
friendlypics2

Release 0.0.2

Kevin S. Phillips

Aug 03, 2020

CONTENTS:

1	friendlypics2	3
2	Overview	11
3	Indices and tables	13
	Python Module Index	15
	Index	17

FRIENDLYPICS2

1.1 friendlypics2 package

1.1.1 Subpackages

friendlypics2.dialogs package

Submodules

friendlypics2.dialogs.about_dlg module

Logic for the application about box

```
class friendlypics2.dialogs.about_dlg.AboutDialog (parent, app_settings)  
    Bases: QDialog  
    Logic for managing about box  
    staticMetaObject = <PySide2.QtCore.QMetaObject object>
```

friendlypics2.dialogs.main_window module

GUI dialog defining behavior of main application window

```
class friendlypics2.dialogs.main_window.ImageItem (file_path)  
    Bases: object  
    Abstraction around an image that appears in the main list view  
    Parameters file_path (pathlib.Path) – path to the file managed by this instance  
    property file_name  
        name of the file managed by this object, excluding the path  
        Type str  
    property thumbnail  
        thumbnail representation of the image, lazy loaded when needed  
        Type QIcon  
class friendlypics2.dialogs.main_window.ImageModel (folder)  
    Bases: QAbstractListModel  
    Qt model that manages a list of images
```

Parameters `folder` (*pathlib.Path*) – path containing images to be presented to the user

canFetchMore (*parent*)

Is there more data to load from our model

Used for lazy loading image thumbnails only when needed

Parameters `parent` (*QModelIndex*) – reference to the parent item containing the images to load

Returns True if there is more data to load from our model, False if not

Return type `bool`

data (*index, role*)

retrieves data for a specific image given a specific role

Parameters

- **index** (*QModelIndex*) – index for the image to query
- **role** – *ItemDataRole* of the role within the view where the data will be used

Returns returns the name of the file when using the Display role, and returns the image thumbnail when using the Decoration role

Return type `str` or `QIcon`

fetchMore (*parent*)

Lazily loads more image data when needed

Triggered by the view class when the user scrolls below the current end of the viewable data

Parameters `parent` (*QModelIndex*) – reference to the parent item containing the images to load

property `max_count`

gets the total number of images managed by this model

Type `int`

rowCount (`_`)

`int`: number of images in the model, lazy loaded as needed

staticMetaObject = `<PySide2.QtCore.QMetaObject object>`

class `friendlypics2.dialogs.main_window.MainWindow`

Bases: `QMainWindow`

Main window interface

closeEvent (*event*)

event handler called when the application is about to close

Parameters `event` (*QCloseEvent*) – reference to the event object being raised

file_open_click (`()`)

callback for file-open menu

file_settings_click (`()`)

event handler for when the file->settings menu is clicked

help_about_click (`()`)

callback for the help-about menu

staticMetaObject = `<PySide2.QtCore.QMetaObject object>`

window_debug_click()
event handler for when the window->debug menu is clicked

friendlypics2.dialogs.settings_dlg module

Logic for the application settings UI

class friendlypics2.dialogs.settings_dlg.**AppSettingsModel** (*data*)
Bases: `QAbstractItemModel`

Interface for rendering application settings stored in a hierarchical format

Loosely based on example code found [here](#)

Parameters **data** (`AppSettings`) – reference to the application settings to be managed by this dialog

columnCount (*parent*)
retrieves the number of fields associated with a given item

Parameters **parent** (`QModelIndex`) – index of the item to retrieve information for contains a reference to the `SettingsItem` it manages

Returns number of fields associated with the given item

Return type `int`

data (*index, role*)
retrieves data for a specific setting given a specific role

Parameters

- **index** (`QModelIndex`) – index for the setting to query
- **role** – `ItemDataRole` of the role within the view where the data will be used

Returns data for the specified setting within the given context, or `None` if no data suitable for the given role can be found

Return type `str`

flags (*index*)
gets bit-field describing how a given setting should be rendered

Parameters **index** (`QModelIndex`) – index of the setting to be analysed

Returns bit-field encoded set of flags describing how the given item should be displayed

Return type `int`

headerData (*section, orientation, role*)
Gets column header text for the view

Parameters

- **section** (`int`) – id of the field / column to retrieve the header for
- **orientation** – `Orientation` of the text direction for the headers
- **role** – `ItemDataRole` of the role within the view where the data will be used

Returns header text for the given section / column

Return type `str`

index (*row, column, parent*)

Gets the index for a given element in the data managed by the model

Parameters

- **row** (*int*) – row of the data set to query within our context, this is the ID of the setting to query
- **column** (*int*) – column of the data set to query within our context, this is the ID of the field of a specific setting to query
- **parent** (*QModelIndex*) – index of the node / setting containing the children to query

Returns reference to the setting associated with the given context

Return type *QModelIndex*

parent (*index*)

Gets the index of the parent item for a given setting

Parameters **index** (*QModelIndex*) – index of the setting to query

Returns reference to the setting owning the given item, or an empty index if the given item has no parent

Return type *QModelIndex*

property root_item

gets the root node of our settings tree

Type *SettingsItem*

rowCount (*parent*)

Number of direct children owned by a given index

Parameters **parent** (*QModelIndex*) – reference to the setting to query

Returns number of children owned by this parent

Return type *int*

setData (*index, value, role*)

Modifies the value of a settings

Loosely based on example code found [here](#)

Parameters

- **index** (*QModelIndex*) – index of the app setting to modify
- **value** (*str*) – new data value for the setting
- **role** – *ItemDataRole* of the role within the view where the data is being edited

Returns True if the data was successfully modified, False if not

Return type *bool*

staticMetaObject = `<PySide2.QtCore.QMetaObject object>`

class friendlypics2.dialogs.settings_dlg.**SettingsDialog** (*parent, settings*)

Bases: *QDialog*

Logic for managing application settings dialog

Parameters

- **parent** (*QWidget*) – Parent widget / dialog that owns the settings dialog

- **settings** (*AppSettings*) – reference to the application settings to be managed by this dialog

staticMetaObject = <PySide2.QtCore.QMetaObject object>

class friendlypics2.dialogs.settings_dlg.**SettingsItem** (*data, parent=None*)
Bases: *object*

Interface to a single application setting or group of application settings

Loosely based on example code found [here](#)

Parameters

- **data** (*tuple*) – fields associated with this setting, providing data for each column in the view
- **parent** (*SettingsItem*) – parent setting that owns this one. Set to None to indicate this item is a root setting with no parent

appendChild (*item*)

Adds a new child element to this setting in the tree

Parameters **item** (*SettingsItem*) – child item related to this one

child (*row*)

Gets a specific child item from this parent

Parameters **row** (*int*) – index of the child item to retrieve

Returns child item with the given index

Return type *SettingsItem*

childCount ()

int: number of children contained within this item

property child_items

list of children owned by this item

Type *list (SettingsItem)*

columnCount ()

int: number of columns or fields associated with this item

data (*column*)

str: data associated with this item for the given field / column. May be None if no data for the specified column exists.

parent ()

SettingItem: parent object that owns this item, or None if this is a root item

row ()

int: gets the relative offset of this item with respect to its siblings

set_data (*value*)

Modifies the data value for this setting

Parameters **value** (*str*) – new value for this setting

Module contents

this sub package contains classes for managing UI dialogs

friendlypics2.misc package

Submodules

friendlypics2.misc.app_helpers module

Helper functions to perform various application level operations

`friendlypics2.misc.app_helpers.is_mac_app_bundle()`
bool: Checks to see if the application is running under a MacOS app bundle

`friendlypics2.misc.app_helpers.is_pyinstaller_bundle()`
bool: Checks to see if the application is running under a pyinstaller bundle

friendlypics2.misc.app_settings module

Interface for persisting application settings

class `friendlypics2.misc.app_settings.AppSettings`

Bases: `object`

Interface for accessing and persisting application settings

property data

reference to the raw settings data managed by this object. Used exclusively for the settings dialog

Type `dict`

property file_version

gets the schema version for the config file

Type `str`

property path

Location of the config file managed by this class

property pinterest_user

user to authenticate with to Pinterest

Type `str`

save()

Saves the current contents of the app settings for later reference

friendlypics2.misc.gui_helpers module

helper methods for use by GUI objects and dialogs

class friendlypics2.misc.gui_helpers.**GuiLogger** (*widget*)

Bases: `logging.Handler`

Custom Python log handler that redirects output to a Qt widget

Parameters **widget** (*QWidget*) – text widget to redirect log output to

emit (*record*)

Displays a log message to the GUI

write (*_*)

Disable disk-based file access

friendlypics2.misc.gui_helpers.**generate_screen_id** (*screen*)

generates a unique identifier for a screen

Parameters **screen** (*QScreen*) – screen to be analysed

Returns unique ID describing the given screen

Return type `str`

friendlypics2.misc.gui_helpers.**load_ui** (*file, parent*)

Loads a UI interface file from disk

Parameters

- **file** (*str*) – path to the UI file to load. Must be relative to the data/ui folder.
- **parent** (*object*) – reference to the Qt dialog object associated with the UI file

friendlypics2.misc.gui_helpers.**settings_group_context** (*settings, group_name*)

Context manager that starts loading settings from a specific sub-group and restores the original settings group once the context goes out of scope

Code within this context will load settings from the specified sub-group

Parameters

- **settings** (*QSettings*) – settings for the running app
- **group_name** (*str*) – name of the settings group to invoke

Module contents

sub package containing helper scripts used throughout the app

friendlypics2.scripts package

Submodules

friendlypics2.scripts.fpics2 module

Command line interface to the project

`friendlypics2.scripts.fpics2.main()`
primary entry point method

Module contents

Command line interfaces associated with the project

1.1.2 Submodules

friendlypics2.main module

Main entry point module for the application

`friendlypics2.main.configure_logging()`
Configure the Python logging system for the app

Only configures the console based logger

`friendlypics2.main.run(args)`
Main entrypoint function

Parameters `args` (*list*) – command line arguments to be passed to the application

Returns return code to report back to the shell with

Return type `int`

friendlypics2.version module

1.1.3 Module contents

main package for the project

OVERVIEW

User friendly image organization tool.

Comments, suggestions and bugs may be reported to the project [maintainer](#)

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

f

- friendlypics2, 10
- friendlypics2.dialogs, 8
- friendlypics2.dialogs.about_dlg, 3
- friendlypics2.dialogs.main_window, 3
- friendlypics2.dialogs.settings_dlg, 5
- friendlypics2.main, 10
- friendlypics2.misc, 9
- friendlypics2.misc.app_helpers, 8
- friendlypics2.misc.app_settings, 8
- friendlypics2.misc.gui_helpers, 9
- friendlypics2.scripts, 10
- friendlypics2.scripts.fpics2, 10
- friendlypics2.version, 10

INDEX

A

AboutDialog (class in *friendlypics2.dialogs.about_dlg*), 3
appendChild() (*friendlypics2.dialogs.settings_dlg.SettingsItem* method), 7
AppSettings (class in *friendlypics2.misc.app_settings*), 8
AppSettingsModel (class in *friendlypics2.dialogs.settings_dlg*), 5

C

canFetchMore() (*friendlypics2.dialogs.main_window.ImageModel* method), 4
child() (*friendlypics2.dialogs.settings_dlg.SettingsItem* method), 7
child_items() (*friendlypics2.dialogs.settings_dlg.SettingsItem* property), 7
childCount() (*friendlypics2.dialogs.settings_dlg.SettingsItem* method), 7
closeEvent() (*friendlypics2.dialogs.main_window.MainWindow* method), 4
columnCount() (*friendlypics2.dialogs.settings_dlg.AppSettingsModel* method), 5
columnCount() (*friendlypics2.dialogs.settings_dlg.SettingsItem* method), 7
configure_logging() (in module *friendlypics2.main*), 10

D

data() (*friendlypics2.dialogs.main_window.ImageModel* method), 4
data() (*friendlypics2.dialogs.settings_dlg.AppSettingsModel* method), 5
data() (*friendlypics2.dialogs.settings_dlg.SettingsItem* method), 7

data() (*friendlypics2.misc.app_settings.AppSettings* property), 8

E

emit() (*friendlypics2.misc.gui_helpers.GuiLogger* method), 9

F

fetchMore() (*friendlypics2.dialogs.main_window.ImageModel* method), 4
file_name() (*friendlypics2.dialogs.main_window.ImageItem* property), 3
file_open_click() (*friendlypics2.dialogs.main_window.MainWindow* method), 4
file_settings_click() (*friendlypics2.dialogs.main_window.MainWindow* method), 4
file_version() (*friendlypics2.misc.app_settings.AppSettings* property), 8
flags() (*friendlypics2.dialogs.settings_dlg.AppSettingsModel* method), 5
friendlypics2 module, 10
friendlypics2.dialogs module, 8
friendlypics2.dialogs.about_dlg module, 3
friendlypics2.dialogs.main_window module, 3
friendlypics2.dialogs.settings_dlg module, 5
friendlypics2.main module, 10
friendlypics2.misc module, 9
friendlypics2.misc.app_helpers module, 8
friendlypics2.misc.app_settings

module, 8
 friendlypics2.misc.gui_helpers
 module, 9
 friendlypics2.scripts
 module, 10
 friendlypics2.scripts.fpics2
 module, 10
 friendlypics2.version
 module, 10

G

generate_screen_id() (in module friendlypics2.misc.gui_helpers), 9
 GuiLogger (class in friendlypics2.misc.gui_helpers), 9

H

headerData() (friendlypics2.dialogs.settings_dlg.AppSettingsModel method), 5
 help_about_click() (friendlypics2.dialogs.main_window.MainWindow method), 4

I

ImageItem (class in friendlypics2.dialogs.main_window), 3
 ImageModel (class in friendlypics2.dialogs.main_window), 3
 index() (friendlypics2.dialogs.settings_dlg.AppSettingsModel method), 5
 is_mac_app_bundle() (in module friendlypics2.misc.app_helpers), 8
 is_pyinstaller_bundle() (in module friendlypics2.misc.app_helpers), 8

L

load_ui() (in module friendlypics2.misc.gui_helpers), 9

M

main() (in module friendlypics2.scripts.fpics2), 10
 MainWindow (class in friendlypics2.dialogs.main_window), 4
 max_count() (friendlypics2.dialogs.main_window.ImageModel property), 4
 module
 friendlypics2, 10
 friendlypics2.dialogs, 8
 friendlypics2.dialogs.about_dlg, 3
 friendlypics2.dialogs.main_window, 3
 friendlypics2.dialogs.settings_dlg, 5

friendlypics2.main, 10
 friendlypics2.misc, 9
 friendlypics2.misc.app_helpers, 8
 friendlypics2.misc.app_settings, 8
 friendlypics2.misc.gui_helpers, 9
 friendlypics2.scripts, 10
 friendlypics2.scripts.fpics2, 10
 friendlypics2.version, 10

P

parent() (friendlypics2.dialogs.settings_dlg.AppSettingsModel method), 6
 parent() (friendlypics2.dialogs.settings_dlg.SettingsItem method), 7
 path() (friendlypics2.misc.app_settings.AppSettings property), 8
 pinterest_user() (friendlypics2.misc.app_settings.AppSettings property), 8

R

root_item() (friendlypics2.dialogs.settings_dlg.AppSettingsModel property), 6
 row() (friendlypics2.dialogs.settings_dlg.SettingsItem method), 7
 rowCount() (friendlypics2.dialogs.main_window.ImageModel method), 4
 rowCount() (friendlypics2.dialogs.settings_dlg.AppSettingsModel method), 6
 run() (in module friendlypics2.main), 10

S

save() (friendlypics2.misc.app_settings.AppSettings method), 8
 set_data() (friendlypics2.dialogs.settings_dlg.SettingsItem method), 7
 setData() (friendlypics2.dialogs.settings_dlg.AppSettingsModel method), 6
 settings_group_context() (in module friendlypics2.misc.gui_helpers), 9
 SettingsDialog (class in friendlypics2.dialogs.settings_dlg), 6
 SettingsItem (class in friendlypics2.dialogs.settings_dlg), 7
 staticMetaObject (friendlypics2.dialogs.about_dlg.AboutDialog attribute), 3

staticMetaObject (friendlypics2.dialogs.main_window.ImageModel attribute), 4

staticMetaObject (friendlypics2.dialogs.main_window.MainWindow attribute), 4

staticMetaObject (friendlypics2.dialogs.settings_dlg.AppSettingsModel attribute), 6

staticMetaObject (friendlypics2.dialogs.settings_dlg.SettingsDialog attribute), 7

T

thumbnail() (friendlypics2.dialogs.main_window.ImageItem property), 3

W

window_debug_click() (friendlypics2.dialogs.main_window.MainWindow method), 4

write() (friendlypics2.misc.gui_helpers.GuiLogger method), 9