



KAN Imagery Catalog

Release v1.6.1

Kan Territory & IT

Jan 21, 2025

CONTENTS:

1	API Reference	3
1.1	src	3
	Python Module Index	29
	Index	31

QGIS plugin to get satellite images from different providers.

API REFERENCE

This page contains auto-generated API reference documentation¹.

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.

¹ 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>	<code>image_id,</code>	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 = None

key = 0.0

property name

property acquisition_date

property incidence_angle

property cloud_coverage**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*FormBase*

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*FormCatalogInfo*

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*)**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(parameters, arguments)</i>	Main entry point.
<i>hide_password(url[, start])</i>	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>	
<code>http_post(url[, host_name, headers, payload, ...])</code>	

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)
```

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

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

```
Ui_frm_settings
```

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
    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

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

<code>pyqt_set_trace()</code>	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

<i>CustomJsonDecimalEncoder</i>	Custom JSON encoder for Decimal objects.
---------------------------------	--

Module Contents

```
class src.utils.encoders.CustomJsonDecimalEncoder(*, 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

<i>PluginError</i>	Base class for custom exceptions.
<i>ProviderError</i>	Custom exception for provider related errors.
<i>HostError</i>	Custom exception for host related errors.
<i>AuthorizationError</i>	Custom exception for authorization related errors.
<i>SettingsError</i>	Custom exception for settings related errors.
<i>DataNotFoundError</i>	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

<i>PLUGIN_NAME</i>

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

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

message

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.
<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

`classFactory(iface)`

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`, 14
- `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`, 25
 - `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), 22
- `__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*), 27
- `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`, 24
- 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 src.utils.encoders), 24
 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), 27
 create_quicklook_layer() (src.gui.custom_widgets.CustomWidgetListItem method), 7
 CRITICAL (src.utils.constants.MessageType attribute), 23
 CustomEnum (class in src.utils.constants), 22
 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), 22
 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 attribute), 14
(method), 14

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), 27

get_layer_by_name() (in module src.utils.qgis_helper), 27

get_or_create_footprints_layer() (in module src.utils.qgis_helper), 27

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 attribute), 14
(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 attribute), 14
(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 module src.utils.qgis_helper), 27

H

hide_password() (in module src.plugin_upload), 15

host (src.gui.custom_widgets.CustomWidgetListItem attribute), 6

HostError, 24

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

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

I

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 attribute), 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), 24

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, 14

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, 25

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, 24
 providers(*src.gui.form_default_collections.FormDefaultCollections* attribute), 8
 pyqt_set_trace() (in module *src.utils.debug*), 23

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), 5
 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_enabled_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* 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, 14
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, 25
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.CustomEnum* 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 attribute*), 9
 up42_settings (*src.gui.form_settings.FormSettings attribute*), 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*), 27