API Documentation

API Documentation

January 14, 2015

Contents

1 Package isySUR

1.1 Modules

- **gui**: Created on Wed Jan 7 22:18:20 2015 Make gui into a package.
  (Section 2, p. 7)
  - **MapGUI**: Module containg the base classes for the GUI.
    (Section 3, p. 8)
  - **mapview**: ..
    (Section 4, p. 32)
    * **downloader** *(Section 5, p. 55)*
    * **geojson**: ..
      (Section 6, p. 56)
    * **mbtsource**: This provider is based on .mbfiles from MapBox.
      (Section 7, p. 58)
    * **source** *(Section 8, p. 60)*
    * **types** *(Section 9, p. 62)*
    * **utils** *(Section 10, p. 64)*
    * **view** *(Section 11, p. 65)*
  - **triangulation**: Created on Tue Jan 13 00:15:11 2015 Simple triangulation class.
    (Section 12, p. 84)
- **isyUtils**: Module containing util functions.
  (Section 13, p. 86)
- **kmlData**: Module containg all necessary information for creating and parsing a kml-xml.
  (Section 14, p. 87)
- **osmAPI**: Base class that builds a connection to the Openstreetmap API.
  (Section 15, p. 92)
- **osmData**: #Basic class that holds the osm-data (consisting of basing elements)
  (Section 16, p. 94)
- **program**: Main pipeline to compute kml from a given SUR(file).
  (Section 17, p. 108)
- **sur**: Basic class to load and store space usage rules.
  (Section 18, p. 110)
- **surTypeManager**: Helper class that leads known sur types (indoor, outdoor, both) from a file and can be queried for a certain rule.
  (Section 19, p. 112)

1.2 Variables

Name	Description
__package__	**Value:** `None`

2 Package isySUR.gui

Created on Wed Jan 7 22:18:20 2015 Make gui into a package.

Author: jpoeppel

2.1 Modules

- **MapGUI**: Module containg the base classes for the GUI.
  (Section 3, p. 8)
- **mapview**: ..
  (Section 4, p. 32)
  - **downloader** *(Section 5, p. 55)*
  - **geojson**: ..
    (Section 6, p. 56)
  - **mbtsource**: This provider is based on .mbfiles from MapBox.
    (Section 7, p. 58)
  - **source** *(Section 8, p. 60)*
  - **types** *(Section 9, p. 62)*
  - **utils** *(Section 10, p. 64)*
  - **view** *(Section 11, p. 65)*
- **triangulation**: Created on Tue Jan 13 00:15:11 2015 Simple triangulation class.
  (Section 12, p. 84)

2.2 Variables

Name	Description
__package__	**Value:** `None`

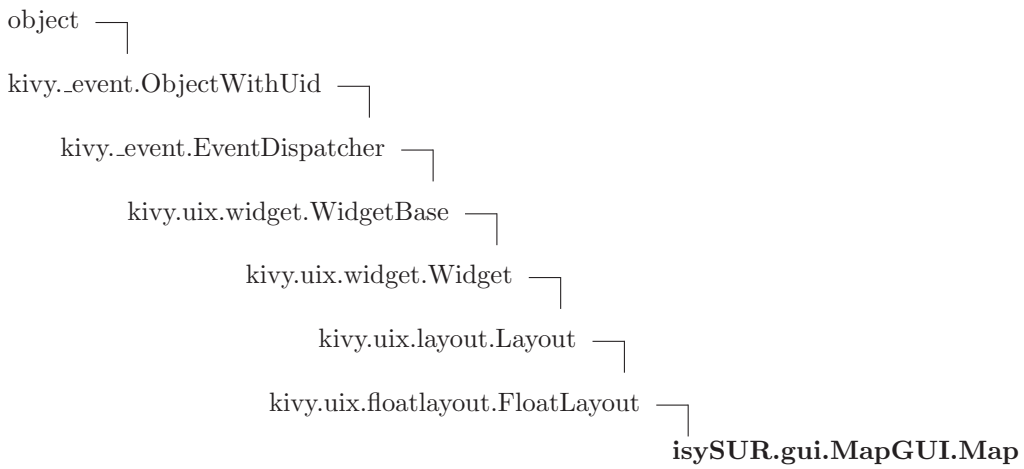
3 Module isySUR.gui.MapGUI

Module containg the base classes for the GUI.

3.1 Variables

Name	Description
__package__	**Value:** 'isySUR.gui'
__warningregistry__	**Value:** {('Not importing directory \'C:\\Python27\\lib\\kivy\': m...

3.2 Class Map



3.2.1 Methods

__init__(*self*, *app*)

Initializies the GUI.

Parameters

`app`: Reference to the Application

(type=kivy.app)

Overrides: object.__init__

setStop(*self*)

Indicator for stopping the SUR calculation Thread.

cleanUpCache(*self*)

Triggers function to delete cache folder.

toast(*self*, *text*, *long_duration*=`False`)

Shows a toast.

Parameters

`duration:` If paramter is True the toast is visible for a long time. Otherwise it has a shorter duration.

(type=Boolean)

open_menu(*self*)

Opens and closes the Main Menu.

open_kmlList(*self*)

Opens and closes the KML List.

showPolygons(*self*, *names*)

Shows all polygones represented by names.

Parameters

`names:` Namelist of Polygons to be displayed on the GUI Map.

(type=[str])

hidePolygons(*self*, *names*)

Hides all polygones represented by names.

Parameters

`names:` Namelist of Polygons to be removed from the GUI Map.

(type=[str])

addPolygon(*self*, *kmlObj*, *kmlName*, *first*=`True`)

Adds all Polygon from one KML Object to the Map.

Parameters

`kmlObj:` KML Data with Placemarks which will be displayed on the Map.

(type=kmlData.KMLObject)

`kmlName:` Name of the kmlObj.

(type=str)

`first:` Decides whether to jump to the first or last added Polygon. If there is only one Polygon in the KML Object and first is True, the Map moves to the Polygon. If first is False, the Map moves to the last added Polygon.

(type=Boolean)

Return Value

Returns whether the map already moved to a polygon and the name of the added Polygon to which the map moves if moved is False.

computeAndShowKmls(*self*, *path*, *queue*)

Calculates all KMLs from a loaded SUR file. The names of the KMLs are added to the KMLList to display all loaded KMLs. And each calculated Polygon of the Placemarks in the KMLs are added to the Map Layer to be displayed. When the calculation is finished, the Map moves to the last added Placemark.

Parameters

`path:` Path to the SUR file

(type=str)

`queue:` Queue in which all calculated KMLs are added (Thread Output)

(type=Queue.Queue)

Inherited from kivy.uix.floatlayout.FloatLayout

add_widget(), do_layout(), remove_widget()

Inherited from kivy.uix.widget.Widget

__eq__(), __hash__(), clear_widgets(), collide_point(), collide_widget(), get_center_x(), get_center_y(), get_parent_window(), get_right(), get_root_window(), get_top(), on_disabled(), on_opacity(), on_touch_down(), on_touch_move(), on_touch_up(), set_center_x(), set_center_y(), set_right(), set_top(), to_local(), to_parent(), to_widget(), to_window()

Inherited from kivy._event.EventDispatcher

__new__(), bind(), create_property(), dispatch(), events(), get_property_observers(), getter(), is_event_type(), properties(), property(), register_event_type(), setter(), unbind(), unregister_event_types()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

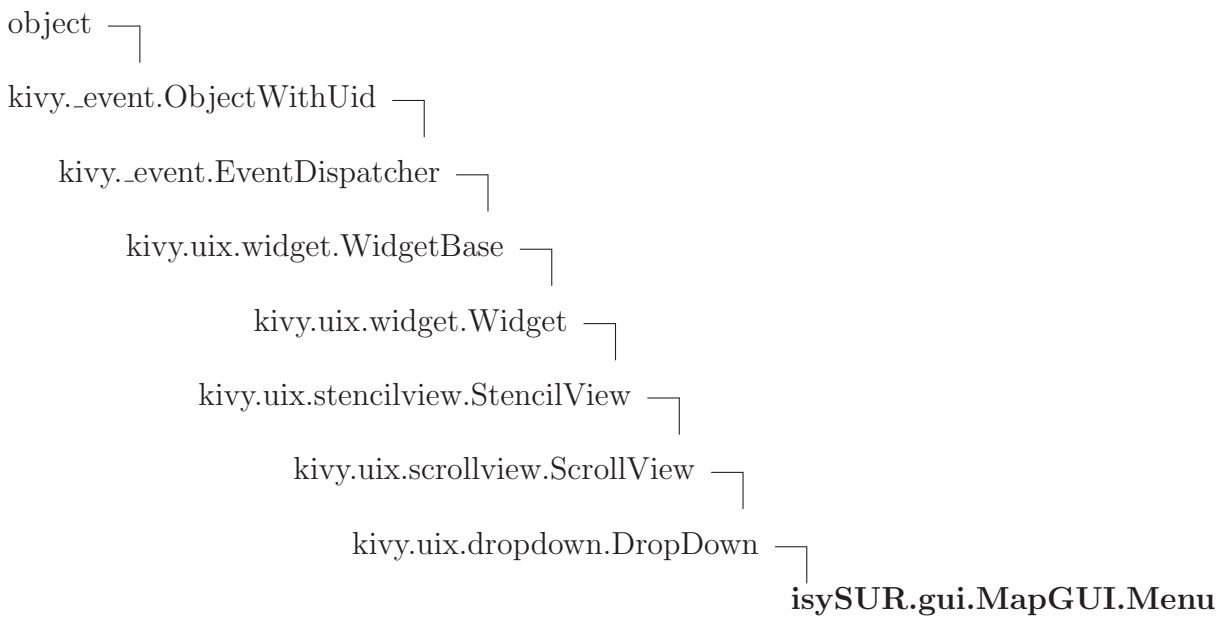
3.2.2 Properties

Name	Description
Inherited from kivy.uix.widget.Widget __self__, proxy_ref	
Inherited from kivy._event.ObjectWithUid uid	
Inherited from object __class__	

3.2.3 Class Variables

Name	Description
text	**Value:** `StringProperty()`
Inherited from kivy.uix.widget.Widget __events__, canvas, center, center_x, center_y, children, cls, disabled, height, id, ids, opacity, parent, pos, pos_hint, right, size, size_hint, size_hint_x, size_hint_y, top, width, x, y	

3.3 Class Menu



3.3.1 Methods

__init__(*self*, *mapview*, *app*)

Initializes the main menu of the GUI.

Parameters

`mapview`: Reference to the main GUI widget.
(type=kivy.floatlayout)

`app`: Reference to the main application.
(type=kivy.app)

Overrides: object.__init__

dismiss_load(*self*)

Dismisses the load popup.

dismiss_save(*self*)

Dismisses the save popup.

dismiss_config(*self*)

Dismisses the config popup.

show_load(*self*, *obj*)

Creates a load popup and displays it.

Parameters

`obj`: Reference to the button which was clicked to open the load popup.

(type=kivy.uix.button)

show_save(*self*, *isConfig*=`False`)

Creates a save popup and displayes it.

Parameters

`obj`: Reference to the Button which was clicked to open the save popup.

(type=kivy.uix.button)

show_config(*self*)

Creates a config popup and displayes it.

load_kml(*self*, *path*, *filename*)

Loads a given kml file.

Parameters

`path`: Path to the selected files.

(type=str)

`filename`: Names of selected files.

(type=[Str])

load_sur(*self*, *path*, *filename*)

Loads a given SUR file.

Parameters

`path:` Path to the selected files.

(type=str)

`filename:` Names of selected files.

(type=[Str])

load_cfg(*self*, *path*, *filename*)

Loads a given config file.

Parameters

`path:` Path to the selected files.

(type=str)

`filename:` Names of selected files.

(type=[Str])

saveConfig(*self*, *path*, *filename*)

Saves the config to the given path and filename.

Parameters

`path:` Path to store location.

(type=str)

`filename:` Name of the new file.

(type=str)

saveKML(*self*, *path*, *filename*)

Saves selected KMLs. If the given path is a directory all selected KMLs are saved separately to the directory. Additional a complete KML containing all KMLs is stored there too. When the store location is a file, all KMLs will be added to one complete KML and stored with the given filename.

Parameters

`path:` Path to store location.

(type=str)

`filename:` Name of the new file.

(type=str)

switchMarkers(*self*, *obj*)

Shows or unshows markers on SUR position.

Parameters

`obj`: Button which changes the marker behaviour.

(type=kivy.uix.button)

Inherited from kivy.uix.dropdown.DropDown

add_widget(), clear_widgets(), dismiss(), on_container(), on_dismiss(), on_key_down(), on_select(), on_touch_down(), on_touch_up(), open(), remove_widget(), select()

Inherited from kivy.uix.scrollview.ScrollView

on__viewport(), on_effect_cls(), on_effect_x(), on_effect_y(), on_touch_move(), simulate_touch_down(), to_local(), to_parent(), update_from_scroll()

Inherited from kivy.uix.widget.Widget

__eq__(), __hash__(), collide_point(), collide_widget(), get_center_x(), get_center_y(), get_parent_window(), get_right(), get_root_window(), get_top(), on_disabled(), on_opacity(), set_center_x(), set_center_y(), set_right(), set_top(), to_widget(), to_window()

Inherited from kivy._event.EventDispatcher

__new__(), bind(), create_property(), dispatch(), events(), get_property_observers(), getter(), is_event_type(), properties(), property(), register_event_type(), setter(), unbind(), unregister_event_types()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

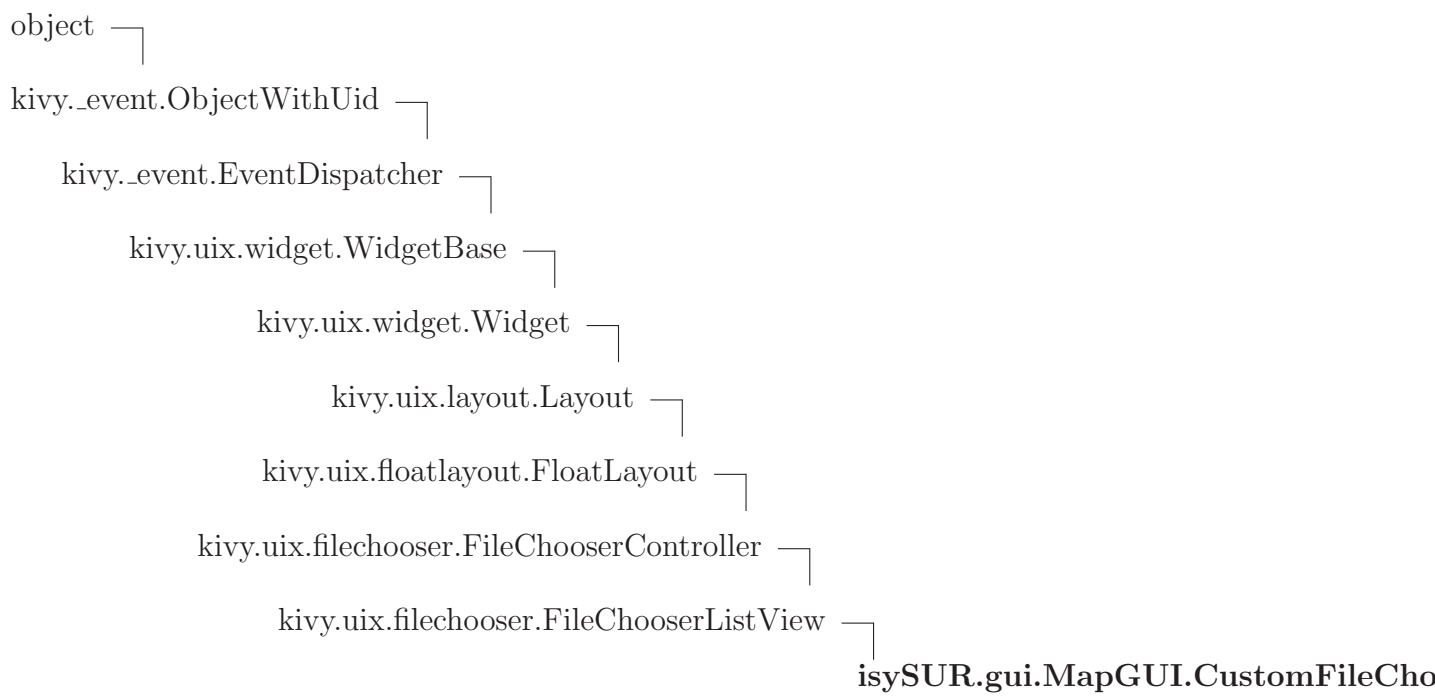
3.3.2 Properties

Name	Description
Inherited from kivy.uix.widget.Widget __self__, proxy_ref	
Inherited from kivy._event.ObjectWithUid uid	
Inherited from object __class__	

3.3.3 Class Variables

Name	Description
loadfile	**Value:** `ObjectProperty(None)`
savefile	**Value:** `ObjectProperty(None)`
text_input	**Value:** `ObjectProperty(None)`
Inherited from kivy.uix.dropdown.DropDown __events__, attach_to, auto_dismiss, auto_width, container, dismiss_on_select, max_height	
Inherited from kivy.uix.scrollview.ScrollView bar_alpha, bar_color, bar_margin, bar_pos, bar_pos_x, bar_pos_y, bar_width, do_scroll, do_scroll_x, do_scroll_y, effect_cls, effect_x, effect_y, hbar, scroll_distance, scroll_timeout, scroll_type, scroll_wheel_distance, scroll_x, scroll_y, vbar, viewport_size	
Inherited from kivy.uix.widget.Widget canvas, center, center_x, center_y, children, cls, disabled, height, id, ids, opacity, parent, pos, pos_hint, right, size, size_hint, size_hint_x, size_hint_y, top, width, x, y	

3.4 Class CustomFileChooser



Implemented this and override the following method to fix path bug.

3.4.1 Methods

open_entry(*self*, *entry*)

Builds the path to the selected item. If it's a directory the filechooser opens it.

Parameters

`entry`: Entry to open

(type=str)

Overrides: kivy.uix.filechooser.FileChooserController.open_entry

Inherited from kivy.uix.filechooser.FileChooserController

__init__(), cancel(), close_subselection(), entry_released(), entry_subselect(), entry_touched(), get_nice_size(), on_entries_cleared(), on_entry_added(), on_remove_subentry(), on_subentry_to_entry(), on_submit(), on_touch_down(), on_touch_up()

Inherited from kivy.uix.floatlayout.FloatLayout

add_widget(), do_layout(), remove_widget()

Inherited from kivy.uix.widget.Widget

__eq__(), __hash__(), clear_widgets(), collide_point(), collide_widget(), get_center_x(), get_center_y(), get_parent_window(), get_right(), get_root_window(), get_top(), on_disabled(), on_opacity(), on_touch_move(), set_center_x(), set_center_y(), set_right(), set_top(), to_local(), to_parent(), to_widget(), to_window()

Inherited from kivy._event.EventDispatcher

__new__(), bind(), create_property(), dispatch(), events(), get_property_observers(), getter(), is_event_type(), properties(), property(), register_event_type(), setter(), unbind(), unregister_event_types()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

3.4.2 Properties

Name	Description
Inherited from kivy.uix.widget.Widget	
__self__, proxy_ref	
Inherited from kivy._event.ObjectWithUid	
uid	
Inherited from object	

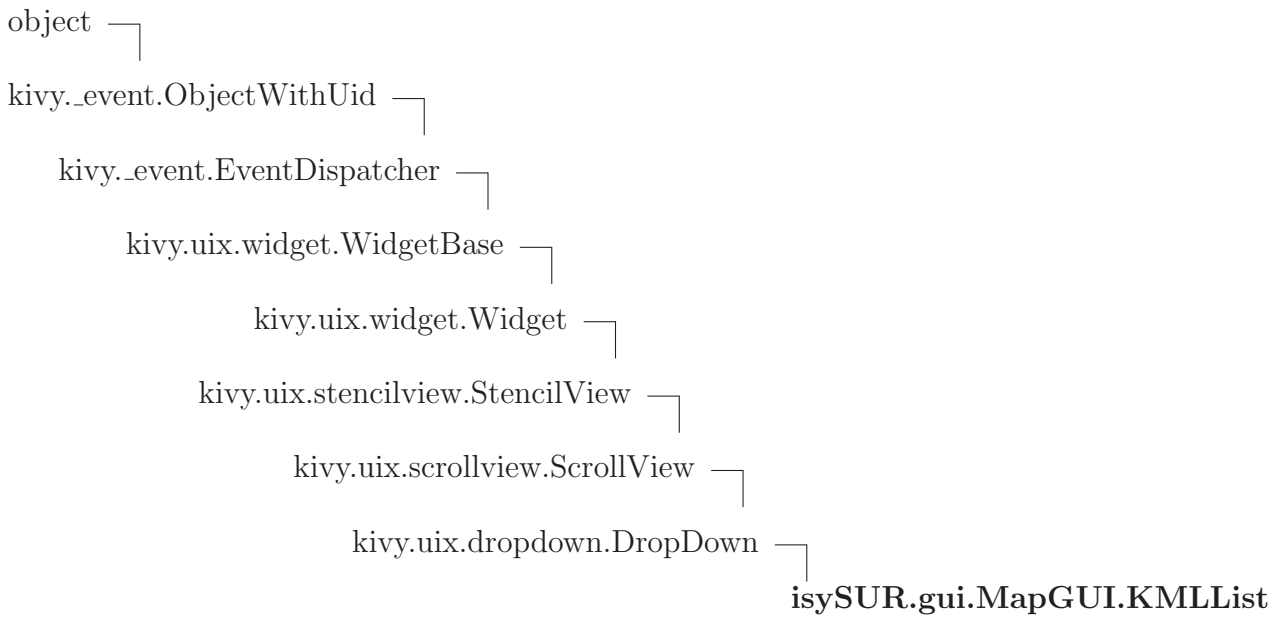
continued on next page

Name	Description
__class__	

3.4.3 Class Variables

Name	Description
Inherited from kivy.uix.filechooser.FileChooserController	
__events__, dirselect, file_encodings, file_system, files, filter_dirs, filters, multiselect, path, progress_cls, rootpath, selection, show_hidden, sort_func	
Inherited from kivy.uix.widget.Widget	
canvas, center, center_x, center_y, children, cls, disabled, height, id, ids, opacity, parent, pos, pos_hint, right, size, size_hint, size_hint_x, size_hint_y, top, width, x, y	

3.5 Class KMLList



3.5.1 Methods

__init__(*self*, *mapview*, *app*)

Initializes the KMLList menu, which displays all loaded KML files.

Parameters

`mapview`: Reference to the main GUI Widget
(type=kivy.floatlayout)

`app`: Reference to the main Application
(type=kivy.app)

Overrides: object.__init__

createList(*self*)

Creates the KML List.

selectBut(*self*, *obj*)

Hides or shows the selected KML on the Map.

Parameters

`obj`: Button which represents a loaded KML.
(type=kivy.uix.button)

addItem(*self*, *name*)

Adds an item to the KML List.

Parameters

`name`: Name of the new item.
(type=str)

Inherited from kivy.uix.dropdown.DropDown

add_widget(), clear_widgets(), dismiss(), on_container(), on_dismiss(), on_key_down(), on_select(), on_touch_down(), on_touch_up(), open(), remove_widget(), select()

Inherited from kivy.uix.scrollview.ScrollView

on__viewport(), on_effect_cls(), on_effect_x(), on_effect_y(), on_touch_move(), simulate_touch_down(), to_local(), to_parent(), update_from_scroll()

Inherited from kivy.uix.widget.Widget

__eq__(), __hash__(), collide_point(), collide_widget(), get_center_x(), get_center_y(), get_parent_window(), get_right(), get_root_window(), get_top(), on_disabled(), on_opacity(),

set_center_x(), set_center_y(), set_right(), set_top(), to_widget(), to_window()

Inherited from kivy._event.EventDispatcher

__new__(), bind(), create_property(), dispatch(), events(), get_property_observers(), getter(), is_event_type(), properties(), property(), register_event_type(), setter(), unbind(), unregister_event_types()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

3.5.2 Properties

Name	Description
Inherited from kivy.uix.widget.Widget __self__, proxy_ref	
Inherited from kivy._event.ObjectWithUid uid	
Inherited from object __class__	

3.5.3 Class Variables

Name	Description
Inherited from kivy.uix.dropdown.DropDown __events__, attach_to, auto_dismiss, auto_width, container, dismiss_on_select, max_height	
Inherited from kivy.uix.scrollview.ScrollView bar_alpha, bar_color, bar_margin, bar_pos, bar_pos_x, bar_pos_y, bar_width, do_scroll, do_scroll_x, do_scroll_y, effect_cls, effect_x, effect_y, hbar, scroll_distance, scroll_timeout, scroll_type, scroll_wheel_distance, scroll_x, scroll_y, vbar, viewport_size	
Inherited from kivy.uix.widget.Widget canvas, center, center_x, center_y, children, cls, disabled, height, id, ids, opacity, parent, pos, pos_hint, right, size, size_hint, size_hint_x, size_hint_y, top, width, x, y	

3.6 Class Toast

3.6.1 Methods

__init__(*self*, *mapview*)

Initializes a new Toast.

Parameters

`mapview`: Reference to the main GUI Widget

(type=kivy.floatlayout)

Overrides: object.__init__

stayVisible(*self*, *text*)

Displayes the toast for an unkown duration.

Parameters

`text`: Text of the toast.

(type=str)

remove(*self*)

Removes a toast after stayVisible() was called.

show(*self*, *text*, *length_long*)

Displayes a toast for the short or long duration.

Parameters

`text:` Text of the toast.

(type=str)

`length_long:` When length_long is True, the toast is visible for a long duration, otherwise it is only visible for a short duration.

(type=Boolean)

Inherited from kivy.uix.label.Label

on_ref_press(), on_touch_down(), texture_update()

Inherited from kivy.uix.widget.Widget

__eq__(), __hash__(), add_widget(), clear_widgets(), collide_point(), collide_widget(), get_center_x(), get_center_y(), get_parent_window(), get_right(), get_root_window(), get_top(), on_disabled(), on_opacity(), on_touch_move(), on_touch_up(), remove_widget(), set_center_x(), set_center_y(), set_right(), set_top(), to_local(), to_parent(), to_widget(), to_window()

Inherited from kivy._event.EventDispatcher

__new__(), bind(), create_property(), dispatch(), events(), get_property_observers(), getter(), is_event_type(), properties(), property(), register_event_type(), setter(), unbind(), unregister_event_types()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

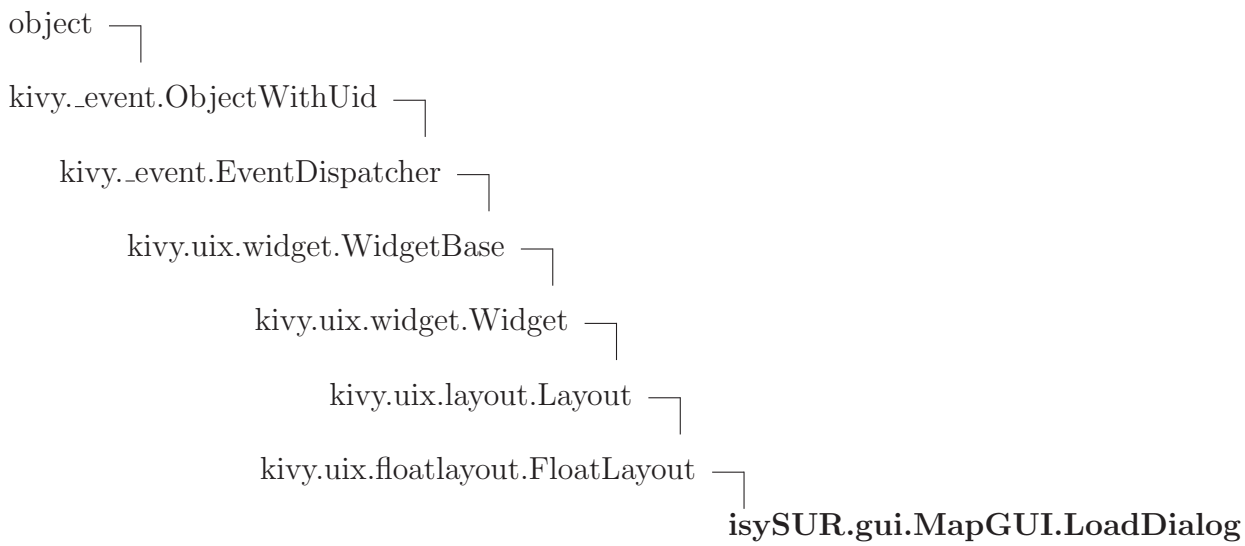
3.6.2 Properties

Name	Description
Inherited from kivy.uix.widget.Widget __self__, proxy_ref	
Inherited from kivy._event.ObjectWithUid uid	
Inherited from object __class__	

3.6.3 Class Variables

Name	Description
Inherited from kivy.uix.label.Label anchors, bold, color, disabled_color, font_name, font_size, halign, italic, line_height, markup, max_lines, mipmap, padding, padding_x, padding_y, refs, shorten, text, text_size, texture, texture_size, valign	
Inherited from kivy.uix.widget.Widget __events__, canvas, center, center_x, center_y, children, cls, disabled, height, id, ids, opacity, parent, pos, pos_hint, right, size, size_hint, size_hint_x, size_hint_y, top, width, x, y	

3.7 Class LoadDialog



3.7.1 Methods

Inherited from kivy.uix.floatlayout.FloatLayout

__init__(), add_widget(), do_layout(), remove_widget()

Inherited from kivy.uix.widget.Widget

__eq__(), __hash__(), clear_widgets(), collide_point(), collide_widget(), get_center_x(), get_center_y(), get_parent_window(), get_right(), get_root_window(), get_top(), on_disabled(), on_opacity(), on_touch_down(), on_touch_move(), on_touch_up(), set_center_x(), set_center_y(), set_right(), set_top(), to_local(), to_parent(), to_widget(), to_window()

Inherited from kivy._event.EventDispatcher

__new__(), bind(), create_property(), dispatch(), events(), get_property_observers(), getter(), is_event_type(), properties(), property(), register_event_type(), setter(), unbind(), unregister_event_types()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

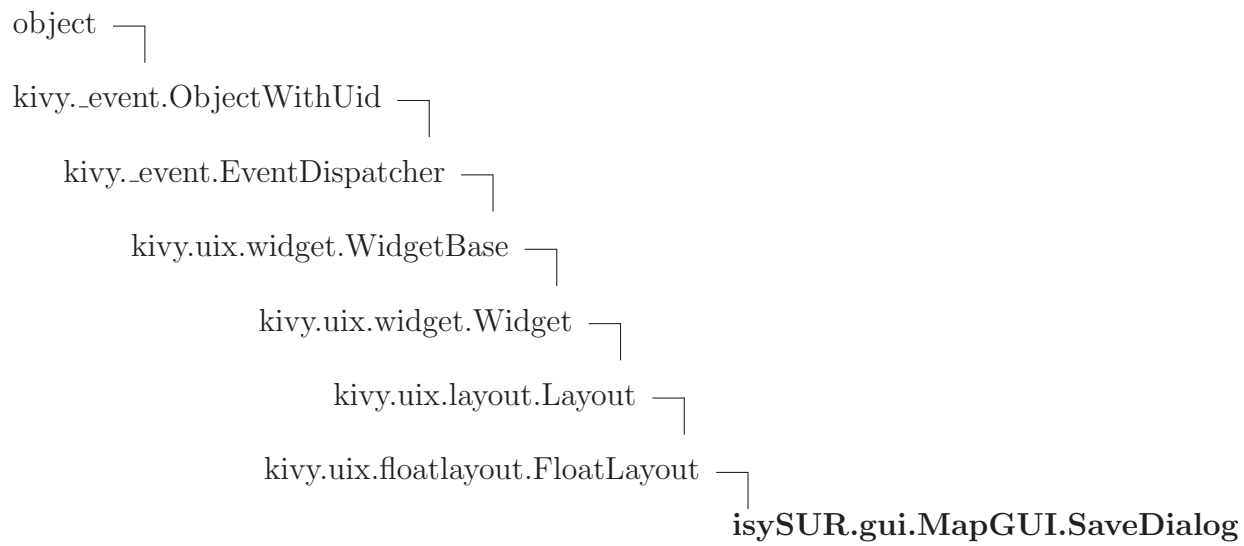
3.7.2 Properties

Name	Description
Inherited from kivy.uix.widget.Widget __self__, proxy_ref	
Inherited from kivy._event.ObjectWithUid uid	
Inherited from object __class__	

3.7.3 Class Variables

Name	Description
load	**Value:** `ObjectProperty(None)`
cancel	**Value:** `ObjectProperty(None)`
test	**Value:** `ObjectProperty(None)`
Inherited from kivy.uix.widget.Widget __events__, canvas, center, center_x, center_y, children, cls, disabled, height, id, ids, opacity, parent, pos, pos_hint, right, size, size_hint, size_hint_x, size_hint_y, top, width, x, y	

3.8 Class SaveDialog



3.8.1 Methods

Inherited from kivy.uix.floatlayout.FloatLayout

__init__(), add_widget(), do_layout(), remove_widget()

Inherited from kivy.uix.widget.Widget

__eq__(), __hash__(), clear_widgets(), collide_point(), collide_widget(), get_center_x(), get_center_y(), get_parent_window(), get_right(), get_root_window(), get_top(), on_disabled(), on_opacity(), on_touch_down(), on_touch_move(), on_touch_up(), set_center_x(), set_center_y(), set_right(), set_top(), to_local(), to_parent(), to_widget(), to_window()

Inherited from kivy._event.EventDispatcher

__new__(), bind(), create_property(), dispatch(), events(), get_property_observers(), getter(), is_event_type(), properties(), property(), register_event_type(), setter(), unbind(), unregister_event_types()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

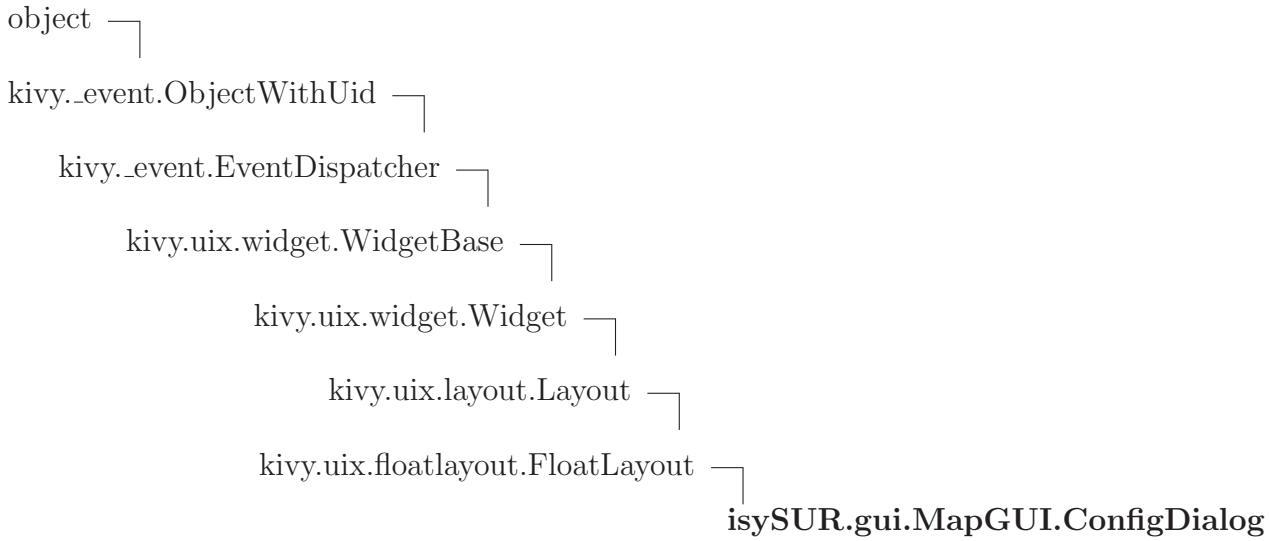
3.8.2 Properties

Name	Description
Inherited from kivy.uix.widget.Widget __self__, proxy_ref	
Inherited from kivy._event.ObjectWithUid uid	
Inherited from object __class__	

3.8.3 Class Variables

Name	Description
save	**Value:** `ObjectProperty(None)`
text_input	**Value:** `ObjectProperty(None)`
cancel	**Value:** `ObjectProperty(None)`
Inherited from kivy.uix.widget.Widget __events__, canvas, center, center_x, center_y, children, cls, disabled, height, id, ids, opacity, parent, pos, pos_hint, right, size, size_hint, size_hint_x, size_hint_y, top, width, x, y	

3.9 Class ConfigDialog



3.9.1 Methods

__init__(*self*, *app*, *save*, *load*, *cancel*)

Initilizes the config dialog.

Parameters

`app:` Reference to the main Application
(type=kivy.app)

`save:` Reference to save function.
(type=kivy.uix.property.ObjectProperty)

`load:` Reference to load function.
(type=kivy.uix.property.ObjectProperty)

`cancel:` Reference to cancel function.
(type=kivy.uix.property.ObjectProperty)

Overrides: object.__init__

addConfigContent(*self*)

Adds the loaded config to the Config Popup.

addContentHeader(*self*)

Adds the ruleAreas to the Config Popup.

addConfigEntry(*self*, *ruleArea*, *rule*)

Adds one config rule to the Config Popup.

Parameters

`ruleArea:` Field of application of the rule.
(type=str)

`rule:` SUR Rule of this entry.
(type=str)

changeRuleArea(*self*, **args*)

Changes the field of application of a rule.

Parameters

`args:` List of arguments from the kivy.uix.checkbox, when selecting the new rule area.
(type=[])

action(*self*, *obj*)

Changes the action of the Action Button in the Config Popup.

Possible actions:

- Create new rule
- Add new rule to config
- Delete selected rules

Parameters

`obj`: Actionbutton

(type=kivy.uix.button)

clearConfig(*self*)

Clears the config popup.

deleteEntry(*self*, **args*)

Adds or removes rule from the deletion list.

Parameters

`args`: List of arguments from the Checkbox when clicked.

(type=[])

Inherited from kivy.uix.floatlayout.FloatLayout

add_widget(), do_layout(), remove_widget()

Inherited from kivy.uix.widget.Widget

__eq__(), __hash__(), clear_widgets(), collide_point(), collide_widget(), get_center_x(), get_center_y(), get_parent_window(), get_right(), get_root_window(), get_top(), on_disabled(), on_opacity(), on_touch_down(), on_touch_move(), on_touch_up(), set_center_x(), set_center_y(), set_right(), set_top(), to_local(), to_parent(), to_widget(), to_window()

Inherited from kivy._event.EventDispatcher

__new__(), bind(), create_property(), dispatch(), events(), get_property_observers(), getter(), is_event_type(), properties(), property(), register_event_type(), setter(), unbind(), unregister_event_types()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

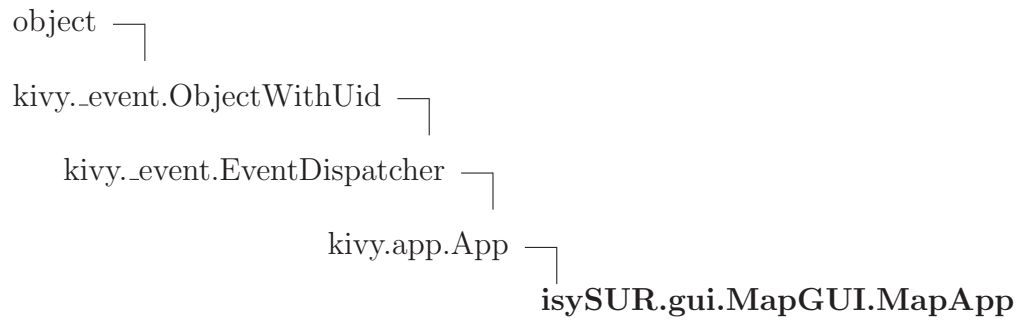
3.9.2 Properties

Name	Description
Inherited from kivy.uix.widget.Widget __self__, proxy_ref	
Inherited from kivy._event.ObjectWithUid uid	
Inherited from object __class__	

3.9.3 Class Variables

Name	Description
Inherited from kivy.uix.widget.Widget __events__, canvas, center, center_x, center_y, children, cls, disabled, height, id, ids, opacity, parent, pos, pos_hint, right, size, size_hint, size_hint_x, size_hint_y, top, width, x, y	

3.10 Class MapApp



3.10.1 Methods

__init__(*self*, *configPath=''*)

Initializes main application class.

Parameters

`configPath`: Path to the config, when loaded on start up.
(type=str)

Overrides: object.__init__

on_stop(*self*)

Stops the SUR calculation if one is running and cleans up the cache when program is closed.

Overrides: kivy.app.App.on_stop

on_start(*self*)

Sets the icon and title of the program on start up.

Overrides: kivy.app.App.on_start

build(*self*)

```
Initializes the application; will be called only once.
If this method returns a widget (tree), it will be used as the root
widget and added to the window.

:return: None or a root :class:‘~kivy.uix.widget.Widget‘ instance
         if no self.root exists.
```

Overrides: kivy.app.App.build extit(inherited documentation)

loadConfig(*self*, *configPath*)

Load the given config.

Parameters

`configPath`: Path to the config

(type=str)

clearConfig(*self*)

Empties the config.

isConfigEmpty(*self*)

Checks whether the config is empty.

Return Value

Return True, when config is empty, otherwise False.

addKML(*self*, *kmlObj*)

Adds a KML to the application. and returns the stored name of the kmlObj.

Parameters

`kmlObj`: KML data to be added.

(type=kmlData.KMLObject)

Return Value

Name of the kmlObj under which it is stored.

getPolygonFromPlacemark(*self*, *placemark*)

Returns the Polygon of a Placemark.

Parameters

`placemark`: Placemark from which the polygon is returned.

(type=kmlData.Placemark)

Return Value

List of Polygon coords

getSelectedPolygons(*self*)

Get all active KMLObjects of the application.

Return Value

Returns a list of selected KMLObjects.

Inherited from kivy.app.App

build_config(), build_settings(), close_settings(), create_settings(), destroy_settings(), display_settings(), get_application_config(), get_application_icon(), get_application_name(), get_running_app(), load_config(), load_kv(), on_config_change(), on_icon(), on_pause(), on_resume(), on_title(), open_settings(), run(), stop()

Inherited from kivy._event.EventDispatcher

__new__(), bind(), create_property(), dispatch(), events(), get_property_observers(), getter(), is_event_type(), properties(), property(), register_event_type(), setter(), unbind(), unregister_event_types()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

3.10.2 Properties

Name	Description
Inherited from kivy.app.App directory, name, user_data_dir	
Inherited from kivy._event.ObjectWithUid uid	
Inherited from object __class__	

3.10.3 Class Variables

Name	Description
Inherited from kivy.app.App __events__, icon, kv_directory, kv_file, settings_cls, title, use_kivy_settings	

4 Package isySUR.gui.mapview

(section) MapView

.. author:: Mathieu Virbel <mat@kivy.org>

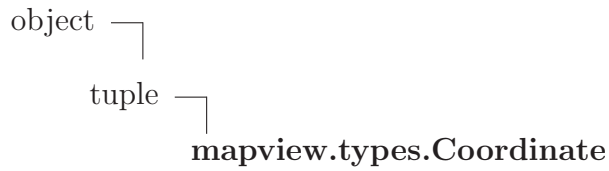
MapView is a Kivy widget that display maps.

Version: 0.2

4.1 Modules

- **downloader** *(Section 5, p. 55)*
- **geojson**: ..
  (Section 6, p. 56)
- **mbtsource**: This provider is based on .mbfiles from MapBox.
  (Section 7, p. 58)
- **source** *(Section 8, p. 60)*
- **types** *(Section 9, p. 62)*
- **utils** *(Section 10, p. 64)*
- **view** *(Section 11, p. 65)*

4.2 Class Coordinate



Coordinate(lat, lon)

4.2.1 Methods

__getnewargs__(*self*)

Return self as a plain tuple. Used by copy and pickle.

Overrides: tuple.__getnewargs__

__getstate__(*self*)

Exclude the OrderedDict from pickling

__new__(*_cls*, *lat*, *lon*)

Create new instance of Coordinate(lat, lon)

Return Value

a new object with type S, a subtype of T

Overrides: object.__new__

__repr__(*self*)

Return a nicely formatted representation string

Overrides: object.__repr__

Inherited from tuple

__add__(), __contains__(), __eq__(), __ge__(), __getattribute__(), __getitem__(), __getslice__(), __gt__(), __hash__(), __iter__(), __le__(), __len__(), __lt__(), __mul__(), __ne__(), __rmul__(), __sizeof__(), count(), index()

Inherited from object

__delattr__(), __format__(), __init__(), __reduce__(), __reduce_ex__(), __setattr__(), __str__(), __subclasshook__()

4.2.2 Properties

Name	Description
lat	Alias for field number 0
lon	Alias for field number 1
Inherited from object __class__	

4.3 Class Bbox

4.3.1 Methods

collide(*self*, **args*)

Inherited from tuple

__add__(), __contains__(), __eq__(), __ge__(), __getattribute__(), __getitem__(), __getnewargs__(), __getslice__(), __gt__(), __hash__(), __iter__(), __le__(), __len__(), __lt__(), __mul__(), __ne__(), __new__(), __repr__(), __rmul__(), __sizeof__(), count(), index()

Inherited from object

__delattr__(), __format__(), __init__(), __reduce__(), __reduce_ex__(), __setattr__(), __str__(), __subclasshook__()

4.3.2 Properties

Name	Description
Inherited from object	
__class__	

4.4 Class MapSource

object
└ **mapview.source.MapSource**

Known Subclasses: isySUR.gui.mapview.mbtsource.MBTilesMapSource

Base class for implementing a map source / provider

4.4.1 Methods

__init__(*self*, *url*='http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', *cache_key*=None, *min_zoom*=0, *max_zoom*=19, *tile_size*=256, *image_ext*='png', *attribution*='\xc2\xa9 OpenStreetMap contributors', *subdomains*='abc')

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

fill_tile(*self*, *tile*)

Add this tile to load within the downloader

from_provider(*key*)

get_col_count(*self*, *zoom*)

Get the number of tiles in a col at this zoom level

get_lat(*self*, *zoom*, *y*)

Get the latitude to the y position in the map source's projection

get_lon(*self*, *zoom*, *x*)

Get the longitude to the x position in the map source's projection

get_max_zoom(*self*)

Return the maximum zoom of this source

get_min_zoom(*self*)

Return the minimum zoom of this source

get_row_count(*self*, *zoom*)

Get the number of tiles in a row at this zoom level

get_x(*self*, *zoom*, *lon*)

Get the x position on the map using this map source's projection (0, 0) is located at the top left.

get_y(*self*, *zoom*, *lat*)

Get the y position on the map using this map source's projection (0, 0) is located at the top left.

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

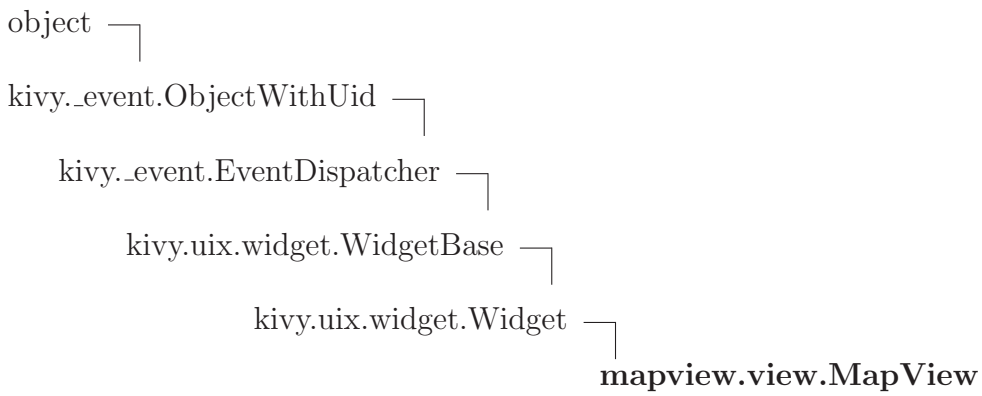
4.4.2 Properties

Name	Description
Inherited from object	
__class__	

4.4.3 Class Variables

Name	Description
providers	**Value:** {'cyclemap': (0, 0, 17, 'http://{s}.tile.opencyclemap.org...

4.5 Class MapView



MapView is the widget that control the map displaying, navigation, and layers management.

4.5.1 Methods

__init__(*self*, ***kwargs*)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

addPolygon(*self*, *name*, *polygon*, *color*, *markerCoords*)

Adds and draws a new polygon onto the map.

Parameters

`name:`	Name of the polygon to be added. *(type=str)*
`polygon:`	List of vertices of the polygon. *(type=[(float, float)])*
`color:`	Style value of KML *(type=dict)*
`markerCoords:`	Coordinates of the SUR. *(type=Tuple(float, float))*

add_layer(*self*, *layer*, *mode='window'*)

Add a new layer to update at the same time the base tile layer. mode can be either "scatter" or "window". If "scatter", it means the layer will be within the scatter transformation. It's perfect if you want to display path / shape, but not for text. If "window", it will have no transformation. You need to position the widget yourself: think as Z-sprite / billboard. Defaults to "window".

Parameters

`layer:` The layer for updating.

(type=kivy.uix.widget.Widget)

`mode:` (Optional) The mode for updating could be "scatter" or "window".

(type=str)

add_marker(*self*, *marker*, *layer=None*)

Add a marker onto the layer. If layer is None, it will be added in the default marker layer. If there is no default marker layer, a new one will be automatically created.

Parameters

`marker:` The marker to be added.

(type=view.MapMarker)

`layer:` (Optional) the layer the marker should be added to.

(type=view.MarkerMapLayer)

add_widget(*self*, *widget*)

```
Add a new widget as a child of this widget.

:Parameters:
    'widget': :class:'Widget'
        Widget to add to our list of children.
    'index': int, defaults to 0
        *(this attribute was added in 1.0.5)*
        Index to insert the widget in the list

>>> from kivy.uix.button import Button
>>> from kivy.uix.slider import Slider
>>> root = Widget()
>>> root.add_widget(Button())
>>> slider = Slider()
>>> root.add_widget(slider)
```

Overrides: kivy.uix.widget.Widget.add_widget extit(inherited documentation)

animated_diff_scale_at(*self*, *d*, *x*, *y*)

bbox_for_zoom(*self*, *vx*, *vy*, *w*, *h*, *zoom*)

center_on(*self*, **args*)

Center the map on the coordinate :class:'Coordinate', or a (lat, lon.

cleanUpCache(*self*)

convertKMLColor(*self*, *kmlColor*)

Convert a KML Color to its rgba value between 0 and 1.

Parameters

`kmlColor`: Color to be converted.

(type=str)

Return Value

Returns the rgba values of kmlColor.

diff_scale_at(*self*, *d*, *x*, *y*)

do_update(*self*, *dt*)

drawPolygon(*self*)

Draws a Polygon onto the Map.

getBBoxOfPolygon(*self*, *polygon*)

This function calculates a boundingbox in respect to the given polygon.

Parameters

`polygon:` The polygon for which the bbox should be computed, e.g. [(1.0,2.0),(2.0,1.0),(1.0,1.0),(1.0,2.0)].

Return Value

The calculated bbox, e.g. [minLat,minLon, maxLat,maxLon].

(type=[float,float,float,float])

get_bbox(*self*, *margin*=`0`)

Returns the bounding box from the bottom/left (lat1, lon1) to top/right (lat2, lon2).

Parameters

`margin:` (Optional) addition margin for the boundingbox.

Return Value

The boundingbox. @rtype mapview.Bbox

get_latlon_at(*self*, *x*, *y*, *zoom*=`None`)

Return the current :class:‘Coordinate‘ within the (x, y) widget coordinate.

Parameters

`x:` The x-coordinate of the point.

(type=float)

`y:` The y-coordinate of the point.

(type=float)

Return Value

The current Coordinate within the (x,y) widget coordinate.

(type=mapview.Coordinate)

get_window_xy_from(*self*, *lat*, *lon*, *zoom*)

Returns the x/y position in the widget absolute coordinates from a lat/lon.

Return Value

The x/y position in the widget.

(type=float,float)

hideMarkers(*self*)

Hides all markers on the Marker Layer.

hidePolygon(*self*, *name*)

Removes a polygon from the Map.

Parameters

`name`: Name of the polygon to be removed.

(type=str)

isPolyInView(*self*, *name*)

This function proves if a polygon is in the current viewspace.

Parameters

`name`: The name of the polygon.

(type=str)

Return Value

True if in viewspace otherwise False.

(type=boolean)

isPolyVisible(*self*, *name*)

This function proves if a polygon is completely visible in viewspace.

Parameters

`name`: The name of the polygon.

(type=str)

Return Value

True if completely in viewspace otherwise False.

(type=boolean)

load_tile(*self*, *x*, *y*, *size*, *zoom*)

load_tile_for_source(*self*, *map_source*, *opacity*, *size*, *x*, *y*, *zoom*)

load_visible_tiles(*self*)

move_tiles_to_background(*self*)

on_map_relocated(*self*, *zoom*, *coord*)

on_map_source(*self*, *instance*, *source*)

on_pos(*self*, *instance*, *pos*)

on_size(*self*, *instance*, *size*)

on_touch_down(*self*, *touch*)

```
Receive a touch down event.

:Parameters:
    `touch`: :class:`~kivy.input.motionevent.MotionEvent` class
        Touch received. The touch is in parent coordinates. See
        :mod:`~kivy.uix.relativelayout` for a discussion on
        coordinate systems.

:Returns:
    bool. If True, the dispatching of the touch event will stop.
```

Overrides: kivy.uix.widget.Widget.on_touch_down extit(inherited documentation)

on_transform(*self*, **args*)

on_zoom(*self*, *instance*, *zoom*)

on_zoom Event handler

remove_all_tiles(*self*)

remove_layer(*self*, *layer*)

Remove the layer.

Parameters

`layer`: The layer to be removed. @type kivy.uix.widget.Widget

remove_marker(*self*, *marker*)

Remove a marker from its layer.

remove_widget(*self*, *widget*)

```
Remove a widget from the children of this widget.

:Parameters:
    `widget`: :class:`Widget`
        Widget to remove from our children list.

>>> from kivy.uix.button import Button
>>> root = Widget()
>>> button = Button()
>>> root.add_widget(button)
>>> root.remove_widget(button)
```

Overrides: kivy.uix.widget.Widget.remove_widget extit(inherited documentation)

scale_at(*self*, *scale*, *x*, *y*)

set_zoom_at(*self*, *zoom*, *x*, *y*, *scale*=`None`)

Sets the zoom level, leaving the (x, y) at the exact same point in the view.

Parameters

`zoom`: tThe zoom level.

`type`: int

`x`: The x-coordinate of the point.

(type=float)

`y`: The y-coordinate of the point.

(type=float)

`scale`: (Optinal) the scalefaktor for the scatter.

(type=int)

showMarkers(*self*)

Shows all markers.

showPolygon(*self*, *name*)

Makes a polygon visible on the Map.

Parameters

`name`: Name of the polygon to be shown.

(type=str)

sync_to(*self*, *other*)

Reflect the lat/lon/zoom of the other MapView to the current one.

tile_in_tile_map(*self*, *tile_x*, *tile_y*)

tile_map_set(*self*, *tile_x*, *tile_y*, *value*)

trigger_update(*self*, *full*)

unload(*self*)

Unload the view and all the layers. It also cancel all the remaining downloads.

zoom_to(*self*, *lat*, *lon*, *zoom*)

Zooms to the given zoom level at the given position.

Parameters

`lat`: Lat-coordinate of the given position.

(type=float)

`lon`: Lon-coordinate of the given position.

(type=float)

`zoom`: Zoom-factor.

(type=int)

zoom_to_Polygon(*self*, *name*, *zoom*)

Zooms to the given zoom level at the given polygon. The zoom parameter is ignored if the user zoomed in more already and the entire polygon is aready visible.

Parameters

`name`: Name of the polygon.

(type=str)

`zoom`: *(type=int)*

Inherited from kivy.uix.widget.Widget

__eq__(), __hash__(), clear_widgets(), collide_point(), collide_widget(), get_center_x(), get_center_y(), get_parent_window(), get_right(), get_root_window(), get_top(), on_disabled(), on_opacity(), on_touch_move(), on_touch_up(), set_center_x(), set_center_y(), set_right(), set_top(), to_local(), to_parent(), to_widget(), to_window()

Inherited from kivy._event.EventDispatcher

__new__(), bind(), create_property(), dispatch(), events(), get_property_observers(), getter(), is_event_type(), properties(), property(), register_event_type(), setter(), unbind(), unregister_event_types()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

4.5.2 Properties

Name	Description
scale	Returns the current scalefaktor. *(type=float)*
viewport_pos	Returns the current viewport position.
Inherited from kivy.uix.widget.Widget __self__, proxy_ref	
Inherited from kivy._event.ObjectWithUid uid	
Inherited from object __class__	

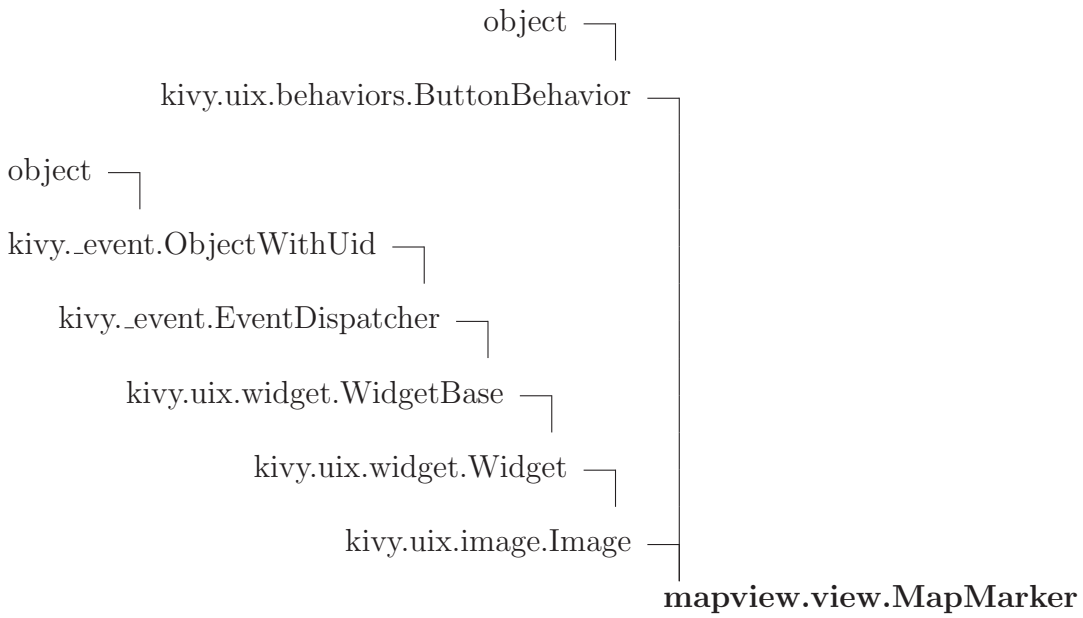
4.5.3 Class Variables

Name	Description
__events__	**Value:** `['on_map_relocated']`
background_color	**Value:** `<kivy.properties.ListProperty object at 0x0000000003B1A888>`
bbox	**Value:** `<kivy.properties.AliasProperty object at 0x0000000003B0F748>`
delta_x	**Value:** `<kivy.properties.NumericProperty object at 0x0000000003B1...`
delta_y	**Value:** `<kivy.properties.NumericProperty object at 0x0000000003B1...`
double_tap_zoom	**Value:** `<kivy.properties.BooleanProperty object at 0x0000000003B1...`
lat	**Value:** `<kivy.properties.NumericProperty object at 0x0000000003AC...`
lon	**Value:** `<kivy.properties.NumericProperty object at 0x0000000003AC...`
map_source	**Value:** `<kivy.properties.ObjectProperty object at 0x0000000003B2B...`

continued on next page

Name	Description
markers	**Value:** <kivy.properties.BooleanProperty object at 0x0000000003B1...
zoom	**Value:** <kivy.properties.NumericProperty object at 0x0000000003B1...
Inherited from kivy.uix.widget.Widget canvas, center, center_x, center_y, children, cls, disabled, height, id, ids, opacity, parent, pos, pos_hint, right, size, size_hint, size_hint_x, size_hint_y, top, width, x, y	

4.6 Class MapMarker



Known Subclasses: mapview.view.MapMarkerPopup

A marker on a map, that must be used on a :class:‘MapMarker‘

4.6.1 Methods

__init__(*self*, ***kwargs*)
x.__init__(...) initializes x; see help(type(x)) for signature
Overrides: object.__init__ extit(inherited documentation)

Inherited from kivy.uix.behaviors.ButtonBehavior

on_press(), on_release(), on_touch_down(), on_touch_move(), on_touch_up(), trigger_action()

Inherited from kivy.uix.image.Image

get_image_ratio(), get_norm_image_size(), on_anim_delay(), on_nocache(), on_texture(), reload(), texture_update()

Inherited from kivy.uix.widget.Widget

__eq__(), __hash__(), add_widget(), clear_widgets(), collide_point(), collide_widget(), get_center_x(), get_center_y(), get_parent_window(), get_right(), get_root_window(), get_top(), on_disabled(), on_opacity(), remove_widget(), set_center_x(), set_center_y(), set_right(), set_top(), to_local(), to_parent(), to_widget(), to_window()

Inherited from kivy._event.EventDispatcher

__new__(), bind(), create_property(), dispatch(), events(), get_property_observers(), getter(), is_event_type(), properties(), property(), register_event_type(), setter(), unbind(), unregister_event_types()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

4.6.2 Properties

Name	Description
default_marker_fn	*(type=str)*
Inherited from kivy.uix.widget.Widget __self__, proxy_ref	
Inherited from kivy._event.ObjectWithUid uid	
Inherited from object __class__	

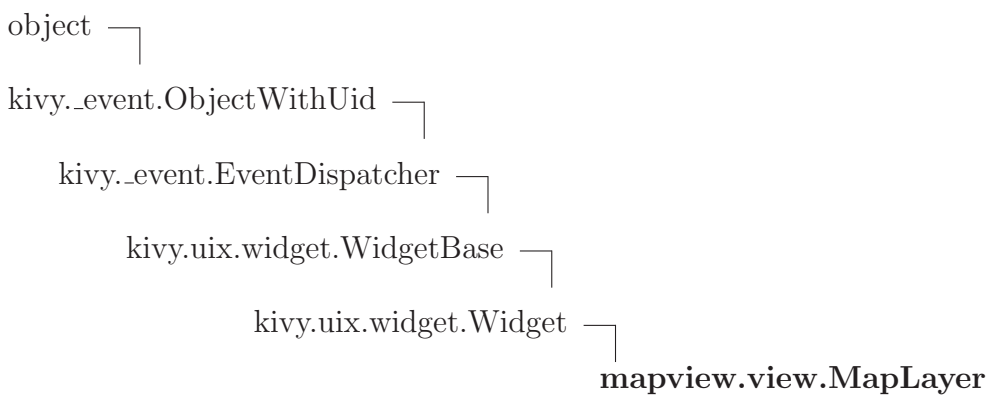
4.6.3 Class Variables

Name	Description
anchor_x	**Value:** <kivy.properties.NumericProperty object at 0x0000000003AC...
anchor_y	**Value:** <kivy.properties.NumericProperty object at 0x0000000003AC...
lat	**Value:** <kivy.properties.NumericProperty object at 0x0000000003AC...

continued on next page

Name	Description
lon	**Value:** <kivy.properties.NumericProperty object at 0x0000000003AC...
visible	**Value:** <kivy.properties.NumericProperty object at 0x0000000003AC...
Inherited from kivy.uix.behaviors.ButtonBehavior last_touch, state	
Inherited from kivy.uix.image.Image allow_stretch, anim_delay, color, image_ratio, keep_data, keep_ratio, mipmap, nocache, norm_image_size, source, texture, texture_size	
Inherited from kivy.uix.widget.Widget __events__, canvas, center, center_x, center_y, children, cls, disabled, height, id, ids, opacity, parent, pos, pos_hint, right, size, size_hint, size_hint_x, size_hint_y, top, width, x, y	

4.7 Class MapLayer



Known Subclasses: mapview.view.MarkerMapLayer, isySUR.gui.mapview.geojson.GeoJsonMapLayer

A map layer, that is repositionned everytime the :class:‘MapView‘ is moved.

4.7.1 Methods

reposition(*self*)

Function called when :class:‘MapView‘ is moved. You must recalculate the position of your children.

unload(*self*)

Called when the view want to completly unload the layer.

Inherited from kivy.uix.widget.Widget

__eq__(), __hash__(), __init__(), add_widget(), clear_widgets(), collide_point(), collide_widget(), get_center_x(), get_center_y(), get_parent_window(), get_right(), get_root_window(), get_top(), on_disabled(), on_opacity(), on_touch_down(), on_touch_move(), on_touch_up(), remove_widget(), set_center_x(), set_center_y(), set_right(), set_top(), to_local(), to_parent(), to_widget(), to_window()

Inherited from kivy._event.EventDispatcher

__new__(), bind(), create_property(), dispatch(), events(), get_property_observers(), getter(), is_event_type(), properties(), property(), register_event_type(), setter(), unbind(), unregister_event_types()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

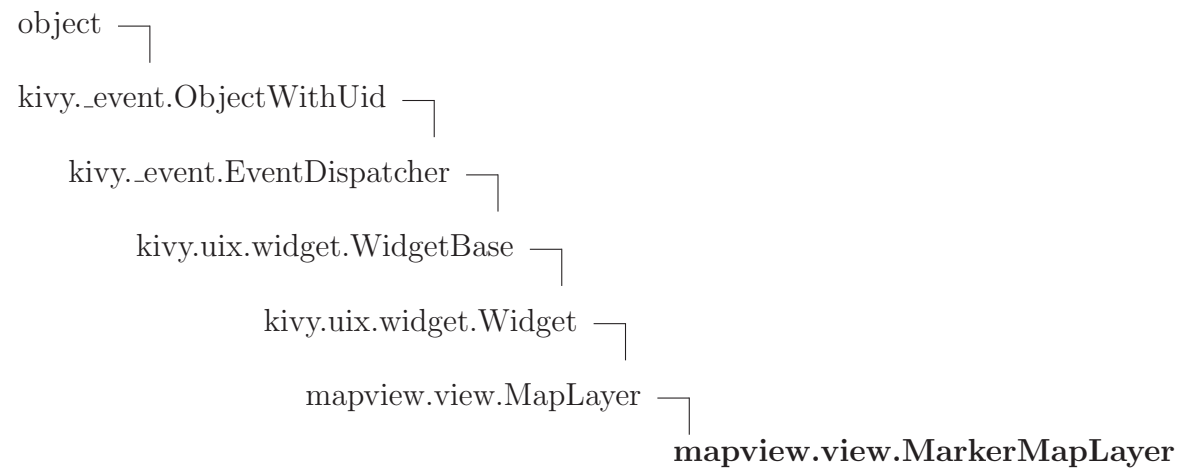
4.7.2 Properties

Name	Description
Inherited from kivy.uix.widget.Widget __self__, proxy_ref	
Inherited from kivy._event.ObjectWithUid uid	
Inherited from object __class__	

4.7.3 Class Variables

Name	Description
viewport_x	**Value:** <kivy.properties.NumericProperty object at 0x0000000003AC...
viewport_y	**Value:** <kivy.properties.NumericProperty object at 0x0000000003AC...
Inherited from kivy.uix.widget.Widget __events__, canvas, center, center_x, center_y, children, cls, disabled, height, id, ids, opacity, parent, pos, pos_hint, right, size, size_hint, size_hint_x, size_hint_y, top, width, x, y	

4.8 Class MarkerMapLayer



A map layer for :class:‘MapMarker‘

4.8.1 Methods

__init__(*self*, ***kwargs*)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

add_widget(*self*, *marker*)

This function adds a marker to the MapLayer.

Parameters

`marker`: The marker to be added.

Overrides: kivy.uix.widget.Widget.add_widget

remove_widget(*self*, *marker*)

This function removes a marker to the MapLayer.

Parameters

`marker`: The marker to be removed.

Overrides: kivy.uix.widget.Widget.remove_widget

reposition(*self*)

This function recalculates the position of all markers on the current Layer, adds new marker if they are visible now and removes marker which not visibile anymore.

Overrides: mapview.view.MapLayer.reposition

set_marker_position(*self*, *mapview*, *marker*)

This function sets the marker position in respect to the current mapview.

Parameters

`mapview:` The current mapview object.
(type=view.MarkerMapLayer)

`marker:` The marker for which the position should be set.
(type=view.Marker)

unload(*self*)

This function deletes all widgets on the current layer.

Overrides: mapview.view.MapLayer.unload

Inherited from kivy.uix.widget.Widget

__eq__(), __hash__(), clear_widgets(), collide_point(), collide_widget(), get_center_x(), get_center_y(), get_parent_window(), get_right(), get_root_window(), get_top(), on_disabled(), on_opacity(), on_touch_down(), on_touch_move(), on_touch_up(), set_center_x(), set_center_y(), set_right(), set_top(), to_local(), to_parent(), to_widget(), to_window()

Inherited from kivy._event.EventDispatcher

__new__(), bind(), create_property(), dispatch(), events(), get_property_observers(), getter(), is_event_type(), properties(), property(), register_event_type(), setter(), unbind(), unregister_event_types()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

4.8.2 Properties

Name	Description
Inherited from kivy.uix.widget.Widget	
__self__, proxy_ref	

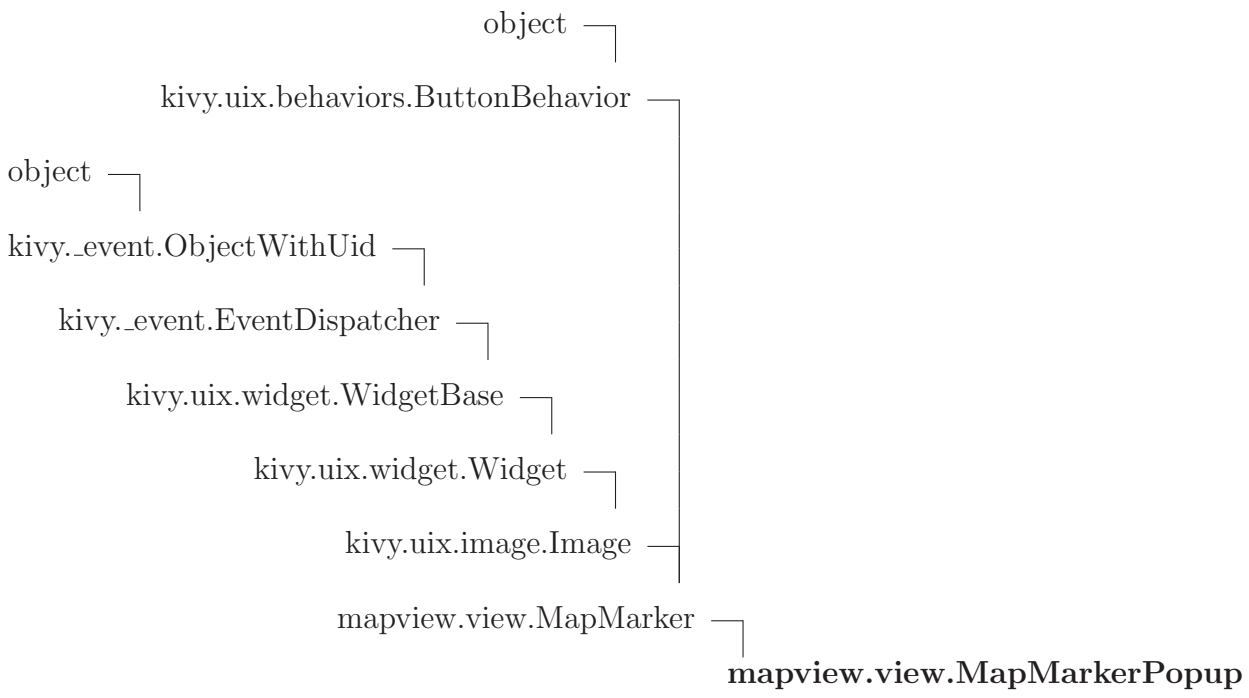
continued on next page

Name	Description
Inherited from kivy._event.ObjectWithUid	
uid	
Inherited from object	
__class__	

4.8.3 Class Variables

Name	Description
Inherited from mapview.view.MapLayer (Section 4.7)	
viewport_x, viewport_y	
Inherited from kivy.uix.widget.Widget	
__events__, canvas, center, center_x, center_y, children, cls, disabled, height, id, ids, opacity, parent, pos, pos_hint, right, size, size_hint, size_hint_x, size_hint_y, top, width, x, y	

4.9 Class MapMarkerPopup



4.9.1 Methods

add_widget(*self*, *widget*)

This function adds a widget to the gui.

Parameters

`widget`: The widget to be added.

(type=kivy.uix.widget.Widget)

Overrides: kivy.uix.widget.Widget.add_widget

on_is_open(*self*, **args*)

on_release Eventhandler

Parameters

`*args`: Eventobject

on_release(*self*, **args*)

on_release Eventhandler

Parameters

`*args`: Eventobject

Overrides: kivy.uix.behaviors.ButtonBehavior.on_release

refresh_open_status(*self*)

This function refreshes the open status from the gui elementes.

remove_widget(*self*, *widget*)

This function removes a widget to the gui.

Parameters

`widget`: The widget to be removed.

(type=kivy.uix.widget.Widget)

Overrides: kivy.uix.widget.Widget.remove_widget

Inherited from mapview.view.MapMarker(Section 4.6)

__init__()

Inherited from kivy.uix.behaviors.ButtonBehavior

on_press(), on_touch_down(), on_touch_move(), on_touch_up(), trigger_action()

Inherited from kivy.uix.image.Image

get_image_ratio(), get_norm_image_size(), on_anim_delay(), on_nocache(), on_texture(), reload(), texture_update()

Inherited from kivy.uix.widget.Widget

__eq__(), __hash__(), clear_widgets(), collide_point(), collide_widget(), get_center_x(), get_center_y(), get_parent_window(), get_right(), get_root_window(), get_top(), on_disabled(), on_opacity(), set_center_x(), set_center_y(), set_right(), set_top(), to_local(), to_parent(), to_widget(), to_window()

Inherited from kivy._event.EventDispatcher

__new__(), bind(), create_property(), dispatch(), events(), get_property_observers(), getter(), is_event_type(), properties(), property(), register_event_type(), setter(), unbind(), unregister_event_types()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

4.9.2 Properties

Name	Description
Inherited from mapview.view.MapMarker (Section 4.6) default_marker_fn	
Inherited from kivy.uix.widget.Widget __self__, proxy_ref	
Inherited from kivy._event.ObjectWithUid uid	
Inherited from object __class__	

4.9.3 Class Variables

Name	Description
is_open	**Value:** <kivy.properties.BooleanProperty object at 0x0000000003AC...
placeholder	**Value:** <kivy.properties.ObjectProperty object at 0x0000000003B08...
popup_size	**Value:** <kivy.properties.ListProperty object at 0x0000000003AC0DC8>
Inherited from mapview.view.MapMarker (Section 4.6) anchor_x, anchor_y, lat, lon, visible	
Inherited from kivy.uix.behaviors.ButtonBehavior	

continued on next page

Name	Description
last_touch, state	
Inherited from kivy.uix.image.Image allow_stretch, anim_delay, color, image_ratio, keep_data, keep_ratio, mipmap, nocache, norm_image_size, source, texture, texture_size	
Inherited from kivy.uix.widget.Widget __events__, canvas, center, center_x, center_y, children, cls, disabled, height, id, ids, opacity, parent, pos, pos_hint, right, size, size_hint, size_hint_x, size_hint_y, top, width, x, y	

5 Module isySUR.gui.mapview.downloader

5.1 Class Downloader

object
└ **isySUR.gui.mapview.downloader.Downloader**

5.1.1 Methods

instance()

__**init**__(*self*, *max_workers*=5, *cap_time*=0.064)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

submit(*self*, *f*, **args*, ***kwargs*)

download_tile(*self*, *tile*)

download(*self*, *url*, *callback*, ***kwargs*)

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

5.1.2 Properties

Name	Description
Inherited from object	
__class__	

6 Module isySUR.gui.mapview.geojson

(section) Geojson layer

.. note:

```
Currently experimental and a work in progress. It requires the new
Kivy's Tesselator, based on libtess2. See
'tesselator branch <https://github.com/kivy/kivy/tree/tesselator>'_
```

6.1 Class GeoJsonMapLayer

object
└ kivy._event.ObjectWithUid
└ kivy._event.EventDispatcher
└ kivy.uix.widget.WidgetBase
└ kivy.uix.widget.Widget
└ mapview.view.MapLayer
└ **isySUR.gui.mapview.geojson.GeoJsonMapLayer**

6.1.1 Methods

reposition(*self*)

Function called when :class:'MapView' is moved. You must recalculate the position of your children.

Overrides: mapview.view.MapLayer.reposition extit(inherited documentation)

on_geojson(*self*, *instance*, *geojson*)

on_source(*self*, *instance*, *value*)

Inherited from mapview.view.MapLayer(Section 4.7)

unload()

Inherited from kivy.uix.widget.Widget

__eq__(), __hash__(), __init__(), add_widget(), clear_widgets(), collide_point(), collide_widget(), get_center_x(), get_center_y(), get_parent_window(), get_right(), get_root_window(), get_top(), on_disabled(), on_opacity(), on_touch_down(), on_touch_move(), on_touch_up(), remove_widget(), set_center_x(), set_center_y(), set_right(), set_top(), to_local(), to_parent(), to_widget(), to_window()

Inherited from kivy._event.EventDispatcher

__new__(), bind(), create_property(), dispatch(), events(), get_property_observers(), getter(), is_event_type(), properties(), property(), register_event_type(), setter(), unbind(), unregister_event_types()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

6.1.2 Properties

Name	Description
Inherited from kivy.uix.widget.Widget __self__, proxy_ref	
Inherited from kivy._event.ObjectWithUid uid	
Inherited from object __class__	

6.1.3 Class Variables

Name	Description
source	**Value:** `StringProperty()`
geojson	**Value:** `ObjectProperty()`
Inherited from mapview.view.MapLayer (Section 4.7) viewport_x, viewport_y	
Inherited from kivy.uix.widget.Widget __events__, canvas, center, center_x, center_y, children, cls, disabled, height, id, ids, opacity, parent, pos, pos_hint, right, size, size_hint, size_hint_x, size_hint_y, top, width, x, y	

7 Module isySUR.gui.mapview.mbtsource

(section) MBTiles provider for MapView

This provider is based on .mbfiles from MapBox. See: http://mbtiles.org/

7.1 Class MBTilesMapSource

object
mapview.source.MapSource
isySUR.gui.mapview.mbtsource.MBTilesMapSource

7.1.1 Methods

__init__(*self*, *filename*)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

fill_tile(*self*, *tile*)

Add this tile to load within the downloader

Overrides: mapview.source.MapSource.fill_tile extit(inherited documentation)

Inherited from mapview.source.MapSource(Section 4.4)

from_provider(), get_col_count(), get_lat(), get_lon(), get_max_zoom(), get_min_zoom(), get_row_count(), get_x(), get_y()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

7.1.2 Properties

Name	Description
Inherited from object	
__class__	

7.1.3 Class Variables

Name	Description
Inherited from mapview.source.MapSource (Section 4.4)	
providers	

8 Module isySUR.gui.mapview.source

8.1 Class MapSource

object
└ **isySUR.gui.mapview.source.MapSource**

Base class for implementing a map source / provider

8.1.1 Methods

__init__(*self*,
url=`'http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png'`,
cache_key=`None`, *min_zoom*=`0`, *max_zoom*=`19`, *tile_size*=`256`,
image_ext=`'png'`, *attribution*=`'\xc2\xa9 OpenStreetMap contributors'`,
subdomains=`'abc'`)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

from_provider(*key*)

get_x(*self*, *zoom*, *lon*)

Get the x position on the map using this map source's projection (0, 0) is located at the top left.

get_y(*self*, *zoom*, *lat*)

Get the y position on the map using this map source's projection (0, 0) is located at the top left.

get_lon(*self*, *zoom*, *x*)

Get the longitude to the x position in the map source's projection

get_lat(*self*, *zoom*, *y*)

Get the latitude to the y position in the map source's projection

get_row_count(*self*, *zoom*)

Get the number of tiles in a row at this zoom level

get_col_count(*self*, *zoom*)

Get the number of tiles in a col at this zoom level

get_min_zoom(*self*)

Return the minimum zoom of this source

get_max_zoom(*self*)

Return the maximum zoom of this source

fill_tile(*self*, *tile*)

Add this tile to load within the downloader

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

8.1.2 Properties

Name	Description
Inherited from object	
__class__	

8.1.3 Class Variables

Name	Description
providers	**Value:** {'cyclemap': (0, 0, 17, 'http://{s}.tile.opencyclemap.org...

9 Module isySUR.gui.mapview.types

9.1 Class Coordinate

object
  tuple
    isySUR.gui.mapview.types.Coordinate

Coordinate(lat, lon)

9.1.1 Methods

__getnewargs__(*self*)

Return self as a plain tuple. Used by copy and pickle.

Overrides: tuple.__getnewargs__

__getstate__(*self*)

Exclude the OrderedDict from pickling

__new__(*_cls*, *lat*, *lon*)

Create new instance of Coordinate(lat, lon)

Return Value

a new object with type S, a subtype of T

Overrides: object.__new__

__repr__(*self*)

Return a nicely formatted representation string

Overrides: object.__repr__

Inherited from tuple

__add__(), __contains__(), __eq__(), __ge__(), __getattribute__(), __getitem__(), __getslice__(), __gt__(), __hash__(), __iter__(), __le__(), __len__(), __lt__(), __mul__(), __ne__(), __rmul__(), __sizeof__(), count(), index()

Inherited from object

__delattr__(), __format__(), __init__(), __reduce__(), __reduce_ex__(), __setattr__(), __str__(), __subclasshook__()

9.1.2 Properties

Name	Description
lat	Alias for field number 0
lon	Alias for field number 1
Inherited from object __class__	

9.2 Class Bbox

9.2.1 Methods

collide(*self*, **args*)

Inherited from tuple

__add__(), __contains__(), __eq__(), __ge__(), __getattribute__(), __getitem__(), __getnewargs__(), __getslice__(), __gt__(), __hash__(), __iter__(), __le__(), __len__(), __lt__(), __mul__(), __ne__(), __new__(), __repr__(), __rmul__(), __sizeof__(), count(), index()

Inherited from object

__delattr__(), __format__(), __init__(), __reduce__(), __reduce_ex__(), __setattr__(), __str__(), __subclasshook__()

9.2.2 Properties

Name	Description
Inherited from object __class__	

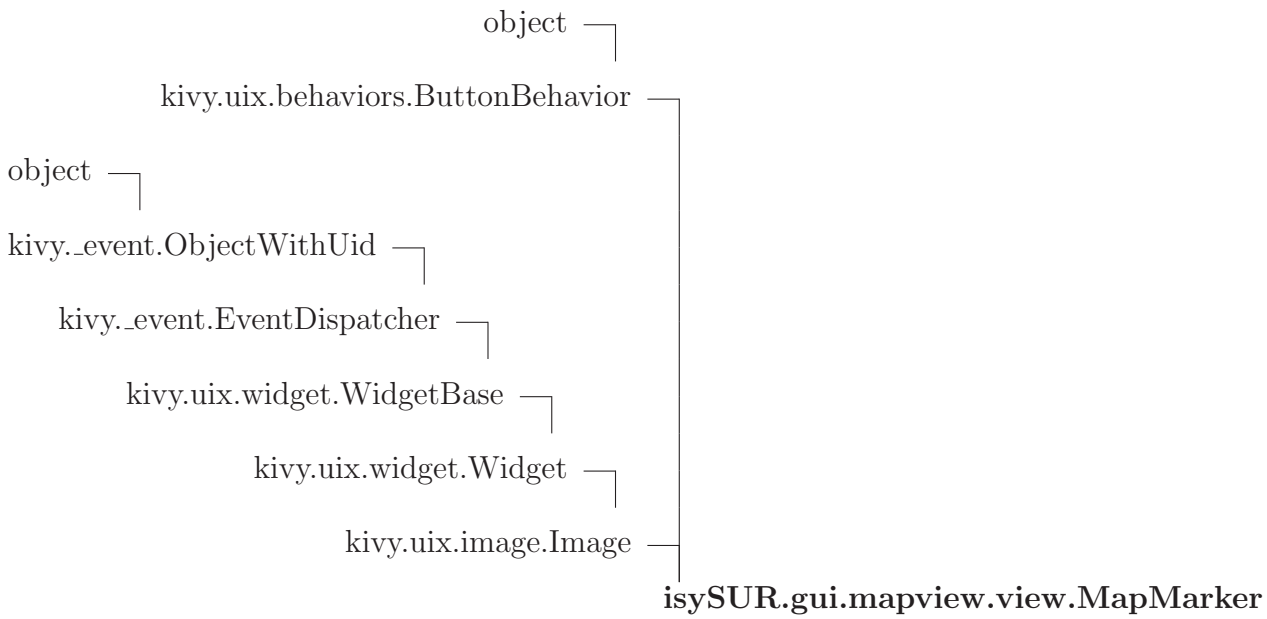
10 Module isySUR.gui.mapview.utils

10.1 Functions

clamp(*x*, *minimum*, *maximum*)

11 Module isySUR.gui.mapview.view

11.1 Class MapMarker



Known Subclasses: isySUR.gui.mapview.view.MapMarkerPopup

A marker on a map, that must be used on a :class:‘MapMarker‘

11.1.1 Methods

__init__(*self*, ***kwargs*)
x.__init__(...) initializes x; see help(type(x)) for signature
Overrides: object.__init__ extit(inherited documentation)

Inherited from kivy.uix.behaviors.ButtonBehavior

on_press(), on_release(), on_touch_down(), on_touch_move(), on_touch_up(), trigger_action()

Inherited from kivy.uix.image.Image

get_image_ratio(), get_norm_image_size(), on_anim_delay(), on_nocache(), on_texture(), reload(), texture_update()

Inherited from kivy.uix.widget.Widget

__eq__(), __hash__(), add_widget(), clear_widgets(), collide_point(), collide_widget(), get_center_x(), get_center_y(), get_parent_window(), get_right(), get_root_window(), get_top(), on_disabled(), on_opacity(), remove_widget(), set_center_x(), set_center_y(), set_right(), set_top(), to_local(), to_parent(), to_widget(), to_window()

Inherited from kivy._event.EventDispatcher

__new__(), bind(), create_property(), dispatch(), events(), get_property_observers(), getter(), is_event_type(), properties(), property(), register_event_type(), setter(), unbind(), unregister_event_types()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

11.1.2 Properties

Name	Description
default_marker_fn	*(type=str)*
Inherited from kivy.uix.widget.Widget __self__, proxy_ref	
Inherited from kivy._event.ObjectWithUid uid	
Inherited from object __class__	

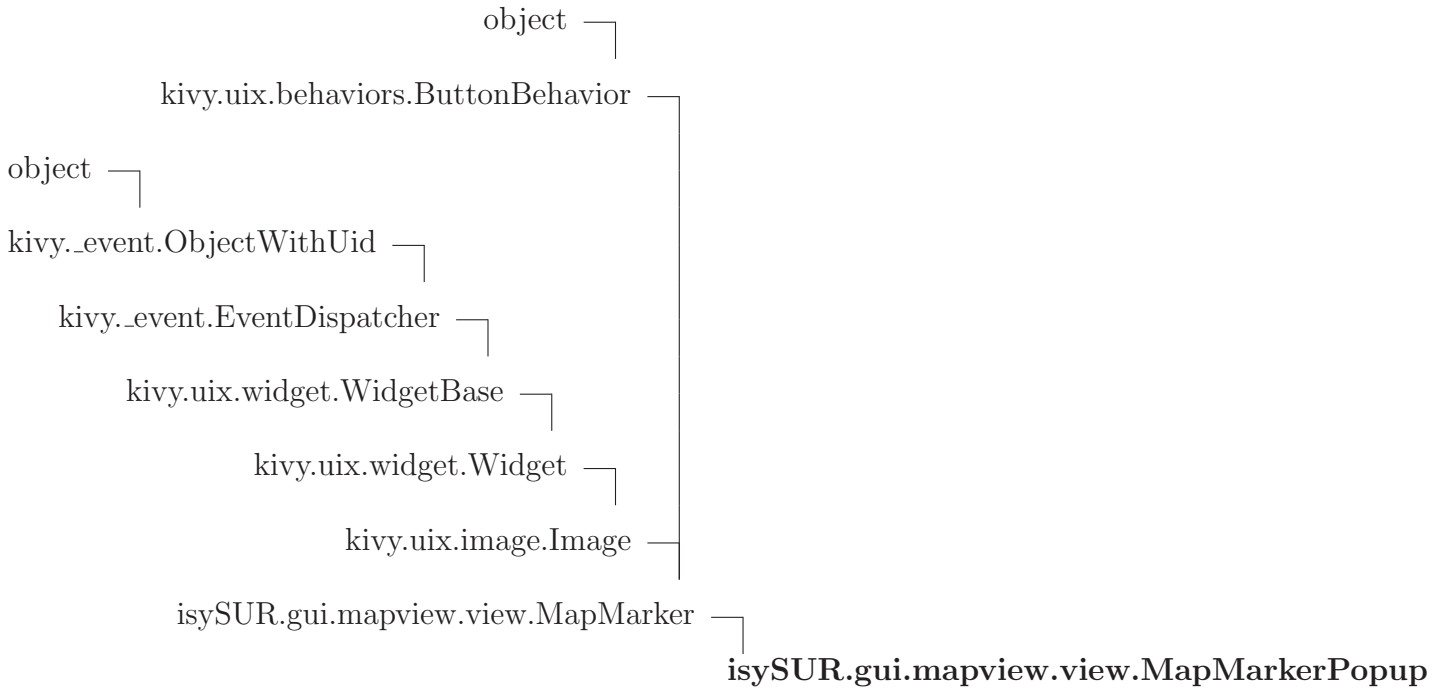
11.1.3 Class Variables

Name	Description
anchor_x	Anchor of the marker on the X axis. Defaults to 0.5, mean the anchor will be at the X center of the image. **Value:** `NumericProperty(0.5)`
anchor_y	Anchor of the marker on the Y axis. Defaults to 0, mean the anchor will be at the Y bottom of the image. **Value:** `NumericProperty(0)`
lat	Latitude of the marker **Value:** `NumericProperty(0)`
lon	Longitude of the marker **Value:** `NumericProperty(0)`
visible	**Value:** `NumericProperty(1)`
Inherited from kivy.uix.behaviors.ButtonBehavior	

continued on next page

Name	Description
last_touch, state	
Inherited from kivy.uix.image.Image allow_stretch, anim_delay, color, image_ratio, keep_data, keep_ratio, mipmap, nocache, norm_image_size, source, texture, texture_size	
Inherited from kivy.uix.widget.Widget __events__, canvas, center, center_x, center_y, children, cls, disabled, height, id, ids, opacity, parent, pos, pos_hint, right, size, size_hint, size_hint_x, size_hint_y, top, width, x, y	

11.2 Class MapMarkerPopup



11.2.1 Methods

add_widget(*self*, *widget*)

This function adds a widget to the gui.

Parameters

`widget`: The widget to be added.

(type=kivy.uix.widget.Widget)

Overrides: kivy.uix.widget.Widget.add_widget

remove_widget(*self*, *widget*)

This function removes a widget to the gui.

Parameters

`widget`: The widget to be removed.

(type=kivy.uix.widget.Widget)

Overrides: kivy.uix.widget.Widget.remove_widget

on_is_open(*self*, **args*)

on_release Eventhandler

Parameters

`*args`: Eventobject

on_release(*self*, **args*)

on_release Eventhandler

Parameters

`*args`: Eventobject

Overrides: kivy.uix.behaviors.ButtonBehavior.on_release

refresh_open_status(*self*)

This function refreshes the open status from the gui elementes.

Inherited from isySUR.gui.mapview.view.MapMarker(Section 11.1)

__init__()

Inherited from kivy.uix.behaviors.ButtonBehavior

on_press(), on_touch_down(), on_touch_move(), on_touch_up(), trigger_action()

Inherited from kivy.uix.image.Image

get_image_ratio(), get_norm_image_size(), on_anim_delay(), on_nocache(), on_texture(), reload(), texture_update()

Inherited from kivy.uix.widget.Widget

__eq__(), __hash__(), clear_widgets(), collide_point(), collide_widget(), get_center_x(), get_center_y(), get_parent_window(), get_right(), get_root_window(), get_top(), on_disabled(), on_opacity(), set_center_x(), set_center_y(), set_right(), set_top(), to_local(), to_parent(), to_widget(), to_window()

Inherited from kivy._event.EventDispatcher

__new__(), bind(), create_property(), dispatch(), events(), get_property_observers(), getter(), is_event_type(), properties(), property(), register_event_type(), setter(), unbind(), unregister_event_types()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

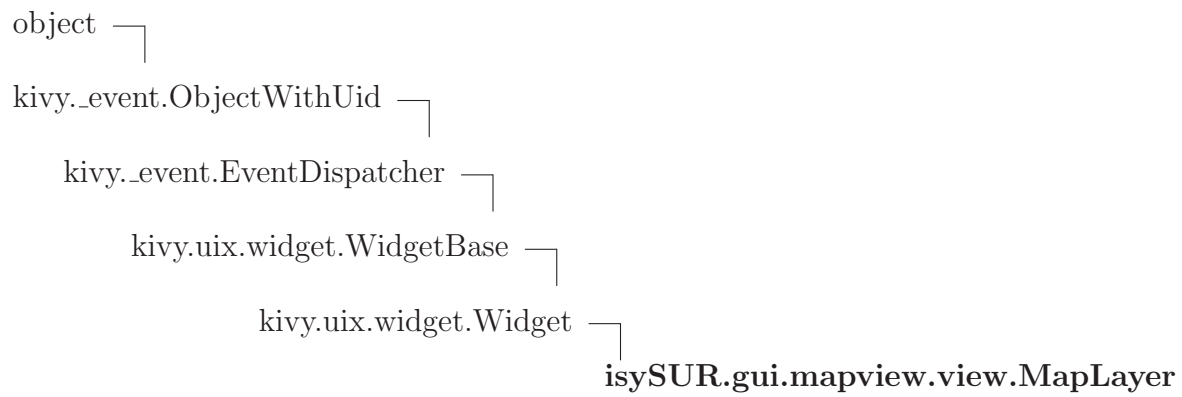
11.2.2 Properties

Name	Description
Inherited from isySUR.gui.mapview.view.MapMarker (Section 11.1) default_marker_fn	
Inherited from kivy.uix.widget.Widget __self__, proxy_ref	
Inherited from kivy._event.ObjectWithUid uid	
Inherited from object __class__	

11.2.3 Class Variables

Name	Description
is_open	**Value:** `BooleanProperty(False)`
placeholder	**Value:** `ObjectProperty(None)`
popup_size	**Value:** `ListProperty([100, 100])`
Inherited from isySUR.gui.mapview.view.MapMarker (Section 11.1) anchor_x, anchor_y, lat, lon, visible	
Inherited from kivy.uix.behaviors.ButtonBehavior last_touch, state	
Inherited from kivy.uix.image.Image allow_stretch, anim_delay, color, image_ratio, keep_data, keep_ratio, mipmap, nocache, norm_image_size, source, texture, texture_size	
Inherited from kivy.uix.widget.Widget __events__, canvas, center, center_x, center_y, children, cls, disabled, height, id, ids, opacity, parent, pos, pos_hint, right, size, size_hint, size_hint_x, size_hint_y, top, width, x, y	

11.3 Class MapLayer



Known Subclasses: isySUR.gui.mapview.view.MarkerMapLayer, isySUR.gui.mapview.view.PolyMapLay

A map layer, that is repositionned everytime the :class:‘MapView‘ is moved.

11.3.1 Methods

reposition(*self*)

Function called when :class:‘MapView‘ is moved. You must recalculate the position of your children.

unload(*self*)

Called when the view want to completly unload the layer.

Inherited from kivy.uix.widget.Widget

__eq__(), __hash__(), __init__(), add_widget(), clear_widgets(), collide_point(), collide_widget(), get_center_x(), get_center_y(), get_parent_window(), get_right(), get_root_window(), get_top(), on_disabled(), on_opacity(), on_touch_down(), on_touch_move(), on_touch_up(), remove_widget(), set_center_x(), set_center_y(), set_right(), set_top(), to_local(), to_parent(), to_widget(), to_window()

Inherited from kivy._event.EventDispatcher

__new__(), bind(), create_property(), dispatch(), events(), get_property_observers(), getter(), is_event_type(), properties(), property(), register_event_type(), setter(), unbind(), unregister_event_types()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

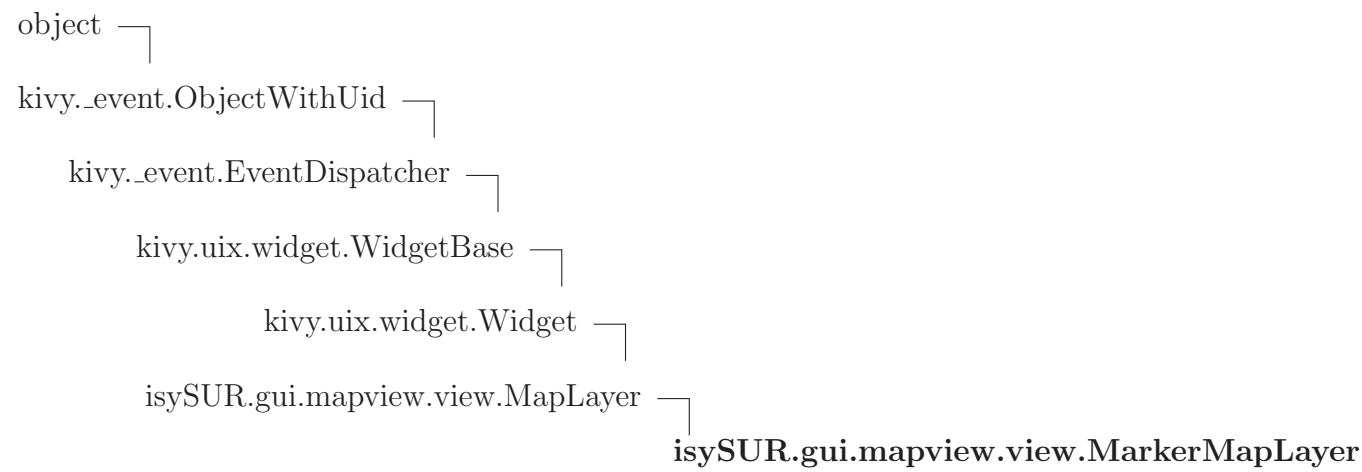
11.3.2 Properties

Name	Description
Inherited from kivy.uix.widget.Widget __self__, proxy_ref	
Inherited from kivy._event.ObjectWithUid uid	
Inherited from object __class__	

11.3.3 Class Variables

Name	Description
viewport_x	**Value:** `NumericProperty(0)`
viewport_y	**Value:** `NumericProperty(0)`
Inherited from kivy.uix.widget.Widget __events__, canvas, center, center_x, center_y, children, cls, disabled, height, id, ids, opacity, parent, pos, pos_hint, right, size, size_hint, size_hint_x, size_hint_y, top, width, x, y	

11.4 Class MarkerMapLayer



A map layer for :class:‘MapMarker‘

11.4.1 Methods

__init__(*self*, ***kwargs*)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

add_widget(*self*, *marker*)

This function adds a marker to the MapLayer.

Parameters

`marker:` The marker to be added.

Overrides: kivy.uix.widget.Widget.add_widget

remove_widget(*self*, *marker*)

This function removes a marker to the MapLayer.

Parameters

`marker:` The marker to be removed.

Overrides: kivy.uix.widget.Widget.remove_widget

reposition(*self*)

This function recalculates the position of all markers on the current Layer, adds new marker if they are visible now and removes marker which not visibile anymore.

Overrides: isySUR.gui.mapview.view.MapLayer.reposition

set_marker_position(*self*, *mapview*, *marker*)

This function sets the marker position in respect to the current mapview.

Parameters

`mapview:` The current mapview object.

(type=view.MarkerMapLayer)

`marker:` The marker for which the position should be set.

(type=view.Marker)

unload(*self*)

This function deletes all widgets on the current layer.

Overrides: isySUR.gui.mapview.view.MapLayer.unload

Inherited from kivy.uix.widget.Widget

__eq__(), __hash__(), clear_widgets(), collide_point(), collide_widget(), get_center_x(), get_center_y(), get_parent_window(), get_right(), get_root_window(), get_top(), on_disabled(), on_opacity(), on_touch_down(), on_touch_move(), on_touch_up(), set_center_x(), set_center_y(), set_right(), set_top(), to_local(), to_parent(), to_widget(), to_window()

Inherited from kivy._event.EventDispatcher

__new__(), bind(), create_property(), dispatch(), events(), get_property_observers(), getter(), is_event_type(), properties(), property(), register_event_type(), setter(), unbind(), unregister_event_types()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

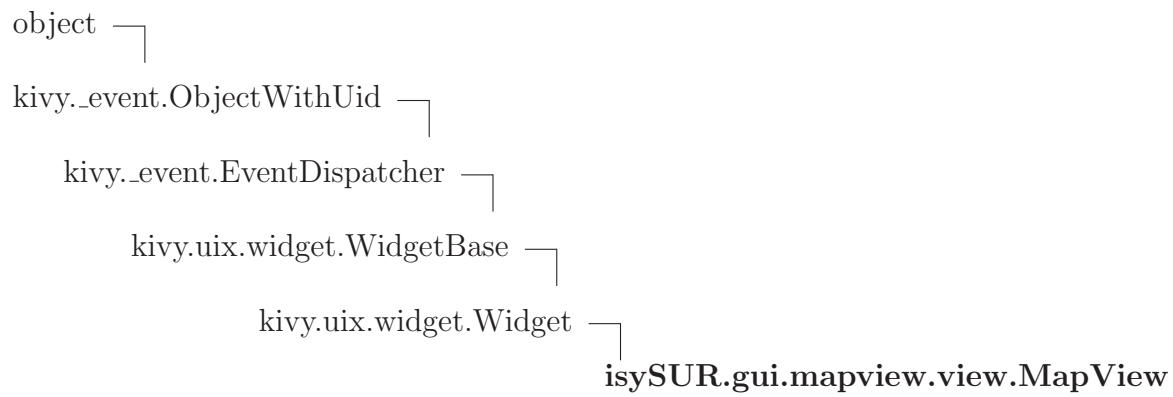
11.4.2 Properties

Name	Description
Inherited from kivy.uix.widget.Widget __self__, proxy_ref	
Inherited from kivy._event.ObjectWithUid uid	
Inherited from object __class__	

11.4.3 Class Variables

Name	Description
Inherited from isySUR.gui.mapview.view.MapLayer (Section 11.3) viewport_x, viewport_y	
Inherited from kivy.uix.widget.Widget __events__, canvas, center, center_x, center_y, children, cls, disabled, height, id, ids, opacity, parent, pos, pos_hint, right, size, size_hint, size_hint_x, size_hint_y, top, width, x, y	

11.5 Class MapView



MapView is the widget that control the map displaying, navigation, and layers management.

11.5.1 Methods

get_bbox(*self*, *margin*=0)

Returns the bounding box from the bottom/left (lat1, lon1) to top/right (lat2, lon2).

Parameters

`margin`: (Optional) addition margin for the boundingbox.

Return Value

The boundingbox. @rtype mapview.Bbox

unload(*self*)

Unload the view and all the layers. It also cancel all the remaining downloads.

get_window_xy_from(*self*, *lat*, *lon*, *zoom*)

Returns the x/y position in the widget absolute coordinates from a lat/lon.

Return Value

The x/y position in the widget.

(type=float,float)

center_on(*self*, **args*)

Center the map on the coordinate :class:‘Coordinate‘, or a (lat, lon.

set_zoom_at(*self*, *zoom*, *x*, *y*, *scale*=`None`)

Sets the zoom level, leaving the (x, y) at the exact same point in the view.

Parameters

`zoom`: tThe zoom level.

`type`: int

`x`: The x-coordinate of the point.

(type=float)

`y`: The y-coordinate of the point.

(type=float)

`scale`: (Optinal) the scalefaktor for the scatter.

(type=int)

zoom_to(*self*, *lat*, *lon*, *zoom*)

Zooms to the given zoom level at the given position.

Parameters

`lat`: Lat-coordinate of the given position.

(type=float)

`lon`: Lon-coordinate of the given position.

(type=float)

`zoom`: Zoom-factor.

(type=int)

zoom_to_Polygon(*self*, *name*, *zoom*)

Zooms to the given zoom level at the given polygon. The zoom parameter is ignored if the user zoomed in more already and the entire polygon is aready visible.

Parameters

`name`: Name of the polygon.

(type=str)

`zoom`: *(type=int)*

on_zoom(*self*, *instance*, *zoom*)

on_zoom Event handler

get_latlon_at(*self*, *x*, *y*, *zoom*=`None`)

Return the current :class:‘Coordinate‘ within the (x, y) widget coordinate.

Parameters

`x:` The x-coordinate of the point.

(type=float)

`y:` The y-coordinate of the point.

(type=float)

Return Value

The current Coordinate within the (x,y) widget coordinate.

(type=mapview.Coordinate)

add_marker(*self*, *marker*, *layer*=`None`)

Add a marker onto the layer. If layer is None, it will be added in the default marker layer. If there is no default marker layer, a new one will be automatically created.

Parameters

`marker:` The marker to be added.

(type=view.MapMarker)

`layer:` (Optional) the layer the marker should be added to.

(type=view.MarkerMapLayer)

drawPolygon(*self*)

Draws a Polygon onto the Map.

isPolyInView(*self*, *name*)

This function proves if a polygon is in the current viewspace.

Parameters

`name:` The name of the polygon.

(type=str)

Return Value

True if in viewspace otherwise False.

(type=boolean)

isPolyVisible(*self*, *name*)

This function proves if a polygon is completely visible in viewspace.

Parameters

`name:` The name of the polygon.

(type=str)

Return Value

True if completely in viewspace otherwise False.

(type=boolean)

addPolygon(*self*, *name*, *polygon*, *color*, *markerCoords*)

Adds and draws a new polygon onto the map.

Parameters

`name:` Name of the polygon to be added.

(type=str)

`polygon:` List of vertices of the polygon.

(type=[(float, float)])

`color:` Style value of KML

(type=dict)

`markerCoords:` Coordinates of the SUR.

(type=Tuple(float, float))

hideMarkers(*self*)

Hides all markers on the Marker Layer.

showMarkers(*self*)

Shows all markers.

getBBoxOfPolygon(*self*, *polygon*)

This function calculates a boundingbox in respect to the given polygon.

Parameters

`polygon:` The polygon for which the bbox should be computed, e.g. [(1.0,2.0),(2.0,1.0),(1.0,1.0),(1.0,2.0)].

Return Value

The calculated bbox, e.g. [minLat,minLon, maxLat,maxLon].

(type=[float,float,float,float])

showPolygon(*self*, *name*)

Makes a polygon visible on the Map.

Parameters

`name`: Name of the polygon to be shown.

(type=str)

hidePolygon(*self*, *name*)

Removes a polygon from the Map.

Parameters

`name`: Name of the polygon to be removed.

(type=str)

convertKMLColor(*self*, *kmlColor*)

Convert a KML Color to its rgba value between 0 and 1.

Parameters

`kmlColor`: Color to be converted.

(type=str)

Return Value

Returns the rgba values of kmlColor.

remove_marker(*self*, *marker*)

Remove a marker from its layer.

add_layer(*self*, *layer*, *mode='window'*)

Add a new layer to update at the same time the base tile layer. mode can be either "scatter" or "window". If "scatter", it means the layer will be within the scatter transformation. It's perfect if you want to display path / shape, but not for text. If "window", it will have no transformation. You need to position the widget yourself: think as Z-sprite / billboard. Defaults to "window".

Parameters

`layer`: The layer for updating.

(type=kivy.uix.widget.Widget)

`mode`: (Optional) The mode for updating could be "scatter" or "window".

(type=str)

remove_layer(*self*, *layer*)

Remove the layer.

Parameters

`layer:` The layer to be removed. @type kivy.uix.widget.Widget

sync_to(*self*, *other*)

Reflect the lat/lon/zoom of the other MapView to the current one.

__init__(*self*, ***kwargs*)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

add_widget(*self*, *widget*)

```
Add a new widget as a child of this widget.

:Parameters:
    `widget`: :class:`Widget`
        Widget to add to our list of children.
    `index`: int, defaults to 0
        *(this attribute was added in 1.0.5)*
        Index to insert the widget in the list

>>> from kivy.uix.button import Button
>>> from kivy.uix.slider import Slider
>>> root = Widget()
>>> root.add_widget(Button())
>>> slider = Slider()
>>> root.add_widget(slider)
```

Overrides: kivy.uix.widget.Widget.add_widget extit(inherited documentation)

remove_widget(*self*, *widget*)

```
Remove a widget from the children of this widget.

:Parameters:
    `widget`: :class:`Widget`
        Widget to remove from our children list.

>>> from kivy.uix.button import Button
>>> root = Widget()
>>> button = Button()
>>> root.add_widget(button)
>>> root.remove_widget(button)
```

Overrides: kivy.uix.widget.Widget.remove_widget extit(inherited documentation)

on_map_relocated(*self*, *zoom*, *coord*)

animated_diff_scale_at(*self*, *d*, *x*, *y*)

diff_scale_at(*self*, *d*, *x*, *y*)

scale_at(*self*, *scale*, *x*, *y*)

on_touch_down(*self*, *touch*)

```
Receive a touch down event.

:Parameters:
    `touch`: :class:`~kivy.input.motionevent.MotionEvent` class
        Touch received. The touch is in parent coordinates. See
        :mod:`~kivy.uix.relativelayout` for a discussion on
        coordinate systems.

:Returns:
    bool. If True, the dispatching of the touch event will stop.
```

Overrides: kivy.uix.widget.Widget.on_touch_down extit(inherited documentation)

on_transform(*self*, **args*)

trigger_update(*self*, *full*)

do_update(*self*, *dt*)

bbox_for_zoom(*self*, *vx*, *vy*, *w*, *h*, *zoom*)

load_visible_tiles(*self*)

load_tile(*self*, *x*, *y*, *size*, *zoom*)

load_tile_for_source(*self*, *map_source*, *opacity*, *size*, *x*, *y*, *zoom*)

move_tiles_to_background(*self*)

remove_all_tiles(*self*)

tile_map_set(*self*, *tile_x*, *tile_y*, *value*)

tile_in_tile_map(*self*, *tile_x*, *tile_y*)

on_size(*self*, *instance*, *size*)

on_pos(*self*, *instance*, *pos*)

on_map_source(*self*, *instance*, *source*)

cleanUpCache(*self*)

Inherited from kivy.uix.widget.Widget

__eq__(), __hash__(), clear_widgets(), collide_point(), collide_widget(), get_center_x(), get_center_y(), get_parent_window(), get_right(), get_root_window(), get_top(), on_disabled(), on_opacity(), on_touch_move(), on_touch_up(), set_center_x(), set_center_y(), set_right(), set_top(), to_local(), to_parent(), to_widget(), to_window()

Inherited from kivy._event.EventDispatcher

__new__(), bind(), create_property(), dispatch(), events(), get_property_observers(), getter(), is_event_type(), properties(), property(), register_event_type(), setter(), unbind(), unregister_event_types()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

11.5.2 Properties

Name	Description
viewport_pos	Returns the current viewport position.
scale	Returns the current scalefaktor. *(type=float)*
Inherited from kivy.uix.widget.Widget __self__, proxy_ref	
Inherited from kivy._event.ObjectWithUid uid	
Inherited from object __class__	

11.5.3 Class Variables

Name	Description
lon	Longitude at the center of the widget **Value:** `NumericProperty()`
lat	Latitude at the center of the widget **Value:** `NumericProperty()`
zoom	Zoom of the widget. Must be between :meth:‘MapSource.get_min_zoom‘ and :meth:‘MapSource.get_max_zoom‘. Default to 0. **Value:** `NumericProperty(0)`
map_source	Provider of the map, default to a empty :class:‘MapSource‘. **Value:** `ObjectProperty(MapSource())`
double_tap_zoom	If True, this will activate the double-tap to zoom. **Value:** `BooleanProperty(False)`
markers	The list of markers which belongs to the current MapView object **Value:** `BooleanProperty(True)`
delta_x	**Value:** `NumericProperty(0)`
delta_y	**Value:** `NumericProperty(0)`
background_color	**Value:** `ListProperty([181/ 255., 208/ 255., 208/ 255., 1])`
__events__	**Value:** `['on_map_relocated']`
bbox	**Value:** `AliasProperty(get_bbox, None, bind= ["lat", "lon", "_zoom"])`
Inherited from kivy.uix.widget.Widget	

continued on next page

Name	Description
canvas, center, center_x, center_y, children, cls, disabled, height, id, ids, opacity, parent, pos, pos_hint, right, size, size_hint, size_hint_x, size_hint_y, top, width, x, y	

12 Module isySUR.gui.triangulation

Created on Tue Jan 13 00:15:11 2015 Simple triangulation class. It tries to triangulate a given polygon using the ear clipping algorithm. Roughly based on Rawlyn's implementation (http://chipmunk-physics.net/forum/viewtopic.php?f=1&t=813&p=3985) but not using pymunk and more flexible to the polygon orientation.

12.1 Variables

Name	Description
__package__	**Value:** `None`

12.2 Class Triangulator

A simple triangulator. Takes a polygon outline with no self intersection or holes and computes a triangulation using ear clipping.

12.2.1 Methods

__init__(*self*)

Constructor for the triangulator. Initialises the clockwise variable, used to store if the polygon points are ordered clockwise or counter-clockwise

calc_area(*self*, *tri*)

Function to compute the given triangle.

Parameters

`tri`: The triangle which area is to be computed

(type=List[a,b,c])

Return Value

Area of the triangle, if there were 3 points given in tri. None otherwise

(type=Float)

is_clockwise(*self*, *points*)

Function to compute if the orientation of a pointlist is clockwise.

Parameters

`points:` The pointlist that is to be checked.

(type=[(Float,Float),])

Return Value

True if the points are orientated clockwise, false if they are orientated counter-clockwise.

(type=Boolean)

triangulate(*self*, *poly*)

Main function performing the triangulation of the given polygon. Does not work for self intersecting polygons!

Parameters

`poly:` Pointlist making up the polygon. Points can be sorted either clockwise or counter-clockwise.

(type=[Tupel(Float,Float),])

Return Value

List of triangles covering the polygon.

(type=[[Tupel(Float,Float),Tupel(Float,Float),Tupel(Float,Float)],])

13 Module isySUR.isyUtils

Module containing util functions.

13.1 Functions

getXYpos(*relativeNullPoint*, *p*)

Calculates X and Y distances in meters.

Parameters

`relativeNullPoint`: The relative null point for the calculation.
(type=Tupel(float,float))

`p`: Point (lat,long) for which the x,y position should be calculated.
(type=Tupel(float,float))

Return Value

Tupel with x,y coordinates of the given point.
(type=Tupel(float,float))

13.2 Variables

Name	Description
__package__	**Value:** `'isySUR'`

14 Module isySUR.kmlData

Module containg all necessary information for creating and parsing a kml-xml.

14.1 Functions

indent(*elem*, *level*=`0`)

Recursive function used to indent xml elements according to their level to allow pretty print.

Parameters

`elem`: The element to be indented

(type=ET.Element)

`level`: Optional parameter representing the current level in the tree of the element

(type=Int)

14.2 Variables

Name	Description
__package__	**Value:** `'isySUR'`

14.3 Class KMLObject

Class representing a kml file. Holds a list of contained placemarks.

14.3.1 Methods

__**init**__(*self*, *name*, *placemarks*=`None`)

Constructor for the KMLObject.

Parameters

`name`: Name of the kml.

(type=String)

`placemarks`: Optional paramter to initialise this KMLObject with a list of placemarks.

(type=[kmlData.Placemark,])

getStyle(*self*, *styleName*)

addStyles(*self*, *styles*)

Function that allows to add styles to the kml. If styles does not include lineColour, or lineWidth, standard values are used.

Parameters

`styles:` The styles that are to be added.

(type={styleID: {"polyColour":value, "lineColour":value, "lineWidth":value},})

addPlacemark(*self*, *placemark*)

Function to add a placemark to this SURObject.

Parameters

`placemark:` The placemark object that is to be added.

Raises

`TypeError` If the given placemark is not a Placemark object.

addPlacemarkList(*self*, *placemarkList*)

Function to add a list of placemarks to this SURObject.

Parameters

`placemarkList:` The list of placemark objects that are to be added.

Raises

`TypeError` If the plcemarkList is not actually a list.

parseKML(*cls*, *filename*)

Classmethod to create a KMLObject from a file.

Parameters

`filename:` The name (including the path) of the file.

(type=String)

Return Value

The parsed KMLObject.

saveAsXML(*self*, *filename*)

Function to save the kml in it's xml representation in a file with the given filename.

Parameters

`filename:` The name of the file this kml should be written to.

(type=String)

getXML(*self*)

Function to return the XML representation for this kml as string.

Return Value

The String-XML representation of this kml object.

14.4 Class Placemark

14.4.1 Methods

__init__(*self*, *name*, *imageName*, *ruleType*=`None`, *pointList*=`None`, *style*=`'#defaultStyle'`, *ruleCoords*=`None`)

Constructor for the Placemark class.

Contains a list of nodes that make up the polygon for this placemark.

Parameters

`name:` The name of the placemark.

(type=String)

`imageName:` The name/src of the image in the placemark description.

(type=String)

`ruleType:` The rule type of the placemark. (Currently not used)

(type=Tupel(key, value))

`pointList:` Optional pointList that contains the points coordinates (lon,lat) that make up the polygon this placemark describes.

`style:` Optional style for the placemark. Relevant for displaying the placemark in googleEarth. (Currently not used)

(type=String.)

`ruleCoords:` Optional rule coordinates (lat,lon).

(type=(Float,Float))

addPoint(*self*, *point*)

Function to add a node to the polygon for the placemark.

Parameters

`node:` The point coordinate (lon,lat) that is to be added to the placemark.

Raises

`TypeError` If point is not a string.

addPointList(*self*, *pointList*)

Function to add a list of nodes to the polygon of the placemark.

Parameters

`pointList:` The list of point coordinates(lon,lat) that are to be added.

Raises

`TypeError` If pointList is not a list.

hasPolygon(*self*)

Function to check if a placemark contains a valid polygon.

A polygon is considered as valid as soon as it contains at least 3 nodes.

Return Value

True if the polygon consists of at least 3 nodes, else False.

getXMLTree(*self*)

Function to get the xmlTree representation of the placemark.

Return Value

A xmlTree (xml.etree) representation of the placemark.

15 Module isySUR.osmAPI

Base class that builds a connection to the Openstreetmap API.

15.1 Variables

Name	Description
__package__	**Value:** `'isySUR'`

15.2 Class osmAPI

15.2.1 Methods

__init__(*self*)

Constructor of the osmAPI making a conntection to Openstreetmap.

getDataFromPoly(*self*, *polyString*)

Function to request parsed data from osm that is within the polygon given by the polyString

Parameters

`polyString`: String containing the outline of the polygon "lat1 lon1 lat2 lon2 ..."

(type="String")

Return Value

The parsed osmData

(type=osmData.OSM)

performRequest(*self*, *boundingBox*, *filterList*=[])

This function requests data from openStreetMap

Parameters

`boundingBox:` a list of the points of the boundingBox
[minLat,minLon,maxLat,maxLon]

(type=[float,float,float,flaot])

`filterList:` (optional) List of tupel of filter-rules
e.g.[('way',['"amenity"="univerity"',..]),..] or
('way',['"building"=""']) for some kind of wild-card

(type=[Tupel(str,[str,..])])

Return Value

an request object with the data-xml in the content property

16 Module isySUR.osmData

#Basic class that holds the osm-data (consisting of basing elements)

16.1 Variables

Name	Description
__package__	**Value:** `'isySUR'`

16.2 Class OSM

16.2.1 Methods

__init__(*self*)

Constructor for the osm data object.

Initialises the dictionaries for the nodes, ways and relations that will be contained in this osmObject.

addNode(*self*, *node*)

Function to add a node to this osm object.

Parameters

`relation:` The node object that is to be added.

Raises

`TypeError` TypeError is raised when something other than a node is passed.

addNodeList(*self*, *nodeList*)

Function to add a list of nodes to this osm object.

Parameters

`nodeList:` The list of node objects that are to be added.

addWay(*self*, *way*)

Function to add a way to this osm object.

Parameters

`relation:` The way object that is to be added.

Raises

`TypeError` TypeError is raised when something other than a way is passed.

addRelation(*self*, *relation*)

Function to add a relation to this osm object.

Parameters

`relation:` The relation object that is to be added.

Raises

`TypeError` TypeError is raised when something other than a relation is passed.

__**eq**__(*self*, *other*)

Override of the equal method for OSM.

Equality is based on the equality of the three dictionaries nodes, ways and relations

Parameters

`other:` The other osm object that this object is to be compared with.

Return Value

True if the other object is equal to this object, else False.

(type=Boolean)

__**ne**__(*self*, *other*)

Override of the not equal method for OSM.

Parameters

`other:` The osm object that this object is to be compared with.

Return Value

True if other is not equal to this object, else False.

(type=Boolean)

getNearestNode(*self*, *point*, *tags*={}, *otherNodes*=[])

```
This function returns the ids of the nodes and its distance which are closest to the

@param point: The point - (lat, lon) - for which the function has
              to compute the closest node.
@type point: Tuple(float,float)

@param tags: A dictionary of tags, given as a key value pair, which
             will be used to filter the nodes. You can use * as wildcard
             for the value or key but NOT both.

             e.g. dict("type":"xyz") or dict("type":"*")
@type tags: dict(str:str)

@param otherNodes: Use only this nodes, given by a list of
                       its IDs, to find the nearest relation.
@type otherNodes: [str,]

@return: The function returns a list distanceResult-Objects (e.g [distObj1,distObj2,
        which holds the following informations:

        - distance (float): If an object is found, it contains the
                            distance to the nearest object

        - nearestObj (str, type):   it contains the ID and the type
                                    of the nearest object

                                 For example:
                                   found object: ("1", osmData.Node)

        - nearestSubObj [(str, type)]:  Is empty: [("-1",None)]
    If nothing is found, the resulting list is empty.
@rtype: [osmData.distanceResult,..]
```

getNearestWay(*self*, *point*, *onlyPolygons*, *tags*={}, *otherWays*=[])

```
This function returns the ids of the ways, the distance which is closest to the give

@param point: The point - (lat, lon) - for which the function has
              to compute the closest way.
@type point: Tuple(float,float)

@param onlyPolygons: True for only using Ways with complete Polygons for computation
                     False for use all

@type onlyPolygons: boolean

@param tags: A dictionary of tags, given as a key value pair, which
             will be used to filter the ways. You can use * as wildcard
             for the value or key but NOT both.

             e.g. dict("type":"xyz") or dict("type":"*")
@type tags: dict(str:str)

@param otherWays: Use only these ways, given by a list of
                        its IDs, to find the nearest way.
@type otherWays: [str,]

@return: The function returns a list of distanceResult-Objects (e.g [distObj1,distOb
        which holds the following informations:

        - distance (float): If an object is found, it contains the
                            distance to the nearest object

        - nearestObj [(str, type)]: If one object is found, it contains the ID and
                                    the type of the nearest object

                                  For example:
                                    found object: ("1", osmData.Way)

        - nearestSubObj [(str, type)]:  If an object is found, it contains the IDs
                                        of the two Nodes, which defines the nearest Ed
                                        There could be several edges, which have the s
                                        distance

                                        For example:
                                          found object: ([["1","2"], osmData.Node),..]
    If nothing is found, the resulting list is empty.
@rtype: [osmData.distanceResult,..]
```

getNearestRelation(*self*, *point*, *tags*={}, *otherRelations*=[])

```
This function returns the ids of the relation,its way and its distance which is clos

@param point: The point - (lat, lon) - for which the function has
              to compute the closest relation.
@type point: Tuple(float,float)

@param tags: A dictionary of tags, given as a key value pair, which
             will be used to filter the realtions. You can use * as wildcard
             for the value or key but NOT both.

             e.g. dict("type":"multipolyon")  or dict("type":"*")
@type tags: dict(str:str)

@param otherRelations: Use only this relations, given by a list of
                       its IDs, to find the nearest relation.
@type otherRelations: [str,]

@return: The function returns a list of distanceResult-Object (e.g [distObj1,distObj
        which holds the following informations:

        - distance (float): If an object is found, it contains the
                            distance to the nearest object

        - nearestObj (str, type): If one object is found, it contains the ID and
                                  the type of the nearest object

                                  For example:
                                    found object: ("1", osmData.Relation)

        - nearestSubObj [(str, type)]:  If an object is found, it contains the ID
                                        and the type of the neares subobject in
                                        the relation.

                                        For example:
                                            found object: [("1", osmData.Relation),..]
    If nothing is found, the resulting list is empty.
@rtype: [osmData.distanceResult,..]
```

16.3 Class Node

object
└ **isySUR.osmData.Node**

16.3.1 Methods

__init__(*self*, *identifier*, *lat*, *lon*, *tags*)

Basic class containing an osm Node

Parameters

`identifier:` The id of the node.
(type=Will be parsed to string)

`lat:` Latitude of the node as float.
(type=Will be parsed to float.)

`lon:` Longitude of the node as float.
(type=Will be parsed to float.)

`tags:` A dictionary containing all the tags for the node.

Overrides: object.__init__

getCoordinateString(*self*)

Returns a string representation of the coordinates for this node.

Return Value

A String with lon, lat. Both with 8 trailing digits

__eq__(*self*, *other*)

Override of the equality method for node.

Equality is based on the equality of the id, longitude, latitude and the tags.

Parameters

`other:` The node this node is to be compared with.

Return Value

True if the other node is equal to this node with respect to the above mentioned fields, else False.
(type=Boolean)

__ne__(*self*, *other*)

Override of the not equal method for node.

Parameters

`other`: The node that this node is to be compared with.

Return Value

True if other is not equal to this node, else False.

(type=Boolean)

getDistance(*self*, *point*)

```
This function computes the distance between two points

@param point: the point the distance should be computed with
@type point: tuple of latitude and longitude (float,float)

@return: The function returns a distanceResult-Object which holds the
        following informations:

        - distance (float): Distance between the current and the given point

        - nearestObj (str, type): The current node
                                    For example: ("1", osmData.Node)

        - nearestSubObj [(str, type)]:  is empty: [("-1", None)]
@rtype: osmData.distanceResult
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

16.3.2 Properties

Name	Description
coords	This function-property returns latitude and longitude as tupel *(type=Tupel(float,float))*
Inherited from object __class__	

16.4 Class Way

object
└ **isySUR.osmData.Way**

16.4.1 Methods

__init__(*self*, *identifier*, *refs*, *tags*, *osmObj*)

Basic class containing an osm Way.

Parameters

`identifier:` The id of the way as a string

`refs:` An ordered list of node id's that make up the way
(type=[str,..])

`tags:` A dictionary containing all the tags for the way
(type=dict(str:str,..))

`osmObj:` Reference to the osmObj, this way is included in.
(type=osmData.OSM)

Overrides: object.__init__

isPolygon(*self*)

This functions prooves if the Way is a polygon

Return Value

true if polygon exists
(type=boolean)

__eq__(*self*, *other*)

Override of the equality method for way.

Equality is based on the equality of the id, the references and the tags.

Parameters

`other:` The relation this relation is to be compared with.

Return Value

True if the other way is equal to this way in id, references and tags, else False.
(type=Boolean)

__ne__(*self*, *other*)

Override of the not equal method for way.

Parameters

`other:` The way that this way is to be compared with.

Return Value

True if other is not equal to this way, else False.

(type=Boolean)

isInside(*self*, *point*, *vertices*=`[]`)

This function proves if a points is envolved in a polygone

Parameters

`point:` x and y-coord of the point

(type=Tupel(float,float))

`vertices:` list of points to calculate with (e.g used for combined polygons)

(type=[Tupel(float,float),..])

Return Value

true if point is inside false if point is outside or on edge or way isn't a polygon

(type=boolean)

getDistance(*self*, *point*)

```
Function that returns the distance of the given point to the current way.

@param point: The point(lat,lon) to which the distance is calculated
@type point: Tuple(float,float)

@return: The function returns a distanceResult-Object which holds the
         following informations:

         - distance (float): The distance between the current way and the given poin

         - nearestObj [(str, type)]: The current way
                                     For example: [("1", osmData.Way)]

         - nearestSubObj [(str, type)]:  The edge from the current way which is clos
                                         For example: [(["1","2"], osmData.Node)]
@rtype: osmData.distanceResult
```

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

16.4.2 Properties

Name	Description
Inherited from object	
__class__	

16.5 Class Relation

object

isySUR.osmData.Relation

16.5.1 Methods

__init__(*self*, *identifier*, *members*, *tags*, *osmObj*)

Basic class containing an osm Relation.

Parameters

`identifier:` The id of the relation.
(type=any type)

`members:` The members of this relation.
(type=A list of tripel [membertype(e.g. way), id of the member, addition tags (e.g. outer)])

`tags:` A dictionary containing all the tags for the relation
(type={key: value,})

`osmObj:` Reference to the osmObj, this way is included in.
(type=osmData.OSM)

Overrides: object.__init__

getDistance(*self*, *point*)

```
Function that returns the distance of the given point to the current relation.

@param point: The point(lat,lon) to which the distance is calculated
@type point: Tuple(float,float)

@return: The function returns a distanceResult-Object which holds the
        following informations:

        - distance (float): Distance between the current and the given point

        - nearestObj (str, type): The current relation
                                    For example: ("1", osmData.Relation)

        - nearestSubObj [(str, type)]:  The nearestSubObject of the current relation
                                        For example: [("3", osmData.Way),..]
@rtype: osmData.distanceResult
```

isInside(*self*, *point*)

This function prooves, if a point is inside a relation.

Parameters

`point:` the point to proove @type point; Tuple(float,float)

Return Value

the result e.g. (True,([1],osmData.Way)) or (True,([1,2,5],osmData.Way)) for polygon combinded of more then one way

(type=Tupel(boolean,Tupel([str/int,..],osmData.Types)))

addPolygon(*self*, *wayList*)

Function to add a polygon to the relation.

Parameters

`wayList:` List of way ids that make up the polygon

(type=A list of ids. The id's can be of any type but must match the type of the actual objects.)

addPolygonList(*self*, *polyList*)

Function to add a list of polygons to the relation.

Parameters

`polyList:` List of polygons. A polygon is given by a list of way Ids that make up the polygon.

(type=A list of lists that contain way Ids.)

hasMember(*self*, *memId*)

Function to query if the given member is present in the relation

Parameters

`memId:` The member id that is to be tested.

(type=Any type)

Return Value

True if this relation has got a member of the given id, else False

(type=Boolean)

__**eq**__(*self*, *other*)

Override of the equality method for relations.

Equality is based on the equality of the id, the members and the tags.

Parameters

`other`: The relation this relation is to be compared with.

Return Value

True if the other relation is equal to this relation in id, members and tags, else False.

(type=Boolean)

__**ne**__(*self*, *other*)

Override of the not equal method for relations.

Parameters

`other`: The relation that this relation is to be compared with.

Return Value

True if other is not equal to this relation, else False.

(type=Boolean)

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

16.5.2 Properties

Name	Description
Inherited from object	
__class__	

16.6 Class distanceResult

object

isySUR.osmData.distanceResult

16.6.1 Methods

__init__(*self*, *distance*, *nearestObj*, *nearestSubObj*=`[('-1', None)]`)

Basic class containing the result of a distance calculation

Parameters

`distance:` The distance to the nearestObj

(type=float)

`nearestObj:` the ID and type of the nearest object e.g. ("1",osmData.Relation)

(type=Tuple(str,type))

`nearestSubObj:` (optional) the nearest subobject of the current nearest object (a way which is a subobject of a relation) e.g. [("2",osmData.Way),...]

(type=[Tuple(str,str)])

Overrides: object.__init__

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

16.6.2 Properties

Name	Description
Inherited from object	
__class__	

17 Module isySUR.program

Main pipeline to compute kml from a given SUR(file).

17.1 Variables

Name	Description
__package__	**Value:** `'isySUR'`

17.2 Class KMLCalculator

17.2.1 Methods

__init__(*self*)

Constructor for the pipeline. Sets up the osmAPI as well as the desired bounding box, that is to be used to request osm data.

computeKMLsAndStore(*self*, *inPath*, *outPath*, *configPath*='')

Function to compute kmls from a given file of SURs. Stores them either in one kml or in individual kmls plus one containing all of them. W

Parameters

`inPath`: Path to the file containing the SURs which areas are to be computed.

(type=String)

`outPath`: Path to the file or directory where the results should be saved. If outPath points to a file, all placemarks are stored in one kml. If outPath points to a directory, one kml for each SUR will be computed plus one, containing all others.

(type=String)

`configPath`: Optional path to a config file, containing information about the classification of rules (indoor, outdoor or both).

(type=String)

calcKML(*self*, *surObj*)

Function to work on a single sur.SUR object and computes it's kml.

Parameters

`surObj:` The sur object whose kml is to be calculated.

(type=sur.SUR)

Return Value

KML object containing the calculated area for the given sur.
Returns None if no polygon could be computed.

(type=kmlData.KMLObject)

18 Module isySUR.sur

Basic class to load and store space usage rules.

18.1 Variables

Name	Description
__package__	**Value:** `'isySUR'`

18.2 Class SUR

18.2.1 Methods

__init__(*self*, *surID*, *name*, *lat*, *lon*, *surClassification='IO'*)

Constructor for the space usage rule object.

Parameters

`surID`: Id of the sur

`name`: The name of the rule. Usually a key-value combination.

(type=String)

`lat`: The latitude that belongs to the SUR.

(type=Float)

`lon`: The longitude that belongs to the SUR.

(type=Float)

`surClassification`: Optional parameter to determine whether this sur can be applied indoor ("I"), Outdoor ("O") or indoor as well as outdoor ("IO"). Default is "IO".

(type=String)

addRuleName(*self*, *name*)

Function to add further rule names to the SUR.

Parameters

`name`: The name for the rule that is to be added.

(type=String)

fromString(*cls*, *s*)

Classmethod that creates a SUR object from the given string.

Parameters

`s:` The string that contains the relevant data. The data should be seperated by ','

(type=String)

Return Value

The created SUR object.

(type=sur.SUR)

fromFile(*cls*, *f*, *configPath*)

Classmethod that creates a list of SUR objects from the given file.

Parameters

`f:` The file handler of the already opened file that contains the SUR data.

(type=file)

`configPath:` Path to the config file that should be used to determine sur classification.

(type=String)

Return Value

A list of all the created SURs.

(type=[sur.SUR,])

19 Module isySUR.surTypeManager

Helper class that leads known sur types (indoor, outdoor, both) from a file and can be queried for a certain rule.

19.1 Variables

Name	Description
__package__	**Value:** `None`

19.2 Class surTypeManager

19.2.1 Methods

__init__(*self*, *configPath*)

Constructor for the type manager. Parses the given config file.

Parameters

`configPath`: Path to the config file that is to be used.

(type=String)

getSURType(*self*, *ruleString*)

Function to query the surType for a given rule. Returns the classification ("I","O","IO") of the sur. If the given ruleString was not found in the config IO is returned.

Parameters

`ruleString`: String of the rule name, e.g. "animal_feeding="no"".

(type=String)

Return Value

Sur classification ("I","O","IO")

(type=String)

20 Module run_isySUR

Created on Mon Dec 22 18:36:46 2014 Main entrance point for the informatiCup program. Name should change once a final name for the program has been found.

Author: adreyer & jpoeppel

20.1 Functions

parseArguments()

Function that takes program parameters and checks for required arguments.

gui(*args*=`None`)

Function that starts the gui version of the program. When importing fails, a message is printed and dealWithImportError is called.

Parameters

`args`: the arguments to give to the gui version. @type argparse.Namespace

dealWithImportError()

Function that gives a prompt to get the parameters for the cli version without sys.argv.

cli(*args*)

Function that starts the command line version of the program.

Parameters

`args`: the arguments to give to the cli version. @type argparse.Namespace

20.2 Variables

Name	Description
__package__	**Value:** `None`

Index