docs.get_document_info(lookup_value, lookup_type='doc_id', return_fields=None,
                                       ignore_exceptions=False, verify_ssl=True)
```

This function obtains the group information for a given document.

Changed in version 2.6.0: Added the `verify_ssl` argument.

**Parameters**

- **lookup\_value** (*int*, *str*) – The value with which to look up the document
- **lookup\_type** (*str*) – Identifies the type of lookup value that has been provided (Default: `doc_id`)
- **return\_fields** (*list*, *None*) – Specific fields to return if not all of the default fields are needed (Optional)
- **ignore\_exceptions** (*bool*) – Determines whether or not exceptions should be ignored (Default: `False`)
- **verify\_ssl** (*bool*) – Determines if API calls should verify SSL certificates (True by default)

**Returns**

A dictionary with the group information

**Raises**

`khorsjx.errors.exceptions.GETRequestError`, `khorsjx.errors.exceptions.InvalidDatasetError`, `khorsjx.errors.exceptions.InvalidLookupTypeError`, `khorsjx.errors.exceptions.LookupMismatchError`

```
khorsjx.content.docs.get_url_for_id(doc_id)
```

This function constructs a full URL for a given Document ID.

**Parameters**

**doc\_id** (*int*, *str*) – The Document ID with which to construct the URL

**Returns**

The fully constructed URL for the document (e.g. <https://community.example.com/docs/DOC-1234>)

**Raises**

`TypeError`, `ValueError`

```
khorosjx.content.docs.overwrite_doc_body(url, body_html, minor_edit=True, ignore_exceptions=False,
                                           verify_ssl=True)
```

This function overwrites the body of a document with new HTML content.

Changed in version 2.6.0: Added the `verify_ssl` argument.

**Parameters**

- **url** (*str*) – The URL of the document to be updated
- **body\_html** – The new HTML body to replace the existing document body
- **minor\_edit** (*bool*) – Determines whether the *Minor Edit* flag should be set (Default: True)
- **ignore\_exceptions** (*bool*) – Determines whether or not exceptions should be ignored (Default: False)
- **verify\_ssl** (*bool*) – Determines if API calls should verify SSL certificates (True by default)

**Returns**

The response of the PUT request used to update the document

**Raises**

`khorosjx.errors.exceptions.ContentPublishError`

```
khorosjx.content.docs.retrieve_connection_info()
```

This function initializes and defines the global variables for the connection information.

Changed in version 3.1.0: Refactored the function to be more efficient.

**Returns**

None

**Raises**

`khorosjx.errors.exceptions.KhorosJXError`, `khorosjx.errors.exceptions.NoCredentialsError`

```
khorosjx.content.docs.verify_core_connection()
```

This function verifies that the core connection information (Base URL and API credentials) has been defined.

Changed in version 3.1.0: Refactored the function to be more pythonic and to avoid depending on a try/except block.

**Returns**

None

**Raises**

`khorosjx.errors.exceptions.KhorosJXError`, `khorosjx.errors.exceptions.NoCredentialsError`

*[Return to Top](#)*

## Events Module (`khorsjx.content.events`)

This module contains functions specific to handling events.

### Module

`khorsjx.content.events`

### Synopsis

Collection of functions relating to events (e.g. <https://community.example.com/event/1234>)

### Usage

```
from khorsjx.content import events
```

### Example

```
content_id = events.get_content_id(url)
```

### Created By

Jeff Shurtliff

### Last Modified

Jeff Shurtliff

### Modified Date

22 Sep 2021

`khorsjx.content.events.get_content_id(url)`

This function obtains the Content ID for a particular event.

### Parameters

**url** (*str*) – The URL of the event

### Returns

The Content ID for the event

### Raises

`ValueError`

`khorsjx.content.events.retrieve_connection_info()`

This function initializes and defines the global variables for the connection information.

Changed in version 3.1.0: Refactored the function to be more efficient.

### Returns

None

### Raises

`khorsjx.errors.exceptions.KhorosJXError`, `khorsjx.errors.exceptions.NoCredentialsError`

`khorsjx.content.events.verify_core_connection()`

This function verifies that the core connection information (Base URL and API credentials) has been defined.

Changed in version 3.1.0: Refactored the function to be more pythonic and to avoid depending on a try/except block.

### Returns

None

### Raises

`khorsjx.errors.exceptions.KhorosJXError`, `khorsjx.errors.exceptions.NoCredentialsError`

*[Return to Top](#)*

## Ideas Module (`khorosjx.content.ideas`)

This module contains functions specific to handling ideas.

### Module

`khorosjx.content.ideas`

### Synopsis

Collection of functions relating to ideas (e.g. <https://community.example.com/idea/1234>)

### Usage

```
from khorosjx.content import ideas
```

### Example

```
content_id = ideas.get_content_id(url)
```

### Created By

Jeff Shurtliff

### Last Modified

Jeff Shurtliff

### Modified Date

22 Sep 2021

`khorosjx.content.ideas.get_content_id(url)`

This function obtains the Content ID for a particular idea.

### Parameters

**url** (*str*) – The URL of the idea

### Returns

The Content ID for the idea

### Raises

`ValueError`

`khorosjx.content.ideas.get_ideas_for_space(browse_id, return_type='list', ignore_exceptions=False)`

This function retrieves ideas for a given space.

### Parameters

- **browse\_id** (*str*, *int*) – The Browse ID of the space to be queried
- **return\_type** (*str*) – Determines if the data should be returned as a `list` (default) or a `dataframe`
- **ignore\_exceptions** (*bool*) – Determines if exceptions encountered should be ignored (False by default)

### Returns

The ideas for the given space in a `list` or a `pandas dataframe`

`khorosjx.content.ideas.retrieve_connection_info()`

This function initializes and defines the global variables for the connection information.

Changed in version 3.1.0: Refactored the function to be more efficient.

### Returns

`None`

**Raises**

`khorojx.errors.exceptions.KhorosJXError`, `khorojx.errors.exceptions.NoCredentialsError`

`khorojx.content.ideas.verify_core_connection()`

This function verifies that the core connection information (Base URL and API credentials) has been defined.

Changed in version 3.1.0: Refactored the function to be more pythonic and to avoid depending on a try/except block.

**Returns**

None

**Raises**

`khorojx.errors.exceptions.KhorosJXError`, `khorojx.errors.exceptions.NoCredentialsError`

*[Return to Top](#)*

## Threads Module (`khorojx.content.threads`)

This module contains functions specific to handling community discussion and question threads.

**Module**

`khorojx.content.threads`

**Synopsis**

Collection of functions relating to discussion and question threads

**Usage**

```
from khorojx.content import threads
```

**Example**

```
content_id = threads.get_content_id(url)
```

**Created By**

Jeff Shurtliff

**Last Modified**

Jeff Shurtliff

**Modified Date**

22 Sep 2021

`khorojx.content.threads.get_content_id(url)`

This function obtains the Content ID for a particular discussion or question thread.

**Parameters**

**url** (*str*) – The URL of the thread

**Returns**

The Content ID for the thread

**Raises**

`ValueError`



**khorosjx.content.threads.retrieve\_connection\_info()**

This function initializes and defines the global variables for the connection information.

Changed in version 3.1.0: Refactored the function to be more efficient.

**Returns**

None

**Raises**

`khorosjx.errors.exceptions.KhorosJXError`, `khorosjx.errors.exceptions.NoCredentialsError`

**khorosjx.content.threads.verify\_core\_connection()**

This function verifies that the core connection information (Base URL and API credentials) has been defined.

Changed in version 3.1.0: Refactored the function to be more pythonic and to avoid depending on a try/except block.

**Returns**

None

**Raises**

`khorosjx.errors.exceptions.KhorosJXError`, `khorosjx.errors.exceptions.NoCredentialsError`

*[Return to Top](#)*

## Videos Module (`khorosjx.content.videos`)

This module contains functions specific to handling videos.

**Module**

`khorosjx.content.videos`

**Synopsis**

Collection of functions relating to videos (e.g. <https://community.example.com/videos/1234>)

**Usage**

```
from khorosjx.content import videos
```

**Example**

```
content_id = videos.get_content_id(url)
```

**Created By**

Jeff Shurtliff

**Last Modified**

Jeff Shurtliff

**Modified Date**

22 Sep 2021

**khorosjx.content.videos.check\_if\_embedded(*lookup\_value*, *lookup\_type*='content\_id')**

This function checks to see if a native video is embedded in a document or just a standalone video.

**Parameters**

- **lookup\_value** (*str*, *int*) – The value with which to look up the video

- **lookup\_type** (*str*) – Defines if the lookup value is a `content_id` (default), `video_id` or `url`

**Returns**

A Boolean value indicating whether or not the video is embedded

**Raises**

*khorojx.errors.exceptions.InvalidLookupTypeError*

`khorojx.content.videos.download_video(video_url, output_path, output_name='', default_type='mp4', verbose=False)`

This function downloads a video file when provided its URLs and an output name and location.

**Parameters**

- **video\_url** (*str*) – The direct download URL with its accompanying authorization token
- **output\_path** (*str*) – The full path to the directory where the video file should be downloaded
- **output\_name** (*str*) – The name of the output file (e.g. `video.mp4`)
- **default\_type** (*str*) – Defines a default file extension (`mp4` by default) if an extension is not found
- **verbose** (*bool*) – Determines if verbose console output should be displayed (`False` by default)

**Returns**

None

`khorojx.content.videos.find_video_attachments(document_attachments)`

This function identifies any attached videos in a collection of document attachments.

**Parameters**

**document\_attachments** (*list*, *dict*) – Attachments associated with a document

**Returns**

A list of dictionaries containing info on any video attachments

`khorojx.content.videos.get_content_id(lookup_value, lookup_type='url')`

This function obtains the Content ID for a particular video.

**Parameters**

- **lookup\_value** (*str*, *int*) – The URL or Video ID associated with the video
- **lookup\_type** – Defines the type of value given to identify the video as `url` (default) or `id`

**Returns**

The Content ID for the video

**Raises**

*ValueError*, *khorojx.errors.exceptions.InvalidLookupTypeError*

`khorojx.content.videos.get_native_videos_for_space(browse_id, return_fields=[], return_type='list', ignore_exceptions=False)`

This function returns information on all native (i.e. non-attachment and not third party) for a given space.

**Parameters**

- **browse\_id** (*int*, *str*) – The Browse ID associated with the space

- **return\_fields** (*list*) – Specific fields to return if not all of the default fields are needed (Optional)
- **return\_type** (*str*) – Determines if the data should be returned in a list or a pandas dataframe (Default: list)
- **ignore\_exceptions** (*bool*) – Determines whether or not exceptions should be ignored (Default: False)

**Returns**

A list of dictionaries or a dataframe containing information for each group

**Raises**

*khorosjx.errors.exceptions.InvalidDatasetError*

`khorosjx.content.videos.get_video_dimensions(lookup_value, lookup_type='content_id')`

This function returns the dimensions of a given video.

**Parameters**

- **lookup\_value** (*str*, *int*) – The value with which to look up the video
- **lookup\_type** (*str*) – Defines if the lookup value is a `content_id` (default), `video_id` or `url`

**Returns**

The video dimensions in string format

**Raises**

*khorosjx.errors.exceptions.InvalidLookupTypeError*

`khorosjx.content.videos.get_video_id(lookup_value, lookup_type='url')`

This function returns the video ID for a video when given its URL or a Content ID.

**Parameters**

- **lookup\_value** (*str*, *int*) – The URL or Content ID with which to identify the video
- **lookup\_type** (*str*) – Defines the type of value given to identify the video as `url` (default) or `content_id`

**Returns**

The Video ID for the video

**Raises**

*khorosjx.errors.exceptions.InvalidLookupTypeError*

`khorosjx.content.videos.get_video_info(lookup_value, lookup_type='content_id')`

This function retrieves information about a given video in JSON format.

**Parameters**

- **lookup\_value** (*str*, *int*) – The value with which to look up the video
- **lookup\_type** (*str*) – Defines whether the lookup value is a `content_id` (default), `video_id` or `url`

**Returns**

The video information in JSON format

**Raises**

*khorosjx.errors.exceptions.InvalidLookupTypeError*

### `khrosjx.content.videos.retrieve_connection_info()`

This function initializes and defines the global variables for the connection information.

Changed in version 3.1.0: Refactored the function to be more efficient.

#### Returns

None

#### Raises

`khrosjx.errors.exceptions.KhorosJXError`, `khrosjx.errors.exceptions.NoCredentialsError`

### `khrosjx.content.videos.verify_core_connection()`

This function verifies that the core connection information (Base URL and API credentials) has been defined.

Changed in version 3.1.0: Refactored the function to be more pythonic and to avoid depending on a try/except block.

#### Returns

None

#### Raises

`khrosjx.errors.exceptions.KhorosJXError`, `khrosjx.errors.exceptions.NoCredentialsError`

[Return to Top](#)

## 1.5.5 Groups Module (`khrosjx.groups`)

This module contains functions for working with security groups (and eventually social groups) such as obtaining and managing group membership.

#### Module

`khrosjx.groups`

#### Synopsis

Collection of functions relating to security groups

#### Usage

```
from khrosjx import groups
```

#### Example

```
group_info = groups.get_group_info(1051)
```

#### Created By

Jeff Shurtliff

#### Last Modified

Jeff Shurtliff

#### Modified Date

22 Sep 2021

```
khrosjx.groups.add_user_to_group(group_id, user_value, lookup_type='id', return_mode='none',  
                                print_results=True, ignore_exceptions=True)
```

This function adds a user to a security group.

Changed in version 3.1.0: Parenthesis were added to the exception classes and the function was refactored to be more efficient.

#### Parameters

- **group\_id** (*int*, *str*) – The Group ID of the security group to which the user should be added
- **user\_value** (*int*, *str*) – The value with which to look up the user (e.g. User ID, email address)
- **lookup\_type** (*str*) – Defines whether the user value is a User ID or an email address (Default: id)
- **return\_mode** (*str*) – Determines what—if anything—should be returned by the function (Default: none)
- **print\_results** (*bool*) – Determines whether or not all results (including success messages) should be printed onscreen
- **ignore\_exceptions** (*bool*) – Determines whether or not exceptions should be ignored (Default: True)

#### Returns

The resulting status code, a Boolean value indicating the success of the operation, or nothing

#### Raises

*[khorosjx.errors.exceptions.POSTRequestError](#)*

```
khorosjx.groups.check_user_membership(user_memberships, groups_to_check, scope='any',  
                                     ignore_exceptions=False)
```

This function checks if a user belongs to one or more security groups.

Changed in version 3.1.0: Parenthesis were added to the exception classes and the function was refactored to be more efficient.

#### Parameters

- **user\_memberships** (*list*, *tuple*) – A list of security groups to which the user belongs
- **groups\_to\_check** (*list*, *tuple*, *str*) – One or more groups (name or ID) against which to compare the user's memberships
- **scope** (*str*) – Determines the result returned for the comparison (Options: any, all or each)
- **ignore\_exceptions** (*bool*) – Determines whether or not exceptions should be ignored (Default: False)

#### Returns

Returns a Boolean value for any and all scopes, or a list of Boolean values for each

#### Raises

*[khorosjx.errors.exceptions.InvalidScopeError](#)*

```
khorosjx.groups.get_all_groups(return_fields=None, return_type='list', ignore_exceptions=False)
```

This function returns information on all security groups found within the environment.

Changed in version 3.1.0: Changed the default `return_fields` value to `None` and adjusted the function accordingly.

#### Parameters

- **return\_fields** (*list*, *None*) – Specific fields to return if not all of the default fields are needed (Optional)
- **return\_type** (*str*) – Determines if the data should be returned in a list or a pandas dataframe (Default: *list*)
- **ignore\_exceptions** (*bool*) – Determines whether or not exceptions should be ignored (Default: *False*)

**Returns**

A list of dictionaries or a dataframe containing information for each group

**Raises**

*khorojx.errors.exceptions.InvalidDatasetError*

`khorojx.groups.get_group_info(group_id, return_fields=None, ignore_exceptions=False)`

This function obtains the group information for a given Group ID.

Changed in version 3.1.0: Changed the default `return_fields` value to `None` and adjusted the function accordingly.

**Parameters**

- **group\_id** (*int*, *str*) – The Group ID of the group whose information will be requested
- **return\_fields** (*list*, *None*) – Specific fields to return if not all of the default fields are needed (Optional)
- **ignore\_exceptions** (*bool*) – Determines whether or not exceptions should be ignored (Default: *False*)

**Returns**

A dictionary with the group information

**Raises**

*khorojx.errors.exceptions.GETRequestError*, *khorojx.errors.exceptions.InvalidDatasetError*

`khorojx.groups.get_group_memberships(group_id, user_type='member', only_id=True, return_type='list', ignore_exceptions=False, quiet=False)`

This function gets the memberships (including administrator membership) for a specific security group.

Changed in version 2.5.3: Added the optional `_quiet` argument to silence missing API field errors.

Changed in version 2.5.1: The `?fields=@all` query string was added to the API URI to ensure all fields are retrieved.

**Parameters**

- **group\_id** (*int*, *str*) – The Group ID for the security group
- **user\_type** (*str*) – Determines if the function should return `admin` or `member` users (Default: `member`)
- **only\_id** (*bool*) – Determines if only the User ID for the members should be returned or full data (Default: *True*)
- **return\_type** (*str*) – Determines if a `list` or `dataframe` should be returned (Default: `list`)
- **ignore\_exceptions** (*bool*) – Determines whether or not exceptions should be ignored (Default: *True*)

- **quiet** (*bool*) – Silences any errors about being unable to locate API fields (False by default)

**Returns**

A list or dataframe of security group memberships

**Raises**

*ValueError*

`khorojx.groups.get_user_memberships(user_lookup, return_values='name', ignore_exceptions=False)`

This function returns the security group memberships for a given user.

Changed in version 3.1.0: Refactored the function to be more efficient.

**Parameters**

- **user\_lookup** (*int, str*) – A User ID or email address that can be used to identify the user
- **return\_values** (*str*) – The type of values that should be returned in the membership list (Default: name)
- **ignore\_exceptions** (*bool*) – Determines whether or not exceptions should be ignored (Default: False)

**Returns**

A list of group memberships for the user

**Raises**

*khorojx.errors.exceptions.UserQueryError*

`khorojx.groups.retrieve_connection_info()`

This function initializes and defines the global variables for the connection information.

Changed in version 3.1.0: Refactored the function to be more efficient.

**Returns**

None

**Raises**

*khorojx.errors.exceptions.KhorosJXError*, *khorojx.errors.exceptions.NoCredentialsError*

`khorojx.groups.verify_core_connection()`

This function verifies that the core connection information (Base URL and API credentials) has been defined.

Changed in version 3.1.0: Refactored the function to be more pythonic and to avoid depending on a try/except block.

**Returns**

None

**Raises**

*khorojx.errors.exceptions.KhorosJXError*, *khorojx.errors.exceptions.NoCredentialsError*

*Return to Top*

### 1.5.6 News Module (`khorosjx.news`)

This module contains functions for working with news streams, including publications and subscriptions.

**Module**

`khorosjx.news`

**Synopsis**

Collection of functions relating to security groups

**Usage**

```
from khorosjx import news
```

**Example**

```
all_publication = khorosjx.news.get_all_publications()
```

**Created By**

Jeff Shurtliff

**Last Modified**

Jeff Shurtliff

**Modified Date**

22 Sep 2021

`khorosjx.news.delete_publication(pub_id, return_json=False)`

This function deletes a publication when given its ID.

**Parameters**

- **pub\_id** (*int*, *str*) – The ID of the publication
- **return\_json** (*bool*) – Determines if the API response should be returned in JSON format (False by default)

**Returns**

The API response (optionally in JSON format)

`khorosjx.news.delete_stream(stream_id, return_json=False)`

This function deletes a stream when given its ID.

**Parameters**

- **stream\_id** (*int*, *str*) – The ID of the stream
- **return\_json** (*bool*) – Determines if the API response should be returned in JSON format (False by default)

**Returns**

The API response (optionally in JSON format)

`khorosjx.news.filter_subscriptions_by_id(sub_id, subscriptions)`

This function filters the returned IDs by a supplied subscription ID when applicable.

Changed in version 3.1.0: Parenthesis were added to the exception classes and the function was refactored to be more efficient.

**Parameters**

- **sub\_id** (*str*) – The subscription ID to use as the filter
- **subscriptions** (*list*) – A list of subscriptions

**Returns**

The subscription that has the supplied subscription ID



**Raises***khorojx.errors.exceptions.SubscriptionNotFoundError*`khorojx.news.get_all_publications(return_fields=None, return_type='list', ignore_exceptions=False)`

This function retrieves all publications within an environment.

Changed in version 3.1.0: Changed the default `return_fields` value to `None` and adjusted the function accordingly.

**Parameters**

- **return\_fields** (*list*, *None*) – Specific fields to return if not all of the default fields are needed (Optional)
- **return\_type** (*str*) – Determines if the data should be returned in a list or a pandas dataframe (Default: `list`)
- **ignore\_exceptions** (*bool*) – Determines whether or not exceptions should be ignored (Default: `False`)

**Returns**

A list of dictionaries or a dataframe containing information for each publication

**Raises***khorojx.errors.exceptions.InvalidDatasetError*`khorojx.news.get_publication(pub_id, return_fields=None, ignore_exceptions=False)`

This function retrieves the information on a single publication when supplied its ID.

Changed in version 3.1.0: Changed the default `return_fields` value to `None` and adjusted the function accordingly.

**Parameters**

- **pub\_id** (*int*, *str*) – The ID of the publication
- **return\_fields** (*list*, *None*) – Specific fields to return if not all of the default fields are needed (Optional)
- **ignore\_exceptions** (*bool*) – Determines whether or not exceptions should be ignored (Default: `False`)

**Returns**

A dictionary with the data for the publication

**Raises***khorojx.errors.exceptions.InvalidDatasetError*, *khorojx.errors.exceptions.GETRequestError*`khorojx.news.get_stream(stream_id, return_fields=None, ignore_exceptions=False)`

This function retrieves the information on a single publication when supplied its ID.

Changed in version 3.1.0: Changed the default `return_fields` value to `None` and adjusted the function accordingly.

**Parameters**

- **stream\_id** (*int*, *str*) – The ID of the stream to retrieve
- **return\_fields** (*list*, *None*) – Specific fields to return if not all of the default fields are needed (Optional)
- **ignore\_exceptions** (*bool*) – Determines whether or not exceptions should be ignored (Default: `False`)

**Returns**

A dictionary with the data for the publication

**Raises**

*khorojx.errors.exceptions.InvalidDatasetError*, *khorojx.errors.exceptions.GETRequestError*

`khorojx.news.get_subscriber_groups(publication_id, subscription_id="", full_uri=False)`

This function identifies the subscriber groups for one or more subscriptions within a publication.

Changed in version 3.1.0: Refactored the function to be more efficient.

**Parameters**

- **publication\_id** (*int*, *str*) – The ID of the publication
- **subscription\_id** (*int*, *str*) – The specific subscription ID for which to return subscriber groups (Optional)
- **full\_uri** (*bool*) – Determines whether or not to return the full URI or just the Group ID (False by default)

**Returns**

A dictionary mapping the subscription IDs to the respective subscriber groups

**Raises**

*khorojx.errors.exceptions.SubscriptionNotFoundError*

`khorojx.news.get_subscriber_ids(subscribers)`

This function pulls the subscriber IDs out of dictionaries and into a single list.

**Parameters**

**subscribers** (*list*) – A list of dictionaries containing subscriber information from which to extract the IDs

**Returns**

A list of IDs for the supplied subscribers

`khorojx.news.get_subscribers(publication_id, subscription_id, return_type='list', only_id=True, return_fields=None, ignore_exceptions=False)`

This function retrieves the individual subscribers (i.e. users) for a given subscription within a publication.

Changed in version 3.1.0: Changed the default `return_fields` value to `None` and adjusted the function accordingly.

**Parameters**

- **publication\_id** (*int*, *str*) – The ID of the publication where the subscription resides
- **subscription\_id** (*int*, *str*) – The ID of the subscription in which to identify the subscribers
- **return\_type** (*str*) – Determines whether the data should be returned as a `list` (default) or a `pandas dataframe`
- **only\_id** (*bool*) – Determines if only the ID of each user should be returned (default) or a dict with all user data
- **return\_fields** (*list*, *None*) – Specific fields to return if not all of the default fields are needed (Optional)
- **ignore\_exceptions** (*bool*) – Determines whether or not exceptions should be ignored (Default: `False`)

**Returns**

A list or pandas dataframe with the subscriber information

`khorosjx.news.get_subscription_data(pub_id)`

This function returns the subscription data for a given publication.

**Parameters**

**pub\_id** (*int*, *str*) – The ID of the publication

**Returns**

A list of dictionaries containing the data for each subscription

`khorosjx.news.get_subscription_ids(pub_id, return_type='str')`

This function compiles a list of subscription IDs for a given publication ID.

**Parameters**

- **pub\_id** (*int*, *str*) – The ID of the publication
- **return\_type** – Determines if the IDs should be returned in *str* (default) or *int* format

**Returns**

A list of subscription IDs

**Raises**

*ValueError*

`khorosjx.news.rebuild_publication(publication_id)`

This function rebuilds a publication.

**Parameters**

**publication\_id** (*int*, *str*) – The ID of the publication to be rebuilt

**Returns**

The response from the API PUT request

**Raises**

*khorosjx.errors.exceptions.PUTRequestError*

`khorosjx.news.retrieve_connection_info()`

This function initializes and defines the global variables for the connection information.

Changed in version 3.1.0: Refactored the function to be more efficient.

**Returns**

None

**Raises**

*khorosjx.errors.exceptions.KhorosJXError*, *khorosjx.errors.exceptions.NoCredentialsError*

`khorosjx.news.update_publication(publication_id, payload)`

This function updates a publication using the supplied JSON payload.

**Parameters**

- **publication\_id** (*int*, *str*) – The ID of the publication to be updated
- **payload** (*dict*) – The JSON payload with which the publication will be updated

**Returns**

The response from the API PUT request

**Raises**

*khorosjx.errors.exceptions.PUTRequestError*

`khorosjx.news.update_stream(stream_id, payload)`

This function updates a stream using the supplied JSON payload.

**Parameters**

- **stream\_id** (*int*, *str*) – The ID of the stream to be updated
- **payload** (*dict*) – The JSON payload with which the stream will be updated

**Returns**

The response from the API PUT request

**Raises**

*khorosjx.errors.exceptions.PUTRequestError*

`khorosjx.news.verify_core_connection()`

This function verifies that the core connection information (Base URL and API credentials) has been defined.

Changed in version 3.1.0: Refactored the function to be more pythonic and to avoid depending on a try/except block.

**Returns**

None

**Raises**

*khorosjx.errors.exceptions.KhorosJXError*, *khorosjx.errors.exceptions.NoCredentialsError*

*[Return to Top](#)*

## 1.5.7 Places Module (khorosjx.places)

This module contains sub-modules containing functions for working with containers known as places, which are identified as either spaces or blogs.

**Package**

`khorosjx.places`

**Synopsis**

This package includes custom exceptions and accompanying function

**Usage**

```
import khorosjx.places
```

**Example**

TBD

**Created By**

Jeff Shurtliff

**Last Modified**

Jeff Shurtliff

**Modified Date**

14 Jan 2020

[Return to Top](#)

## Base Places Module (`khorojx.places.base`)

This module contains core functions relating to places. (i.e. spaces and blogs)

### Module

`khorojx.places.base`

### Synopsis

Collection of core functions relating to places (i.e. spaces and blogs)

### Usage

```
import khorojx.places.base as places_core
```

### Example

```
place_info = khorojx.spaces.core.get_place_info(browse_id)
```

### Created By

Jeff Shurtliff

### Last Modified

Jeff Shurtliff

### Modified Date

22 Sep 2021

`khorojx.places.base.get_browse_id(container_id, return_type='str')`

This function retrieves the Browse ID (aka Place ID) for a place given its Container ID.

### Parameters

- **container\_id** (*int*, *str*) – The Space ID for the space to query
- **return\_type** (*str*) – Determines whether to return the value as a `str` or an `int` (Default: `str`)

### Returns

The Browse ID (aka Place ID) for the place

### Raises

`GETRequestError`

`khorojx.places.base.get_place_id(container_id, return_type='str')`

This function retrieves the Place ID (aka Browse ID) for a place given its Container ID.

Changed in version 3.1.0: Made improvements to proactively avoid raising any `NameError` exceptions.

### Parameters

- **container\_id** (*int*, *str*) – The Container ID for the space to query
- **return\_type** (*str*) – Determines whether to return the value as a `str` or an `int` (Default: `str`)

### Returns

The Place ID (aka Browse ID) for the place

**Raises**

GETRequestError

`khorosjx.places.base.get_place_info(place_id, return_fields=None, ignore_exceptions=False)`

This function obtains the place information for a given Place ID. (aka Browse ID)

Changed in version 3.1.0: Changed the default `return_fields` value to `None` and adjusted the function accordingly.

**Parameters**

- **place\_id** (*int*, *str*) – The Place ID (aka Browse ID) of the place whose information will be requested
- **return\_fields** (*list*, *None*) – Specific fields to return if not all of the default fields are needed (Optional)
- **ignore\_exceptions** (*bool*) – Determines whether or not exceptions should be ignored (Default: `False`)

**Returns**

A dictionary with the place information

**Raises**

GETRequestError, InvalidDatasetError

`khorosjx.places.base.get_places_list_from_file(full_path, file_type='csv', has_headers=True,  
id_column='', id_type='browse_id',  
excel_sheet_name='', filter_info=None)`

This function retrieves a list of place identifiers from a file.

**Parameters**

- **full\_path** (*str*) – The full path to the file to import
- **file\_type** – Defines if the file to be imported is a csv (Default), `xlsx`, `xls` or `txt` file.
- **has\_headers** (*bool*) – Defines if the import file uses column headers (True by default)
- **id\_column** (*str*) – Defines the column name (if applicable) which contains the place identifier (Null by default)
- **id\_type** (*str*) – Defines if the ID type is a `browse_id` (Default) or `container_id`

---

**Todo:** Add functionality for this parameter within the function

---

- **excel\_sheet\_name** (*str*) – The sheet name to retrieve if an Excel file is supplied (First sheet imported by default)
- **filter\_info** (*dict*, *None*) – Dictionary used to apply any filter to the imported data if necessary (Null by default)

**Returns**

A list of place identifiers

**Raises**

InvalidFileTypeError

`khorosjx.places.base.get_uri_for_id(destination_id, browse_id=True)`

This function generates the full URI for a place given an identifier.

**Parameters**

- **destination\_id** (*int*, *str*) – A Jive ID or Place ID (aka Browse ID) for a place
- **browse\_id** (*bool*) – Defines whether or not the identifier provided is a Browse ID (True by default)

**Returns**

The full URI in string format

`khorojx.places.base.retrieve_connection_info()`

This function initializes and defines the global variables for the connection information.

New in version 3.1.0.

**Returns**

None

**Raises**

`khorojx.errors.exceptions.KhorosJXError`, `khorojx.errors.exceptions.NoCredentialsError`

`khorojx.places.base.verify_core_connection()`

This function verifies that the core connection information (Base URL and API credentials) has been defined.

Changed in version 3.1.0: Refactored the function to be more pythonic and to avoid depending on a try/except block.

**Returns**

None

**Raises**

`khorojx.errors.exceptions.KhorosJXError`, `khorojx.errors.exceptions.NoCredentialsError`

*[Return to Top](#)*

## Blogs Module (`khorojx.places.blogs`)

This module contains functions for working with blogs (meaning the containers rather than the individual blog posts) such as identifying posts within a blog, etc.

**Module**

`khorojx.places.blogs`

**Synopsis**

Collection of core places functions that are specific to blogs

**Usage**

```
import khorojx.places.blogs
```

**Example**

```
blog_info = khorojx.places.blogs.get_blog_info(browse_id)
```

**Created By**

Jeff Shurtliff

**Last Modified**

Jeff Shurtliff

**Modified Date**

22 Sep 2021

`khorojsx.places.blogs.get_blog_info(place_id, return_fields=None, ignore_exceptions=False)`

This function obtains the blog information for a given Place ID. (aka Browse ID)

Changed in version 3.1.0: Changed the default `return_fields` value to `None` and adjusted the function accordingly.

**Parameters**

- **place\_id** (*int*, *str*) – The Place ID (aka Browse ID) of the blog whose information will be requested
- **return\_fields** (*list*, *None*) – Specific fields to return if not all of the default fields are needed (Optional)
- **ignore\_exceptions** (*bool*) – Determines whether or not exceptions should be ignored (Default: `False`)

**Returns**

A dictionary with the blog information

**Raises**

`GETRequestError`, `InvalidDatasetError`

`khorojsx.places.blogs.retrieve_connection_info()`

This function initializes and defines the global variables for the connection information.

New in version 3.1.0.

**Returns**

`None`

**Raises**

`khorojsx.errors.exceptions.KhorosJXError`, `khorojsx.errors.exceptions.NoCredentialsError`

`khorojsx.places.blogs.verify_core_connection()`

This function verifies that the core connection information (Base URL and API credentials) has been defined.

Changed in version 3.1.0: Refactored the function to be more pythonic and to avoid depending on a try/except block.

**Returns**

`None`

**Raises**

`khorojsx.errors.exceptions.KhorosJXError`, `khorojsx.errors.exceptions.NoCredentialsError`

*[Return to Top](#)*



## Spaces Module (`khorojx.places.spaces`)

This module contains functions for working with spaces, such as identifying content within spaces, space permissions, permitted content types, etc.

### Module

`khorojx.places.spaces`

### Synopsis

Collection of core places functions that are specific to spaces

### Usage

```
import khorojx.places.spaces
```

### Example

```
space_info = khorojx.places.spaces.get_space_info(browse_id)
```

### Created By

Jeff Shurtliff

### Last Modified

Jeff Shurtliff

### Modified Date

22 Sep 2021

`khorojx.places.spaces.get_permitted_content_types(id_value, id_type='browse_id', return_type='list')`

This function returns the permitted content types for a given space.

### Parameters

- **id\_value** (*int*, *str*) – The space identifier as a Browse ID (default), Place ID or Container ID
- **id\_type** (*str*) – Determines if the id\_value is a browse\_id (Default), place\_id or container\_id
- **return\_type** (*str*) – Determines if the result should be returned in list (Default), tuple or str format

### Returns

The permitted content types in list, tuple or string format

### Raises

`khorojx.errors.exceptions.SpaceNotFoundError`, `khorojx.errors.exceptions.GETRequestError`

`khorojx.places.spaces.get_space_content_permissions(id_value, id_type='browse_id', return_type='list')`

This function returns all of the defined permissions (aka `appliedEntitlements`) for a specific space.

### Parameters

- **id\_value** (*int*, *str*) – The space identifier as a Browse ID (default), Place ID or Space ID
- **id\_type** (*str*) – Determines if the id\_value is a browse\_id (Default), place\_id or space\_id
- **return\_type** (*str*) – Determines if the result should be returned as a list (Default) or pandas dataframe

#### Returns

The list or dataframe with the space permissions

#### Raises

*khorojx.errors.exceptions.SpaceNotFoundError*, *khorojx.errors.exceptions.GETRequestError*

`khorojx.places.spaces.get_space_info(place_id, return_fields=None, ignore_exceptions=False)`

This function obtains the space information for a given Place ID. (aka Browse ID)

Changed in version 3.1.0: Changed the default `return_fields` value to `None` and adjusted the function accordingly.

#### Parameters

- **place\_id** (*int*, *str*) – The Place ID (aka Browse ID) of the space whose information will be requested
- **return\_fields** (*list*, *None*) – Specific fields to return if not all of the default fields are needed (Optional)
- **ignore\_exceptions** (*bool*) – Determines whether or not exceptions should be ignored (Default: `False`)

#### Returns

A dictionary with the space information

#### Raises

*khorojx.errors.exceptions.GETRequestError*, *khorojx.errors.exceptions.InvalidDatasetError*

`khorojx.places.spaces.get_space_permissions(id_value, id_type='browse_id', return_type='list')`

This **deprecated** function returns all of the defined permissions (aka `appliedEntitlements`) for a space.

Deprecated since version 2.6.0: The function has been renamed to be *khorojx.places.spaces.get\_space\_content\_permissions()*.

#### Parameters

- **id\_value** (*int*, *str*) – The space identifier as a Browse ID (default), Place ID or Space ID
- **id\_type** (*str*) – Determines if the `id_value` is a `browse_id` (Default), `place_id` or `space_id`
- **return\_type** (*str*) – Determines if the result should be returned as a `list` (Default) or `pandas dataframe`

#### Returns

The list or dataframe with the space permissions

#### Raises

*khorojx.errors.exceptions.SpaceNotFoundError*, *khorojx.errors.exceptions.GETRequestError*

`khorojx.places.spaces.retrieve_connection_info()`

This function initializes and defines the global variables for the connection information.

Changed in version 3.1.0: Refactored the function to be more efficient.

#### Returns

`None`

**Raises**

`khorojx.errors.exceptions.KhorosJXError`, `khorojx.errors.exceptions.NoCredentialsError`

`khorojx.places.spaces.verify_core_connection()`

This function verifies that the core connection information (Base URL and API credentials) has been defined.

Changed in version 3.1.0: Refactored the function to be more pythonic and to avoid depending on a try/except block.

**Returns**

None

**Raises**

`khorojx.errors.exceptions.KhorosJXError`, `khorojx.errors.exceptions.NoCredentialsError`

[Return to Top](#)

## 1.5.8 Spaces Module (`khorojx.spaces`)

This module contains functions for working with spaces, such as identifying content within spaces, etc.

**Module**

`khorojx.spaces`

**Synopsis**

Collection of deprecated functions relating to spaces/places

**Usage**

```
import khorojx.spaces
```

**Example**

```
space_info = khorojx.spaces.get_space_info(1234)
```

**Created By**

Jeff Shurtliff

**Last Modified**

Jeff Shurtliff

**Modified Date**

24 Mar 2020

`khorojx.spaces.get_browse_id(space_id, return_type='str')`

This function retrieves the Browse ID (aka Place ID) for a space given its ID.

Deprecated since version 2.0.0: The `khorojx.places.spaces.get_browse_id()` function should be used.

**Parameters**

- **space\_id** (`int`, `str`) – The Space ID for the space to query
- **return\_type** (`str`) – Determines whether to return the value as a `str` or an `int` (Default: `str`)

**Returns**

The Browse ID (aka Place ID) for the space

**Raises**

[\*khorosjx.errors.exceptions.GETRequestError\*](#)

`khorosjx.spaces.get_permitted_content_types(id_value, id_type='browse_id', return_type='list')`

This function returns the permitted content types for a given space.

Deprecated since version 2.0.0: The [\*khorosjx.places.spaces.get\\_permitted\\_content\\_types\(\)\*](#) function should be used.

**Parameters**

- **id\_value** (*int*, *str*) – The space identifier as a Browse ID (default), Place ID or Space ID
- **id\_type** (*str*) – Determines if the id\_value is a browse\_id (Default), place\_id or space\_id
- **return\_type** (*str*) – Determines if the result should be returned in list (Default), tuple or str format

**Returns**

The permitted content types in list, tuple or string format

**Raises**

`khorosjx.errors.exceptions.SpaceNotFountError`, [\*khorosjx.errors.exceptions.GETRequestError\*](#)

`khorosjx.spaces.get_place_id(space_id, return_type='str')`

This function retrieves the Place ID (aka Browse ID) for a space given its ID.

Deprecated since version 2.0.0: The `khorosjx.places.spaces.get_place_id()` function should be used.

**Parameters**

- **space\_id** (*int*, *str*) – The Space ID for the space to query
- **return\_type** (*str*) – Determines whether to return the value as a str or an int (Default: str)

**Returns**

The Place ID (aka Browse ID) for the space

**Raises**

`GETRequestError`

`khorosjx.spaces.get_space_info(place_id, return_fields=[], ignore_exceptions=False)`

This function obtains the space information for a given Space ID.

Deprecated since version 2.0.0: The [\*khorosjx.places.spaces.get\\_space\\_info\(\)\*](#) function should be used.

**Parameters**

- **place\_id** (*int*, *str*) – The Place ID (aka Browse ID) of the space whose information will be requested
- **return\_fields** (*list*) – Specific fields to return if not all of the default fields are needed (Optional)
- **ignore\_exceptions** (*bool*) – Determines whether or not exceptions should be ignored (Default: False)

**Returns**

A dictionary with the space information

**Raises**

`khorojx.errors.exceptions.GETRequestError`, `khorojx.errors.exceptions.InvalidDatasetError`

`khorojx.spaces.get_space_permissions(id_value, id_type='browse_id', return_type='list')`

This function returns all of the defined permissions for a specific space.

Deprecated since version 2.0.0: The `khorojx.places.spaces.get_space_permissions()` function should be used.

**Parameters**

- **id\_value** (*int*, *str*) – The space identifier as a Browse ID (default), Place ID or Space ID
- **id\_type** (*str*) – Determines if the id\_value is a browse\_id (Default), place\_id or space\_id
- **return\_type** (*str*) – Determines if the result should be returned as a list (Default) or pandas dataframe

**Returns**

The list or dataframe with the space permissions

**Raises**

`khorojx.errors.exceptions.SpaceNotFountError`, `khorojx.errors.exceptions.GETRequestError`

`khorojx.spaces.get_spaces_list_from_file(full_path, file_type='csv', has_headers=True, id_column="", id_type='browse_id', excel_sheet_name="", filter_info={})`

This function retrieves a list of space identifiers from a file.

Deprecated since version 2.0.0: The `khorojx.places.base.get_places_list_from_file()` function should be used.

**Parameters**

- **full\_path** (*str*) – The full path to the file to import
- **file\_type** – Defines if the file to be imported is a csv (Default), `xlsx`, `xls` or `txt` file.
- **has\_headers** (*bool*) – Defines if the import file uses column headers (True by default)
- **id\_column** (*str*) – Defines the column name (if applicable) which contains the space identifier (Null by default)
- **id\_type** (*str*) – Defines if the ID type is a browse\_id (Default) or place\_id (aka container\_id)
- **excel\_sheet\_name** (*str*) – The sheet name to retrieve if an Excel file is supplied (First sheet imported by default)
- **filter\_info** (*dict*) – Dictionary used to apply any filter to the imported data if necessary (Null by default)

**Returns**

A list of space identifiers

**Raises**

`khorojx.errors.exceptions.InvalidFileTypeError`

*Return to Top*

### 1.5.9 Users Module (khorosjx.users)

This module contains functions for working with users, such as obtaining their account/profile information, getting a count of their created content, etc.

**Module**

khorosjx.users

**Synopsis**

Collection of functions relating to user accounts and profiles

**Usage**

```
import khorosjx
```

**Example**

```
user_info = khorosjx.users.get_people_followed(user_id)
```

**Created By**

Jeff Shurtliff

**Last Modified**

Jeff Shurtliff

**Modified Date**

22 Sep 2021

`khorosjx.users.get_json_field(json_data, field_names)`

This function retrieves a value for a specific field from the JSON data for a user.

Changed in version 3.1.0: Refactored the function to be more efficient.

**Parameters**

- **json\_data** (*dict*) – The JSON data from which the field value must be retrieved
- **field\_names** (*tuple*, *list*) – The field name along with any parent field paths

**Returns**

The value for the specific field in its original format

**Raises**

*ValueError*, *TypeError*, *KeyError*

`khorosjx.users.get_people_followed(user_id, ignore_exceptions=False, return_type=<class 'list'>, start_index=0)`

This function returns a list of users followed by a particular user.

**Parameters**

- **user\_id** (*int*) – The User ID for the user against which to check
- **ignore\_exceptions** (*bool*) – Determines whether non-200 API responses should raise an exception (Default: *False*)
- **return\_type** (*type*) – Determines whether to return a comma-separated string, tuple or list (Default: *list*)
- **start\_index** (*int*) – The startIndex for the API call (Default: *0*)

**Returns**

The User IDs of all people followed by the queried user

`khorojx.users.get_primary_email(lookup_value, lookup_type='id')`

This function obtains the primary email address for a user by looking up their User ID or username.

Changed in version 3.1.0: Updated the `khorojx.users._validate_lookup_type()` function call to use the new function name.

#### Parameters

- **lookup\_value** (*str*) – The User ID or username for which to look up the user
- **lookup\_type** (*str*) – Determines if the User ID or username should be used to find the user (Default: `id`)

#### Returns

The primary email address for the user

#### Raises

*khorojx.errors.exceptions.InvalidLookupTypeError*

`khorojx.users.get_profile_url(lookup_value, lookup_type='id')`

This function constructs the URL to a user's profile.

Changed in version 3.1.0: Removed a hardcoded URL with the interpolated `base_url` variable.

#### Parameters

- **lookup\_value** (*str*) – The lookup value to locate the user
- **lookup\_type** (*str*) – Determines if the lookup value is a User ID, email address or username (Default: `id`)

#### Returns

The URL of the user's profile

`khorojx.users.get_recent_logins(count=100, start_index=0)`

This function returns the most recent logins that have occurred in the environment.

#### Parameters

- **count** – The maximum number of results to return in the given GET request (Default: 100)
- **start\_index** – The `startIndex` value in the GET request (Default: 0)

#### Returns

The login data in JSON format

#### Raises

*khorojx.errors.exceptions.UserQueryError*

`khorojx.users.get_user_content_count(user_id, start_index=0)`

This function obtains the number of content items created by a particular user.

Changed in version 3.1.0: Updated the `khorojx.users._get_paginated_content_count()` function call to use the new function name.

#### Parameters

- **user\_id** (*int*) – The User ID of the user
- **start\_index** (*int*) – The `startIndex` value in the REST API call (Default: 0)

#### Returns

The count of content found for the user in integer format

`khorojx.users.get_user_id(lookup_value, lookup_type='email')`

This function obtains the User ID for a user by querying the API against the user's email address or username.

Changed in version 3.1.0: Updated the `khorojx.users._validate_lookup_type()` function call to use the new function name.

**Parameters**

- **lookup\_value** (*str*) – Email address or username of the user
- **lookup\_type** (*str*) – Determines if the lookup value is an email or username (Default: email)

**Returns**

The User ID for the user

**Raises**

*khorojx.errors.exceptions.InvalidLookupTypeError*, *khorojx.errors.exceptions.GETRequestError*

`khorojx.users.get_username(lookup_value, lookup_type='id')`

This function obtains the username for a user by looking up their User ID or email address.

Changed in version 3.1.0: Updated the `khorojx.users._validate_lookup_type()` function call to use the new function name.

**Parameters**

- **lookup\_value** (*str*) – The User ID or email address for which to look up the user
- **lookup\_type** (*str*) – Determines if the User ID or email address should be used to find the user (Default: id)

**Returns**

The username for the user

`khorojx.users.parse_user_fields(json_data)`

This function populates a dictionary with the user information retrieved from the API response.

Changed in version 3.1.0: Refactored the function to be more efficient.

**Parameters**

**json\_data** (*dict*) – The user data retrieved from the API in JSON format

**Returns**

Dictionary of user information that has been validated and normalized

`khorojx.users.retrieve_connection_info()`

This function initializes and defines the global variables for the connection information.

Changed in version 3.1.0: Refactored the function to be more efficient.

**Returns**

None

**Raises**

*khorojx.errors.exceptions.KhorosJXError*, *khorojx.errors.exceptions.NoCredentialsError*

`khorojx.users.verify_core_connection()`

This function verifies that the core connection information (Base URL and API credentials) has been defined.

Changed in version 3.1.0: Refactored the function to be more pythonic and to avoid depending on a try/except block.



**Returns**

None

**Raises**

`khrosjx.errors.exceptions.KhorosJXError`, `khrosjx.errors.exceptions.NoCredentialsError`

[Return to Top](#)

## 1.6 Supporting Modules

This section provides details around the supporting modules used in the **khrosjx** package, which are listed below.

- **Tools and Utilities**
  - *Core Utilities Module* (`khrosjx.utils.core_utils`)
  - *Dataframe Utilities Module* (`khrosjx.utils.df_utils`)
  - *Helper Module* (`khrosjx.utils.helper`)
  - *Tests Module* (`khrosjx.utils.tests`)
  - *Version Module* (`khrosjx.utils.version`)
- **Classes and Exceptions**
  - *Classes Module* (`khrosjx.utils.classes`)
  - *Errors Module* (`khrosjx.errors`)
    - \* *Exceptions Module* (`khrosjx.errors.exceptions`)
    - \* *Handlers Module* (`khrosjx.errors.handlers`)

### 1.6.1 Tools and Utilities

This section includes modules that contain tools and utilities leveraged by other scripts.

#### Core Utilities Module (`khrosjx.utils.core_utils`)

This module includes various utilities to assist in converting dictionaries to JSON, formatting timestamps, etc.

**Module**`khrosjx.utils.core_utils`**Synopsis**

Useful tools and utilities to assist in managing a Khoros JX (formerly Jive-x) or Jive-n community

**Usage**

```
import khrosjx
```

**Example**

```
timestamp = khorosjx.utils.core_utils.get_timestamp(time_format="delimited")
```

**Created By**

Jeff Shurtliff

**Last Modified**

Jeff Shurtliff

**Modified Date**

Dec 2019

```
khorosjx.utils.core_utils.add_to_master_list(single_list, master_list)
```

This function appends items in a list to the master list.

**Parameters**

- **single\_list** (*list*) – List of dictionaries from the paginated query
- **master\_list** (*list*) – Master list of dictionaries containing group information

**Returns**

The master list with the appended data

```
khorosjx.utils.core_utils.convert_dict_list_to_dataframe(dict_list, column_names=[])
```

This function converts a list of dictionaries into a pandas dataframe.

**Parameters**

- **dict\_list** (*list*) – List of dictionaries
- **column\_names** (*list*) – The column names for the dataframe (Optional)

**Returns**

A pandas dataframe of the data

```
khorosjx.utils.core_utils.convert_dict_to_json(data)
```

This function converts a dictionary to JSON so that it can be traversed similar to a converted requests response.

**Parameters**

**data** (*dict*) – Dictionary to be converted to JSON

**Returns**

The dictionary data in JSON format

**Raises**

TypeError

```
khorosjx.utils.core_utils.convert_single_pair_dict_list(dict_list)
```

This function converts a list of single-pair dictionaries into a normal list.

**Parameters**

**dict\_list** (*list*) – A list of single-pair dictionaries

**Returns**

A normal list with the value from each dictionary

```
khorosjx.utils.core_utils.eprint(*args, **kwargs)
```

This function behaves the same as the `print()` function but is leveraged to print errors to `sys.stderr`.

```
khorosjx.utils.core_utils.get_format_syntax(syntax_nickname)
```

This function obtains the appropriate datetime format syntax for a format nickname. (e.g. delimited)

**Parameters**

**syntax\_nickname** (*str*) – The nickname of a datetime format

**Returns**

The proper datetime format as a string

`khorojx.utils.core_utils.get_timestamp(time_format='split')`

This function obtains the current timestamp in the local timezone.

**Parameters**

**time\_format** (*str*) – The format for the timestamp that will be returned (default: split)

**Returns**

The current timestamp in %Y-%m-%d %H:%M:%S format as a string

`khorojx.utils.core_utils.identify_dataset(query_uri)`

This function identifies the appropriate field dataset by examining a query URI.

**Parameters**

**query\_uri** (*str*) – The API query URI to be examined

**Returns**

The appropriate dataset name in string format

`khorojx.utils.core_utils.print_if_verbose(msg, verbose_enabled=False)`

This function prints a message onscreen only if verbose mode is enabled.

**Parameters**

- **msg** (*str*) – The message to print onscreen
- **verbose\_enabled** (*bool*) – Determines if verbose mode is enabled (False by default)

**Returns**

None

`khorojx.utils.core_utils.remove_comment_wrappers_from_html(html_string)`

This function removes comment wrappers (i.e. <!-- and -->) from an HTML string.

New in version 2.5.1.

**Parameters**

**html\_string** (*str*) – The HTML string from which to remove comment wrappers

**Returns**

The HTML string with comment wrappers removed

`khorojx.utils.core_utils.validate_timestamp(timestamp, time_format='delimited',  
replace_invalid=True)`

This function validates a timestamp string to ensure that it matches a prescribed syntax.

**Parameters**

- **timestamp** (*str*) – The timestamp in string format
- **time\_format** (*str*) – The format for the supplied timestamp (default: delimited)
- **replace\_invalid** (*bool*) – States if an invalid timestamp should be replaced with a default value (Default: True)

**Returns**

A valid timestamp string, either what was provided or a default timestamp

**Raises**

ValueError

[Return to Top](#)

## Dataframe Utilities Module (khorosjx.utils.df\_utils)

This module includes various utilities to assist in creating, importing, exporting and manipulating pandas dataframes.

**Module**

khorosjx.utils.df\_utils

**Synopsis**

Useful tools and utilities to assist in importing, manipulating and exporting pandas dataframes

**Usage**

```
from khorosjx import df_utils
```

**Example**

TBD

**Created By**

Jeff Shurtliff

**Last Modified**

Jeff Shurtliff

**Modified Date**

18 Dec 2019

khorosjx.utils.df\_utils.**convert\_dict\_list\_to\_dataframe**(*dict\_list*, *column\_names*=[])

This function converts a list of dictionaries into a pandas dataframe.

**Parameters**

- **dict\_list** (*list*) – List of dictionaries
- **column\_names** (*list*) – The column names for the dataframe (Optional)

**Returns**

A pandas dataframe of the data

khorosjx.utils.df\_utils.**import\_csv**(*file\_path*, *delimiter*=',', *column\_names*=[], *columns\_to\_return*=[], *has\_headers*=True)

This function imports a CSV file to generate a dataframe.

**Parameters**

- **file\_path** (*str*) – The absolute path to the CSV file to be imported
- **delimiter** (*str*) – The column delimiter utilized in the CSV
- **column\_names** (*list*) – The column names to use with the imported dataframe (Optional)
- **columns\_to\_return** – Determines which of the columns should actually be returned (Default: all columns)

- **has\_headers** (*bool*) – Defines whether or not the data in the file has column headers (Default: True)

**Returns**

The imported data as a pandas dataframe

**Raises**

FileNotFoundError, TypeError

```
khorosjx.utils.df_utils.import_excel(file_path, excel_sheet="", use_first_sheet=False, column_names=[],  
                                     columns_to_return=[], has_headers=True)
```

This function imports a Microsoft Excel file to generate a dataframe.

**Parameters**

- **file\_path** (*str*) – The absolute path to the Excel file to be imported
- **excel\_sheet** (*str*) – The name of the specific sheet in the file to import
- **use\_first\_sheet** (*bool*) – Defines whether or not the first sheet in the file should be used (Default: False)
- **column\_names** (*list*) – The column names to use with the imported dataframe (Optional)
- **columns\_to\_return** – Determines which of the columns should actually be returned (Default: all columns)
- **has\_headers** (*bool*) – Defines whether or not the data in the file has column headers (Default: True)

**Returns**

The imported data as a pandas dataframe

**Raises**

FileNotFoundError, TypeError

[Return to Top](#)

## Helper Module (khorosjx.utils.helper)

This module includes allows a “helper” configuration file to be imported and parsed to facilitate the use of the library (e.g. defining the base URL and API credentials) and defining additional settings.

**Module**

khorosjx.utils.helper

**Synopsis**

Module that allows the khorosjx library to leverage a helper file and/or script

**Usage**

```
from khorosjx.utils import helper
```

**Example**

```
helper_cfg = helper.import_yaml_file('/path/to/jxhelper.yml')
```

**Created By**

Jeff Shurtliff

**Last Modified**

Jeff Shurtliff

**Modified Date**

29 Apr 2020

**class** khorosjx.utils.helper.**HelperParsing**

This class is used to help parse values imported from a YAML configuration file.

khorosjx.utils.helper.**import\_yaml\_file**(*file\_path*)

This function imports a YAML (.yaml) helper config file.

Changed in version 2.5.1: Changed the name and replaced the `yaml.load` function call with `yaml.safe_load` to be more secure.

**Parameters**

**file\_path** (*str*) – The file path to the YAML file

**Returns**

The parsed configuration data

**Raises**

FileNotFoundError

khorosjx.utils.helper.**parse\_helper\_cfg**(*helper\_cfg*, *file\_type*='yaml')

This is the primary function used to parse the helper config file.

**Parameters**

- **helper\_cfg** – The raw data loaded from the config file
- **file\_type** – Indicates the type of configuration file (Default: `yaml`)

**Returns**

None (Defines global variables)

**Raises**

CredentialsUnpackingError, InvalidHelperArgumentsError, HelperFunctionNotFoundError

khorosjx.utils.helper.**retrieve\_helper\_settings**()

This function returns a dictionary of the defined helper settings.

**Returns**

Dictionary of helper variables with nicknames

[Return to Top](#)

## Tests Module (khorosjx.utils.tests)

This module includes unit tests for the package that are performed using `pytest`.

### Test Module Import (khorosjx.utils.tests.test\_init\_module)

**Module**

khorosjx.utils.tests.test\_init\_module

**Synopsis**

This module is used by `pytest` to verify that primary modules can be imported successfully

**Created By**

Jeff Shurtliff

**Last Modified**

Jeff Shurtliff

**Modified Date**

22 Nov 2019

`khrosjx.utils.tests.test_init_module.init_module_operation()`

This function imports the primary modules for the package and returns True when successful.

`khrosjx.utils.tests.test_init_module.set_package_path()`

This function adds the high-level khrosjx directory to the sys.path list.

`khrosjx.utils.tests.test_init_module.test_init_module()`

This function tests to confirm that all primary modules are able to be imported successfully.

*[Return to Top](#)*

## Version Module (`khrosjx.utils.version`)

This module is the primary source of the current version of the khrosjx package, and includes two simple functions to return either the full version or the major.minor (i.e. X.Y) version.

**Module**`khrosjx.utils.version`**Synopsis**

This simple script contains the package version

**Usage**

```
from .utils import version
```

**Example**

```
__version__ = version.get_full_version()
```

**Created By**

Jeff Shurtliff

**Last Modified**

Jeff Shurtliff

**Modified Date**

23 Sep 2021

`khrosjx.utils.version.get_full_version()`

This function returns the current full version of the khrosjx package.

**Returns**

The current full version (i.e. X.Y.Z) in string format

`khrosjx.utils.version.get_latest_stable()`

This function returns the latest stable version of the khrosjx package.

**Returns**

The latest stable version in string format

`khorosjx.utils.version.get_major_minor_version()`

This function returns the current major.minor (i.e. X.Y) version of the khorosjx package.

**Returns**

The current major.minor (i.e. X.Y) version in string format

`khorosjx.utils.version.latest_version()`

This function defines if the current version matches the latest stable version on PyPI.

**Returns**

Boolean value indicating if the versions match

`khorosjx.utils.version.warn_when_not_latest()`

This function displays a [RuntimeWarning](#) if the running version doesn't match the latest stable version.

**Returns**

None

*[Return to Top](#)*

## 1.6.2 Classes and Exceptions

This section includes modules that contain the classes and exceptions used in the package.

### Classes Module (`khorosjx.utils.classes`)

This module contains nearly all classes utilized by other modules within the library.

**Module**

`khorosjx.utils.classes`

**Synopsis**

Collection of classes relating to the khorosjx library

**Usage**

```
from khorosjx.utils.classes import Users
```

**Created By**

Jeff Shurtliff

**Last Modified**

Jeff Shurtliff

**Modified Date**

06 Mar 2020

**class** `khorosjx.utils.classes.Content`

This class includes content-related lists, dictionaries and other utilities.

**class** `khorosjx.utils.classes.FieldLists`

This class provides lists of JSON fields for various API data sets.



**class** khorosjx.utils.classes.Groups

This class provides various mappings to security group-related information.

**class** khorosjx.utils.classes.Platform

This class provides various mappings to Jive-related information such as environments, URLs, etc.

**class** khorosjx.utils.classes.TimeUtils

This class contains dictionaries and other utilities to assist with time-related function calls.

**class** khorosjx.utils.classes.Users

This class includes user-related lists, dictionaries and other utilities.

**class** UserJSON

This class maps the field names in the users table to the JSON field names.

*[Return to Top](#)*

## Errors Module (khorosjx.errors)

This module contains all of the exception classes and error handling functions leveraged throughout the library.

**Package**

khorosjx.errors

**Synopsis**

This module includes custom exceptions and accompanying function

**Usage**

```
import khorosjx.errors (Imported by default in primary package)
```

**Example**

```
raise errors.exceptions.BadCredentialsError
```

**Created By**

Jeff Shurtliff

**Last Modified**

Jeff Shurtliff

**Modified Date**

08 Jan 2020

## Exceptions Module (`khorsjx.errors.exceptions`)

This sub-module contains all of the exception classes leveraged in functions throughout the library.

### Module

`khorsjx.errors.exceptions`

### Synopsis

Collection of exception classes relating to the khorsjx library

### Usage

```
import khorsjx.errors.exceptions
```

### Example

```
raise khorsjx.errors.exceptions.BadCredentialsError
```

### Created By

Jeff Shurtliff

### Last Modified

Jeff Shurtliff

### Modified Date

23 Sep 2021

**exception** `khorsjx.errors.exceptions.APIConnectionError(*args, **kwargs)`

This exception is used when the API query could not be completed due to connection aborts and/or timeouts.

**exception** `khorsjx.errors.exceptions.BadCredentialsError(*args, **kwargs)`

This exception is used when the supplied API credentials are incorrect.

**exception** `khorsjx.errors.exceptions.ContentNotFoundError(*args, **kwargs)`

This exception is used when an API query for content returns a 404 status code.

**exception** `khorsjx.errors.exceptions.ContentPublishError(*args, **kwargs)`

This exception is used when content is unable to publish successfully.

**exception** `khorsjx.errors.exceptions.CredentialsUnpackingError(*args, **kwargs)`

This exception is used when the tuple containing API credentials cannot be unpacked.

**exception** `khorsjx.errors.exceptions.CurrentlyUnsupportedError(*args, **kwargs)`

This exception is used when an operation is attempted that is not yet supported.

**exception** `khorsjx.errors.exceptions.DatasetNotFoundError(*args, **kwargs)`

This exception is used when a dataset was not provided and/or cannot be found.

**exception** `khorsjx.errors.exceptions.GETRequestError(*args, **kwargs)`

This exception is used for generic GET request errors when there isn't a more specific exception.

**exception** `khorsjx.errors.exceptions.HelperFunctionNotFoundError(*args, **kwargs)`

This exception is used when a function referenced in the helper config file does not exist.

**exception** `khorsjx.errors.exceptions.IncompleteCredentialsError(*args, **kwargs)`

This exception is used when a tuple containing API credentials is missing a username or password.

**exception** `khorsjx.errors.exceptions.InvalidDatasetError(*args, **kwargs)`

This exception is used when a supplied dataset is invalid.

**exception** `khorsjx.errors.exceptions.InvalidEndpointError(*args, **kwargs)`

This exception is used when an invalid API endpoint / service is provided.

**exception** `khrosjx.errors.exceptions.InvalidFileTypeError(*args, **kwargs)`

This exception is used when a supplied file type is invalid and cannot be used.

**exception** `khrosjx.errors.exceptions.InvalidHelperArgumentsError(*args, **kwargs)`

This exception is used when the helper function was supplied arguments instead of keyword arguments.

**exception** `khrosjx.errors.exceptions.InvalidKhorosJXModuleError(*args, **kwargs)`

This exception is used when an invalid module is attempted to be initialized from the primary `__init__` file.

**exception** `khrosjx.errors.exceptions.InvalidLookupTypeError(*args, **kwargs)`

This exception is used when an invalid API lookup type is provided.

**exception** `khrosjx.errors.exceptions.InvalidRequestTypeError(*args, **kwargs)`

This exception is used when an invalid API request type is provided.

**exception** `khrosjx.errors.exceptions.InvalidScopeError(*args, **kwargs)`

This exception is used when a supplied scope is invalid and cannot be found.

**exception** `khrosjx.errors.exceptions.KhorosJXError`

This is the base class for Khoros JX exceptions.

**exception** `khrosjx.errors.exceptions.LookupMismatchError(*args, **kwargs)`

This exception is used when an a lookup value doesn't match the supplied lookup type.

**exception** `khrosjx.errors.exceptions.MissingBaseUrlError(*args, **kwargs)`

This exception is used when a valid base URL has not been defined.

**exception** `khrosjx.errors.exceptions.NoCredentialsError(*args, **kwargs)`

This exception is used when credentials weren't found when utilizing the core functions.

**exception** `khrosjx.errors.exceptions.NotFoundResponseError(*args, **kwargs)`

This exception is used when an API query returns a 404 response and there isn't a more specific class.

**exception** `khrosjx.errors.exceptions.POSTRequestError(*args, **kwargs)`

This exception is used for generic POST request errors when there isn't a more specific exception.

**exception** `khrosjx.errors.exceptions.PUTRequestError(*args, **kwargs)`

This exception is used for generic PUT request errors when there isn't a more specific exception.

**exception** `khrosjx.errors.exceptions.SpaceNotFoundError(*args, **kwargs)`

This exception is used when an API query for a space returns a 404 response.

**exception** `khrosjx.errors.exceptions.SubscriptionNotFoundError(*args, **kwargs)`

This exception is used when a subscription referenced in a function does not exist.

**exception** `khrosjx.errors.exceptions.UserNotFoundError(*args, **kwargs)`

This exception is used when an API query for a user returns a 404 status code.

**exception** `khrosjx.errors.exceptions.UserQueryError(*args, **kwargs)`

This exception is used when an API query returns an unidentified non-200 response.

**exception** `khrosjx.errors.exceptions.WrongCredentialTypeError(*args, **kwargs)`

This exception is used when a username or password is not in string format.

[Return to Top](#)

## Handlers Module (`khorsjx.errors.handlers`)

This sub-module contains various error handling functions that are leveraged throughout the library.

### Module

`khorsjx.errors.handlers`

### Synopsis

Collection of error handler functions relating to the khorsjx library

### Usage

```
from khorsjx.errors import handlers
```

### Example

```
successful_response = check_api_response(response)
```

### Created By

Jeff Shurtliff

### Last Modified

Jeff Shurtliff

### Modified Date

22 Sep 2021

`khorsjx.errors.handlers.bad_lookup_type(lookup_type, good_examples)`

This function raises the `InvalidLookupTypeError` exception and provides a custom message.

`khorsjx.errors.handlers.check_api_response(response, request_type='get', ignore_exceptions=False)`

This function checks an API response to determine if it was successful

### Parameters

- **response** (*class*) – The API response obtained via the requests package
- **request\_type** (*str*) – The type of API request that was performed. (Default: `get`)
- **ignore\_exceptions** (*bool*) – Determines whether or not exceptions should be ignored and not raised (Default: `False`)

### Returns

A Boolean value indicating whether or not the API request was deemed successful

### Raises

`khorsjx.errors.exceptions.BadCredentialsError`, `khorsjx.errors.exceptions.GETRequestError`, `khorsjx.errors.exceptions.POSTRequestError`, `khorsjx.errors.exceptions.PUTRequestError`

`khorsjx.errors.handlers.check_json_for_error(json_data, data_type='space')`

This function checks to see if JSON from an API response contains an error.

Changed in version 3.1.0: Parenthesis were added to the exception classes and the function was refactored to be more efficient.

### Parameters

- **json\_data** (*dict*) – The API response data in JSON format
- **data\_type** (*str*) – Determines what type of data was being queried (Default: `space`)

### Returns

None

**Raises**

`khorojx.errors.exceptions.GETRequestError` `khorojx.errors.exceptions.SpaceNotFoundError`, `khorojx.errors.exceptions.NotFoundResponseError`

[Return to Top](#)

## 1.7 Change Log

This page documents the additions, changes, fixes, deprecations and removals made in each release.

### 1.7.1 v3.2.0

**Release Date: 2021-09-23**

#### Added

##### Primary Modules

Additions to the *primary modules*.

- Added the `khorojx.core.ensure_absolute_url()` function.

##### Supporting Modules

Additions to the *supporting modules*.

- Added the `khorojx.errors.exceptions.MissingBaseUrlError` exception class.

#### Changed

##### Primary Modules

Changes to the *primary modules*.

- Added the `return_url` parameter to the `khorojx.core.set_base_url()` function to determine if the base URL should be returned by the function.
- The following functions were updated to leverage the `khorojx.core.ensure_absolute_url()` function:
  - `khorojx.core.get_request_with_retries()`
  - `khorojx.core._api_request_with_payload()`
  - `khorojx.core.post_request_with_retries()`
  - `khorojx.core.put_request_with_retries()`
  - `khorojx.core.delete()`

## 1.7.2 v3.1.0

Release Date: 2021-09-22

### Added

#### Primary Modules

Additions to the *primary modules*.

- Added the `khorsjx.places.base.retrieve_connection_info()` function.
- Added the `khorsjx.places.blogs.retrieve_connection_info()` function.

### Changed

#### Primary Modules

Changes to the *primary modules*.

- Refactored the following functions to be more pythonic and to avoid depending on a try/except block, and to define the related global variables at the module level:
  - `khorsjx.core.verify_connection()`
  - `khorsjx.admin.verify_core_connection()`
  - `khorsjx.content.base.verify_core_connection()`
  - `khorsjx.content.docs.verify_core_connection()`
  - `khorsjx.content.events.verify_core_connection()`
  - `khorsjx.content.ideas.verify_core_connection()`
  - `khorsjx.content.threads.verify_core_connection()`
  - `khorsjx.content.videos.verify_core_connection()`
  - `khorsjx.groups.verify_core_connection()`
  - `khorsjx.news.verify_core_connection()`
  - `khorsjx.places.base.verify_core_connection()`
  - `khorsjx.places.blogs.verify_core_connection()`
  - `khorsjx.places.spaces.verify_core_connection()`
  - `khorsjx.users.verify_core_connection()`
- Refactored the following functions to be more efficient and removed one of the exception references in the docstring:
  - `khorsjx.admin.retrieve_connection_info()`
  - `khorsjx.content.base.retrieve_connection_info()`
  - `khorsjx.content.docs.retrieve_connection_info()`
  - `khorsjx.content.events.retrieve_connection_info()`
  - `khorsjx.content.ideas.retrieve_connection_info()`

- `khrosjx.content.threads.retrieve_connection_info()`
- `khrosjx.content.videos.retrieve_connection_info()`
- `khrosjx.groups.retrieve_connection_info()`
- `khrosjx.news.retrieve_connection_info()`
- `khrosjx.users.retrieve_connection_info()`

- **Refactored the following functions to be more efficient:**

- `khrosjx.core.get_query_url()`
- `khrosjx.core.get_request_with_retries()`
- `khrosjx.core.get_api_version()`
- `khrosjx.groups.check_user_membership()`
- `khrosjx.groups.add_user_to_group()`
- `khrosjx.news.get_subscriber_groups()`
- `khrosjx.places.base.get_places_list_from_file()`
- `khrosjx.users.get_json_field()`
- `khrosjx.users.parse_user_fields()`

- Changed the default `return_fields` value to `None` and made related adjustments in the following functions:

- `khrosjx.core.get_fields_from_api_response()`
- `khrosjx.core.get_paginated_results()`
- `khrosjx.content.base.get_paginated_content()`
- `khrosjx.content.base.get_document_info()`
- `khrosjx.groups.get_group_info()`
- `khrosjx.groups.get_all_groups()`
- `khrosjx.news.get_all_publications()`
- `khrosjx.news.get_publication()`
- `khrosjx.news.get_stream()`
- `khrosjx.news.get_subscribers()`
- `khrosjx.places.base.get_place_info()`
- `khrosjx.places.spaces.get_space_info()`

- Changed the default `categories` and `tags` values to `None` in the `khrosjx.content.docs.create_document()` and adjusted the function accordingly.
- The name of the raised exception was added to the error message in the `khrosjx.core._api_request_with_payload()` function.
- Renamed the `khrosjx.users.__validate_lookup_type()` function to be `khrosjx.users._validate_lookup_type()` instead. (Single underscore prefix)
- Renamed the `khrosjx.users.__get_paginated_content_count()` function to be `khrosjx.users._get_paginated_content_count()` instead. (Single underscore prefix)
- Renamed the `khrosjx.users.__get_followed()` function to be `khrosjx.users._get_followed()` instead. (Single underscore prefix)

- Updated the `khorojsx.users._validate_lookup_type()` function call in the following functions to use the new function name:
  - `khorojsx.users.get_user_id()`
  - `khorojsx.users.get_username()`
  - `khorojsx.users.get_primary_email()`
- Updated the `khorojsx.users._get_paginated_content_count()` function call in `khorojsx.users.get_user_content_count()` to use the new function name.
- Added a TODO to rename the following functions:
  - `khorojsx.content.base.__convert_lookup_value()`
  - `khorojsx.content.base.__trim_attachments_info()`
  - `khorojsx.places.base.__verify_browse_id()`

## Supporting Modules

Changes to the *supporting modules*.

- Moved the function `khorojsx.errors.handlers._raise_exception_for_status_code()` function out to the module level from within `khorojsx.errors.handlers.check_api_response()`.

## Fixed

### Primary Modules

Fixes in the *primary modules*.

- Fixed how the `query_url` variable is defined in the `khorojsx.core.get_data()` function to proactively avoid raising any `NameError` exceptions.
- Adjusted a dictionary lookup in the following functions to proactively avoid raising a `KeyError` exception:
  - `khorojsx.groups._get_paginated_groups()`
  - `khorojsx.content.docs.get_document_attachments()`
- Added parenthesis to the exception classes in the following functions:
  - `khorojsx.core.set_credentials()`
  - `khorojsx.content.docs.delete_document()`
  - `khorojsx.groups.check_user_membership()`
  - `khorojsx.groups.add_user_to_group()`
  - `khorojsx.news.filter_subscriptions_by_id()`
  - `khorojsx.users._get_followed()`
- Refactored the `khorojsx.core.get_base_url()` function to properly utilize the `base_url` global variable.
- Removed a hardcoded URL in the `khorojsx.users.get_profile_url()` with the interpolated `base_url` variable.
- Made some minor syntax improvements in the `khorojsx.content.base.get_content_id()` function.



- Made improvements to the `khoro.places.base.get_place_id()` function to proactively avoid raising any `NameError` exceptions.

## Supporting Modules

Changes to the *supporting modules*.

- Added parenthesis to the exception classes in the following functions:
  - `khorojx.errors.handlers.check_json_for_error()`

## 1.7.3 v3.0.0

Release Date: 2021-09-20

### Added

#### General

- Added the `codeql-analysis.yml` workflow.
- Merged dependabot pull requests to mitigate security vulnerabilities with `twine` dependency packages.

### Changed

#### General

- Started over with the `requirements.txt` file and renamed the original file to be `original-requirements.txt`.
- Added the `install_requires` configuration to the `setup.py` file.
- Added Python version 3.9 to `pythonpackage.yml`.

### Fixed

#### General

- Fixed a minor grammatical error in the `examples/khorosjx_helper.yml` file.

## 1.7.4 v2.5.3

Release Date: 2020-05-01

### Changed

#### Primary Modules

Changes to the *primary modules*.

- Added the optional `quiet` argument to the `khorojx.core.get_fields_from_api_response()` function which silences any errors for being unable to locate API fields.
- Added the optional `quiet` argument to the `khorojx.core.get_paginated_results()` function which silences any errors for being unable to locate API fields.
- Added the optional `quiet` argument to the `khorojx.groups.get_group_memberships()` function which silences any errors for being unable to locate API fields.
- Added the optional `quiet` argument to the `khorojx.groups._add_paginated_members()` function which silences any errors for being unable to locate API fields.
- Removed the unnecessary variable definition of `added_to_group` within the `khorojx.groups.add_user_to_group()` function.
- Renamed the `khorojx.core._get_filter_syntax()` function to be `khorojx.core._get_filter_syntax()` instead.
- Renamed the `khorojx.core._api_request_with_payload()` function to be `khorojx.core._api_request_with_payload()` instead.

### Documentation

Changes to the documentation.

- Added a docstring to the `khorojx.core._get_filter_syntax()` function.

### Fixed

#### Primary Modules

Fixes to the *primary modules*.

- Changed the filter string `emails.value` to be `email.value` in the `khorojx.core.get_fields_from_api_response()` function.

## 1.7.5 v2.5.2

Release Date: 2020-04-29

### Fixed

#### Supporting Modules

Fixes to the *supporting modules*.

- Fixed the `khorsjx.utils.helper._convert_yaml_to_bool()` function to only perform its operations if the passed value is not a Boolean value to prevent the following `AttributeError` exception from occurring: `AttributeError: 'bool' object has no attribute 'lower'`

## 1.7.6 v2.5.1

Release Date: 2020-04-29

### Added

#### Supporting Modules

Additions to the *supporting modules*.

- Added the `khorsjx.utils.core_utils.remove_comment_wrappers_from_html()` function.

### General

- Added *PyCharm Python Security Scanner* to the `pythonpackage.yml` file.
- Updated to `bleach v3.1.4` as recommended by `GitHub` to include some security fixes.

### Changed

#### Primary Modules

Changes to the *primary modules*.

- Renamed the `khorsjx.places.spaces.get_space_permissions()` function to be `khorsjx.places.spaces.get_space_content_permissions()` instead and added a `DeprecationWarning` to the original.
- Moved the `khorsjx.places.spaces.__get_paginated_content_permissions()` function from within the `khorsjx.places.spaces.get_space_permissions()` function to the module level.
- Moved the `khorsjx.places.spaces.verify_core_connection.__get_info()` function to the module level as `khorsjx.places.spaces.retrieve_connection_info()`.
- Renamed the `khorsjx.groups.__add_paginated_members()` function to be `khorsjx.groups._add_paginated_members()` instead.

- Renamed the `khrosjx.groups.__get_paginated_groups()` function to be `khrosjx.groups._get_paginated_groups()` instead.
- Added the `?fields=@all` query string to the API URI in the `khrosjx.groups.get_group_members()` function to ensure all fields are retrieved.

### Supporting Modules

Changes to the *supporting modules*.

- Replaced the `yaml.load()` function call with `yaml.safe_load()` in `khrosjx.utils.helper.import_yaml_file()` as it is a better security practice.
- Renamed the `khrosjx.utils.helper.__get_connection_info()` function to be `khrosjx.utils.helper._get_connection_info()` instead.
- Renamed the `khrosjx.utils.helper.__get_credentials_from_module()` function to be `khrosjx.utils.helper._get_credentials_from_module()` instead.
- Renamed the `khrosjx.utils.helper.__parse_function_arguments()` function to be `khrosjx.utils.helper._parse_function_arguments()` instead.
- Renamed the `khrosjx.utils.helper.__get_console_color_settings()` function to be `khrosjx.utils.helper._get_console_color_settings()` instead.
- Renamed the `khrosjx.utils.helper.__get_modules_to_import()` function to be `khrosjx.utils.helper._get_modules_to_import()` instead.
- Renamed the `khrosjx.utils.helper.__convert_yaml_to_bool()` function to be `khrosjx.utils.helper._convert_yaml_to_bool()` instead.

### Fixed

#### Primary Modules

Fixes to the *primary modules*.

- Added error handling to the `khrosjx.core.get_fields_from_api_response()` function to prevent it from failing with an `IndexError` exception.

#### Documentation

Fixes to the documentation.

- Fixed a typo and added hyperlinks to raised exceptions in the `khrosjx.places.spaces.get_permitted_content_types()` function docstring.
- Fixed a typo in the docstring for the `khrosjx.news.filter_subscriptions_by_id()` function.

## 1.7.7 v2.5.0

Release Date: 2020-03-25

### Changed

#### Primary Modules

Changes to the *primary modules*.

- Updated the `khorsjx.core.get_request_with_retries()` function to raise the custom `khorsjx.errors.exceptions.APIConnectionError` exception class rather than the built-in `ConnectionError` exception class.
- Removed the `import warnings` line from the `khorsjx.core` module as it was not being used.
- Moved the `khorsjx.admin.verify_core_connection.get_info()` function to the module level as `khorsjx.admin.retrieve_connection_info()`.
- Moved the `khorsjx.content.base.verify_core_connection.get_info()` function to the module level as `khorsjx.content.base.retrieve_connection_info()`.
- Moved the `khorsjx.content.docs.verify_core_connection.get_info()` function to the module level as `khorsjx.content.docs.retrieve_connection_info()`.
- Moved the `khorsjx.content.docs.__perform_overwrite_operation()` function to be at the module level instead of within the `khorsjx.content.docs.overwrite_doc_body()` function.
- Moved the `khorsjx.content.events.verify_core_connection.get_info()` function to the module level as `khorsjx.content.events.retrieve_connection_info()`.
- Moved the `khorsjx.content.ideas.verify_core_connection.get_info()` function to the module level as `khorsjx.content.ideas.retrieve_connection_info()`.
- Moved the `khorsjx.content.threads.verify_core_connection.get_info()` function to the module level as `khorsjx.content.threads.retrieve_connection_info()`.
- Moved the `khorsjx.content.videos.verify_core_connection.get_info()` function to the module level as `khorsjx.content.videos.retrieve_connection_info()`.
- Moved the `khorsjx.content.videos.__append_videos()` function to be at the module level instead of within the `khorsjx.content.videos.get_native_videos_for_space()` function.
- Moved the `khorsjx.groups.verify_core_connection.get_info()` function to the module level as `khorsjx.groups.retrieve_connection_info()`.
- Moved the `khorsjx.groups.__add_paginated_members()` function to be at the module level instead of within the `khorsjx.groups.get_group_memberships()` function.
- Moved the `khorsjx.groups.__get_paginated_groups()` function to be at the module level instead of within the `khorsjx.groups.get_all_groups()` function.
- Updated the `khorsjx.groups.get_group_memberships()` function to leverage the `khorsjx.utils.df_utils.convert_dict_list_to_dataframe()` function rather than the deprecated `khorsjx.utils.core_utils.convert_dict_list_to_dataframe()` function.
- Moved the `khorsjx.news.verify_core_connection.get_info()` function to the module level as `khorsjx.news.retrieve_connection_info()`.
- Moved the `khorsjx.news.get_subscriber_groups.__filter_subscriptions_by_id()` function to the module level as `khorsjx.news.filter_subscriptions_by_id()`.

- Moved the `khorojx.news.get_subscriber_groups.__get_subscriber_ids()` function to the module level as `khorojx.news.get_subscriber_ids()`.
- Moved the `khorojx.users.verify_core_connection.get_info()` function to the module level as `khorojx.users.retrieve_connection_info()`.
- Moved the `khorojx.users.get_user_content_count.__get_count()` function to the module level as `khorojx.users.__get_paginated_content_count()`.

## Documentation

Changes to the documentation.

- **Updated the exception class references within docstrings to be hyperlinks to the class details in these functions:**

- `khorojx.init_module()`
- `khorojx.init_helper()`
- `khorojx.admin.retrieve_connection_info()`
- `khorojx.admin.verify_core_connection()`
- `khorojx.core.get_data()`
- `khorojx.core.get_fields_from_api_response()`
- `khorojx.core.get_paginated_results()`
- `khorojx.core.get_request_with_retries()`
- `khorojx.core.post_request_with_retries()`
- `khorojx.core.put_request_with_retries()`
- `khorojx.core.set_credentials()`
- `khorojx.core.set_base_url()`
- `khorojx.core.verify_connection()`
- `khorojx.core.__api_request_with_payload()`
- `khorojx.content.get_content_id()`
- `khorojx.content.get_document_attachments()`
- `khorojx.content.get_document_info()`
- `khorojx.content.override_doc_body()`
- `khorojx.content.__convert_lookup_value()`
- `khorojx.content.base.get_content_id()`
- `khorojx.content.base.retrieve_connection_info()`
- `khorojx.content.base.verify_core_connection()`
- `khorojx.content.base.__convert_lookup_value()`
- `khorojx.content.docs.create_document()`
- `khorojx.content.docs.delete_document()`
- `khorojx.content.docs.get_content_id()`

- `khorosjx.content.docs.get_document_attachments()`
- `khorosjx.content.docs.get_document_info()`
- `khorosjx.content.docs.get_url_for_id()`
- `khorosjx.content.docs.override_doc_body()`
- `khorosjx.content.docs.retrieve_connection_info()`
- `khorosjx.content.docs.verify_core_connection()`
- `khorosjx.content.docs.__perform_overwrite_operation()`
- `khorosjx.content.events.get_content_id()`
- `khorosjx.content.events.retrieve_connection_info()`
- `khorosjx.content.events.verify_core_connection()`
- `khorosjx.content.ideas.retrieve_connection_info()`
- `khorosjx.content.ideas.verify_core_connection()`
- `khorosjx.content.threads.get_content_id()`
- `khorosjx.content.threads.retrieve_connection_info()`
- `khorosjx.content.threads.verify_core_connection()`
- `khorosjx.content.videos.check_if_embedded()`
- `khorosjx.content.videos.get_content_id()`
- `khorosjx.content.videos.get_native_videos_for_space()`
- `khorosjx.content.videos.get_video_dimensions()`
- `khorosjx.content.videos.get_video_id()`
- `khorosjx.content.videos.get_video_info()`
- `khorosjx.content.videos.retrieve_connection_info()`
- `khorosjx.content.videos.verify_core_connection()`
- `khorosjx.groups.add_user_to_group()`
- `khorosjx.groups.check_user_membership()`
- `khorosjx.groups.get_all_groups()`
- `khorosjx.groups.get_group_info()`
- `khorosjx.groups.get_group_memberships()`
- `khorosjx.groups.get_user_memberships()`
- `khorosjx.groups.retrieve_connection_info()`
- `khorosjx.groups.verify_core_connection()`
- `khorosjx.news.filter_subscriptions_by_id()`
- `khorosjx.news.get_all_publications()`
- `khorosjx.news.get_publication()`
- `khorosjx.news.get_stream()`
- `khorosjx.news.get_subscriber_groups()`

- `khorosjx.news.get_subscription_ids()`
- `khorosjx.news.rebuild_publication()`
- `khorosjx.news.retrieve_connection_info()`
- `khorosjx.news.update_publication()`
- `khorosjx.news.update_stream()`
- `khorosjx.news.verify_core_connection()`
- `khorosjx.spaces.get_browse_id()`
- `khorosjx.spaces.get_permitted_content_types()`
- `khorosjx.spaces.get_space_info()`
- `khorosjx.spaces.get_space_permissions()`
- `khorosjx.spaces.get_spaces_list_from_file()`
- `khorosjx.users.get_json_field()`
- `khorosjx.users.get_primary_email()`
- `khorosjx.users.get_user_id()`
- `khorosjx.users.retrieve_connection_info()`
- `khorosjx.users.verify_core_connection()`
- `khorosjx.users.__get_paginated_content_count()`
- `khorosjx.users.__validate_lookup_type()`

### Fixed

### Security

Fixes relating to security vulnerabilities.

- Updated the version of the bleach package in `requirements.txt` to be 3.1.2 to mitigate an identified [mutation XSS vulnerability](#) that was reported by GitHub.

### Documentation

Fixes to the documentation.

- Corrected a typo in the docstring for the `khorosjx.core.get_base_url()` function.



## 1.7.8 v2.4.1

Release Date: 2020-03-23

### Fixed

### Primary Modules

Additions to the *primary modules*.

- Added the `verify_core_connection()` function call to the `khorsjx.content.base.__convert_lookup_value()` function to resolve the `NameError: name 'base_url' is not defined` error.
- Added missing docstrings to the `khorsjx.content.ideas.get_ideas_for_space()` function.

## 1.7.9 v2.4.0

Release Date: 2020-03-16

### Added

### Primary Modules

Additions to the *primary modules*.

- Added the `khorsjx.content.ideas.get_ideas_for_space()` function.
- Added the `idea_fields` list to the `khorsjx.utils.classes.FieldLists` class.
- Added the `khorsjx.utils.version.warn_when_not_latest()` function call in the main `khorsjx` module.

### Supporting Modules

Additions to the *supporting modules*.

- Added the `khorsjx.utils.version.get_latest_stable()` function.
- Added the `khorsjx.utils.version.latest_version()` function.
- Added the `khorsjx.utils.version.warn_when_not_latest()` function.

### Changed

#### Primary Modules

Changes to the *primary modules*.

- Introduced the `all_fields` argument into the `khorsjx.content.base.get_paginated_content()` function.
- Updated the `datasets` dictionary in the `khorsjx.utils.classes.Content` class to include the `idea` key value pair.
- Updated the `khorsjx.groups.__get_paginated_groups()` function to use the `khorsjx.utils.df_utils.convert_dict_list_to_dataframe()` function instead of the deprecated `khorsjx.utils.core_utils.convert_dict_list_to_dataframe()` function.
- Removed obsolete `import re` statement from the `khorsjx.groups.__get_paginated_groups()` function.

### Fixed

#### Primary Modules

Fixes to the *primary modules*.

- Fixed a minor syntax issue in the `khorsjx.content.base.get_content_id()` function.

### Documentation

Changes to the documentation.

- Fixed a typo in the `README.md` file.

## 1.7.10 v2.3.1

**Release Date: 2020-02-24**

### Changed

#### General

- Adjusted the `python_requires` value in `setup.py` to reject version 3.8.1 and above as the `numpy` and `pandas` packages do not currently support that version.
- Upgraded the `bleach` package to version 3.1.1 to mitigate a security alert for a [mutation XSS](#) vulnerability and updated the `requirements.txt` file accordingly.

## 1.7.11 v2.3.0

Release Date: 2020-02-11

### Added

#### Primary Modules

Additions to the *primary modules*.

- Added the `khrosjx.news` module with the following functions:

- `khrosjx.news.verify_core_connection()`
- `khrosjx.news.get_all_publications()`
- `khrosjx.news.get_publication()`
- `khrosjx.news.delete_publication()`
- `khrosjx.news.get_subscription_data()`
- `khrosjx.news.get_subscription_ids()`
- `khrosjx.news.get_subscriber_groups()`
- `khrosjx.news.get_subscribers()`
- `khrosjx.news.rebuild_publication()`
- `khrosjx.news.get_stream()`
- `khrosjx.news.update_stream()`
- `khrosjx.news.delete_stream()`

#### Supporting Modules

Additions to the *supporting modules*.

- Added the `publication_fields`, `subscription_fields` and `stream_fields` lists to the `khrosjx.utils.classes.FieldLists` class.
- Added the `khrosjx.errors.exceptions.SubscriptionNotFoundError` exception class.

### Changed

#### Primary Modules

Changes to the *primary modules*.

- Added the `khrosjx.news` module to the `__all__` special variable in the primary `khrosjx` module.
- Updated the `khrosjx.init_module()` function to be compatible with the `khrosjx.news` module.
- Updated the `khrosjx.core.get_data()` function to include the `all_fields` argument. (False by default)
- Referenced the `khrosjx.errors.exceptions.POSTRequestError` exception class in the docstring for the `khrosjx.core.post_request_with_retries()` function.

- Referenced the `khorsjx.errors.exceptions.PUTRequestError` exception class in the docstring for the `khorsjx.core.put_request_with_retries()` function.

### Supporting Modules

Changes to the *supporting modules*.

- Added a `DeprecationWarning` to the `khorsjx.utils.core_utils.convert_dict_list_to_dataframe()` function.

### Fixed

### Primary Modules

Fixes to the *primary modules*.

- Changed the `json_payload` type to `dict` in the docstring for the `khorsjx.core.put_request_with_retries()` and `khorsjx.core.post_request_with_retries()` functions.

### Removed

### General

- Removed the `MANIFEST.in` file as the `VERSION` file is no longer used.

## 1.7.12 v2.2.0

Release Date: 2020-01-22

### Added

### Primary Modules

Additions to the *primary modules*.

- Added the `khorsjx.core.get_base_url()` function.
- Added the `khorsjx.core.get_query_url()` function.
- Added the `khorsjx.core.__get_filter_syntax()` function.
- Added the `khorsjx.content.videos.get_video_id()` function.
- Added the `khorsjx.content.videos.get_native_videos_for_space()` function.
- Added the `khorsjx.content.videos.find_video_attachments()` function.
- Added the `khorsjx.content.videos.__construct_url_from_id()` function.
- Added the `khorsjx.content.videos.check_if_embedded()` function.

- Added the `khorsjx.content.videos.get_video_dimensions()` function.

## Supporting Modules

Additions to the *supporting modules*.

- Added the `video_fields` list to the `khorsjx.utils.classes.FieldLists` class.
- Added the `video` key value pair to the `datasets` dictionary within the `khorsjx.utils.classes.Content` class.
- Added the `khorsjx.errors.handlers.bad_lookup_type()` function.
- Added the `khorsjx.errors.exceptions.ContentNotFoundError` exception.

## Changed

### General

- Updated `setup.py` to define `version` in the `setuptools.setup()` function using the `__version__` value from `khorsjx.utils.version()`.

## Primary Modules

Changes to the *primary modules*.

- Updated the `khorsjx.core.get_paginated_results()` function to allow additional filters to be supplied as a tuple containing the element and criteria (e.g. ('type', 'video') or a list of tuples for multiple filters.
- Made minor adjustment to the `khorsjx.groups.get_all_groups()` function.
- Updated the `khorsjx.content.videos.get_content_id()` function to allow a URL or Video ID to be supplied.
- Updated the `khorsjx.content.base.get_content_id()` function to raise the `khorsjx.errors.exceptions.ContentNotFoundError` exception instead of a generic `KeyError` exception.

## 1.7.13 v2.1.0

**Release Date: 16 Jan 2020**

### Added

#### General

- Added the `__version__` global variable in the `khorsjx(__init__.py)` module.

## Primary Modules

Additions to the *primary modules*.

- Added the `khrosjx.core.delete()` function.
- Added the `khrosjx.content.docs.create_document()` function.
- Added the `khrosjx.content.docs.delete_document()` function.
- Added the `khrosjx.places.base.get_uri_for_id()` function.
- Added the `khrosjx.content.docs.get_url_for_id()` function.

## Supporting Modules

Additions to the *supporting modules*.

- Added the `khrosjx.utils.helper.__get_modules_to_import()` function.
- Added the `khrosjx.errors.exceptions.DatasetNotFoundError` exception class.
- Added the `uri_dataset_mapping` and `security_group_uri_map` dictionaries to the `khrosjx.utils.classes.Content` class.
- Added the `khrosjx.utils.core_utils.identify_dataset()` function with the accompanying internal functions:
  - `khrosjx.utils.core_utils.__get_security_group_dataset()`
  - `khrosjx.utils.core_utils.__get_invite_dataset()`
  - `khrosjx.utils.core_utils.__get_metadata_dataset()`
  - `khrosjx.utils.core_utils.__get_moderation_dataset()`
  - `khrosjx.utils.core_utils.__get_search_dataset()`
  - `khrosjx.utils.core_utils.__get_support_center_dataset()`
  - `khrosjx.utils.core_utils.__get_tile_dataset()`
- Added the `khrosjx.utils.version` module containing the source `__version__` and the following functions:
  - `khrosjx.utils.version.get_full_version()`
  - `khrosjx.utils.version.get_major_minor_version()`

## Changed

### General

- Updated `setup.py` to obtain the version value from the `khrosjx.utils.version()` function.
- Updated `docs/conf.py` to obtain the version value from the `khrosjx.utils.version()` function.

## Primary Modules

Changes to the *primary modules*.

- Updated the `khorsjx.places` module to proactively import the `khorsjx.places.base`, `khorsjx.places.blogs` and `khorsjx.places.spaces` modules.
- Updated the `khorsjx.content.docs.get_content_id()` function to accept both URLs and Document IDs.
- Updated the `khorsjx.init_helper()` function to handle the modules section of the YAML configuration file.
- Added error handling for invalid file types in the `khorsjx.init_helper()` function.
- Updated the `khorsjx.init_module()` function to properly handle the all string within an iterable.
- Updated the `khorsjx.core.get_fields_from_api_response()` to reference the datasets dictionary that was moved into the `khorsjx.utils.classes.Content` class.

## Supporting Modules

Changes to the *supporting modules*.

- Updated the `khorsjx.utils.helper.parse_helper_cfg()` and `khorsjx.utils.helper.retrieve_helper_settings()` functions to leverage the `khorsjx.utils.helper.__get_modules_to_import()` function.
- Added the `accepted_import_modules` and `all_modules` lists to the `khorsjx.utils.helper.HelperParsing` class.
- Moved the datasets dictionary from the `khorsjx.core.get_fields_from_api_response()` function into the `khorsjx.utils.classes.Content` class.

## Documentation

Changes to the documentation.

- Adjusted the docstring for the `khorsjx.errors.exceptions.InvalidDatasetError` exception class to differentiate it from the `khorsjx.errors.exceptions.DatasetNotFoundError` exception class.

## Examples

Changes to the example files found in the `examples` directory within the GitHub repository.

- Added the modules section to the `khorsjx_helper.yml` file.

## 1.7.14 v2.0.0

Release Date: 8 Jan 2020

### Added

### Primary Modules

Additions to the *primary modules*.

- Created the ***khorosjx.places*** module with the following sub-modules and functions:
  - ***khorosjx.places.base***
    - \* *khorosjx.places.base.verify\_core\_connection()*
    - \* *khorosjx.places.base.get\_browse\_id()*
    - \* *khorosjx.places.base.get\_place\_id()*
    - \* *khorosjx.places.base.get\_place\_info()*
    - \* *khorosjx.places.base.get\_places\_list\_from\_file()*
  - ***khorosjx.places.spaces***
    - \* *khorosjx.places.spaces.verify\_core\_connection()*
    - \* *khorosjx.places.spaces.get\_space\_info()*
    - \* *khorosjx.places.spaces.get\_permitted\_content\_types()*
    - \* *khorosjx.places.spaces.get\_space\_permissions()*
  - ***khorosjx.places.blogs***
    - \* *khorosjx.places.blogs.verify\_core\_connection()*
    - \* *khorosjx.places.blogs.get\_blog\_info()*
- Created the ***khorosjx.content*** module with the following sub-modules and functions:
  - ***khorosjx.content.base***
    - \* *khorosjx.content.base.verify\_core\_connection()*
    - \* *khorosjx.content.base.get\_content\_id()*
    - \* *khorosjx.content.base.\_\_convert\_lookup\_value()*
    - \* *khorosjx.content.base.\_\_trim\_attachments\_info()*
  - ***khorosjx.content.docs***
    - \* *khorosjx.content.docs.verify\_core\_connection()*
    - \* *khorosjx.content.docs.get\_content\_id()*
    - \* *khorosjx.content.docs.overwrite\_doc\_body()*
    - \* *khorosjx.content.docs.get\_document\_info()*
    - \* *khorosjx.content.docs.get\_document\_attachments()*
  - ***khorosjx.content.events***
    - \* *khorosjx.content.events.verify\_core\_connection()*



- \* `khorosjx.content.events.get_content_id()`
  - **`khorosjx.content.ideas`**
    - \* `khorosjx.content.ideas.verify_core_connection()`
    - \* `khorosjx.content.ideas.get_content_id()`
  - **`khorosjx.content.threads`**
    - \* `khorosjx.content.threads.verify_core_connection()`
    - \* `khorosjx.content.threads.get_content_id()`
  - **`khorosjx.content.videos`**
    - \* `khorosjx.content.videos.verify_core_connection()`
    - \* `khorosjx.content.videos.get_content_id()`
- Added the `khorosjx.content.videos.download_video()` function.

## Supporting Modules

Additions to the *supporting modules*.

- Added the `khorosjx.utils.core_utils.print_if_verbose()` function.
- Added the `permitted_video_file_types` list to the `khorosjx.utils.classes.Content` class.

## Documentation

Additions to the documentation.

- Added “Return to Top” hyperlinks on the *primary modules*, *supporting modules* and *change log* pages.
- Added the `khorosjx.utils.df_utils` and `khorosjx.errors` modules to the *supporting modules* page.

## Changed

### General

- Changed the Development Status PyPI classifier in the `setup.py` file to be 5 - Production/Stable.

## Primary Modules

Changes to the *primary modules*.

- Included the `blog` and `place` datasets in the dictionary within the `khorosjx.core.get_fields_from_api_response()` function.

### Supporting Modules

Changes to the *supporting modules*.

- Added `df_utils` and `helper` to `__all__` in the `khrosjx.utils` module.

### Documentation

Changes to the documentation.

- Updated the *Primary Modules* page to show functions within the `__init__.py` files.
- Added deprecated directives to docstrings of deprecated functions.
- Adjusted the docstrings on the `khrosjx.init_helper()` function.
- Restructured the table of contents at the top of the *Supporting Modules* page.
- Updated the short-term and long-term items on the *Roadmap* page.

### Fixed

#### Primary Modules

Fixes applied in the *primary modules*.

- Fixed the try/except in the `khrosjx.content.docs.get_document_attachments()` function to account for both `KeyError` and `IndexError` exceptions instead of just the latter.

#### Supporting Modules

Fixes applied in the *supporting modules*.

- Fixed the `khrosjx.errors.handlers.check_api_response()` function so that a 502 response code displays a simple `Site Temporarily Unavailable` error rather than displaying the entire raw HTML response from the API.

### Documentation

Fixes applied to the documentation.

- Fixed an issue with the header block docstring for the `khrosjx.utils.classes` module.

### Deprecated

#### Primary Modules

Deprecations in the *primary modules*.

- Deprecated and moved the functions below to the `khrosjx/content/__init__.py` file from the removed `khrosjx/content.py` file. (The deprecated functions will be removed in v3.0.0.)
  - `khrosjx.content.get_content_id()`
  - `khrosjx.content.override_doc_body()`

- `khorsjx.content.__convert_lookup_value()`
- `khorsjx.content.get_document_info()`
- `khorsjx.content.__trim_attachments_info()`
- `khorsjx.content.get_document_attachments()`
- Deprecated the `khorsjx.spaces.get_space_info()` function.
- Deprecated the `khorsjx.spaces.get_place_id()` function.
- Deprecated the `khorsjx.spaces.get_browse_id()` function.
- Deprecated the `khorsjx.spaces.__verify_browse_id()` function.
- Deprecated the `khorsjx.spaces.get_spaces_list_from_file()` function.
- Deprecated the `khorsjx.spaces.get_permitted_content_types()` function.
- Deprecated the `khorsjx.spaces.get_space_permissions()` function.
- Deprecated the `khorsjx.spaces.__get_unique_permission_fields()` function.
- Deprecated the `khorsjx.spaces.__generate_permissions_dataframe()` function.

## Removed

### Primary Modules

Removals in the *primary modules*.

- The `khorsjx.content` module has been removed. (See the previous sections for additional context.)

*Return to Top*

## 1.7.15 v1.7.0

Release Date: 2019-12-21

## Added

### Primary Modules

Additions to the *primary modules*.

- Added the `khorsjx.spaces.get_spaces_list_from_file()` function.
-

## Supporting Modules

Additions to the *supporting modules*.

- Added the new `khrosjx.utils.df_utils` module to house all pandas-related functions and utilities.
- Added the `khrosjx.utils.df_utils.convert_dict_list_to_dataframe()` function. (Moved from the `khrosjx.utils.core_utils` module.)
- Added the `khrosjx.utils.df_utils.import_csv()` function.
- Added the `khrosjx.utils.df_utils.import_excel()` function.
- Added the `khrosjx.errors.exceptions.InvalidFileTypeError` exception class.

## Changed

### Primary Modules

Changes to the *primary modules*.

- Updated the `khrosjx.core.get_fields_from_api_response()` function to handle the `resources.html.ref` field.

---

## Supporting Modules

Changes to the *supporting modules*.

- Updated the `khrosjx.utils.core_utils.convert_dict_list_to_dataframe()` function to leverage the `khrosjx.utils.df_utils.convert_dict_list_to_dataframe()` function in the new module.
- Updated the `place_fields` list in the `khrosjx.utils.classes.FieldLists` class to include the `resources.html.ref` field.

## Fixed

### Primary Modules

Fixes in the *primary modules*.

- Fixed a logic error in the `khrosjx.core.get_fields_from_api_response()` function which was preventing custom-curated fields for nested values from returning properly.

## Deprecated

### Supporting Modules

Deprecations in the *supporting modules*.

- Deprecated the `khrosjx.utils.core_utils.convert_dict_list_to_dataframe()` function as it now resides in the `khrosjx.utils.df_utils.convert_dict_list_to_dataframe()` function within the new module.

[Return to Top](#)

## 1.7.16 v1.6.0

**Release Date: 2019-12-17**

### Added

#### Primary Modules

Additions to the *primary modules*.

- Added the `khorsjx.spaces.get_permitted_content_types()` function.
  - Added the internal `khorsjx.spaces._verify_browse_id()` function.
  - Added the `khorsjx.spaces.get_space_permissions()` function.
  - Added the internal `khorsjx.spaces.__get_unique_permission_fields()` function.
  - Added the internal `khorsjx.spaces.__generate_permissions_dataframe()` function.
- 

#### Supporting Modules

Additions to the *supporting modules*.

- Added the `khorsjx.errors.handlers.check_json_for_error()` function.
  - Added the `khorsjx.errors.exceptions.NotFoundResponseError` exception class.
  - Added the `khorsjx.errors.exceptions.SpaceNotFoundError` exception class.
- 

#### Documentation

Addition to the documentation in this release.

- Added a [Roadmap](#) page to list upcoming enhancements and changes.

### Changed

#### Primary Modules

Changes to the *primary modules*.

- Updated the `khorsjx.init_module()` function to allow the `all` string to be passed which imports all modules.
-

### Supporting Modules

Changes to the *supporting modules*.

- Added the optional `column_names` keyword argument in the `khorsjx.utils.core_utils.convert_dict_list_to_dataframe()` function.
- 

### Documentation

Changes to the documentation in this release.

- Changed the project name from KhorosJX to Khoros JX Python Library in the `conf.py` script.
- Made adjustments to the `toctree` directives on the *index* page.
- Changed the **Latest Release** badge in the `README.md` file to be labeled **Latest Stable Release** instead.

### Fixed

#### Primary Modules

Fixes to the *primary modules*.

- Removed `helper` from `__all__` in the *Init Module* (`khorsjx`).
  - Fixed how to query URL was generated in the `khorsjx.core.get_api_info()` function.
  - Fixed a docstring error in the `khorsjx.core.put_request_with_retries()` function.
  - Fixed a minor docstring error in `khorsjx.groups.add_user_to_group()` function.
  - Fixed a docstring error in the `khorsjx.users.get_people_followed()` function.
  - Added the missing `verify_core_connection()` function call in `khorsjx.users.get_recent_logins()` function. (See [Issue #1](#))
- 

### Supporting Modules

Fixes to the *supporting modules*.

- Removed the `**kwargs` argument in the `super()` call within all custom exceptions.
-

## Documentation

Fixes in the documentation in this release.

- Fixed minor typos in the [README.md](#) file.
- Fixed a minor typo in the `khorsjx.utils.core_utils.convert_dict_list_to_dataframe()` function docstring.

[Return to Top](#)

## 1.7.17 v1.5.0

**Release Date: 2019-12-05**

### Added

#### Primary Modules

Additions to the *primary modules*.

- Added to the `datasets` dictionary and a “field not found” error message in the `khorsjx.core.get_fields_from_api_response()` function.
  - Added the `khorsjx.content.get_document_info()` function.
  - Added the `khorsjx.content.get_document_attachments()` function.
  - Added the internal `khorsjx.content.__convert_lookup_value()` function.
  - Added the internal `khorsjx.content.__trim_attachments_info()` function.
  - Added the `khorsjx.spaces.get_space_info()` function.
  - Added the `khorsjx.spaces.get_place_id()` and `khorsjx.spaces.get_browse_id()` functions.
  - Added the internal `khorsjx.users.__validate_lookup_type()` function.
- 

#### Supporting Modules

Additions to the *supporting modules*.

- Added the `place_fields` and `document_fields` lists to the `khorsjx.utils.classes.FieldLists` class.
  - Added the `khorsjx.errors.exceptions.LookupMismatchError` exception class.
  - Added the `khorsjx.errors.exceptions.CurrentlyUnsupportedError` exception class.
-

### Documentation

- Added the section on how to *Obtain the primary email address* within the *Managing Users* page now that the function is available.

### Changed

- Updated the `khrosjx.users.get_user_id()` and `khrosjx.users.get_username()` functions to leverage the internal `khrosjx.users.__validate_lookup_type()` function.
- Updated the `khrosjx.users.get_user_id()` function to confirm that an email address was provided if the 'email' lookup type is supplied.
- Updated the header block docstring at the top of the `khrosjx.spaces()` module.
- Updated the header block docstring at the top of the `khrosjx.errors.exceptions()` module.

### Fixed

- Fixed a variable name error in the `khrosjx.users.get_username()` function.

*Return to Top*

## 1.7.18 v1.4.0

**Release Date: 2019-11-30**

### Added

#### Primary Modules

Additions to the *primary modules*.

- Added the `khrosjx.core.get_paginated_results()` function.
  - Added docstrings to the `khrosjx.core.get_fields_from_api_response()` function.
  - Added the `khrosjx.groups.get_group_memberships()` function.
-



## Supporting Modules

Additions to the *supporting modules*.

- Added the `khorsjx.utils.core_utils.add_to_master_list()` function.
- Added the `khorsjx.utils.core_utils.convert_single_pair_dict_list()` function.
- Added docstrings to the `khorsjx.utils.core_utils.convert_dict_list_to_dataframe()` function.
- Added the new `khorsjx.utils.classes.Groups` class which contains the `membership_types` and `user_type_mapping` dictionaries.
- Added the `people_fields` list to the `khorsjx.utils.classes.FieldLists` class.

## Changed

### Supporting Modules

Changes in the *supporting modules*.

- Added a TODO note to move the `khorsjx.utils.core_utils.eprint()` function to the `khorsjx.errors.handlers` module.

## Documentation

- Changed the structure of the changelog to be more concise. (i.e. less sub-sections)

## Developer Changes

- Changed the **Development Status classifier** from Alpha to Beta in the `setup.py` file.

## Removed

### Primary Modules

Removals in the *primary modules*.

- Removed the nested `add_to_master_list()` function from within the `khorsjx.groups.get_all_groups()` function.

*Return to Top*

## 1.7.19 v1.3.0

Release Date: 2019-11-27

### Added

#### Primary Modules

Additions to the *primary modules*.

- Added the `init_helper()` function to the *Init Module* (*khrosjx*) to initialize a helper configuration file.
- 

#### Supporting Modules

Additions to the *supporting modules*.

- Added the new *Helper Module* (*khrosjx.utils.helper*) which allows a “helper” configuration file to be imported and parsed to facilitate the use of the library (e.g. defining the base URL and API credentials) and defining additional settings.
  - Added the *khrosjx.errors.exceptions.InvalidHelperArgumentsError* exception class.
  - Added the *khrosjx.errors.exceptions.HelperFunctionNotFoundError* exception class.
- 

#### Examples

- Added a new *examples* directory containing the *khrosjx\_helper.yml* file which demonstrates how the helper configuration file should be formatted.
- 

#### Documentation

- Added the *Using the Helper Utility* page to address the new Helper Utility that was introduced.
- Added the *Helper Module* (*khrosjx.utils.helper*) to the *Supporting Modules* page.
- Added a “See Also” section mentioning the Helper Utility on the *Core Functionality* page.

*Return to Top*

## 1.7.20 v1.2.0

Release Date: 2019-11-24

### Added

#### Primary Modules

Additions to the *primary modules*.

- Added the `khrosjx.core.__api_request_with_payload()` function to leverage for both POST and PUT requests.
  - Added the `khrosjx.core.post_request_with_retries()` function for POST requests, which leverages the private function above.
  - Added the `khrosjx.groups.add_user_to_group()` function.
  - Added the `khrosjx.groups.check_user_membership()` function.
- 

#### Supporting Modules

Additions to the *supporting modules*.

- Added the `khrosjx.utils.core_utils.eprint()` function to print error messages to stderr and onscreen.
  - Added the `khrosjx.errors.exceptions.POSTRequestError`, `khrosjx.errors.exceptions.InvalidScopeError`, `khrosjx.errors.exceptions.InvalidLookupTypeError`, `khrosjx.errors.exceptions.InvalidEndpointError`, `khrosjx.errors.exceptions.InvalidRequestTypeError` and `khrosjx.errors.exceptions.APIConnectionError` exception classes.
- 

#### Documentation

- Added the *Core Functionality* page with instructions on leveraging the core functionality of the library. (Page is still a work in progress)
- Added the *Managing Users* page with instructions on managing users by leveraging the library. (Page is still a work in progress)
- Added the Basic Usage page with the intent of inserting it into more than one page.

### Changed

#### General

- Updated the classifiers in `setup.py` to specifically reference Python 3.6, 3.7 and 3.8.
-

### Primary Modules

Changes to existing functions in the *primary modules*.

- Updated the `khrosjx.core.get_data()` function to accept username as an identifier for the people endpoint.
  - Updated the `khrosjx.core.get_request_with_retries()` function to include the `return_json` optional argument. (Disabled by default)
  - Refactored the `khrosjx.core.put_request_with_retries()` function to leverage the `khrosjx.core.__api_request_with_payload()` function.
  - Updated the `khrosjx.users.get_user_id()` function to accept a username as well as an email address.
- 

### Supporting Modules

Changes to existing functions in the *supporting modules*.

- Expanded the functionality of the `khrosjx.errors.handlers.check_api_response()` function.
- 

### Documentation

- Updated the *Introduction* page to insert the *Basic Usage* content.
- Added the Basic Usage page with the intent of inserting it into more than one page.

*Return to Top*

## 1.7.21 v1.1.1

**Release Date: 2019-11-23**

### Added

- Added default messages to all of the exception classes in the *Exceptions Module* (`khrosjx.errors.exceptions`).
- Added docstrings to the `khrosjx.content.override_doc_body()` function.

## Changed

- Updated the build workflow (`pythonpackage.yml`) to also test Python 3.8 for compatibility.
- Changed the structure of the change log to match the best practices from [keepachangelog.com](https://keepachangelog.com).
- Made minor [PEP8](https://pep8.org) compliance edits to the *Classes Module* (`khorsjx.utils.classes`).

## Removed

- The `khorsjx.errors.raise_exceptions()` function is no longer necessary as the exception classes now have default messages and it has been removed from the `khorsjx.errors` module (`__init__.py`) and the *Handlers Module* (`khorsjx.errors.handlers`).
- Removed the `khorsjx.errors.exceptions.ExceptionMapping` and `khorsjx.errors.exceptions.ExceptionGrouping` classes as they are no longer used.

*[Return to Top](#)*

## 1.7.22 v1.1.0

**Release Date: 2019-11-22**

## Added

### Primary Modules

Additions to the *primary modules*.

- Added the `khorsjx.core.put_request_with_retries()` function.
  - Added the `ignore_exceptions` parameter in the `khorsjx.core.get_data()` function and replaced the built-in `ValueError` exception with the custom `khorsjx.errors.exceptions.GETRequestError` exception class.
  - Added the `khorsjx.core.get_fields_from_api_response()` function.
  - Added the `khorsjx.content.override_doc_body()` function.
  - Added the `khorsjx.groups.get_user_memberships()` function.
  - Added the `khorsjx.groups.get_group_info()` function.
  - Added the `khorsjx.groups.get_all_groups()` function.
  - Added the `khorsjx.users.get_recent_logins()` function.
-

## Supporting Modules

Additions to the *supporting modules*.

- Added the `khorsjx.utils.core_utils.convert_dict_list_to_dataframe()` function.
- Added the `khorsjx.errors.exceptions.ContentPublishError`, `khorsjx.errors.exceptions.BadCredentialsError`, `khorsjx.errors.exceptions.GETRequestError` and `khorsjx.errors.exceptions.PUTRequestError` exception classes.
- Added the new *Handlers Module* (`khorsjx.errors.handlers`) which includes a new `khorsjx.errors.handlers.check_api_response()` function.
- Created the new *Tests Module* (`khorsjx.utils.tests`) for unit tests to leverage with `pytest`.

## Changed

- Updated the *Supporting Modules* documentation page to reference the new modules.
- Reformatted the *Change Log* documentation page to follow the [Sphinx Style Guide](#).

## Deprecated

- The `raise_exception()` function in the `khorsjx.errors` module now displays a `DeprecationWarning` as it has been moved into the new *Handlers Module* (`khorsjx.errors.handlers`).
- Added a `PendingDeprecationWarning` warning on the `khorsjx.errors.handlers.raise_exception()` function as it will be deprecated in a future release. (See *v1.1.1*)

## Fixed

- Added the `khorsjx.core.verify_connection()` function call to the `khorsjx.core.get_data()` function.

[Return to Top](#)

## 1.7.23 v1.0.1.post1

**Release Date: 2019-11-19**

## Changed

- Created a new *Introduction* page with the existing home page content and added a *Navigation* (i.e. Table of Contents) to the home page.
- Changed all *auxilliary modules* references to be *supporting modules* instead.
- Added a *Reporting Issues* section to the *Introduction* page and to the `README` file.

[Return to Top](#)

### 1.7.24 v1.0.1

Release Date: 2019-11-19

#### Changed

- Removed the version from the individual module header blocks as all will adhere to the primary versioning.

#### Fixed

- Added missing `from . import core` in the `khrosjx.admin`, `khrosjx.groups` and `khrosjx.spaces` modules.

*[Return to Top](#)*

## 1.8 Roadmap

This page lists enhancements that are planned for upcoming releases in both the short-term and long-term.

**Attention:** The enhancements listed on this page are subject to change without notice.

### 1.8.1 Short-Term

The following list includes enhancements, changes and other items currently planned for short-term release.

- Create the `khrosjx.content.base.get_content_info()` function.
- Create the `khrosjx.content.base.get_content_for_space()` function.
- Create the `khrosjx.content.base.get_content_count_for_space()` function.
- Create the `khrosjx.content.threads.get_attachments()` function.

### 1.8.2 Long-Term

The following list includes enhancements, changes and other items currently planned for long-term release.

- Support for OAuth authentication as opposed to only basic authentication.
- Integrate the library with the standard `logging` package.





## MODULE INDEX AND SEARCH

- [genindex](#)
- [modindex](#)
- [search](#)



## PYTHON MODULE INDEX

### k

- `khorosjx`, 18
- `khorosjx.admin`, 25
- `khorosjx.content`, 26
  - `khorosjx.content.base`, 28
  - `khorosjx.content.docs`, 30
  - `khorosjx.content.events`, 34
  - `khorosjx.content.ideas`, 35
  - `khorosjx.content.threads`, 36
  - `khorosjx.content.videos`, 37
- `khorosjx.core`, 19
- `khorosjx.errors`, 69
  - `khorosjx.errors.exceptions`, 70
  - `khorosjx.errors.handlers`, 72
- `khorosjx.groups`, 40
- `khorosjx.news`, 44
- `khorosjx.places`, 48
  - `khorosjx.places.base`, 49
  - `khorosjx.places.blogs`, 51
  - `khorosjx.places.spaces`, 53
- `khorosjx.spaces`, 55
- `khorosjx.users`, 58
- `khorosjx.utils.classes`, 68
- `khorosjx.utils.core_utils`, 61
- `khorosjx.utils.df_utils`, 64
- `khorosjx.utils.helper`, 65
- `khorosjx.utils.tests.test_init_module`, 66
- `khorosjx.utils.version`, 67



## A

`add_to_master_list()` (in module `khorojsx.utils.core_utils`), 62  
`add_user_to_group()` (in module `khorojsx.groups`), 40  
`APIConnectionError`, 70

## B

`bad_lookup_type()` (in module `khorojsx.errors.handlers`), 72  
`BadCredentialsError`, 70

## C

`check_api_response()` (in module `khorojsx.errors.handlers`), 72  
`check_if_embedded()` (in module `khorojsx.content.videos`), 37  
`check_json_for_error()` (in module `khorojsx.errors.handlers`), 72  
`check_user_membership()` (in module `khorojsx.groups`), 41  
`connect()` (in module `khorojsx.core`), 19  
`Content` (class in `khorojsx.utils.classes`), 68  
`ContentNotFoundError`, 70  
`ContentPublishError`, 70  
`convert_dict_list_to_dataframe()` (in module `khorojsx.utils.core_utils`), 62  
`convert_dict_list_to_dataframe()` (in module `khorojsx.utils.df_utils`), 64  
`convert_dict_to_json()` (in module `khorojsx.utils.core_utils`), 62  
`convert_single_pair_dict_list()` (in module `khorojsx.utils.core_utils`), 62  
`create_document()` (in module `khorojsx.content.docs`), 30  
`CredentialsUnpackingError`, 70  
`CurrentlyUnsupportedError`, 70

## D

`DatasetNotFoundError`, 70  
`delete()` (in module `khorojsx.core`), 20  
`delete_document()` (in module `khorojsx.content.docs`), 31

`delete_publication()` (in module `khorojsx.news`), 44  
`delete_stream()` (in module `khorojsx.news`), 44  
`download_video()` (in module `khorojsx.content.videos`), 38

## E

`ensure_absolute_url()` (in module `khorojsx.core`), 20  
`eprint()` (in module `khorojsx.utils.core_utils`), 62

## F

`FieldLists` (class in `khorojsx.utils.classes`), 68  
`filter_subscriptions_by_id()` (in module `khorojsx.news`), 44  
`find_video_attachments()` (in module `khorojsx.content.videos`), 38

## G

`get_all_groups()` (in module `khorojsx.groups`), 41  
`get_all_publications()` (in module `khorojsx.news`), 45  
`get_api_info()` (in module `khorojsx.core`), 20  
`get_api_version()` (in module `khorojsx.core`), 20  
`get_base_url()` (in module `khorojsx.core`), 21  
`get_blog_info()` (in module `khorojsx.places.blogs`), 52  
`get_browse_id()` (in module `khorojsx.places.base`), 49  
`get_browse_id()` (in module `khorojsx.spaces`), 55  
`get_connection_info()` (in module `khorojsx.core`), 21  
`get_content_id()` (in module `khorojsx.content`), 27  
`get_content_id()` (in module `khorojsx.content.base`), 29  
`get_content_id()` (in module `khorojsx.content.docs`), 31  
`get_content_id()` (in module `khorojsx.content.events`), 34  
`get_content_id()` (in module `khorojsx.content.ideas`), 35  
`get_content_id()` (in module `khorojsx.content.threads`), 36  
`get_content_id()` (in module `khorojsx.content.videos`), 38  
`get_data()` (in module `khorojsx.core`), 21

`get_document_attachments()` (in module `khorojx.content`), 27

`get_document_attachments()` (in module `khorojx.content.docs`), 32

`get_document_info()` (in module `khorojx.content`), 27

`get_document_info()` (in module `khorojx.content.docs`), 32

`get_fields_from_api_response()` (in module `khorojx.core`), 21

`get_format_syntax()` (in module `khorojx.utils.core_utils`), 62

`get_full_version()` (in module `khorojx.utils.version`), 67

`get_group_info()` (in module `khorojx.groups`), 42

`get_group_memberships()` (in module `khorojx.groups`), 42

`get_ideas_for_space()` (in module `khorojx.content.ideas`), 35

`get_json_field()` (in module `khorojx.users`), 58

`get_latest_stable()` (in module `khorojx.utils.version`), 67

`get_major_minor_version()` (in module `khorojx.utils.version`), 67

`get_native_videos_for_space()` (in module `khorojx.content.videos`), 38

`get_paginated_content()` (in module `khorojx.content.base`), 29

`get_paginated_results()` (in module `khorojx.core`), 22

`get_people_followed()` (in module `khorojx.users`), 58

`get_permitted_content_types()` (in module `khorojx.places.spaces`), 53

`get_permitted_content_types()` (in module `khorojx.spaces`), 56

`get_place_id()` (in module `khorojx.places.base`), 49

`get_place_id()` (in module `khorojx.spaces`), 56

`get_place_info()` (in module `khorojx.places.base`), 50

`get_places_list_from_file()` (in module `khorojx.places.base`), 50

`get_platform_version()` (in module `khorojx.core`), 23

`get_primary_email()` (in module `khorojx.users`), 59

`get_profile_url()` (in module `khorojx.users`), 59

`get_publication()` (in module `khorojx.news`), 45

`get_query_url()` (in module `khorojx.core`), 23

`get_recent_logins()` (in module `khorojx.users`), 59

`get_request_with_retries()` (in module `khorojx.core`), 23

`get_space_content_permissions()` (in module `khorojx.places.spaces`), 53

`get_space_info()` (in module `khorojx.places.spaces`), 54

`get_space_info()` (in module `khorojx.spaces`), 56

`get_space_permissions()` (in module `khorojx.places.spaces`), 54

`get_space_permissions()` (in module `khorojx.spaces`), 57

`get_spaces_list_from_file()` (in module `khorojx.spaces`), 57

`get_stream()` (in module `khorojx.news`), 45

`get_subscriber_groups()` (in module `khorojx.news`), 46

`get_subscriber_ids()` (in module `khorojx.news`), 46

`get_subscribers()` (in module `khorojx.news`), 46

`get_subscription_data()` (in module `khorojx.news`), 47

`get_subscription_ids()` (in module `khorojx.news`), 47

`get_timestamp()` (in module `khorojx.utils.core_utils`), 63

`get_uri_for_id()` (in module `khorojx.places.base`), 50

`get_url_for_id()` (in module `khorojx.content.docs`), 32

`get_user_content_count()` (in module `khorojx.users`), 59

`get_user_id()` (in module `khorojx.users`), 59

`get_user_memberships()` (in module `khorojx.groups`), 43

`get_username()` (in module `khorojx.users`), 60

`get_video_dimensions()` (in module `khorojx.content.videos`), 39

`get_video_id()` (in module `khorojx.content.videos`), 39

`get_video_info()` (in module `khorojx.content.videos`), 39

`GETRequestError`, 70

`Groups` (class in `khorojx.utils.classes`), 68

## H

`HelperFunctionNotFoundError`, 70

`HelperParsing` (class in `khorojx.utils.helper`), 66

## I

`identify_dataset()` (in module `khorojx.utils.core_utils`), 63

`import_csv()` (in module `khorojx.utils.df_utils`), 64

`import_excel()` (in module `khorojx.utils.df_utils`), 65

`import_yaml_file()` (in module `khorojx.utils.helper`), 66

`IncompleteCredentialsError`, 70

`init_helper()` (in module `khorojx`), 18

`init_module()` (in module `khorojx`), 19

`init_module_operation()` (in module `khorojx.utils.tests.test_init_module`), 67

`InvalidDatasetError`, 70

InvalidEndpointError, 70  
 InvalidFileTypeError, 70  
 InvalidHelperArgumentsError, 71  
 InvalidKhorosJXModuleError, 71  
 InvalidLookupTypeError, 71  
 InvalidRequestTypeError, 71  
 InvalidScopeError, 71

## K

khorosjx  
     module, 18  
 khorosjx.admin  
     module, 25  
 khorosjx.content  
     module, 26  
 khorosjx.content.base  
     module, 28  
 khorosjx.content.docs  
     module, 30  
 khorosjx.content.events  
     module, 34  
 khorosjx.content.ideas  
     module, 35  
 khorosjx.content.threads  
     module, 36  
 khorosjx.content.videos  
     module, 37  
 khorosjx.core  
     module, 19  
 khorosjx.errors  
     module, 69  
 khorosjx.errors.exceptions  
     module, 70  
 khorosjx.errors.handlers  
     module, 72  
 khorosjx.groups  
     module, 40  
 khorosjx.news  
     module, 44  
 khorosjx.places  
     module, 48  
 khorosjx.places.base  
     module, 49  
 khorosjx.places.blogs  
     module, 51  
 khorosjx.places.spaces  
     module, 53  
 khorosjx.spaces  
     module, 55  
 khorosjx.users  
     module, 58  
 khorosjx.utils.classes  
     module, 68  
 khorosjx.utils.core\_utils

    module, 61  
 khorosjx.utils.df\_utils  
     module, 64  
 khorosjx.utils.helper  
     module, 65  
 khorosjx.utils.tests.test\_init\_module  
     module, 66  
 khorosjx.utils.version  
     module, 67  
 KhorosJXError, 71

## L

latest\_version() (*in module khorosjx.utils.version*),  
     68  
 LookupMismatchError, 71

## M

MissingBaseUrlError, 71  
 module  
     khorosjx, 18  
     khorosjx.admin, 25  
     khorosjx.content, 26  
     khorosjx.content.base, 28  
     khorosjx.content.docs, 30  
     khorosjx.content.events, 34  
     khorosjx.content.ideas, 35  
     khorosjx.content.threads, 36  
     khorosjx.content.videos, 37  
     khorosjx.core, 19  
     khorosjx.errors, 69  
     khorosjx.errors.exceptions, 70  
     khorosjx.errors.handlers, 72  
     khorosjx.groups, 40  
     khorosjx.news, 44  
     khorosjx.places, 48  
     khorosjx.places.base, 49  
     khorosjx.places.blogs, 51  
     khorosjx.places.spaces, 53  
     khorosjx.spaces, 55  
     khorosjx.users, 58  
     khorosjx.utils.classes, 68  
     khorosjx.utils.core\_utils, 61  
     khorosjx.utils.df\_utils, 64  
     khorosjx.utils.helper, 65  
     khorosjx.utils.tests.test\_init\_module, 66  
     khorosjx.utils.version, 67

## N

NoCredentialsError, 71  
 NotFoundResponseError, 71

## O

overwrite\_doc\_body() (*in module khorosjx.content*),  
     28

`overwrite_doc_body()` (in module `khorosjx.content.docs`), 33

## P

`parse_helper_cfg()` (in module `khorosjx.utils.helper`), 66

`parse_user_fields()` (in module `khorosjx.users`), 60

`Platform` (class in `khorosjx.utils.classes`), 69

`post_request_with_retries()` (in module `khorosjx.core`), 23

`POSTRequestError`, 71

`print_if_verbose()` (in module `khorosjx.utils.core_utils`), 63

`put_request_with_retries()` (in module `khorosjx.core`), 24

`PUTRequestError`, 71

## R

`rebuild_publication()` (in module `khorosjx.news`), 47

`remove_comment_wrappers_from_html()` (in module `khorosjx.utils.core_utils`), 63

`retrieve_connection_info()` (in module `khorosjx.admin`), 26

`retrieve_connection_info()` (in module `khorosjx.content.base`), 29

`retrieve_connection_info()` (in module `khorosjx.content.docs`), 33

`retrieve_connection_info()` (in module `khorosjx.content.events`), 34

`retrieve_connection_info()` (in module `khorosjx.content.ideas`), 35

`retrieve_connection_info()` (in module `khorosjx.content.threads`), 36

`retrieve_connection_info()` (in module `khorosjx.content.videos`), 39

`retrieve_connection_info()` (in module `khorosjx.groups`), 43

`retrieve_connection_info()` (in module `khorosjx.news`), 47

`retrieve_connection_info()` (in module `khorosjx.places.base`), 51

`retrieve_connection_info()` (in module `khorosjx.places.blogs`), 52

`retrieve_connection_info()` (in module `khorosjx.places.spaces`), 54

`retrieve_connection_info()` (in module `khorosjx.users`), 60

`retrieve_helper_settings()` (in module `khorosjx.utils.helper`), 66

## S

`set_base_url()` (in module `khorosjx.core`), 24

`set_credentials()` (in module `khorosjx.core`), 25

`set_package_path()` (in module `khorosjx.utils.tests.test_init_module`), 67

`SpaceNotFoundError`, 71

`SubscriptionNotFoundError`, 71

## T

`test_init_module()` (in module `khorosjx.utils.tests.test_init_module`), 67

`TimeUtils` (class in `khorosjx.utils.classes`), 69

## U

`update_publication()` (in module `khorosjx.news`), 47

`update_stream()` (in module `khorosjx.news`), 48

`UserNotFoundError`, 71

`UserQueryError`, 71

`Users` (class in `khorosjx.utils.classes`), 69

`Users.UserJSON` (class in `khorosjx.utils.classes`), 69

## V

`validate_timestamp()` (in module `khorosjx.utils.core_utils`), 63

`verify_connection()` (in module `khorosjx.core`), 25

`verify_core_connection()` (in module `khorosjx.admin`), 26

`verify_core_connection()` (in module `khorosjx.content.base`), 30

`verify_core_connection()` (in module `khorosjx.content.docs`), 33

`verify_core_connection()` (in module `khorosjx.content.events`), 34

`verify_core_connection()` (in module `khorosjx.content.ideas`), 36

`verify_core_connection()` (in module `khorosjx.content.threads`), 37

`verify_core_connection()` (in module `khorosjx.content.videos`), 40

`verify_core_connection()` (in module `khorosjx.groups`), 43

`verify_core_connection()` (in module `khorosjx.news`), 48

`verify_core_connection()` (in module `khorosjx.places.base`), 51

`verify_core_connection()` (in module `khorosjx.places.blogs`), 52

`verify_core_connection()` (in module `khorosjx.places.spaces`), 55

`verify_core_connection()` (in module `khorosjx.users`), 60

## W

`warn_when_not_latest()` (in module `khorosjx.utils.version`), 68

`WrongCredentialTypeError`, 71