



# **KAN Imagery Catalog**

***Release v1.6.5***

**Kan Territory & IT**

**Jul 16, 2025**



**CONTENTS:**

<b>1</b>	<b>API Reference</b>	<b>3</b>
1.1	src . . . . .	3
	<b>Python Module Index</b>	<b>29</b>
	<b>Index</b>	<b>31</b>



QGIS plugin to get satellite images from different providers.



## API REFERENCE

This page contains auto-generated API reference documentation<sup>1</sup>.

### 1.1 src

This script initializes the plugin, making it known to QGIS.

#### 1.1.1 Submodules

##### src.core

##### Submodules

##### src.core.catalogs

Catalogs module.

##### Functions

<code>get_custom_query</code> (→ dict)	Get custom query for a specific provider.
<code>get_catalog</code> (→ dict)	Get catalog data from a specific provider.
<code>get_catalogs_from_collection</code> (→ list)	Get catalog list from a specific provider.
<code>get_thumbnail</code> (→ dict)	Get catalog thumbnail from a specific provider.
<code>get_quicklook</code> (→ dict)	Get catalog quicklook from a specific provider.
<code>get_download</code> (→ dict)	Get catalog download from a specific provider.
<code>get_download_url</code> (**kwargs)	Get catalog download url from a specific provider.

##### Module Contents

`src.core.catalogs.get_custom_query(provider: str, max_cloud_coverage: int) → dict`

Get custom query for a specific provider.

`src.core.catalogs.get_catalog(provider: str, host_name: str, search_params: dict, max_cloud_coverage: int, collection_names: list) → dict`

Get catalog data from a specific provider.

`src.core.catalogs.get_catalogs_from_collection(collections: list, provider: str, search_params: dict) → list`

Get catalog list from a specific provider.

---

<sup>1</sup> Created with sphinx-autoapi

`src.core.catalogs.get_thumbnail(provider: str, collection_name: str, host_name: str, image_id: str, feature_data: dict) → dict`

Get catalog thumbnail from a specific provider.

`src.core.catalogs.get_quicklook(provider: str, host_name: str, image_id: str, feature_data: dict) → dict`

Get catalog quicklook from a specific provider.

`src.core.catalogs.get_download(provider: str, host_name: str, search_params: dict) → dict`

Get catalog download from a specific provider.

`src.core.catalogs.get_download_url(**kwargs)`

Get catalog download url from a specific provider.

## src.core.collections

Collections module.

### Functions

<code>get_collections(→ dict)</code>	Get collections from a specific provider.
--------------------------------------	---

## Module Contents

`src.core.collections.get_collections(provider: str, search_params: dict = None) → dict`

Get collections from a specific provider.

## src.core.providers

Providers module.

### Functions

<code>check_credentials(→ bool)</code>	Check credentials from a specific provider.
--	---

## Module Contents

`src.core.providers.check_credentials(provider: str, credentials: dict) → bool`

Check credentials from a specific provider.

## src.core.results

This module contains the functions to create the results layers.

### Functions

<code>create_quicklook(image_path, layer_name)</code>	image_id, Create results layers.
---	----------------------------------



## Module Contents

`src.core.results.create_quicklook(image_path, image_id, layer_name)`

Create results layers.

## src.core.settings

PluginSettings module.

## Classes

*PluginSettings*

PluginSettings class.

## Module Contents

**class** `src.core.settings.PluginSettings`

PluginSettings class.

**language:** `str` = 'english'

**back\_days:** `int` = 20

**cloud\_coverage:** `int` = 10

**max\_catalog\_results:** `int` = 10

**max\_features\_results:** `int` = 5

**download\_path:** `str` = './downloads'

**\_last\_cleaning\_date:** `str` | `None` = `None`

**\_provider\_settings:** `dict`

**\_selected\_collections:** `list` = []

**\_\_post\_init\_\_()**

Init settings.

**save()**

Save settings.

**clean\_temporary\_files\_if\_needed()**

Checks whether temporary files need to be deleted based on the days before the last activity date and delete files.

**update\_last\_cleaning\_date()**

Update last activity date.

**get\_active\_providers()** → `list`

Get active providers.

**property provider\_settings:** `dict`

**property selected\_collections:** `list`

**property last\_cleaning\_date:** `datetime.datetime`

## src.gui

### Submodules

#### src.gui.custom\_widgets

Custom widgets module.

### Classes

---

<i>CustomWidgetListItem</i>	Custom widget for list item.
-----------------------------	------------------------------

---

### Module Contents

```
class src.gui.custom_widgets.CustomWidgetListItem(parent=None, provider_name="", host_name="",
                                                    collection_name="", feature_data=None,
                                                    thumbnail=None, acquisition_date="",
                                                    incidence_angle=0.0, cloud_coverage=None,
                                                    image_id=None, feature_index=None,
                                                    footprints_layer=None, closing_plugin=None)
```

Bases: PyQt5.QtWidgets.QWidget, ui.custom\_widget\_list\_item.Ui\_CustomWidgetListItem

Custom widget for list item.

**closing\_plugin** = None

**parent** = None

**thread\_quicklooks**

**provider** = ''

**host** = ''

**feature\_data** = None

**footprints\_layer** = None

**image\_id** = None

**feature\_index** = None

**\_name** = '\_'

**\_acquisition\_date** = ''

**\_incidence\_angle** = 0.0

**\_cloud\_coverage**

**key** = 0.0

**property name**

**property acquisition\_date**

**property incidence\_angle**

**property cloud\_coverage**

**get\_date\_for\_label()**

Get date for label display.

**set\_thumbnail(*image\_bytes*)**

Set thumbnail image in label.

**download\_images()**

Download image action.

**view\_details()**

View details action.

**show\_quicklook()**

Show quicklook action.

**get\_quicklook\_image(*provider: str, host\_name: str, image\_id: str, feature\_data: dict*)**

Get quicklook from host.

**create\_quicklook\_layer(*params: dict*)**

Create a layer with the quicklook.

## src.gui.form\_base

Base form module for all forms in the plugin.

### Classes

<i>FormBase</i>	Base form class for all forms in the plugin.
-----------------	--

### Module Contents

**class** src.gui.form\_base.**FormBase**(*parent=None, accept\_btn=False, cancel\_btn=False, close\_btn=True, closing\_plugin=None*)

Bases: PyQt5.QtWidgets.QDialog

Base form class for all forms in the plugin.

**parent** = None

**set\_control\_buttons**(*accept\_btn, cancel\_btn, close\_btn*)

Set control buttons for the form (Accept, Cancel, Close).

**btn\_close\_click()**

Event handler for the Close button click.

## src.gui.form\_catalog\_info

Form Catalog Info Module.

## Classes

<i>FormCatalogInfo</i>	Form Catalog Info Class.
------------------------	--------------------------

## Module Contents

**class** `src.gui.form_catalog_info.FormCatalogInfo`(*parent=None, data=None, closing\_plugin=None*)  
Bases: `gui.form_base.FormBase`, `ui.frm_catalog_info.Ui_frm_catalog_info`

Form Catalog Info Class.

**normalize\_dict\_data**(*data*)  
Normalize data to be loaded in table.

**load\_data**(*data*)  
Load data in table.

## `src.gui.form_default_collections`

Collection selection form module.

## Classes

<i>FormDefaultCollections</i>	Collection selection form class.
-------------------------------	----------------------------------

## Module Contents

**class** `src.gui.form_default_collections.FormDefaultCollections`(*parent=None, closing\_plugin=None*)  
Bases: `gui.form_base.FormBase`, `ui.frm_default_collections.Ui_frm_default_collections`

Collection selection form class.

**settings**

**selected\_collections**

**providers**

**btn\_accept\_clicked**()  
Event handler for accept button click.

**btn\_add\_selected\_clicked**()  
Event handler for add selected button click.

**btn\_remove\_selected\_clicked**()  
Event handler for remove selected button click.

**btn\_filter\_results\_clicked**()  
Event handler for filter results button click.

**load\_filtered\_collections**(*data*)  
Load filtered collections in table.

**load\_selected\_collections**()  
Load selected collections in table.

## src.gui.form\_settings

Form settings module.

### Classes

*FormSettings*

Form PluginSettings Class.

### Module Contents

**class** src.gui.form\_settings.**FormSettings**(parent=None, closing\_plugin=None)

Bases: gui.form\_base.FormBase, ui.frm\_settings.Ui\_frm\_settings

Form PluginSettings Class.

**settings**

**up42\_settings**

**sentinelhub\_settings**

**planet\_settings**

**up42\_is\_valid**

**sentinelhub\_is\_valid**

**planet\_is\_valid**

**update\_cloud\_coverage\_label()**

Update cloud coverage label text.

**btn\_accept\_clicked()**

Event handler for accept button click.

**txt\_default\_back\_days\_text\_changed()**

Event handler for default back days text changed.

**txt\_max\_catalog\_results\_text\_changed()**

Event handler for max catalog results text changed.

**txt\_max\_features\_results\_text\_changed()**

Event handler for max features results text changed.

**btn\_up42\_check\_credentials\_clicked()**

Event handler for check credentials button click.

**btn\_sentinelhub\_check\_credentials\_clicked()**

Event handler for check credentials button click.

## src.gui.helpers

### Submodules

#### src.gui.helpers.forms

Form helpers functions module.

## Functions

<code>load_combobox(obj_combobox, value_member, ...)</code>	<code>key_member,</code>	Load a QComboBox with provided data.
<code>set_form_stylesheet(parent)</code>		Set stylesheet to a form.
<code>set_elided_text_to_label(label, text)</code>		Set elided text to a QLabel if the text is too long.
<code>set_tooltips(form)</code>		Set tooltips to all QLineEdit widgets in a form.
<code>get_shadow_effect()</code>		Get shadow effect for widgets.
<code>init_tableview(obj_tableview, row_count, ...)</code>	<code>column_count,</code>	Initialize tableview with provided data, headers and columns.
<code>load_table_data(obj_table, data[, headers, columns, ...])</code>		Load table with provided data, headers and columns.
<code>check_int_not_empty(txt_widget)</code>		Check if the value is an integer and greater than 0 in a QLineEdit.

## Module Contents

`src.gui.helpers.forms.load_combobox(obj_combobox, key_member, value_member, lst_data, block_signals=False)`

Load a QComboBox with provided data.

`src.gui.helpers.forms.set_form_stylesheet(parent)`

Set stylesheet to a form.

`src.gui.helpers.forms.set_elided_text_to_label(label, text)`

Set elided text to a QLabel if the text is too long.

`src.gui.helpers.forms.set_tooltips(form)`

Set tooltips to all QLineEdit widgets in a form.

`src.gui.helpers.forms.get_shadow_effect()`

Get shadow effect for widgets.

`src.gui.helpers.forms.init_tableview(obj_tableview, column_count, row_count, headers, default_row_height=25)`

Initialize tableview with provided data, headers and columns.

`src.gui.helpers.forms.load_table_data(obj_table, data, headers=None, columns=[], fn_custom_items=[], cols_to_stretch=[], cols_width=[], cols_to_hide=[], add_dict_object=False, literal_values=False, default_row_height=25, default_alignment=None, checkable_columns=[], num_decimals=2)`

Load table with provided data, headers and columns.

`src.gui.helpers.forms.check_int_not_empty(txt_widget)`

Check if the value is an integer and greater than 0 in a QLineEdit.

## `src.gui.helpers.worker`

Worker thread class.

## Classes

<i>WorkerThread</i>	Worker thread class.
---------------------	----------------------

## Module Contents

**class** `src.gui.helpers.worker.WorkerThread`

Bases: `PyQt5.QtCore.QThread`

Worker thread class.

**finished**

**progress\_updated**

**error\_signal**

**warning\_signal**

**info\_signal**

**start**(*process, dict\_params*)

Start thread.

**run**()

Run thread.

## **src.kan\_imagery\_catalog**

KAN Imagery Catalog QGIS plugin main module.

## Classes

<i>KANImageryCatalog</i>	QGIS Plugin Implementation.
--------------------------	-----------------------------

## Module Contents

**class** `src.kan_imagery_catalog.KANImageryCatalog(iface)`

QGIS Plugin Implementation.

**iface**

**plugin\_dir**

**actions** = []

**menu**

**toolbar**

**pluginIsActive** = False

**dockwidget** = None

**frm\_settings** = None

**frm\_collection\_settings = None**

**tr**(*message*)

Get the translation for a string using Qt translation API.

We implement this ourselves since we do not inherit QObject.

**Parameters**

**message** (*str*, *QString*) – String for translation.

**Returns**

Translated version of message.

**Return type**

QString

**add\_action**(*icon\_path*, *text*, *callback*, *enabled\_flag=True*, *add\_to\_menu=True*, *add\_to\_toolbar=True*, *status\_tip=None*, *whats\_this=None*, *parent=None*)

Add a toolbar icon to the toolbar.

**Parameters**

- **icon\_path** (*str*) – Path to the icon for this action. Can be a resource path (e.g. ‘:/plugins/foo/bar.png’) or a normal file system path.
- **text** (*str*) – Text that should be shown in menu items for this action.
- **callback** (*function*) – Function to be called when the action is triggered.
- **enabled\_flag** (*bool*) – A flag indicating if the action should be enabled by default. Defaults to True.
- **add\_to\_menu** (*bool*) – Flag indicating whether the action should also be added to the menu. Defaults to True.
- **add\_to\_toolbar** (*bool*) – Flag indicating whether the action should also be added to the toolbar. Defaults to True.
- **status\_tip** (*str*) – Optional text to show in a popup when mouse pointer hovers over the action.
- **parent** (*QWidget*) – Parent widget for the new action. Defaults None.
- **whats\_this** – Optional text to show in the status bar when the mouse pointer hovers over the action.

**Returns**

The action that was created. Note that the action is also added to self.actions list.

**Return type**

QAction

**initGui()**

Create the menu entries and toolbar icons inside the QGIS GUI.

**onClosePlugin()**

Cleanup necessary items here when plugin dockwidget is closed

**unload()**

Removes the plugin menu item and icon from QGIS GUI.

**run()**

Run method that loads and starts the plugin



## src.kan\_imagery\_catalog\_dock

KAN Imagery Catalog QGIS plugin dock widget module.

### Classes

---

*KANImageryCatalogDock*KANImageryCatalogDock Dockwidget class.

---

### Module Contents

```

class src.kan_imagery_catalog_dock.KANImageryCatalogDock(parent=None)
    Bases: PyQt5.QtWidgets.QDockWidget, FORM_CLASS
    KANImageryCatalogDock Dockwidget class.
    closing_plugin
    error_signal
    warning_signal
    info_signal
    parent = None
    settings
    collections = []
    sort_ascending = True
    loading_spinner
    thread_get_catalogs
    thread_get_thumbnails
    thread_load_collections_cache
    show_ended_search_message = True
    show_info(title, message)
        Show info message.
    show_warning(title, message)
        Show warning message.
    show_error(title, message)
        Show error message.
    set_form_state(is_busy, show_spinner=False)
        Sets the form state (enabled/disabled) when the process is running in a separate thread.
    closeEvent(event)
        Run close plugin event.
    txt_max_catalog_results_text_changed()
        Event handler for textbox 'txt_max_catalog_results'.

```

**update\_cloud\_coverage\_label()**

Event handler for slider 'slider\_cloud\_coverage'.

**chk\_search\_by\_dataframe\_update()**

Event handler for checkbox 'chk\_search\_by\_dataframe'.

**btn\_update\_layers\_list\_clicked()**

Event handler for button 'btn\_update\_layers\_list'.

**btn\_settings\_clicked()**

Event handler for button 'btn\_settings'.

**btn\_select\_catalogs\_clicked()**

Event handler for button 'btn\_select\_catalogs'.

**load\_collections\_cache()**

Load collections in cache

**show\_collections\_form()**

Show form to select collections.

**btn\_sort\_results\_clicked()**

Event handler for button 'btn\_sort\_results'.

**btn\_show\_hide\_search\_area\_clicked()**

Event handler for button 'btn\_show\_hide\_search\_area'.

**btn\_show\_hide\_catalogs\_clicked()**

Event handler for button 'btn\_show\_hide\_catalogs'.

**btn\_show\_hide\_filters\_clicked()**

Event handler for button 'btn\_show\_hide\_filters'.

**btn\_get\_data\_clicked()**

Event handler for button 'btn\_get\_data'.

**get\_data\_finished()**

Event handler for thread 'thread\_get\_catalogs' finished.

**get\_results(*bounding\_box, cloud\_coverage, date\_from, date\_to, max\_catalog\_results*)**

Get results from selected catalogs with selected filters.

**get\_thumbnails\_in\_background(*catalogs\_data: list*)**

Get thumbnails in background thread.

**update\_progress(*progress\_data*)**

Update thread results in Toc and DockWidget.

**add\_item\_to\_results(*provider\_name, host\_name, collection\_name, feature\_data, acquisition\_date, incidence\_angle, cloud\_coverage, image\_id, feature\_index, thumbnail*)**

Add item to results list.

**add\_feature\_to\_footprints\_layer(*coordinates, footprint\_id*)**

Add feature to footprints layer.

## src.plugin\_upload

This script uploads a plugin package to the plugin repository. Authors: A. Pasotti, V. Picavet git sha : \$TemplateVCS-Format

### Attributes

<i>PROTOCOL</i>
<i>SERVER</i>
<i>PORT</i>
<i>ENDPOINT</i>
<i>VERBOSE</i>
<i>parser</i>

### Functions

<i>main</i> (parameters, arguments)	Main entry point.
<i>hide_password</i> (url[, start])	Returns the http url with password part replaced with '*'.

### Module Contents

```
src.plugin_upload.PROTOCOL = 'https'
src.plugin_upload.SERVER = 'plugins.qgis.org'
src.plugin_upload.PORT = '443'
src.plugin_upload.ENDPOINT = '/plugins/RPC2/'
src.plugin_upload.VERBOSE = False
```

```
src.plugin_upload.main(parameters, arguments)
```

Main entry point.

#### Parameters

- **parameters** – Command line parameters.
- **arguments** – Command line arguments.

```
src.plugin_upload.hide_password(url, start=6)
```

Returns the http url with password part replaced with '\*'.

#### Parameters

- **url** (*str*) – URL to upload the plugin to.
- **start** (*int*) – Position of start of password.

```
src.plugin_upload.parser
```

**src.resources****Attributes**

---

*qt\_resource\_data**qt\_resource\_name**qt\_resource\_struct\_v1**qt\_resource\_struct\_v2**qt\_version**rcc\_version*

---

**Functions**

---

*qInitResources()**qCleanupResources()*

---

**Module Contents**

```
src.resources.qt_resource_data =  
b'\x00\x00\x16\x8b\x89PNG\r\n\x1a\n\x00\x00\x00\rIHDR\x00\x00\x00 \x00\x00\x00...  
  
src.resources.qt_resource_name = b'\x00\x08\naZ\xa7\x00i\x00c\x00o\x00n\x00.\x00p\x00n\  
x00g\x00\t\nlxC\x00r\x00e\x00s\x00o\x00u\x00...  
  
src.resources.qt_resource_struct_v1 = b'\x00\x00\x00\x00\x00\x02\x00\x00\x00\x02\x00\x00\  
x00\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00...  
  
src.resources.qt_resource_struct_v2 = b'\x00\x00\x00\x00\x00\x02\x00\x00\x00\x02\x00\x00\  
x00\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00...  
  
src.resources.qt_version  
  
src.resources.rcc_version = 1  
  
src.resources.qInitResources()  
  
src.resources.qCleanupResources()
```

**src.services****Submodules****src.services.element84**

Module for Element84 API calls

## Attributes

<code>DOWNLOAD_URL</code>
---------------------------

## Functions

<code>get_collections()</code>	Get collections from Element84 API
<code>get_catalog(→ dict)</code>	Get catalog data from Element84 API
<code>get_thumbnail(collection_name, image_id)</code>	Get catalog thumbnail from Element84 API
<code>get_quicklook(image_id, feature_data)</code>	Get catalog quicklook from Element84 API

## Module Contents

```
src.services.element84.DOWNLOAD_URL = 'https://element84.com/'
```

```
src.services.element84.get_collections()
```

```
    Get collections from Element84 API
```

```
src.services.element84.get_catalog(search_params: dict) → dict
```

```
    Get catalog data from Element84 API
```

```
src.services.element84.get_thumbnail(collection_name: str, image_id: str)
```

```
    Get catalog thumbnail from Element84 API
```

```
src.services.element84.get_quicklook(image_id: str, feature_data: dict)
```

```
    Get catalog quicklook from Element84 API
```

## src.services.microsoft

Module for MicrosoftPlanetary API calls

## Attributes

<code>DOWNLOAD_URL</code>
---------------------------

## Functions

<code>get_collections()</code>	Get collections from MicrosoftPlanetary API
<code>get_catalog(→ dict)</code>	Get catalog data from MicrosoftPlanetary API
<code>get_thumbnail(collection_name, feature_data)</code>	Get catalog thumbnail from MicrosoftPlanetary API
<code>get_quicklook(host_name, image_id, feature_data)</code>	Get catalog quicklook from MicrosoftPlanetary API

## Module Contents

```
src.services.microsoft.DOWNLOAD_URL = 'https://planetarycomputer.microsoft.com/'
```

`src.services.microsoft.get_collections()`

Get collections from MicrosoftPlanetary API

`src.services.microsoft.get_catalog(search_params: dict) → dict`

Get catalog data from MicrosoftPlanetary API

`src.services.microsoft.get_thumbnail(collection_name: str, feature_data: dict)`

Get catalog thumbnail from MicrosoftPlanetary API

`src.services.microsoft.get_quicklook(host_name: str, image_id: str, feature_data: dict)`

Get catalog quicklook from MicrosoftPlanetary API

## src.services.sentinel\_hub

Module for Sentinel Hub API calls

### Attributes

<code>DOWNLOAD_URL</code>
---------------------------

### Functions

<code>get_token(client_id, client_secret)</code>	Get token from Sentinel Hub API
<code>get_collections(token)</code>	Get collections from Sentinel Hub API
<code>get_catalog(→ dict)</code>	Get catalog data from Sentinel Hub API
<code>get_thumbnail(token, host_name, image_id)</code>	Get catalog thumbnail from Sentinel Hub API
<code>get_quicklook(token, host_name, image_id)</code>	Get catalog quicklook from Sentinel Hub API

### Module Contents

`src.services.sentinel_hub.DOWNLOAD_URL = 'https://www.sentinel-hub.com/'`

`src.services.sentinel_hub.get_token(client_id, client_secret)`

Get token from Sentinel Hub API

`src.services.sentinel_hub.get_collections(token)`

Get collections from Sentinel Hub API

`src.services.sentinel_hub.get_catalog(token: str, host_name: str, search_params: dict) → dict`

Get catalog data from Sentinel Hub API

`src.services.sentinel_hub.get_thumbnail(token: str, host_name: str, image_id: str)`

Get catalog thumbnail from Sentinel Hub API

`src.services.sentinel_hub.get_quicklook(token: str, host_name: str, image_id: str)`

Get catalog quicklook from Sentinel Hub API

## src.services.up42

Module for Up42 API calls

## Attributes

<code>DOWNLOAD_URL</code>
---------------------------

## Functions

<code>get_token(→ str)</code>	Get token from UP42 API
<code>get_collections()</code>	Get collections from UP42 API
<code>get_catalog(→ dict)</code>	Get catalog data from UP42 API
<code>get_thumbnail(token, host_name, image_id)</code>	Get catalog thumbnail from UP42 API
<code>get_quicklook(token, host_name, image_id)</code>	Get catalog quicklook from UP42 API

## Module Contents

```
src.services.up42.DOWNLOAD_URL = 'https://console.up42.com/catalog/new-order'
```

```
src.services.up42.get_token(username: str, password: str) → str
    Get token from UP42 API
```

```
src.services.up42.get_collections()
    Get collections from UP42 API
```

```
src.services.up42.get_catalog(token: str, host_name: str, search_params: dict) → dict
    Get catalog data from UP42 API
```

```
src.services.up42.get_thumbnail(token: str, host_name: str, image_id: str)
    Get catalog thumbnail from UP42 API
```

```
src.services.up42.get_quicklook(token: str, host_name: str, image_id: str)
    Get catalog quicklook from UP42 API
```

## src.services.utils

Module for API auxiliary functions

## Attributes

<code>REQUEST_TIMEOUT</code>
------------------------------

## Functions

<code>encode_base64(→ str)</code>	Encode a string to Base64
<code>http_get(url[, host_name, headers, result_type, ...])</code>	Make a GET request to the specified URL.
<code>http_post(url[, host_name, headers, payload, ...])</code>	Make a POST request to the specified URL.

## Module Contents

`src.services.utils.REQUEST_TIMEOUT = 120`

`src.services.utils.encode_base64(text_to_encode: str) → str`

Encode a string to Base64

`src.services.utils.http_get(url, host_name="", headers={}, result_type='json', raise_for_status=False)`

Make a GET request to the specified URL.

`src.services.utils.http_post(url, host_name="", headers={}, payload={}, result_type='json', raise_for_status=False)`

Make a POST request to the specified URL.

## src.ui

### Submodules

`src.ui.custom_widget_list_item`

### Attributes

---

`app`

---

## Classes

---

`Ui_CustomWidgetListItem`

---

## Module Contents

`class src.ui.custom_widget_list_item.Ui_CustomWidgetListItem`

Bases: object

`setupUi(CustomWidgetListItem)`

`retranslateUi(CustomWidgetListItem)`

`src.ui.custom_widget_list_item.app`

`src.ui.frm_catalog_info`

### Attributes

---

`app`

---



## Classes

*Ui\_frm\_catalog\_info*

## Module Contents

```
class src.ui.frm_catalog_info.Ui_frm_catalog_info
```

```
    Bases: object
```

```
        setupUi(frm_catalog_info)
```

```
        retranslateUi(frm_catalog_info)
```

```
src.ui.frm_catalog_info.app
```

**src.ui.frm\_default\_collections**

## Attributes

*app*

## Classes

*Ui\_frm\_default\_collections*

## Module Contents

```
class src.ui.frm_default_collections.Ui_frm_default_collections
```

```
    Bases: object
```

```
        setupUi(frm_default_collections)
```

```
        retranslateUi(frm_default_collections)
```

```
src.ui.frm_default_collections.app
```

**src.ui.frm\_settings**

## Attributes

*app*

## Classes

<i>Ui_frm_settings</i>
------------------------

## Module Contents

```
class src.ui.frm_settings.Ui_frm_settings
```

```
    Bases: object
```

```
        setupUi(frm_settings)
```

```
        retranslateUi(frm_settings)
```

```
src.ui.frm_settings.app
```

## src.utils

### Submodules

#### src.utils.constants

Constants module.

## Attributes

<i>RESULTS_GROUP_NAME</i>
---------------------------

<i>RESULTS_LAYER_NAME</i>
---------------------------

<i>DEFAULT_MESSAGE_DURATION</i>
---------------------------------

<i>DEFAULT_SOURCE_CRS</i>
---------------------------

<i>DEFAULT_TARGET_CRS</i>
---------------------------

## Classes

<i>CustomEnum</i>	Custom Enum class.
<i>StyleVariables</i>	Style variables for qss stylesheet.
<i>MessageType</i>	Qgis Message types.

## Module Contents

```
src.utils.constants.RESULTS_GROUP_NAME = 'kan_imagery_catalog_preview'
```

```
src.utils.constants.RESULTS_LAYER_NAME = 'kan_imagery_catalog_footprints'
```

```
src.utils.constants.DEFAULT_MESSAGE_DURATION = 5
```

```
src.utils.constants.DEFAULT_SOURCE_CRS = 'EPSG:3857'
```

```
src.utils.constants.DEFAULT_TARGET_CRS = 'EPSG:4326'
```

```
class src.utils.constants.CustomEnum(*args, **kwargs)
```

Bases: `enum.Enum`

Custom Enum class.

`__repr__()` → str

Override the default repr behavior.

`__str__()` → str

Override the default str behavior.

`__eq__(other)`

Override the default eq behavior.

**classmethod** `to_dict()` → dict

Convert enum to dict.

```
class src.utils.constants.StyleVariables(*args, **kwargs)
```

Bases: `CustomEnum`

Style variables for qss stylesheet.

`FONT_SIZE_DEFAULT` = '12px'

`FONT_SIZE_SMALL` = '8px'

`COLOR_MAIN_DARK` = '#6597AF'

`COLOR_MAIN_LIGHT` = '#F5F5F5'

`COLOR_TEXT_DEFAULT` = '#2D2D2D'

`COLOR_TEXT_LIGHT` = '#E8E8E8'

`COLOR_BACKGROUND_DEFAULT` = '#E8E8E8'

```
class src.utils.constants.MessageType
```

Qgis Message types.

`INFO` = 0

`WARNING` = 1

`CRITICAL` = 2

`SUCCESS` = 3

## src.utils.debug

Debugging utilities module.

## Functions

---

`pyqt_set_trace()`Set a tracepoint in the Python debugger that works with Qt.

---

## Module Contents

`src.utils.debug.pyqt_set_trace()`

Set a tracepoint in the Python debugger that works with Qt.

## src.utils.encoders

Custom JSON encoders for the application.

## Classes

---

`CustomJsonDecimalEncoder`Custom JSON encoder for Decimal objects.

---

## Module Contents

```
class src.utils.encoders.CustomJsonDecimalEncoder(* (Keyword-only parameters separator (PEP
3102)), skipkeys=False, ensure_ascii=True,
check_circular=True, allow_nan=True,
sort_keys=False, indent=None, separators=None,
default=None)
```

Bases: `json.JSONEncoder`

Custom JSON encoder for Decimal objects.

`default(o)`

Encode Decimal objects as float.

## src.utils.exceptions

Custom exceptions module.

## Exceptions

---

`PluginError`

Base class for custom exceptions.

`ProviderError`

Custom exception for provider related errors.

`HostError`

Custom exception for host related errors.

`AuthorizationError`

Custom exception for authorization related errors.

`SettingsError`

Custom exception for settings related errors.

`DataNotFoundError`Custom exception for data related errors.

---

## Module Contents

```
exception src.utils.exceptions.PluginError(message="", message_type=MessageType.CRITICAL)
```

Bases: `Exception`

Base class for custom exceptions.

**message\_type**

**message = ''**

**exception** `src.utils.exceptions.ProviderError`(*message=tr('Error to get results from provider.')*)

Bases: *PluginError*

Custom exception for provider related errors.

**exception** `src.utils.exceptions.HostError`(*message=tr('Error to get results from host.')*)

Bases: *PluginError*

Custom exception for host related errors.

**exception** `src.utils.exceptions.AuthorizationError`(*message=tr('Error to get results from authorization.')*)

Bases: *PluginError*

Custom exception for authorization related errors.

**exception** `src.utils.exceptions.SettingsError`(*message=tr('Please check the plugin settings.')*)

Bases: *PluginError*

Custom exception for settings related errors.

**exception** `src.utils.exceptions.DataNotFoundError`(*message=tr('No data found.')*)

Bases: *PluginError*

Custom exception for data related errors.

## src.utils.general

General utilities for the KAN Imagery Catalog plugin.

### Attributes

*PLUGIN\_NAME*

### Functions

<i>_get_plugin_name</i> ([missing])	Get plugin name from metadata.txt file.
<i>get_plugin_dir</i> ()	Get plugin directory.
<i>clean_temporary_files</i> ()	Clean temporary files

### Module Contents

`src.utils.general._get_plugin_name`(*missing='KAN Imagery Catalog'*)

Get plugin name from metadata.txt file.

`src.utils.general.get_plugin_dir`()

Get plugin directory.

`src.utils.general.PLUGIN_NAME` = 'KAN Imagery Catalog'

```
src.utils.general.clean_temporary_files()
```

Clean temporary files

## src.utils.helpers

Helper functions module.

### Functions

<code>open_url(→ bool)</code>	Open a URL in the browser.
<code>tr(message)</code>	Get the translation for a string using Qt translation API.
<code>normalize_text(text)</code>	Remove accents from text.

### Module Contents

```
src.utils.helpers.open_url(url: str, params: dict | None = None) → bool
```

Open a URL in the browser.

```
src.utils.helpers.tr(message)
```

Get the translation for a string using Qt translation API.

```
src.utils.helpers.normalize_text(text)
```

Remove accents from text.

## src.utils.qgis\_helper

QGIS helper functions module.

### Attributes

<code>message</code>
----------------------

### Functions

<code>get_bounding_box_transformed(bbox, source_crs, target_crs)</code>	Get bounding box from canvas and transform it if necessary to the default target CRS.
<code>get_bounding_box_canvas()</code>	Get bounding box from canvas.
<code>get_selected_feature_bounding_box(layer_id[, ...])</code>	Get bounding box from feature. By default, the first feature in the array is used if no feature is selected.
<code>get_valid_project_layers_to_search()</code>	Get vector layers names from current project excluding the ones in the results group.
<code>save_setting(setting_name, value)</code>	Save setting.
<code>read_setting(setting_name[, default_value])</code>	Read setting.
<code>save_json_setting(setting_name, value)</code>	Save json setting.
<code>read_json_setting(setting_name[, default_value])</code>	Read json setting.
<code>error_message(title, text)</code>	Show error message.
<code>warning_message(title, text)</code>	Show warning message.
<code>info_message(title, text)</code>	Show info message.

continues on next page

Table 46 – continued from previous page

<code>success_message(title, text)</code>	Show success message.
<code>get_or_create_group(group_name)</code>	Get or create group.
<code>get_layer_by_name(layer_name)</code>	Get layer by name.
<code>get_or_create_footprints_layer(layer_name, group_name)</code>	Get or create footprints layer.
<code>add_feature_to_layer(coordinates, feature_id, layer)</code>	Add feature to layer.
<code>get_georeferenced_image(feature, image_path)</code>	Get georeferenced image from feature and image path.
<code>create_quicklook_layer(footprints_crs, layer_name, ...)</code>	Create a layer with a quicklook image.
<code>zoom_selected_features(layer)</code>	Zoom to selected features in layer.

## Module Contents

`src.utils.qgis_helper.message`

`src.utils.qgis_helper.get_bounding_box_transformed(bbox, source_crs, target_crs)`

Get bounding box from canvas and transform it if necessary to the default target CRS.

`src.utils.qgis_helper.get_bounding_box_canvas()`

Get bounding box from canvas.

`src.utils.qgis_helper.get_selected_feature_bounding_box(layer_id, default_first=True)`

Get bounding box from feature. By default, the first feature in the array is used if no feature is selected.

`src.utils.qgis_helper.get_valid_project_layers_to_search()`

Get vector layers names from current project excluding the ones in the results group.

`src.utils.qgis_helper.save_setting(setting_name, value)`

Save setting.

`src.utils.qgis_helper.read_setting(setting_name, default_value=None)`

Read setting.

`src.utils.qgis_helper.save_json_setting(setting_name, value)`

Save json setting.

`src.utils.qgis_helper.read_json_setting(setting_name, default_value=None)`

Read json setting.

`src.utils.qgis_helper.error_message(title, text)`

Show error message.

`src.utils.qgis_helper.warning_message(title, text)`

Show warning message.

`src.utils.qgis_helper.info_message(title, text)`

Show info message.

`src.utils.qgis_helper.success_message(title, text)`

Show success message.

`src.utils.qgis_helper.get_or_create_group(group_name)`

Get or create group.

`src.utils.qgis_helper.get_layer_by_name(layer_name)`

Get layer by name.

`src.utils.qgis_helper.get_or_create_footprints_layer(layer_name, group_name)`

Get or create footprints layer.

`src.utils.qgis_helper.add_feature_to_layer(coordinates, feature_id, layer)`

Add feature to layer.

`src.utils.qgis_helper.get_georeferenced_image(feature, image_path)`

Get georeferenced image from feature and image path.

`src.utils.qgis_helper.create_quicklook_layer(footprints_crs, layer_name, group_name, feature, image_path=None)`

Create a layer with a quicklook image.

`src.utils.qgis_helper.zoom_selected_features(layer)`

Zoom to selected features in layer.

## 1.1.2 Functions

<code>classFactory(iface)</code>	Load KANImageryCatalog class from file KANImageryCatalog.
----------------------------------	---

## 1.1.3 Package Contents

`src.classFactory(iface)`

Load KANImageryCatalog class from file KANImageryCatalog.

### Parameters

**iface** (*QgsInterface*) – A QGIS interface instance.

==



## PYTHON MODULE INDEX

### S

- `src`, 3
- `src.core`, 3
  - `src.core.catalogs`, 3
  - `src.core.collections`, 4
  - `src.core.providers`, 4
  - `src.core.results`, 4
  - `src.core.settings`, 5
- `src.gui`, 6
  - `src.gui.custom_widgets`, 6
  - `src.gui.form_base`, 7
  - `src.gui.form_catalog_info`, 7
  - `src.gui.form_default_collections`, 8
  - `src.gui.form_settings`, 9
  - `src.gui.helpers`, 9
    - `src.gui.helpers.forms`, 9
    - `src.gui.helpers.worker`, 10
  - `src.kan_imagery_catalog`, 11
  - `src.kan_imagery_catalog_dock`, 13
- `src.plugin_upload`, 15
- `src.resources`, 16
- `src.services`, 16
  - `src.services.element84`, 16
  - `src.services.microsoft`, 17
  - `src.services.sentinel_hub`, 18
  - `src.services.up42`, 18
  - `src.services.utils`, 19
- `src.ui`, 20
  - `src.ui.custom_widget_list_item`, 20
  - `src.ui.frm_catalog_info`, 20
  - `src.ui.frm_default_collections`, 21
  - `src.ui.frm_settings`, 21
- `src.utils`, 22
  - `src.utils.constants`, 22
  - `src.utils.debug`, 23
  - `src.utils.encoders`, 24
  - `src.utils.exceptions`, 24
  - `src.utils.general`, 25
  - `src.utils.helpers`, 26
  - `src.utils.qgis_helper`, 26



## Symbols

- `__eq__()` (*src.utils.constants.CustomEnum* method), 23
- `__post_init__()` (*src.core.settings.PluginSettings* method), 5
- `__repr__()` (*src.utils.constants.CustomEnum* method), 23
- `__str__()` (*src.utils.constants.CustomEnum* method), 23
- `_acquisition_date` (*src.gui.custom\_widgets.CustomWidgetListItem* attribute), 6
- `_cloud_coverage` (*src.gui.custom\_widgets.CustomWidgetListItem* attribute), 6
- `_get_plugin_name()` (in module *src.utils.general*), 25
- `_incidence_angle` (*src.gui.custom\_widgets.CustomWidgetListItem* attribute), 6
- `_last_cleaning_date` (*src.core.settings.PluginSettings* attribute), 5
- `_name` (*src.gui.custom\_widgets.CustomWidgetListItem* attribute), 6
- `_provider_settings` (*src.core.settings.PluginSettings* attribute), 5
- `_selected_collections` (*src.core.settings.PluginSettings* attribute), 5
- A**
- `acquisition_date` (*src.gui.custom\_widgets.CustomWidgetListItem* property), 6
- `actions` (*src.kan\_imagery\_catalog.KANImageryCatalog* attribute), 11
- `add_action()` (*src.kan\_imagery\_catalog.KANImageryCatalog* method), 12
- `add_feature_to_footprints_layer()` (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock* method), 14
- `add_feature_to_layer()` (in module *src.utils.qgis\_helper*), 28
- `add_item_to_results()` (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock* method), 14
- `app` (in module *src.ui.custom\_widget\_list\_item*), 20
- `app` (in module *src.ui.frm\_catalog\_info*), 21
- `app` (in module *src.ui.frm\_default\_collections*), 21
- `app` (in module *src.ui.frm\_settings*), 22
- `AuthorizationError`, 25
- B**
- `back_days` (*src.core.settings.PluginSettings* attribute), 5
- `btn_accept_clicked()` (*src.gui.form\_default\_collections.FormDefaultCollections* method), 8
- `btn_accept_clicked()` (*src.gui.form\_settings.FormSettings* method), 9
- `btn_add_selected_clicked()` (*src.gui.form\_default\_collections.FormDefaultCollections* method), 8
- `btn_close_click()` (*src.gui.form\_base.FormBase* method), 7
- `btn_filter_results_clicked()` (*src.gui.form\_default\_collections.FormDefaultCollections* method), 8
- `btn_get_data_clicked()` (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock* method), 14
- `btn_remove_selected_clicked()` (*src.gui.form\_default\_collections.FormDefaultCollections* method), 8
- `btn_select_catalogs_clicked()` (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock* method), 14
- `btn_sentinelhub_check_credentials_clicked()` (*src.gui.form\_settings.FormSettings* method), 9
- `btn_settings_clicked()` (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock* method), 14
- `btn_show_hide_catalogs_clicked()` (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock* method), 14
- `btn_show_hide_filters_clicked()` (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock* method), 14
- `btn_show_hide_search_area_clicked()` (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock* method), 14

btn\_sort\_results\_clicked() (src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock method), 14  
 btn\_up42\_check\_credentials\_clicked() (src.gui.form\_settings.FormSettings method), 9  
 btn\_update\_layers\_list\_clicked() (src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock method), 14  
**C**  
 check\_credentials() (in module src.core.providers), 4  
 check\_int\_not\_empty() (in module src.gui.helpers.forms), 10  
 chk\_search\_by\_dataframe\_update() (src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock method), 14  
 classFactory() (in module src), 28  
 clean\_temporary\_files() (in module src.utils.general), 25  
 clean\_temporary\_files\_if\_needed() (src.core.settings.PluginSettings method), 5  
 closeEvent() (src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock method), 13  
 closing\_plugin(src.gui.custom\_widgets.CustomWidgetListItem attribute), 6  
 closing\_plugin(src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock attribute), 13  
 cloud\_coverage (src.core.settings.PluginSettings attribute), 5  
 cloud\_coverage (src.gui.custom\_widgets.CustomWidgetListItem property), 6  
 collections (src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock attribute), 13  
 COLOR\_BACKGROUND\_DEFAULT (src.utils.constants.StyleVariables attribute), 23  
 COLOR\_MAIN\_DARK (src.utils.constants.StyleVariables attribute), 23  
 COLOR\_MAIN\_LIGHT (src.utils.constants.StyleVariables attribute), 23  
 COLOR\_TEXT\_DEFAULT (src.utils.constants.StyleVariables attribute), 23  
 COLOR\_TEXT\_LIGHT (src.utils.constants.StyleVariables attribute), 23  
 create\_quicklook() (in module src.core.results), 5  
 create\_quicklook\_layer() (in module src.utils.qgis\_helper), 28  
 create\_quicklook\_layer() (src.gui.custom\_widgets.CustomWidgetListItem method), 7  
 CRITICAL (src.utils.constants.MessageType attribute), 23  
 CustomEnum (class in src.utils.constants), 23  
 CustomJsonDecimalEncoder (class in src.utils.encoders), 24  
 CustomWidgetListItem (class in src.gui.custom\_widgets), 6  
**D**  
 DataNotFoundError, 25  
 default() (src.utils.encoders.CustomJsonDecimalEncoder method), 24  
 DEFAULT\_MESSAGE\_DURATION (in module src.utils.constants), 22  
 DEFAULT\_SOURCE\_CRS (in module src.utils.constants), 22  
 DEFAULT\_TARGET\_CRS (in module src.utils.constants), 23  
 dockwidget (src.kan\_imagery\_catalog.KANImageryCatalog attribute), 11  
 download\_images() (src.gui.custom\_widgets.CustomWidgetListItem method), 7  
 download\_path (src.core.settings.PluginSettings attribute), 5  
 DOWNLOAD\_URL (in module src.services.element84), 17  
 DOWNLOAD\_URL (in module src.services.microsoft), 17  
 DOWNLOAD\_URL (in module src.services.sentinel\_hub), 18  
 DOWNLOAD\_URL (in module src.services.up42), 19  
**E**  
 encode\_base64() (in module src.services.utils), 20  
 ENDPOINT (in module src.plugin\_upload), 15  
 error\_message() (in module src.utils.qgis\_helper), 27  
 error\_signal (src.gui.helpers.worker.WorkerThread attribute), 11  
 error\_signal (src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock attribute), 13  
**F**  
 feature\_data (src.gui.custom\_widgets.CustomWidgetListItem attribute), 6  
 feature\_index (src.gui.custom\_widgets.CustomWidgetListItem attribute), 6  
 finished (src.gui.helpers.worker.WorkerThread attribute), 11  
 FONT\_SIZE\_DEFAULT (src.utils.constants.StyleVariables attribute), 23  
 FONT\_SIZE\_SMALL (src.utils.constants.StyleVariables attribute), 23  
 footprints\_layer (src.gui.custom\_widgets.CustomWidgetListItem attribute), 6  
 FormBase (class in src.gui.form\_base), 7  
 FormCatalogInfo (class in src.gui.form\_catalog\_info), 8  
 FormDefaultCollections (class in src.gui.form\_default\_collections), 8  
 FormSettings (class in src.gui.form\_settings), 9

`frm_collection_settings`  
(*src.kan\_imagery\_catalog.KANImageryCatalog*  
*attribute*), 11

`frm_settings` (*src.kan\_imagery\_catalog.KANImageryCatalog*  
*attribute*), 11

## G

`get_active_providers()`  
(*src.core.settings.PluginSettings* *method*),  
5

`get_bounding_box_canvas()` (in *module*  
*src.utils.qgis\_helper*), 27

`get_bounding_box_transformed()` (in *module*  
*src.utils.qgis\_helper*), 27

`get_catalog()` (in *module src.core.catalogs*), 3

`get_catalog()` (in *module src.services.element84*), 17

`get_catalog()` (in *module src.services.microsoft*), 18

`get_catalog()` (in *module src.services.sentinel\_hub*),  
18

`get_catalog()` (in *module src.services.up42*), 19

`get_catalogs_from_collection()` (in *module*  
*src.core.catalogs*), 3

`get_collections()` (in *module src.core.collections*), 4

`get_collections()` (in *module*  
*src.services.element84*), 17

`get_collections()` (in *module src.services.microsoft*),  
17

`get_collections()` (in *module*  
*src.services.sentinel\_hub*), 18

`get_collections()` (in *module src.services.up42*), 19

`get_custom_query()` (in *module src.core.catalogs*), 3

`get_data_finished()`  
(*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock*  
*method*), 14

`get_date_for_label()`  
(*src.gui.custom\_widgets.CustomWidgetListItem*  
*method*), 7

`get_download()` (in *module src.core.catalogs*), 4

`get_download_url()` (in *module src.core.catalogs*), 4

`get_georeferenced_image()` (in *module*  
*src.utils.qgis\_helper*), 28

`get_layer_by_name()` (in *module*  
*src.utils.qgis\_helper*), 27

`get_or_create_footprints_layer()` (in *module*  
*src.utils.qgis\_helper*), 28

`get_or_create_group()` (in *module*  
*src.utils.qgis\_helper*), 27

`get_plugin_dir()` (in *module src.utils.general*), 25

`get_quicklook()` (in *module src.core.catalogs*), 4

`get_quicklook()` (in *module src.services.element84*),  
17

`get_quicklook()` (in *module src.services.microsoft*), 18

`get_quicklook()` (in *module*  
*src.services.sentinel\_hub*), 18

`get_quicklook()` (in *module src.services.up42*), 19

`get_quicklook_image()`  
(*src.gui.custom\_widgets.CustomWidgetListItem*  
*method*), 7

`get_results()` (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock*  
*method*), 14

`get_selected_feature_bounding_box()` (in *module*  
*src.utils.qgis\_helper*), 27

`get_shadow_effect()` (in *module*  
*src.gui.helpers.forms*), 10

`get_thumbnail()` (in *module src.core.catalogs*), 3

`get_thumbnail()` (in *module src.services.element84*),  
17

`get_thumbnail()` (in *module src.services.microsoft*), 18

`get_thumbnail()` (in *module*  
*src.services.sentinel\_hub*), 18

`get_thumbnail()` (in *module src.services.up42*), 19

`get_thumbnails_in_background()`  
(*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock*  
*method*), 14

`get_token()` (in *module src.services.sentinel\_hub*), 18

`get_token()` (in *module src.services.up42*), 19

`get_valid_project_layers_to_search()` (in *mod-*  
*ule src.utils.qgis\_helper*), 27

## H

`hide_password()` (in *module src.plugin\_upload*), 15

`host` (*src.gui.custom\_widgets.CustomWidgetListItem* *at-*  
*tribute*), 6

`HostError`, 25

`http_get()` (in *module src.services.utils*), 20

`http_post()` (in *module src.services.utils*), 20

`iface` (*src.kan\_imagery\_catalog.KANImageryCatalog*  
*attribute*), 11

`image_id` (*src.gui.custom\_widgets.CustomWidgetListItem*  
*attribute*), 6

`incidence_angle` (*src.gui.custom\_widgets.CustomWidgetListItem*  
*property*), 6

`INFO` (*src.utils.constants.MessageType* *attribute*), 23

`info_message()` (in *module src.utils.qgis\_helper*), 27

`info_signal` (*src.gui.helpers.worker.WorkerThread* *at-*  
*tribute*), 11

`info_signal` (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock*  
*attribute*), 13

`init_tableview()` (in *module src.gui.helpers.forms*),  
10

`initGui()` (*src.kan\_imagery\_catalog.KANImageryCatalog*  
*method*), 12

## K

`KANImageryCatalog` (class in  
*src.kan\_imagery\_catalog*), 11

KANImageryCatalogDock (class in  
src.kan\_imagery\_catalog\_dock), 13  
key (src.gui.custom\_widgets.CustomWidgetListItem attribute), 6

## L

language (src.core.settings.PluginSettings attribute), 5  
last\_cleaning\_date (src.core.settings.PluginSettings property), 5  
load\_collections\_cache()  
(src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock method), 14  
load\_combobox() (in module src.gui.helpers.forms), 10  
load\_data() (src.gui.form\_catalog\_info.FormCatalogInfo method), 8  
load\_filtered\_collections()  
(src.gui.form\_default\_collections.FormDefaultCollections method), 8  
load\_selected\_collections()  
(src.gui.form\_default\_collections.FormDefaultCollections method), 8  
load\_table\_data() (in module src.gui.helpers.forms), 10  
loading\_spinner (src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock attribute), 13

## M

main() (in module src.plugin\_upload), 15  
max\_catalog\_results  
(src.core.settings.PluginSettings attribute), 5  
max\_features\_results  
(src.core.settings.PluginSettings attribute), 5  
menu (src.kan\_imagery\_catalog.KANImageryCatalog attribute), 11  
message (in module src.utils.qgis\_helper), 27  
message (src.utils.exceptions.PluginError attribute), 25  
message\_type (src.utils.exceptions.PluginError attribute), 24  
MessageType (class in src.utils.constants), 23  
module  
src, 3  
src.core, 3  
src.core.catalogs, 3  
src.core.collections, 4  
src.core.providers, 4  
src.core.results, 4  
src.core.settings, 5  
src.gui, 6  
src.gui.custom\_widgets, 6  
src.gui.form\_base, 7  
src.gui.form\_catalog\_info, 7  
src.gui.form\_default\_collections, 8

src.gui.form\_settings, 9  
src.gui.helpers, 9  
src.gui.helpers.forms, 9  
src.gui.helpers.worker, 10  
src.kan\_imagery\_catalog, 11  
src.kan\_imagery\_catalog\_dock, 13  
src.plugin\_upload, 15  
src.resources, 16  
src.services, 16  
src.services.element84, 16  
src.services.microsoft, 17  
src.services.sentinel\_hub, 18  
src.services.up42, 18  
src.services.utils, 19  
src.ui, 20  
src.ui.custom\_widget\_list\_item, 20  
src.ui.frm\_catalog\_info, 20  
src.ui.frm\_default\_collections, 21  
src.ui.frm\_settings, 21  
src.utils, 22  
src.utils.constants, 22  
src.utils.debug, 23  
src.utils.encoders, 24  
src.utils.exceptions, 24  
src.utils.general, 25  
src.utils.helpers, 26  
src.utils.qgis\_helper, 26

## N

name (src.gui.custom\_widgets.CustomWidgetListItem property), 6  
normalize\_dict\_data()  
(src.gui.form\_catalog\_info.FormCatalogInfo method), 8  
normalize\_text() (in module src.utils.helpers), 26

## O

onClosePlugin() (src.kan\_imagery\_catalog.KANImageryCatalog method), 12  
open\_url() (in module src.utils.helpers), 26

## P

parent (src.gui.custom\_widgets.CustomWidgetListItem attribute), 6  
parent (src.gui.form\_base.FormBase attribute), 7  
parent (src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock attribute), 13  
parser (in module src.plugin\_upload), 15  
planet\_is\_valid (src.gui.form\_settings.FormSettings attribute), 9  
planet\_settings (src.gui.form\_settings.FormSettings attribute), 9  
plugin\_dir (src.kan\_imagery\_catalog.KANImageryCatalog attribute), 11



PLUGIN\_NAME (in module *src.utils.general*), 25  
 PluginError, 24  
 pluginIsActive (*src.kan\_imagery\_catalog.KANImageryCatalog* attribute), 11  
 PluginSettings (class in *src.core.settings*), 5  
 PORT (in module *src.plugin\_upload*), 15  
 progress\_updated (*src.gui.helpers.worker.WorkerThread* attribute), 11  
 PROTOCOL (in module *src.plugin\_upload*), 15  
 provider (*src.gui.custom\_widgets.CustomWidgetListItem* attribute), 6  
 provider\_settings (*src.core.settings.PluginSettings* property), 5  
 ProviderError, 25  
 providers (*src.gui.form\_default\_collections.FormDefaultCollections* attribute), 8  
 pyqt\_set\_trace() (in module *src.utils.debug*), 24

## Q

qCleanupResources() (in module *src.resources*), 16  
 qInitResources() (in module *src.resources*), 16  
 qt\_resource\_data (in module *src.resources*), 16  
 qt\_resource\_name (in module *src.resources*), 16  
 qt\_resource\_struct\_v1 (in module *src.resources*), 16  
 qt\_resource\_struct\_v2 (in module *src.resources*), 16  
 qt\_version (in module *src.resources*), 16

## R

rcc\_version (in module *src.resources*), 16  
 read\_json\_setting() (in module *src.utils.qgis\_helper*), 27  
 read\_setting() (in module *src.utils.qgis\_helper*), 27  
 REQUEST\_TIMEOUT (in module *src.services.utils*), 20  
 RESULTS\_GROUP\_NAME (in module *src.utils.constants*), 22  
 RESULTS\_LAYER\_NAME (in module *src.utils.constants*), 22  
 retranslateUi() (*src.ui.custom\_widget\_list\_item.Ui\_CustomWidgetListItem* method), 20  
 retranslateUi() (*src.ui.frm\_catalog\_info.Ui\_frm\_catalog\_info* method), 21  
 retranslateUi() (*src.ui.frm\_default\_collections.Ui\_frm\_default\_collections* method), 21  
 retranslateUi() (*src.ui.frm\_settings.Ui\_frm\_settings* method), 22  
 run() (*src.gui.helpers.worker.WorkerThread* method), 11  
 run() (*src.kan\_imagery\_catalog.KANImageryCatalog* method), 12

## S

save() (*src.core.settings.PluginSettings* method), 5  
 save\_json\_setting() (in module *src.utils.qgis\_helper*), 27  
 save\_setting() (in module *src.utils.qgis\_helper*), 27  
 selected\_collections (*src.core.settings.PluginSettings* property),  
 selected\_collections (*src.gui.form\_default\_collections.FormDefaultCollections* attribute), 8  
 sentinelhub\_is\_valid (*src.gui.form\_settings.FormSettings* attribute), 9  
 sentinelhub\_settings (*src.gui.form\_settings.FormSettings* attribute), 9  
 SERVER (in module *src.plugin\_upload*), 15  
 set\_control\_buttons() (*src.gui.form\_base.FormBase* method), 7  
 set\_elided\_text\_to\_label() (in module *src.gui.helpers.forms*), 10  
 set\_form\_state() (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock* method), 13  
 set\_form\_stylesheet() (in module *src.gui.helpers.forms*), 10  
 set\_thumbnail() (*src.gui.custom\_widgets.CustomWidgetListItem* method), 7  
 set\_tooltips() (in module *src.gui.helpers.forms*), 10  
 settings (*src.gui.form\_default\_collections.FormDefaultCollections* attribute), 8  
 settings (*src.gui.form\_settings.FormSettings* attribute), 9  
 settings (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock* attribute), 13  
 SettingsError, 25  
 setupUi() (*src.ui.custom\_widget\_list\_item.Ui\_CustomWidgetListItem* method), 20  
 setupUi() (*src.ui.frm\_catalog\_info.Ui\_frm\_catalog\_info* method), 21  
 setupUi() (*src.ui.frm\_default\_collections.Ui\_frm\_default\_collections* method), 21  
 setupUi() (*src.ui.frm\_settings.Ui\_frm\_settings* method), 22  
 show\_collections\_form() (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock* method), 14  
 show\_ended\_search\_message (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock* attribute), 13  
 show\_error() (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock* method), 13  
 show\_info() (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock* method), 13  
 show\_quicklook() (*src.gui.custom\_widgets.CustomWidgetListItem* method), 7  
 show\_warning() (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock* method), 13  
 sort\_ascending (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock* method), 13

*attribute*), 13

src

- module, 3

src.core

- module, 3

src.core.catalogs

- module, 3

src.core.collections

- module, 4

src.core.providers

- module, 4

src.core.results

- module, 4

src.core.settings

- module, 5

src.gui

- module, 6

src.gui.custom\_widgets

- module, 6

src.gui.form\_base

- module, 7

src.gui.form\_catalog\_info

- module, 7

src.gui.form\_default\_collections

- module, 8

src.gui.form\_settings

- module, 9

src.gui.helpers

- module, 9

src.gui.helpers.forms

- module, 9

src.gui.helpers.worker

- module, 10

src.kan\_imagery\_catalog

- module, 11

src.kan\_imagery\_catalog\_dock

- module, 13

src.plugin\_upload

- module, 15

src.resources

- module, 16

src.services

- module, 16

src.services.element84

- module, 16

src.services.microsoft

- module, 17

src.services.sentinel\_hub

- module, 18

src.services.up42

- module, 18

src.services.utils

- module, 19

src.ui

module, 20

src.ui.custom\_widget\_list\_item

- module, 20

src.ui.frm\_catalog\_info

- module, 20

src.ui.frm\_default\_collections

- module, 21

src.ui.frm\_settings

- module, 21

src.utils

- module, 22

src.utils.constants

- module, 22

src.utils.debug

- module, 23

src.utils.encoders

- module, 24

src.utils.exceptions

- module, 24

src.utils.general

- module, 25

src.utils.helpers

- module, 26

src.utils.qgis\_helper

- module, 26

start() (*src.gui.helpers.worker.WorkerThread* method), 11

StyleVariables (*class in src.utils.constants*), 23

SUCCESS (*src.utils.constants.MessageType* attribute), 23

success\_message() (*in module src.utils.qgis\_helper*), 27

## T

thread\_get\_catalogs

- (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock* attribute), 13

thread\_get\_thumbnails

- (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock* attribute), 13

thread\_load\_collections\_cache

- (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock* attribute), 13

thread\_quicklooks (*src.gui.custom\_widgets.CustomWidgetListItem* attribute), 6

to\_dict() (*src.utils.constants.CustomButton* class method), 23

toolbar (*src.kan\_imagery\_catalog.KANImageryCatalog* attribute), 11

tr() (*in module src.utils.helpers*), 26

tr() (*src.kan\_imagery\_catalog.KANImageryCatalog* method), 12

txt\_default\_back\_days\_text\_changed()

- (*src.gui.form\_settings.FormSettings* method), 9



txt\_max\_catalog\_results\_text\_changed()  
     (*src.gui.form\_settings.FormSettings method*), 9  
 txt\_max\_catalog\_results\_text\_changed()  
     (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock  
     method*), 13  
 txt\_max\_features\_results\_text\_changed()  
     (*src.gui.form\_settings.FormSettings method*), 9

## U

Ui\_CustomWidgetListItem (class in  
     *src.ui.custom\_widget\_list\_item*), 20  
 Ui\_frm\_catalog\_info (class in  
     *src.ui.frm\_catalog\_info*), 21  
 Ui\_frm\_default\_collections (class in  
     *src.ui.frm\_default\_collections*), 21  
 Ui\_frm\_settings (class in *src.ui.frm\_settings*), 22  
 unload() (*src.kan\_imagery\_catalog.KANImageryCatalog  
     method*), 12  
 up42\_is\_valid (*src.gui.form\_settings.FormSettings at-  
     tribute*), 9  
 up42\_settings (*src.gui.form\_settings.FormSettings at-  
     tribute*), 9  
 update\_cloud\_coverage\_label()  
     (*src.gui.form\_settings.FormSettings method*), 9  
 update\_cloud\_coverage\_label()  
     (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock  
     method*), 13  
 update\_last\_cleaning\_date()  
     (*src.core.settings.PluginSettings method*),  
     5  
 update\_progress() (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock  
     method*), 14

## V

VERBOSE (in module *src.plugin\_upload*), 15  
 view\_details() (*src.gui.custom\_widgets.CustomWidgetListItem  
     method*), 7

## W

WARNING (*src.utils.constants.MessageType attribute*), 23  
 warning\_message() (in module *src.utils.qgis\_helper*),  
     27  
 warning\_signal (*src.gui.helpers.worker.WorkerThread  
     attribute*), 11  
 warning\_signal (*src.kan\_imagery\_catalog\_dock.KANImageryCatalogDock  
     attribute*), 13  
 WorkerThread (class in *src.gui.helpers.worker*), 11

## Z

zoom\_selected\_features() (in module  
     *src.utils.qgis\_helper*), 28