



KAN Imagery Catalog

Release v0.1.3

Fernando D. Gómez :: Kan Territory & IT

Oct 27, 2023

CONTENTS:

1	API Reference	3
1.1	src	3
	Python Module Index	27
	Index	29

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 Subpackages

`src.core`

Submodules

`src.core.catalogs`

Catalogs module.

Module Contents

Functions

<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> (provider)	Get catalog download url from a specific provider.

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

Get catalog thumbnail from a specific provider.

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

Get catalog download url from a specific provider.

`src.core.collections`

Collections module.

Module Contents

Functions

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

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

Get collections from a specific provider.

`src.core.providers`

Providers module.

Module Contents

Functions

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

`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.

Module Contents

Functions

<code>create_results_layers(provider_name, host, ...)</code>	Create results layers.
<code>create_quicklook(provider_name, host, image_id, layer_name)</code>	Create results layers.

`src.core.results.create_results_layers(provider_name, host, feature_data, layer_name, group_name, footprints_layer)`

Create results layers.

`src.core.results.create_quicklook(provider_name, host, image_id, layer_name)`

Create results layers.

src.core.settings

PluginSettings module.

Module Contents

Classes

<code>PluginSettings</code>	PluginSettings class.
-----------------------------	-----------------------

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

PluginSettings class.

```
    property provider_settings: dict
```

```
    property selected_collections: list
```

```
    property last_cleaning_date: datetime.datetime
```

```
    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
_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.
```

`src.gui`

Subpackages

`src.gui.helpers`

Submodules

`src.gui.helpers.forms`

Form helpers functions module.

Module Contents

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.

```
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.

Submodules

```
src.gui.custom_widgets
```

Custom widgets module.

Module Contents

Classes

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

```
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.

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.

get_quicklook()

Get quicklook action.

src.gui.form_base

Module Contents

Classes

FormBase

class src.gui.form_base.**FormBase**(parent=None, accept_btn=False, cancel_btn=False, close_btn=True, closing_plugin=None)

Bases: PyQt5.QtWidgets.QDialog

set_control_buttons(accept_btn, cancel_btn, close_btn)

btn_close_click()

src.gui.form_catalog_info

Form Catalog Info Module.

Module Contents

Classes

FormCatalogInfo

Form Catalog Info Class.

```
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.

Module Contents

Classes

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

```
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.

```
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.

Module Contents

Classes

<i>FormSettings</i>	Form PluginSettings Class.
---------------------	----------------------------

```
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.

```
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.

`src.services`

Submodules

`src.services.sentinel_hub`

Module for Sentinel Hub API calls

Module Contents

Functions

<i>get_token</i> (client_id, client_secret)	Get token from Sentinel Hub API
<i>get_token_info</i> (client_id)	Get token info from Sentinel Hub API

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

Get token from Sentinel Hub API

```
src.services.sentinel_hub.get_token_info(client_id)
```

Get token info from Sentinel Hub API

src.services.up42

Module for Up42 API calls

Module Contents

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

Attributes

<code>REQUEST_TIMEOUT</code>
<code>DOWNLOAD_URL</code>

```
src.services.up42.REQUEST_TIMEOUT = 120
```

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

```
src.services.up42.get_token(project_id: str, api_key: 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

Module Contents

Functions

<code>encode_base64</code> (\rightarrow str)	Encode a string to Base64
---	---------------------------

`src.services.utils.encode_base64(text_to_encode: str) \rightarrow str`
Encode a string to Base64

`src.ui`

Submodules

`src.ui.custom_widget_list_item`

Module Contents

Classes

<code>Ui_CustomWidgetListItem</code>

Attributes

<code>app</code>

```
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`

Module Contents

Classes

Ui_frm_catalog_info

Attributes

app

```
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`

Module Contents

Classes

Ui_frm_default_collections

Attributes

app

```
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

Module Contents

Classes

<i>Ui_frm_settings</i>

Attributes

<i>app</i>

```

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

Module Contents

Classes

<i>CustomEnum</i>	Generic enumeration.
<i>StyleVariables</i>	Generic enumeration.
<i>MessageType</i>	

Attributes

DEBUG_MODE

RESULTS_GROUP_NAME

RESULTS_LAYER_NAME

```
src.utils.constants.DEBUG_MODE = True

src.utils.constants.RESULTS_GROUP_NAME = 'kan_imagery_catalog_preview'

src.utils.constants.RESULTS_LAYER_NAME = 'kan_imagery_catalog_footprints'

class src.utils.constants.CustomEnum
    Bases: enum.Enum
    Generic enumeration.
    Derive from this class to define new enumerations.
    __repr__() → str
        Return repr(self).
    __str__() → str
        Return str(self).
    __eq__(other)
        Return self==value.
    classmethod listar()
    classmethod to_dict() → dict
    classmethod to_key_value_list() → list

class src.utils.constants.StyleVariables
    Bases: CustomEnum
    Generic enumeration.
    Derive from this class to define new enumerations.
    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
```

```
    INFO = 0
    WARNING = 1
    CRITICAL = 2
    SUCCESS = 3
```

```
src.utils.debug
```

Module Contents

Functions

```
pyqt_set_trace()
```

```
src.utils.debug.pyqt_set_trace()
```

src.utils.encoders

Custom JSON encoders for the application.

Module Contents

Classes

```
CustomJsonDecimalEncoder
```

Custom JSON encoder for Decimal objects.

```
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.

Module Contents

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

Bases: `Exception`

Base class for custom exceptions.

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

Bases: `PluginError`

Custom exception for provider related errors.

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

Bases: `PluginError`

Custom exception for host related errors.

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

Bases: `PluginError`

Custom exception for authorization related errors.

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

Bases: `PluginError`

Custom exception for settings related errors.

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

Bases: `PluginError`

Custom exception for data related errors.

src.utils.general

General utilities for the KAN Imagery Catalog plugin.

Module Contents

Functions

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

Attributes

PLUGIN_NAME

`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`

`src.utils.general.clean_temporary_files()`

`src.utils.helpers`

Module Contents

Functions

open_url(\rightarrow bool)

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

`src.utils.qgis_helper`

QGIS helper functions module.

Module Contents

Functions

<code>get_bounding_box_canvas()</code>	Get bounding box from canvas.
<code>get_bounding_box_selected_feature(layer_name)</code>	Get bounding box from selected feature.
<code>get_single_polygon_layers()</code>	Get single polygon layers from current project.
<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>get_current_crs()</code>	Get crs from active layer.
<code>create_layer(layer_name, group_name)</code>	Create a layer.
<code>add_feature_to_layer(feature_data, feature_id, layer)</code>	Add feature to layer.
<code>create_quicklook_layer(layer_name, group_name, ...[, ...])</code>	Create a layer with a quicklook image.

`src.utils.qgis_helper.get_bounding_box_canvas()`

Get bounding box from canvas.

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

Get bounding box from selected feature.

`src.utils.qgis_helper.get_single_polygon_layers()`

Get single polygon layers from current project.

`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.get_current_crs()`

Get crs from active layer.

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

Create a layer.

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

Add feature to layer.

`src.utils.qgis_helper.create_quicklook_layer(layer_name, group_name, feature, crs,
layer_type='Polygon', image_path=None)`

Create a layer with a quicklook image.

1.1.2 Submodules

`src.kan_imagery_catalog`

Module Contents

Classes

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

class `src.kan_imagery_catalog.KANImageryCatalog(iface)`

QGIS Plugin Implementation.

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`

Module Contents

Classes

`KANImageryCatalogDock`

KANImageryCatalogDock Dockwidget class.

```
class src.kan_imagery_catalog_dock.KANImageryCatalogDock(parent=None)
    Bases: PyQt5.QtWidgets.QDockWidget, FORM_CLASS
    KANImageryCatalogDock Dockwidget class.

closing_plugin

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'.

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_results(layer_name, cloud_coverage, date_from, date_to, max_catalog_results)
    Get results from selected catalogs with selected filters.

add_item_to_results(provider_name, host_name, collection_name, feature_data, acquisition_date,
                    incidence_angle, cloud_coverage, image_id, feature_index)
    Add item to results list.

add_feature_to_footprints_layer(feature_data, footprint_id)
```

src.plugin_upload

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

Module Contents

Functions

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

Attributes

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

```
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`

Module Contents

Functions

`qInitResources()`

`qCleanupResources()`

Attributes

`qt_resource_data`

`qt_resource_name`

`qt_resource_struct_v1`

`qt_resource_struct_v2`

`qt_version`

`rcc_version`

```
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...
```

```
src.resources.qt_resource_struct_v2 = b'\x00\x00\x00\x00\x00\x02\x00\x00\x00\x02\x00\x00\x00\x00\x01\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()
```

1.1.3 Package Contents

Functions

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

`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`, 5
 - `src.core.settings`, 5
- `src.gui`, 6
 - `src.gui.custom_widgets`, 7
 - `src.gui.form_base`, 8
 - `src.gui.form_catalog_info`, 8
 - `src.gui.form_default_collections`, 9
 - `src.gui.form_settings`, 10
 - `src.gui.helpers`, 6
 - `src.gui.helpers.forms`, 6
 - `src.kan_imagery_catalog`, 20
 - `src.kan_imagery_catalog_dock`, 21
 - `src.plugin_upload`, 23
 - `src.resources`, 24
 - `src.services`, 10
 - `src.services.sentinel_hub`, 10
 - `src.services.up42`, 11
 - `src.services.utils`, 12
 - `src.ui`, 12
 - `src.ui.custom_widget_list_item`, 12
 - `src.ui.frm_catalog_info`, 13
 - `src.ui.frm_default_collections`, 13
 - `src.ui.frm_settings`, 14
 - `src.utils`, 14
 - `src.utils.constants`, 14
 - `src.utils.debug`, 16
 - `src.utils.encoders`, 16
 - `src.utils.exceptions`, 17
 - `src.utils.general`, 17
 - `src.utils.helpers`, 18
 - `src.utils.qgis_helper`, 18

INDEX

Symbols

`__eq__()` (*src.utils.constants.CustomEnum* method), 15
`__post_init__()` (*src.core.settings.PluginSettings* method), 6
`__repr__()` (*src.utils.constants.CustomEnum* method), 15
`__str__()` (*src.utils.constants.CustomEnum* method), 15
`_get_plugin_name()` (in module *src.utils.general*), 18
`_last_cleaning_date` (*src.core.settings.PluginSettings* attribute), 6
`_provider_settings` (*src.core.settings.PluginSettings* attribute), 6
`_selected_collections` (*src.core.settings.PluginSettings* attribute), 6

A

`acquisition_date` (*src.gui.custom_widgets.CustomWidgetListItem* property), 8
`add_action()` (*src.kan_imagery_catalog.KANImageryCatalogDock* method), 20
`add_feature_to_footprints_layer()` (*src.kan_imagery_catalog_dock.KANImageryCatalogDock* method), 22
`add_feature_to_layer()` (in module *src.utils.qgis_helper*), 20
`add_item_to_results()` (*src.kan_imagery_catalog_dock.KANImageryCatalogDock* method), 22
`app` (in module *src.ui.custom_widget_list_item*), 12
`app` (in module *src.ui.frm_catalog_info*), 13
`app` (in module *src.ui.frm_default_collections*), 13
`app` (in module *src.ui.frm_settings*), 14
`AuthorizationError`, 17

B

`back_days` (*src.core.settings.PluginSettings* attribute), 5
`btn_accept_clicked()` (*src.gui.form_default_collections.FormDefaultCollections* method), 9

`btn_accept_clicked()` (*src.gui.form_settings.FormSettings* method), 10
`btn_add_selected_clicked()` (*src.gui.form_default_collections.FormDefaultCollections* method), 9
`btn_close_click()` (*src.gui.form_base.FormBase* method), 8
`btn_filter_results_clicked()` (*src.gui.form_default_collections.FormDefaultCollections* method), 9
`btn_get_data_clicked()` (*src.kan_imagery_catalog_dock.KANImageryCatalogDock* method), 22
`btn_remove_selected_clicked()` (*src.gui.form_default_collections.FormDefaultCollections* method), 9
`btn_select_catalogs_clicked()` (*src.kan_imagery_catalog_dock.KANImageryCatalogDock* method), 22
`btn_settings_clicked()` (*src.kan_imagery_catalog_dock.KANImageryCatalogDock* method), 22
`btn_show_hide_catalogs_clicked()` (*src.kan_imagery_catalog_dock.KANImageryCatalogDock* method), 22
`btn_show_hide_filters_clicked()` (*src.kan_imagery_catalog_dock.KANImageryCatalogDock* method), 22
`btn_show_hide_search_area_clicked()` (*src.kan_imagery_catalog_dock.KANImageryCatalogDock* method), 22
`btn_sort_results_clicked()` (*src.kan_imagery_catalog_dock.KANImageryCatalogDock* method), 22
`btn_up42_check_credentials_clicked()` (*src.gui.form_settings.FormSettings* method), 10
`btn_update_layers_list_clicked()` (*src.kan_imagery_catalog_dock.KANImageryCatalogDock* method), 22

C

check_credentials() (in module *src.core.providers*), 4

check_int_not_empty() (in module *src.gui.helpers.forms*), 7

chk_search_by_dataframe_update() (*src.kan_imagery_catalog_dock.KANImageryCatalogDock* method), 22

classFactory() (in module *src*), 25

clean_temporary_files() (in module *src.utils.general*), 18

clean_temporary_files_if_needed() (*src.core.settings.PluginSettings* method), 6

closeEvent() (*src.kan_imagery_catalog_dock.KANImageryCatalogDock* method), 22

closing_plugin(*src.kan_imagery_catalog_dock.KANImageryCatalogDock* attribute), 22

cloud_coverage (*src.core.settings.PluginSettings* attribute), 5

cloud_coverage (*src.gui.custom_widgets.CustomWidgetListItem* property), 8

COLOR_BACKGROUND_DEFAULT (*src.utils.constants.StyleVariables* attribute), 15

COLOR_MAIN_DARK (*src.utils.constants.StyleVariables* attribute), 15

COLOR_MAIN_LIGHT (*src.utils.constants.StyleVariables* attribute), 15

COLOR_TEXT_DEFAULT (*src.utils.constants.StyleVariables* attribute), 15

COLOR_TEXT_LIGHT (*src.utils.constants.StyleVariables* attribute), 15

create_layer() (in module *src.utils.qgis_helper*), 20

create_quicklook() (in module *src.core.results*), 5

create_quicklook_layer() (in module *src.utils.qgis_helper*), 20

create_results_layers() (in module *src.core.results*), 5

CRITICAL (*src.utils.constants.MessageType* attribute), 16

CustomEnum (class in *src.utils.constants*), 15

CustomJsonDecimalEncoder (class in *src.utils.encoders*), 16

CustomWidgetListItem (class in *src.gui.custom_widgets*), 7

D

DataNotFoundError, 17

DEBUG_MODE (in module *src.utils.constants*), 15

default() (*src.utils.encoders.CustomJsonDecimalEncoder* method), 16

download_images() (*src.gui.custom_widgets.CustomWidgetListItem* method), 8

download_path (*src.core.settings.PluginSettings* attribute), 5

DOWNLOAD_URL (in module *src.services.up42*), 11

E

encode_base64() (in module *src.services.utils*), 12

ENDPOINT (in module *src.plugin_upload*), 23

error_message() (in module *src.utils.qgis_helper*), 19

F

FONT_SIZE_DEFAULT (*src.utils.constants.StyleVariables* attribute), 15

FONT_SIZE_SMALL (*src.utils.constants.StyleVariables* attribute), 15

FormBase (class in *src.gui.form_base*), 8

FormCatalogInfo (class in *src.gui.form_catalog_info*), 8

FormDefaultCollections (class in *src.gui.form_default_collections*), 9

FormSettings (class in *src.gui.form_settings*), 10

G

get_bounding_box_canvas() (in module *src.utils.qgis_helper*), 19

get_bounding_box_selected_feature() (in module *src.utils.qgis_helper*), 19

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

get_catalog() (in module *src.services.up42*), 11

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

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

get_collections() (in module *src.services.up42*), 11

get_current_crs() (in module *src.utils.qgis_helper*), 20

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

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

get_layer_by_name() (in module *src.utils.qgis_helper*), 20

get_or_create_footprints_layer() (in module *src.utils.qgis_helper*), 20

get_or_create_group() (in module *src.utils.qgis_helper*), 20

get_plugin_dir() (in module *src.utils.general*), 18

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

get_quicklook() (in module *src.services.up42*), 11

get_quicklook() (*src.gui.custom_widgets.CustomWidgetListItem* method), 8

get_results() (*src.kan_imagery_catalog_dock.KANImageryCatalogDock* method), 22

get_shadow_effect() (in module *src.gui.helpers.forms*), 7

get_single_polygon_layers() (in module *src.utils.qgis_helper*), 19

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

get_thumbnail() (in module *src.services.up42*), 11

get_token() (in module *src.services.sentinel_hub*), 10
 get_token() (in module *src.services.up42*), 11
 get_token_info() (in module *src.services.sentinel_hub*), 11

H

hide_password() (in module *src.plugin_upload*), 23
 HostError, 17

I

incidence_angle(*src.gui.custom_widgets.CustomWidgetListItem* property), 8
 INFO(*src.utils.constants.MessageType* attribute), 16
 info_message() (in module *src.utils.qgis_helper*), 19
 init_tableview() (in module *src.gui.helpers.forms*), 7
 initGui() (*src.kan_imagery_catalog.KANImageryCatalog* method), 21

K

KANImageryCatalog (class in *src.kan_imagery_catalog*), 20
 KANImageryCatalogDock (class in *src.kan_imagery_catalog_dock*), 21

L

language(*src.core.settings.PluginSettings* attribute), 5
 last_cleaning_date (*src.core.settings.PluginSettings* property), 5
 listar() (*src.utils.constants.CustomEnum* class method), 15
 load_combobox() (in module *src.gui.helpers.forms*), 6
 load_data() (*src.gui.form_catalog_info.FormCatalogInfo* method), 9
 load_filtered_collections() (*src.gui.form_default_collections.FormDefaultCollections* method), 9
 load_selected_collections() (*src.gui.form_default_collections.FormDefaultCollections* method), 9
 load_table_data() (in module *src.gui.helpers.forms*), 7

M

main() (in module *src.plugin_upload*), 23
 max_catalog_results (*src.core.settings.PluginSettings* attribute), 5
 max_features_results (*src.core.settings.PluginSettings* attribute), 5
 MessageType (class in *src.utils.constants*), 15
 module
 src, 3

src.core, 3
src.core.catalogs, 3
src.core.collections, 4
src.core.providers, 4
src.core.results, 5
src.core.settings, 5
src.gui, 6
src.gui.custom_widgets, 7
src.gui.form_base, 8
src.gui.form_catalog_info, 8
src.gui.form_default_collections, 9
src.gui.form_settings, 10
src.gui.helpers, 6
src.gui.helpers.forms, 6
src.kan_imagery_catalog, 20
src.kan_imagery_catalog_dock, 21
src.plugin_upload, 23
src.resources, 24
src.services, 10
src.services.sentinel_hub, 10
src.services.up42, 11
src.services.utils, 12
src.ui, 12
src.ui.custom_widget_list_item, 12
src.ui.frm_catalog_info, 13
src.ui.frm_default_collections, 13
src.ui.frm_settings, 14
src.utils, 14
src.utils.constants, 14
src.utils.debug, 16
src.utils.encoders, 16
src.utils.exceptions, 17
src.utils.general, 17
src.utils.helpers, 18
src.utils.qgis_helper, 18

N

name (*src.gui.custom_widgets.CustomWidgetListItem* property), 7
 normalize_dict_data() (*src.gui.form_catalog_info.FormCatalogInfo* method), 9

O

onClosePlugin() (*src.kan_imagery_catalog.KANImageryCatalog* method), 21
 open_url() (in module *src.utils.helpers*), 18

P

parser (in module *src.plugin_upload*), 24
 PLUGIN_NAME (in module *src.utils.general*), 18
 PluginError, 17
 PluginSettings (class in *src.core.settings*), 5
 PORT (in module *src.plugin_upload*), 23

PROTOCOL (in module *src.plugin_upload*), 23
provider_settings (*src.core.settings.PluginSettings*
property), 5
ProviderError, 17
pyqt_set_trace() (in module *src.utils.debug*), 16

Q

qCleanupResources() (in module *src.resources*), 24
qInitResources() (in module *src.resources*), 24
qt_resource_data (in module *src.resources*), 24
qt_resource_name (in module *src.resources*), 24
qt_resource_struct_v1 (in module *src.resources*), 24
qt_resource_struct_v2 (in module *src.resources*), 24
qt_version (in module *src.resources*), 24

R

rcc_version (in module *src.resources*), 24
read_json_setting() (in module
src.utils.qgis_helper), 19
read_setting() (in module *src.utils.qgis_helper*), 19
REQUEST_TIMEOUT (in module *src.services.up42*), 11
RESULTS_GROUP_NAME (in module *src.utils.constants*), 15
RESULTS_LAYER_NAME (in module *src.utils.constants*), 15
retranslateUi() (*src.ui.custom_widget_list_item.Ui_CustomWidgetListItem*
method), 12
retranslateUi() (*src.ui.frm_catalog_info.Ui_frm_catalog_info*
method), 13
retranslateUi() (*src.ui.frm_default_collections.Ui_frm_default_collections*
method), 13
retranslateUi() (*src.ui.frm_settings.Ui_frm_settings*
method), 14
run() (*src.kan_imagery_catalog.KANImageryCatalog*
method), 21

S

save() (*src.core.settings.PluginSettings* method), 6
save_json_setting() (in module
src.utils.qgis_helper), 19
save_setting() (in module *src.utils.qgis_helper*), 19
selected_collections
(*src.core.settings.PluginSettings* property),
5
SERVER (in module *src.plugin_upload*), 23
set_control_buttons()
(*src.gui.form_base.FormBase* method), 8
set_elided_text_to_label() (in module
src.gui.helpers.forms), 7
set_form_stylesheet() (in module
src.gui.helpers.forms), 7
set_thumbnail() (*src.gui.custom_widgets.CustomWidgetListItem*
method), 8
set_tooltips() (in module *src.gui.helpers.forms*), 7
SettingsError, 17
setupUi() (*src.ui.custom_widget_list_item.Ui_CustomWidgetListItem*
method), 12
setupUi() (*src.ui.frm_catalog_info.Ui_frm_catalog_info*
method), 13
setupUi() (*src.ui.frm_default_collections.Ui_frm_default_collections*
method), 13
setupUi() (*src.ui.frm_settings.Ui_frm_settings*
method), 14
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, 5
src.core.settings
module, 5
src.gui
module, 6
src.gui.custom_widgets
module, 7
src.gui.form_base
module, 8
src.gui.frm_catalog_info
module, 8
src.gui.frm_default_collections
module, 9
src.gui.frm_settings
module, 10
src.gui.helpers
module, 6
src.gui.helpers.forms
module, 6
src.kan_imagery_catalog
module, 20
src.kan_imagery_catalog_dock
module, 21
src.plugin_upload
module, 23
src.resources
module, 24
src.services
module, 10
src.services.sentinel_hub
module, 10
src.Services.up42
module, 11
src.services.utils
module, 12

src.ui
 module, 12
 src.ui.custom_widget_list_item
 module, 12
 src.ui.frm_catalog_info
 module, 13
 src.ui.frm_default_collections
 module, 13
 src.ui.frm_settings
 module, 14
 src.utils
 module, 14
 src.utils.constants
 module, 14
 src.utils.debug
 module, 16
 src.utils.encoders
 module, 16
 src.utils.exceptions
 module, 17
 src.utils.general
 module, 17
 src.utils.helpers
 module, 18
 src.utils.qgis_helper
 module, 18
 StyleVariables (class in src.utils.constants), 15
 SUCCESS (src.utils.constants.MessageType attribute), 16
 success_message() (in module src.utils.qgis_helper),
 19

T

to_dict() (src.utils.constants.CustomEnum class
 method), 15
 to_key_value_list()
 (src.utils.constants.CustomEnum class
 method), 15
 tr() (src.kan_imagery_catalog.KANImageryCatalog
 method), 20
 txt_default_back_days_text_changed()
 (src.gui.form_settings.FormSettings method),
 10
 txt_max_catalog_results_text_changed()
 (src.gui.form_settings.FormSettings method),
 10
 txt_max_catalog_results_text_changed()
 (src.kan_imagery_catalog_dock.KANImageryCatalogDock
 method), 22
 txt_max_features_results_text_changed()
 (src.gui.form_settings.FormSettings method),
 10

U

Ui_CustomWidgetListItem (class in

 src.ui.custom_widget_list_item), 12
 Ui_frm_catalog_info (class in
 src.ui.frm_catalog_info), 13
 Ui_frm_default_collections (class in
 src.ui.frm_default_collections), 13
 Ui_frm_settings (class in src.ui.frm_settings), 14
 unload() (src.kan_imagery_catalog.KANImageryCatalog
 method), 21
 update_cloud_coverage_label()
 (src.gui.form_settings.FormSettings method),
 10
 update_cloud_coverage_label()
 (src.kan_imagery_catalog_dock.KANImageryCatalogDock
 method), 22
 update_last_cleaning_date()
 (src.core.settings.PluginSettings method),
 6

V

VERBOSE (in module src.plugin_upload), 23
 view_details() (src.gui.custom_widgets.CustomWidgetListItem
 method), 8

W

WARNING (src.utils.constants.MessageType attribute), 16
 warning_message() (in module src.utils.qgis_helper),
 19