
File Manager

Release 0.1.0

KivyMD Team

Nov 09, 2020

CONTENTS

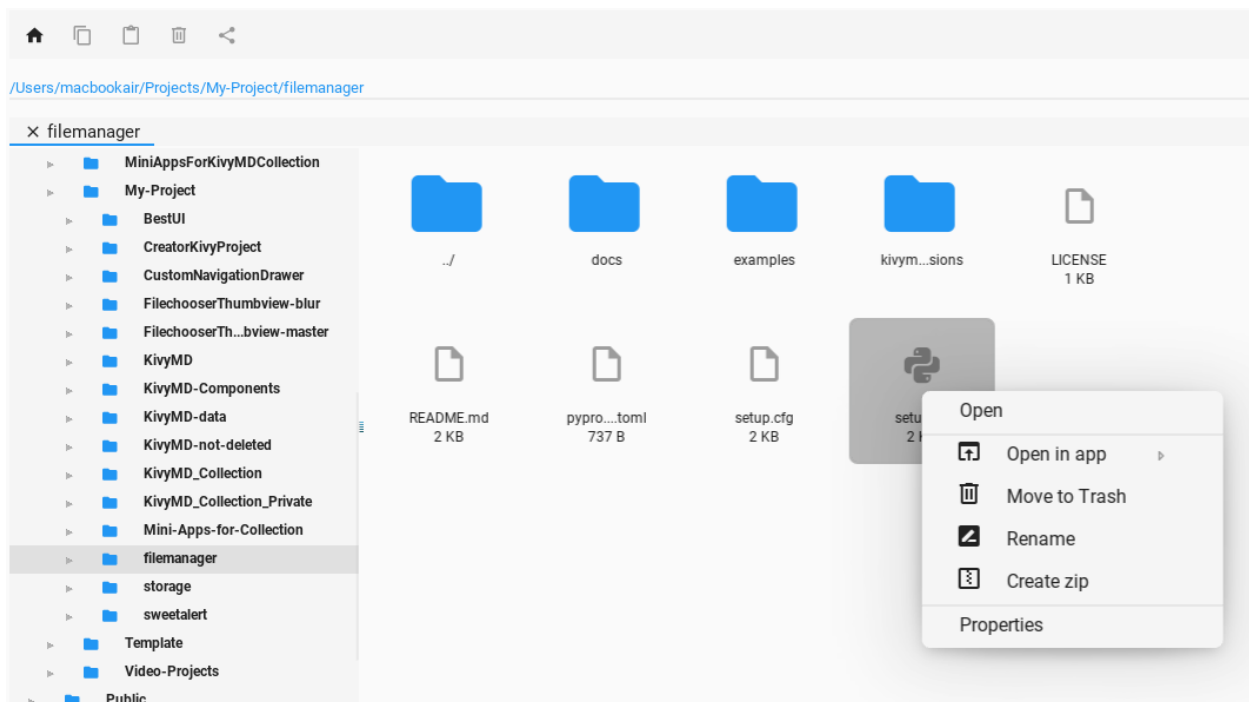
1	Contents	1
1.1	Components	1
1.2	About	9
1.3	kivymd_extensions.filemanager.libs.plugins	10
2	Indices and tables	13
	Python Module Index	15
	Index	17

CONTENTS

1.1 Components

1.1.1 File Manager

File manager for the desktop.



Warning: We have not tested the file manager for Windows OS. Therefore, we do not guarantee the absence of bugs in this OS. Contact [technical support](#) if you have any problems using the file manager.

Usage

```
from kivymd.app import MDApp

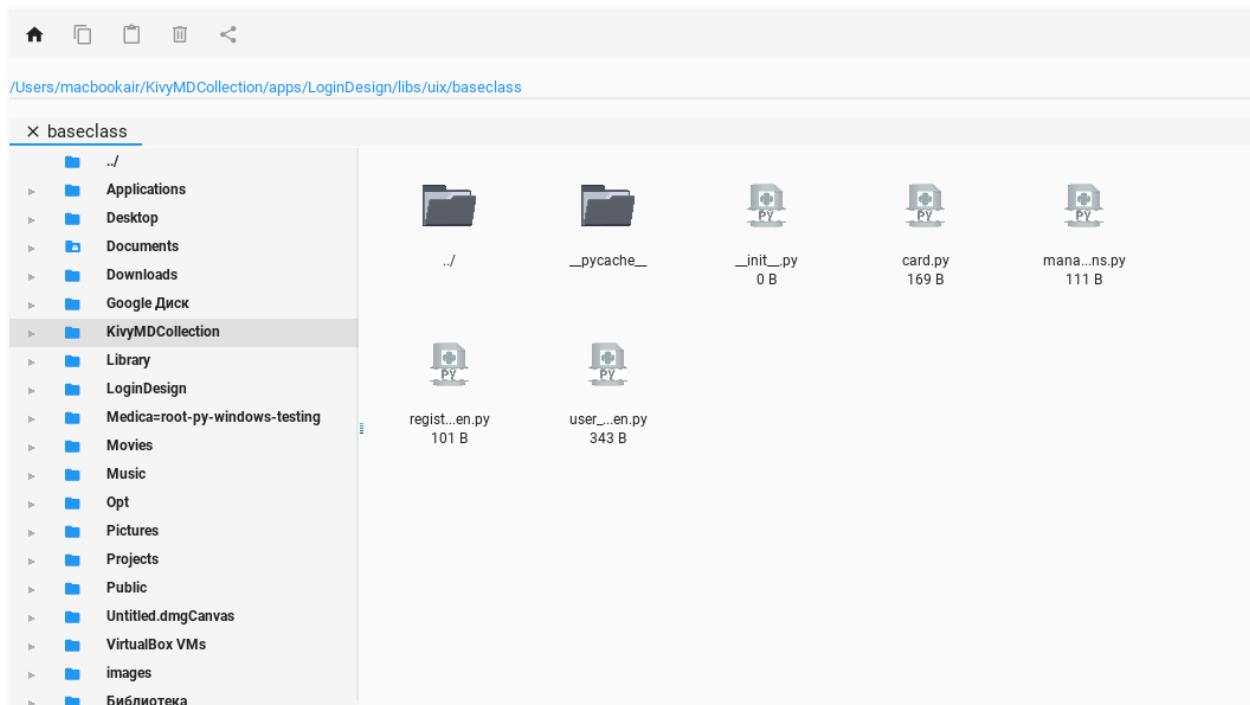
from kivymd_extensions.filemanager import FileManager

class Example(MDApp):
    def on_start(self):
        FileManager().open()
```

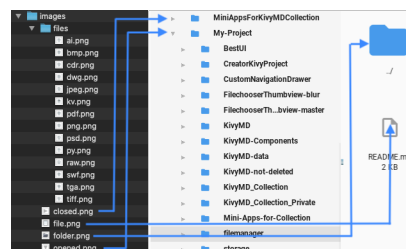
Customization

If you want to use custom icons in the file manager, you need to specify the path to the icon theme:

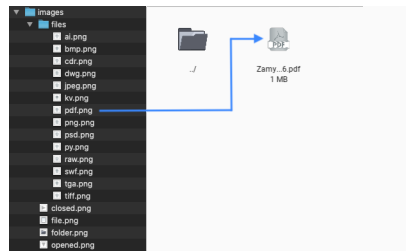
```
FileManager(path_to_skin="path/to/images").open()
```



The resource directory structure for the file manager theme looks like this:

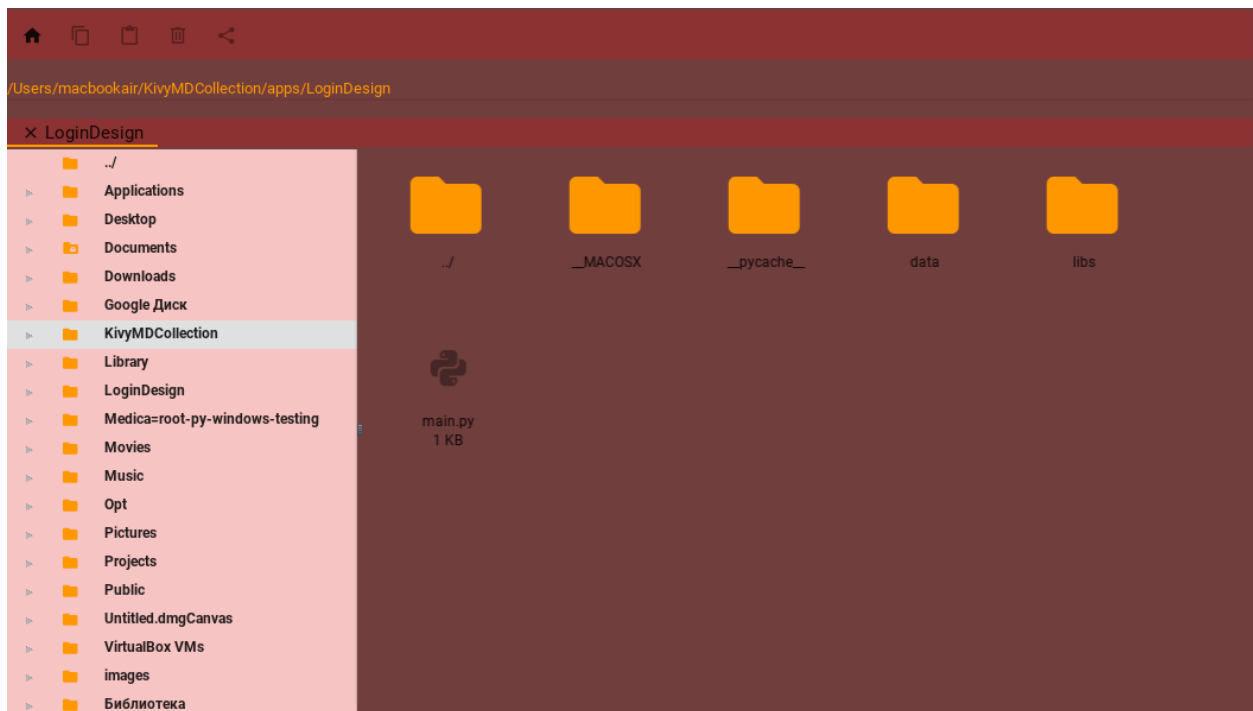


The files directory contains images whose names correspond to the file types:



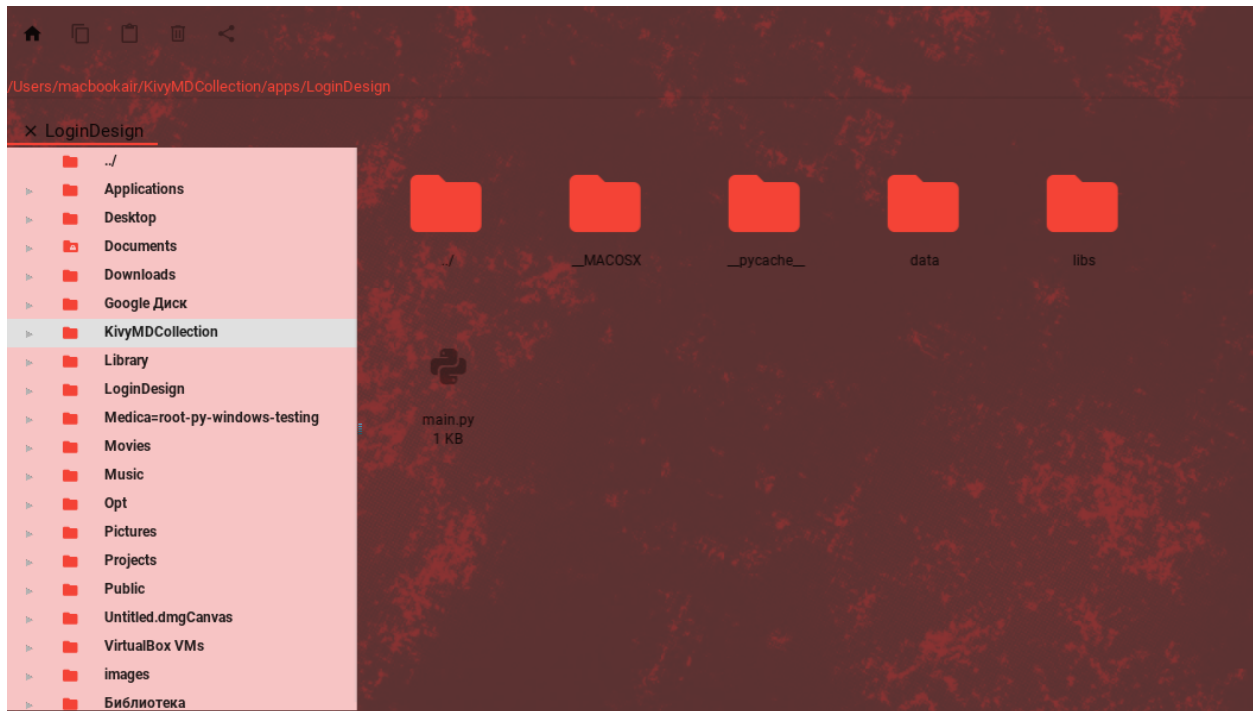
Color

```
FileManager(bg_color=(1, 0, 0, 0.2)).open()
```



Texture

```
FileManager(bg_texture="path/to/texture.png").open()
```



Warning: If you are using `bg_texture` parameter then `bg_color` parameter will be ignored.

Events

on_tab_switch Called when switching tabs.

on_tap_file Called when the file is clicked.

on_tap_dir Called when the folder is clicked.

on_context_menu Called at the end of any actions of the context menu, be it copying, archiving files and other actions.

```
from kivy.md.app import MDApp

from kivy.md.extensions.filemanager import FileManager

class Example(MDApp):
    def on_context_menu(self, instance_file_manager, name_context_plugin):
        print("Event 'on_context_menu'", instance_file_manager, name_context_plugin)

    def on_tap_file(self, instance_file_manager, path):
        print("Event 'on_tap_file'", instance_file_manager, path)

    def on_tap_dir(self, instance_file_manager, path):
        print("Event 'on_tap_dir'", instance_file_manager, path)

    def on_tab_switch(
        self,
        instance_file_manager,
```

(continues on next page)

(continued from previous page)

```

        instance_tabs,
        instance_tab,
        instance_tab_label,
        tab_text,
    ):
        print(
            "Event 'on_tab_switch'",
            instance_file_manager,
            instance_tabs,
            instance_tab,
            instance_tab_label,
            tab_text,
        )

    def build(self):
        self.theme_cls.primary_palette = "Red"

    def on_start(self):
        manager = FileManager()
        manager.bind(
            on_tap_file=self.on_tap_file,
            on_tap_dir=self.on_tap_dir,
            on_tab_switch=self.on_tab_switch,
            on_context_menu=self.on_context_menu,
        )
        manager.open()

```

```
Example().run()
```

API - `kivymd_extensions.filemanager.filemanager`

class `kivymd_extensions.filemanager.filemanager.RightContentCls` (***kwargs*)
 Same as `IRightBody`, but allows the widget to receive touch events instead of triggering the `Listitem`'s ripple effect

class `kivymd_extensions.filemanager.filemanager.FileManagerTab` (***kwargs*)
 Class implementing content for a tab.

manager
FileManager object.

path
 Path to root directory for instance *manager*.
path is an *StringProperty* and defaults to ''.

class `kivymd_extensions.filemanager.filemanager.FileManagerTextFieldSearch` (***kw*)
 The class implements a text field for searching files.

See `rule FileManagerTextFieldSearch` in `kivymd_extensions/filemanager/filemanager.kv` file.

hint_text
 See `:attr:~kivy.uix.textinput.TextInput.hint_text`

background_normal
 See `:attr:~kivy.uix.textinput.TextInput.background_normal`

background_active

See :attr:`~kivy.uix.textinput.TextInput.background_active`

type

Search files by name or extension. Available options are 'name', 'ext'.

icon is an `OptionProperty` and defaults to 'name'.

manager

See `FileManager` object.

on_enter (*self, instance, value*)

Called when the user hits 'Enter' in text field.

create_menu (*self, interval*)**set_type_search** (*self, instance_context_menu, instance_context_item*)

class `kivymd_extensions.filemanager.filemanager.FileManagerFilesSearchResultsDialog` (**kwargs)

The class implements displaying a list with the results of file search.

data_results**manager**

See `FileManager` object.

on_open (*self*)**on_dismiss** (*self*)**go_to_directory_found_file** (*self, path_to_found_file*)

class `kivymd_extensions.filemanager.filemanager.FileManagerSettingsLeftWidgetItem` (**kwargs)

Pseudo-interface for widgets that go in the left container for ListItems that support it.

Implements nothing and requires no implementation, for annotation only.

class `kivymd_extensions.filemanager.filemanager.FileManagerSettingsColorItem` (**kwargs)

Overrides `add_widget` in a `Listitem` to include support for `I*Body` widgets when the appropriate containers are present.

color

class `kivymd_extensions.filemanager.filemanager.FileManagerSettings` (**kwargs)

Box layout class. See module documentation for more information.

class `kivymd_extensions.filemanager.filemanager.FileManagerTextFieldSearchDialog` (**kwargs)

Base class for context menu windows.

manager

class `kivymd_extensions.filemanager.filemanager.ContextMenuBehavior` (**kwargs)

Events

on_enter Fired when mouse enter the bbox of the widget.

on_leave Fired when the mouse exit the widget.

on_enter (*self*)

Fired when mouse enter the bbox of the widget.

on_leave (*self*)

Fired when the mouse exit the widget.

```

class kivymd_extensions.filemanager.filemanager.ContextMenuItemMore(**kwargs)
    Context menu item.

    icon
        Icon of item.

        icon is an StringProperty and defaults to ''.

class kivymd_extensions.filemanager.filemanager.ContextMenuItem(**kwargs)
    Context menu item.

    icon
        Icon of item.

        icon is an StringProperty and defaults to ''.

class kivymd_extensions.filemanager.filemanager.FileManager(**kwargs)

    Events

    on_tab_switch Called when switching tabs.
    on_tap_file Called when the file is clicked.
    on_tap_dir Called when the folder is clicked.
    on_context_menu Called at the end of any actions of the context menu, be it copying, archiving files and other
        actions.
    on_open_plugin_dialog Description.
    on_dismiss_plugin_dialog Description.

    path
        The path to the directory in which the file manager will open by default.

        path is an StringProperty and defaults to os.getcwd().

    context_menu_open
        Open or close context menu.

        context_menu_open is an BooleanProperty and defaults to False.

    path_to_skin
        Path to directory with custom images.

        path_to_skin is an StringProperty and defaults to ''.

    bg_color
        Background color of file manager in the format (r, g, b, a).

        bg_color is a ListProperty and defaults to [].

    bg_texture
        Background texture of file manager.

        bg_texture is a StringProperty and defaults to ''.

    auto_dismiss = False

    add_color_panel(self)

    apply_palette(self)
        Applies the color theme from the settings file when opening the file manager window.

```

apply_properties_on_show_settings (*self*)

Applies the settings from the “settings.ini” file to the checkboxes of items on the settings panel.

show_taskbar (*self*)

hide_taskbar (*self*)

show_settings (*self*, *instance_button*)

Opens the settings panel.

hide_settings (*self*, *theme_panel*)

Closes the settings panel.

set_path (*self*, *path*)

Sets the directory path for the *FileChooserIconLayout* class.

get_formatting_text_for_tab (*self*, *text*)

add_tab (*self*, *path_to_file*)

Adds a new tab in the file manager.

Parameters *path_to_file* – The path to the file or folder that was right-clicked.

remove_tab (*self*, *instance_tabs*, *instance_tab_label*, *instance_tab*, *instance_tab_bar*, *instance_carousel*)

Removes an open tab in the file manager.

Parameters

- **instance_tabs** – <kivymd.uix.tab.MDTabs object>
- **instance_tab_label** – <kivymd.uix.tab.MDTabsLabel object>
- **instance_tab** – <__main__.Tab object>
- **instance_tab_bar** – <kivymd.uix.tab.MDTabsBar object>
- **instance_carousel** – <kivymd.uix.tab.MDTabsCarousel object>

create_header_menu (*self*)

Creates a menu in the file manager header.

update_background_search_field (*self*, *instance*, *value*)

open_context_menu (*self*, *entry_object*, *type_chooser*)

Opens a context menu on right-clicking on a file or folder.

tap_on_file_dir (*self*, **touch*)

Called when the file/dir is clicked.

call_context_menu_plugin (*self*, *name_plugin*, *entry_object*)

tap_to_context_menu_item (*self*, *instance_menu*, *instance_item*, *entry_object*)

dismiss_context_menu (*self*)

context_menu_dismiss (*self*, **args*)

Called when closing the context menu.

get_menu_right_click (*self*, *entry_object*, *type_chooser*)

Returns a list of dictionaries for creating context menu items.

get_icon_file (*self*, *path_to_file*)

Method that returns the icon path for the file.

update_files (*self*, *instance_pludin_dialog*, *path*)

is_dir (*self*, *directory*, *filename*)

on_tap_file (*self*, **args*)
Called when the file is clicked.

on_tap_dir (*self*, **args*)
Called when the folder is clicked.

on_tab_switch (*self*, **args*)
Called when switching tab.

on_open_plugin_dialog (*self*, **args*)

on_dismiss_plugin_dialog (*self*, **args*)

on_context_menu (*self*, **args*)
Called at the end of any actions of the context menu, be it copying, archiving files and other actions.

on_open (*self*)
Called when the ModalView is opened.

1.2 About

1.2.1 License

Refer to [LICENSE](#).

MIT License

Copyright (c) 2020 KivyMD Team and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3 kivymd_extensions.filemanager.libs.plugins

1.3.1 API - kivymd_extensions.filemanager.libs.plugins

```
class kivymd_extensions.filemanager.libs.plugins.PluginBaseDialog(**kwargs)
    Base class for context menu windows.

    bg_color
    bg_background
    instance_context_menu
    instance_manager
```

1.3.2 Submodules

kivymd_extensions.filemanager.libs.plugins.contextmenu

API - kivymd_extensions.filemanager.libs.plugins.contextmenu

```
class kivymd_extensions.filemanager.libs.plugins.contextmenu.ContextMenuPlugin(instance_manager,
en-
try_object=None)

    dismiss_plugin_dialog(self, instance_plugin_dialog, name_plugin)
    main(self, name_plugin)
```

Submodules

kivymd_extensions.filemanager.libs.plugins.contextmenu.properties

API - kivymd_extensions.filemanager.libs.plugins.contextmenu.properties

```
class kivymd_extensions.filemanager.libs.plugins.contextmenu.properties.DialogProperties(**kwargs)
    Base class for context menu windows.

    get_first_created(self)
    get_last_opened(self)
    get_last_changed(self)
    get_file_size(self)
    get_access_string(self)
    set_access(self, interval)
    on_instance_context_menu(self, instance, value)
    on_pre_open(self)
```

kivymd_extensions.filemanager.libs.plugins.contextmenu.remove**API - kivymd_extensions.filemanager.libs.plugins.contextmenu.remove**

class kivymd_extensions.filemanager.libs.plugins.contextmenu.remove.**DialogMoveToTrash** (**kwargs)
Base class for context menu windows.

kivymd_extensions.filemanager.libs.plugins.contextmenu.rename**API - kivymd_extensions.filemanager.libs.plugins.contextmenu.rename**

class kivymd_extensions.filemanager.libs.plugins.contextmenu.rename.**DialogRename** (**kwargs)
Base class for context menu windows.

set_focus (self, interval)

rename_file (self, new_file_name)

on_open (self)

kivymd_extensions.filemanager.libs.plugins.contextmenu.ziparchive**API - kivymd_extensions.filemanager.libs.plugins.contextmenu.ziparchive**

class kivymd_extensions.filemanager.libs.plugins.contextmenu.ziparchive.**DialogZipArchive** (**kwargs)
Base class for context menu windows.

set_progress_value (self, count_files, count)

set_name_packed_file (self, file_name)

on_open (self)

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

k

- kivymd_extensions.filemanager.filemanager,
1
- kivymd_extensions.filemanager.libs.plugins,
10
- kivymd_extensions.filemanager.libs.plugins.contextmenu,
10
- kivymd_extensions.filemanager.libs.plugins.contextmenu.properties,
10
- kivymd_extensions.filemanager.libs.plugins.contextmenu.remove,
11
- kivymd_extensions.filemanager.libs.plugins.contextmenu.rename,
11
- kivymd_extensions.filemanager.libs.plugins.contextmenu.ziparchive,
11

INDEX

A

add_color_panel() (kivymd_extensions.filemanager.filemanager.FileManager method), 7

add_tab() (kivymd_extensions.filemanager.filemanager.FileManager method), 8

apply_palette() (kivymd_extensions.filemanager.filemanager.FileManager method), 7

apply_properties_on_show_settings() (kivymd_extensions.filemanager.filemanager.FileManager method), 7

auto_dismiss(kivymd_extensions.filemanager.filemanager.FileManager attribute), 7

ContextMenuBehavior (class in kivymd_extensions.filemanager.filemanager), 6

ContextMenuItem (class in kivymd_extensions.filemanager.filemanager), 7

ContextMenuItemMore (class in kivymd_extensions.filemanager.filemanager), 6

ContextMenuPlugin (class in kivymd_extensions.filemanager.libs.plugins.contextmenu), 10

create_header_menu() (kivymd_extensions.filemanager.filemanager.FileManager method), 8

create_menu() (kivymd_extensions.filemanager.filemanager.FileManager method), 6

B

background_active (kivymd_extensions.filemanager.filemanager.FileManager attribute), 5

background_normal (kivymd_extensions.filemanager.filemanager.FileManager attribute), 5

bg_background(kivymd_extensions.filemanager.libs.plugins.PluginBaseDialog attribute), 10

bg_color(kivymd_extensions.filemanager.filemanager.FileManager attribute), 7

bg_color(kivymd_extensions.filemanager.libs.plugins.PluginBaseDialog attribute), 10

bg_texture(kivymd_extensions.filemanager.filemanager.FileManager attribute), 7

data_results(kivymd_extensions.filemanager.filemanager.FileManager attribute), 6

DialogMoveToTrash (class in kivymd_extensions.filemanager.libs.plugins.contextmenu.remove), 11

DialogProperties (class in kivymd_extensions.filemanager.libs.plugins.contextmenu.properties), 10

DialogRename (class in kivymd_extensions.filemanager.libs.plugins.contextmenu.rename), 11

DialogZipArchive (class in kivymd_extensions.filemanager.libs.plugins.contextmenu.ziparchive), 11

C

call_context_menu_plugin() (kivymd_extensions.filemanager.filemanager.FileManager method), 8

color(kivymd_extensions.filemanager.filemanager.FileManager attribute), 6

context_menu_dismiss() (kivymd_extensions.filemanager.filemanager.FileManager method), 8

context_menu_open (kivymd_extensions.filemanager.filemanager.FileManager attribute), 7

dismiss_context_menu() (kivymd_extensions.filemanager.filemanager.FileManager method), 8

dismiss_plugin_dialog() (kivymd_extensions.filemanager.libs.plugins.contextmenu.ContextMenuItem method), 10

FileManager (class in kivymd_extensions.filemanager.filemanager), 7

FileManagerFilesSearchResultsDialog (class in kivymd_extensions.filemanager.filemanager), 7

F

6
 FileManagerSettings (class in
 kivymd_extensions.filemanager.filemanager), 6
 FileManagerSettingsColorItem (class in
 kivymd_extensions.filemanager.filemanager), 6
 FileManagerSettingsLeftWidgetItem (class
 in kivymd_extensions.filemanager.filemanager),
 6
 FileManagerTab (class in
 kivymd_extensions.filemanager.filemanager), 5
 FileManagerTextFieldSearch (class in
 kivymd_extensions.filemanager.filemanager), 5
 FileManagerTextFieldSearchDialog (class in
 kivymd_extensions.filemanager.filemanager), 6

G
 get_access_string()
 (kivymd_extensions.filemanager.libs.plugins.contextmenu.properties.DialogProperties
 method), 10
 get_file_size() (kivymd_extensions.filemanager.libs.plugins.contextmenu.properties.DialogProperties
 method), 10
 get_first_created()
 (kivymd_extensions.filemanager.libs.plugins.contextmenu.properties.DialogProperties
 method), 10
 get_formatting_text_for_tab()
 (kivymd_extensions.filemanager.filemanager.FileManager
 method), 8
 get_icon_file() (kivymd_extensions.filemanager.filemanager.FileManager
 method), 8
 get_last_changed()
 (kivymd_extensions.filemanager.libs.plugins.contextmenu.properties.DialogProperties
 method), 10
 get_last_opened()
 (kivymd_extensions.filemanager.libs.plugins.contextmenu.properties.DialogProperties
 method), 10
 get_menu_right_click()
 (kivymd_extensions.filemanager.filemanager.FileManager
 method), 8
 go_to_directory_found_file()
 (kivymd_extensions.filemanager.filemanager.FileManagerFilesSearchResultsDialog
 method), 6

H
 hide_settings() (kivymd_extensions.filemanager.filemanager.FileManager
 method), 8
 hide_taskbar() (kivymd_extensions.filemanager.filemanager.FileManager
 method), 8
 hint_text (kivymd_extensions.filemanager.filemanager.FileManagerTextFieldSearch
 attribute), 5

I
 icon (kivymd_extensions.filemanager.filemanager.ContextMenuItemMore
 attribute), 7
 instance_context_menu
 (kivymd_extensions.filemanager.libs.plugins.PluginBaseDialog
 attribute), 10
 instance_manager (kivymd_extensions.filemanager.libs.plugins.PluginBaseDialog
 attribute), 10
 is_dir() (kivymd_extensions.filemanager.filemanager.FileManager
 method), 9

K
 kivymd_extensions.filemanager.filemanager
 module, 1
 kivymd_extensions.filemanager.libs.plugins
 module, 10
 kivymd_extensions.filemanager.libs.plugins.contextmenu
 module, 10
 kivymd_extensions.filemanager.libs.plugins.contextmenu.properties.DialogProperties
 module, 10
 kivymd_extensions.filemanager.libs.plugins.contextmenu.properties.DialogProperties
 module, 11
 kivymd_extensions.filemanager.libs.plugins.contextmenu.properties.DialogProperties
 module, 11

M
 manager (kivymd_extensions.filemanager.filemanager.FileManagerFilesSearchResultsDialog
 method), 10
 manager (kivymd_extensions.filemanager.filemanager.FileManagerFilesSearchResultsDialog
 attribute), 10
 manager (kivymd_extensions.filemanager.filemanager.FileManagerFilesSearchResultsDialog
 attribute), 5
 manager (kivymd_extensions.filemanager.filemanager.FileManagerFilesSearchResultsDialog
 attribute), 6
 manager (kivymd_extensions.filemanager.filemanager.FileManagerFilesSearchResultsDialog
 attribute), 6
 module
 kivymd_extensions.filemanager.filemanager,
 kivymd_extensions.filemanager.libs.plugins,
 10
 kivymd_extensions.filemanager.libs.plugins.contextmenu,
 10
 kivymd_extensions.filemanager.libs.plugins.contextmenu.properties.DialogProperties,
 10
 kivymd_extensions.filemanager.libs.plugins.contextmenu.properties.DialogProperties,
 11
 kivymd_extensions.filemanager.libs.plugins.contextmenu.properties.DialogProperties,
 11

O

`on_context_menu()` (method), 8
`on_context_menu()` (kivymd_extensions.filemanager.filemanager.FileManager method), 11
`on_dismiss()` (kivymd_extensions.filemanager.filemanager.FileManager method), 9
`on_dismiss()` (kivymd_extensions.filemanager.filemanager.FileManager method), 6
`on_dismiss_plugin_dialog()` (kivymd_extensions.filemanager.filemanager.FileManager method), 9
`on_enter()` (kivymd_extensions.filemanager.filemanager.FileManager method), 6
`on_enter()` (kivymd_extensions.filemanager.filemanager.FileManager method), 6
`on_instance_context_menu()` (kivymd_extensions.filemanager.libs.plugins.contextmenu.ziparchive.ZipArchive method), 11
`on_leave()` (kivymd_extensions.filemanager.filemanager.FileManager method), 6
`on_open()` (kivymd_extensions.filemanager.filemanager.FileManager method), 9
`on_open()` (kivymd_extensions.filemanager.filemanager.FileManager method), 6
`on_open()` (kivymd_extensions.filemanager.libs.plugins.contextmenu.ziparchive.ZipArchive method), 11
`on_open()` (kivymd_extensions.filemanager.libs.plugins.contextmenu.ziparchive.ZipArchive method), 11
`on_open_plugin_dialog()` (kivymd_extensions.filemanager.filemanager.FileManager method), 9
`on_pre_open()` (kivymd_extensions.filemanager.libs.plugins.contextmenu.ziparchive.ZipArchive method), 10
`on_tab_switch()` (kivymd_extensions.filemanager.filemanager.FileManager method), 9
`on_tap_dir()` (kivymd_extensions.filemanager.filemanager.FileManager method), 9
`on_tap_file()` (kivymd_extensions.filemanager.filemanager.FileManager method), 9
`open_context_menu()` (kivymd_extensions.filemanager.filemanager.FileManager method), 8

P

`path()` (kivymd_extensions.filemanager.filemanager.FileManager attribute), 7
`path()` (kivymd_extensions.filemanager.filemanager.FileManagerTab attribute), 5
`path_to_skin()` (kivymd_extensions.filemanager.filemanager.FileManager attribute), 7
`PluginBaseDialog` (class in kivymd_extensions.filemanager.libs.plugins), 10

R

`remove_tab()` (kivymd_extensions.filemanager.filemanager.FileManager