

8INF700

Generated by Doxygen 1.8.13

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	Class Documentation	5
3.1	Winter20_8INF700.Alarm.Alarm Class Reference	5
3.1.1	Detailed Description	6
3.2	Winter20_8INF700.Application.Application Class Reference	6
3.2.1	Detailed Description	10
3.2.2	Member Function Documentation	10
3.2.2.1	__ButtonsAction()	11
3.2.2.2	__ChangeBackLightIntensity()	11
3.2.2.3	__ChangeSettingsListsValues()	11
3.2.2.4	__ExportSettingsToCSV()	11
3.2.2.5	__FillLists()	12
3.2.2.6	__ImportAlarmSettingsToGUI()	12
3.2.2.7	__initialiseSongInfoUpdate()	12
3.2.2.8	__initialiseWeatherDateHourUpdate()	12
3.2.2.9	__initiateAlarmTimer()	13
3.2.2.10	__InitiateBacklight()	13
3.2.2.11	__initiateHourAndWeatherUpdate()	13
3.2.2.12	__initiateMessengerTimers()	13

3.2.2.13	__setActualPrecipIntensity()	13
3.2.2.14	__setActualPrecipitationProb()	14
3.2.2.15	__setActualPrecipType()	14
3.2.2.16	__setActualTemp()	14
3.2.2.17	__setActualWindDir()	14
3.2.2.18	__setActualWindSpeed()	14
3.2.2.19	__setAlarmTimers()	15
3.2.2.20	__setAlarmTimerTime()	15
3.2.2.21	__setDailyMaxFeeledTemp()	15
3.2.2.22	__setDailyMinFeeledTemp()	15
3.2.2.23	__setFeeledTemp()	15
3.2.2.24	__setWeatherDesc()	16
3.2.2.25	__StartMessengerTimer()	16
3.2.2.26	__StopMessengerTimer()	16
3.2.2.27	__UpdateMessengerNotificationInterface()	16
3.2.2.28	__UpdateNotification()	17
3.2.2.29	Alarm()	17
3.2.2.30	ApplySettingsChange()	17
3.2.2.31	DeleteOldCSV()	17
3.2.2.32	Interface()	18
3.2.2.33	setDate()	18
3.2.2.34	setHour()	18
3.2.2.35	setWeather()	18
3.2.2.36	StopSpotifyActualisation()	19
3.2.2.37	StopWeatherAndHourActualisation()	19
3.2.2.38	UpdateSpotify()	19
3.3	Winter20_8INF700.Location.Location Class Reference	19
3.3.1	Detailed Description	20
3.3.2	Member Function Documentation	20
3.3.2.1	getLatitude()	20

3.3.2.2	getLongitude()	20
3.4	Winter20_8INF700.MessengerNotificationsGUI.MessengerNotifGui Class Reference	21
3.4.1	Detailed Description	22
3.4.2	Member Function Documentation	22
3.4.2.1	DeleteNotifications()	22
3.4.2.2	getWidget()	22
3.4.2.3	UpdateLists()	23
3.4.2.4	UpdateNotificationList()	23
3.5	Winter20_8INF700.SpotifyData.Spotify Class Reference	23
3.5.1	Detailed Description	24
3.5.2	Member Function Documentation	24
3.5.2.1	__FetchAlbum()	24
3.5.2.2	__FetchAlbumCoverURL()	25
3.5.2.3	__FetchArtists()	25
3.5.2.4	__FetchSongTitle()	25
3.5.2.5	__getAnHeader()	25
3.5.2.6	__getJSONData()	26
3.5.2.7	__setJSONData()	26
3.5.2.8	__setSpotifyAuthToken()	26
3.5.2.9	CheckSongInfo()	26
3.5.2.10	getAlbum()	27
3.5.2.11	setAlbum()	27
3.5.2.12	setAlbumCover()	27
3.5.2.13	setArtists()	27
3.5.2.14	setAuthToken()	27
3.5.2.15	setSongTitle()	28
3.6	Winter20_8INF700.MessengerUnreadMessageFetcher.UnreadMessageFetcher Class Reference	28
3.6.1	Detailed Description	29
3.6.2	Member Function Documentation	29
3.6.2.1	__getaUserName()	29

3.6.2.2	__getUnReadMessageDict()	29
3.6.2.3	__setThreadList()	30
3.6.2.4	__setUnreadMessagesList()	30
3.6.2.5	EmptyLists()	30
3.6.2.6	getAuthorName()	30
3.6.2.7	getConversationPic()	31
3.6.2.8	getMessage()	31
3.6.2.9	getNewUnreadMessageNumber()	31
3.6.2.10	getTotalUnreadMessageNumber()	31
3.6.2.11	resetNewUnreadMessageNb()	32
3.6.2.12	setLists()	32
3.7	Winter20_8INF700.WeatherClock.Weather Class Reference	32
3.7.1	Member Function Documentation	33
3.7.1.1	getDailyMaxFeeledTemp()	33
3.7.1.2	getDailyMinFeeledTemp()	33
3.7.1.3	getFeeledTemp()	33
3.7.1.4	getPrecipitationIntensity()	34
3.7.1.5	getPrecipitationProb()	34
3.7.1.6	getPrecipitationType()	34
3.7.1.7	getTemp()	34
3.7.1.8	getWeatherDesc()	34
3.7.1.9	getWindDir()	35
3.7.1.10	getWindSpeed()	35
3.7.1.11	setDailyForecastData()	35
3.7.1.12	Update()	35
3.7.2	Member Data Documentation	35
3.7.2.1	__actualWeatherData	36

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Winter20_8INF700.Location.Location	19
Winter20_8INF700.SpotifyData.Spotify	23
Winter20_8INF700.MessengerUnreadMessageFetcher.UnreadMessageFetcher	28
Winter20_8INF700.WeatherClock.Weather	32
QDialog	
Winter20_8INF700.MessengerNotificationsGUI.MessengerNotifGui	21
QWidget	
Winter20_8INF700.Alarm.Alarm	5
Winter20_8INF700.Application.Application	6

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Winter20_8INF700.Alarm.Alarm	
The class of the alarm clock fonctionnality	5
Winter20_8INF700.Application.Application	
The main class of the program Contains the graphical user interface and the alarm clock fonctionnality	6
Winter20_8INF700.Location.Location	
Simple class to get the coordinates of the user's city	19
Winter20_8INF700.MessengerNotificationsGUI.MessengerNotifGui	
The graphical user interface class of the messenger notification fonctionnality The role of this class is to generate the graphical notifications list for the main interface	21
Winter20_8INF700.SpotifyData.Spotify	
Spotify (p. 23)'s data class for the Spotify (p. 23) fonctionnality	23
Winter20_8INF700.MessengerUnreadMessageFetcher.UnreadMessageFetcher	
The data class for the messenger notification fonctionnality The role of this class is to fetch the notifications' data	28
Winter20_8INF700.WeatherClock.Weather	32

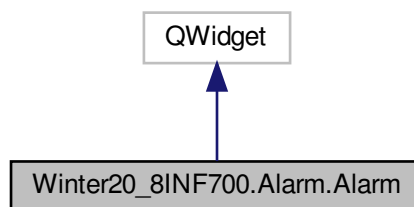
Chapter 3

Class Documentation

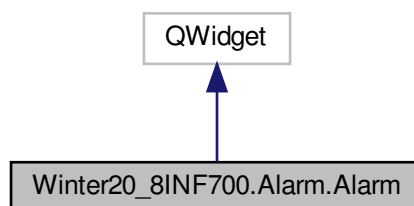
3.1 Winter20_8INF700.Alarm.Alarm Class Reference

The class of the alarm clock fonctionnality.

Inheritance diagram for Winter20_8INF700.Alarm.Alarm:



Collaboration diagram for Winter20_8INF700.Alarm.Alarm:



Public Member Functions

- `def __init__(self)`

3.1.1 Detailed Description

The class of the alarm clock fonctionnality.

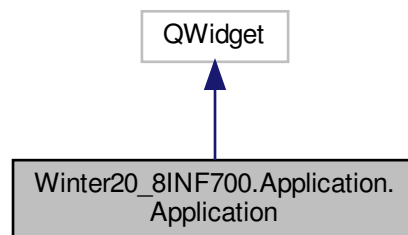
The documentation for this class was generated from the following file:

- `Alarm.py`

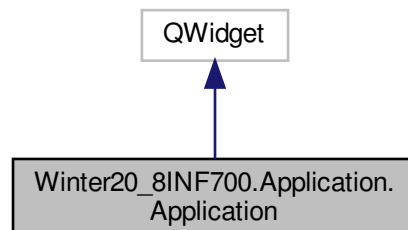
3.2 Winter20_8INF700.Application.Application Class Reference

The main class of the program Contains the graphical user interface and the alarm clock fonctionnality.

Inheritance diagram for `Winter20_8INF700.Application.Application`:



Collaboration diagram for `Winter20_8INF700.Application.Application`:



Public Member Functions

- def **__init__** (self)
 - def **Interface** (self)
Public method to initiate the graphical user interface.
- def **MessengerButtonAction** (self)
- def **setHour** (self)
 - def **setDate** (self)
Private method to set the content of the hour Label.
 - def **setDate** (self)
Private method to set the content of the date Label.
- def **StopWeatherAndHourActualisation** (self)
Public method to stop weather, hour and date timers.
- def **StopSpotifyActualisation** (self)
Public method to stop the Spotify's songs informations actualisation fonctionnality.
- def **UpdateSpotify** (self)
Public method to update the Spotify fonctionnality's interface.
- def **setWeather** (self)
Public method to set the weather's data by calling all the private methods for setting the weather data used for the weather-clock fonctionnality's interface.
- def **DeleteOldCSV** (self)
Public method to delete the csv because the csv module doesn't have an overwrite mode.
- def **ApplySettingsChange** (self)
Public method for calling all the private methods required to apply the settings change.
- def **Alarm** (self)
Public method to play the alarm ring tone.

Public Attributes

- **mainLayout**
- **buttonLayout**
- **spotifyButton**
- **weatherAndHourButton**
- **weatherHourContent**
- **dateAndHourLayout**
- **hourLabel**
- **dateLabel**
- **actualTemperatureLabel**
- **actualFeeledTemperatureLabel**
- **weatherDescriptionLabel**
- **actualPrecipitationProbabilityLabel**
- **actualPrecipitationIntensityLabel**
- **actualPrecipitationTypeLabel**
- **actualWindSpeedLabel**
- **actualWindDirectionLabel**
- **dailyMaxTempLabel**
- **dailyMinTempLabel**
- **spotify**
- **songInfoLayout**
- **songTitleLabel**
- **artistsLabel**
- **albumTitleLabel**
- **albumCoverLabel**

- **albumCover**
- **dateWeatherLayout**
- **weatherWidget**
- **mainWidget**
- **layout**
- **timerAlarmTimer**
- **t**

Private Member Functions

- **def __UpdateMessengerNotificationInterface (self)**
Private method to update the content of the Messenger notification's interface.
- **def __UpdateNotification (self)**
*Private method used by the instance of the **Application** (p. 6) class to update the graphical and data notification structures of the MessengerNotifGui and UnreadMessageFetcher.*
- **def __initiateMessengerTimers (self)**
Private method used to initiate two QTimer's used for calling the methods used to update the notifications.
- **def __StartMessengerTimer (self)**
Private method used to start two QTimer's used for calling the methods used to update the notifications.
- **def __StopMessengerTimer (self)**
Private method used to stop two QTimer's used for calling the methods used to update the notifications.
- **def __ButtonsAction (self, selectionId)**
Private method used to specify what to do depending the actual widget showed by the QStackWidget and wich button the user is pressing.
- **def __setActualTemp (self)**
Private method to set the content of the actual temperature label.
- **def __setFeeledTemp (self)**
Private method to set the content of the actual feeled temperature label.
- **def __setWeatherDesc (self)**
Private method to set the content of the actual weather description label.
- **def __setActualPrecipitationProb (self)**
Private method to set the content of the actual precipitation probability label.
- **def __setActualPrecipIntensity (self)**
Private method to set the content of the actual precipitation intensity label.
- **def __setActualPrecipType (self)**
Private method to set the content of the actual precipitation type label.
- **def __setActualWindSpeed (self)**
Private method to set the content of the actual wind speed label.
- **def __setActualWindDir (self)**
Private method to set the content of the actual wind direction label.
- **def __setDailyMaxFeeledTemp (self)**
Private method to set the content of the daily maximum feeled temperature label.
- **def __setDailyMinFeeledTemp (self)**
Private method to set the content of the daily minimum feeled temperature label.
- **def __initialiseWeatherDateHourUpdate (self)**
Private method to set the hour, the date and the weather at the beginning of the program.
- **def __initialiseSongInfoUpdate (self)**
Private method to do the same task as the __initialiseWeatherDateHourUpdate method but for the Spotify's song informations.
- **def __initiateHourAndWeatherUpdate (self)**

This method does the same task as `__initialiseSongInfoUpdate` and `__initialiseWeatherDateHourUpdate` but for the weather and clock functionality.

- `def __FillLists (self)`
Private method to set the lists for the alarm clock by reading the saved settings into a csv.
- `def __ImportAlarmSettingsToGUI (self)`
Private method to display the content of the alarm clock' data lists into the alarm clock' interface.
- `def __ChangeSettingsListsValues (self)`
Private method to set the alarm clock's settings of the interface into the data structures when the user apply the change.
- `def __ExportSettingsToCSV (self)`
Private method to write the new alarm clock's settings into a new csv weather-clock functionality's interface.
- `def __initiateAlarmTimer (self)`
Private method for initiating the alarm timer useless? arg: none.
- `def __setAlarmTimers (self)`
Public method to set timers required to have a timeout to the time where the alarm clock has to ring.
- `def __setAlarmTimerTime (self)`
Private method to determine the number of milisecond before the next alarm.
- `def __InitiateBacklight (self)`
Public method to initiate the backlight adjustment.
- `def __ChangeBackLightIntensity (self)`
Private method to set the opposite backlight mode of the actual one.
- `def __initialiseRaspotify (self)`

Private Attributes

- `__messenger`
- `__timeUntilNextAlarmCheck`
- `__alarmToday`
- `__fileName`
- `__activeDaysList`
- `__alarmActivationHoursList`
- `__alarmActivationMinuteList`
- `__changingContentWidget`
- `__alarmButton`
- `__separatorLine`
- `__hourLayout`
- `__weatherLayout`
- `__weatherDescriptionLayout`
- `__weatherDataLayout`
- `__temperatureDataLayout`
- `__precipitationAndWindDataLayout`
- `__windDataLayout`
- `__precipitationDataLayout`
- `__weatherClockWidget`
- `__weatherCondition`
- `__spotifyWidget`
- `__alarmSettingsLayout`
- `__dailySettingsLayout`
- `__mondaySettingsLayout`
- `__tuesdaySettingsLayout`
- `__wednesdaySettingsLayout`
- `__thursdaySettingsLayout`

- `__fridaySettingsLayout`
- `__saturdaySettingsLayout`
- `__sundaySettingsLayout`
- `__weekAlarmCheckBox`
- `__mondayAlarmCheckBox`
- `__tuesdayAlarmCheckBox`
- `__wednesdayAlarmCheckBox`
- `__thursdayAlarmCheckBox`
- `__fridayAlarmCheckBox`
- `__saturdayAlarmCheckBox`
- `__sundayAlarmCheckBox`
- `__weekWidget`
- `__mondayWidget`
- `__tuesdayWidget`
- `__wednesdayWidget`
- `__thursdayWidget`
- `__fridayWidget`
- `__saturdayWidget`
- `__sundayWidget`
- `__alarmWidget`
- `__confirmButton`
- `__messengerNotificationButton`
- `__buttonsWidget`
- `__weatherDateHourLayout`
- `__temperatureSectionLabel`
- `__feeledTemperatureSectionLabel`
- `__precipitationSectionLabel`
- `__windSectionLabel`
- `__scrollZone`
- `__messengerListTimer`
- `__messengerguiListTimer`
- `__displayDateTimer`
- `__displayHourTimer`
- `__displayWeatherTimer`
- `__songInfoActualisationTimer`
- `__spotifyInterfaceTimer`
- `__apiTokenTimer`
- `__updateTimerCheckTimeTimer`
- `__alarmTimer`
- `__isDayMode`
- `__backlightActualisationTimer`

3.2.1 Detailed Description

The main class of the program Contains the graphical user interface and the alarm clock fonctionnality.

3.2.2 Member Function Documentation

3.2.2.1 __ButtonsAction()

```
def Winter20_8INF700.Application.Application.__ButtonsAction (
    self,
    selectionId ) [private]
```

Private method used to specify what to do depending the actual widget showed by the QStackWidget and wich button the user is pressing.

arg: selectionId: The id associated to the pressed button.

return: none

3.2.2.2 __ChangeBackLightIntensity()

```
def Winter20_8INF700.Application.Application.__ChangeBackLightIntensity (
    self ) [private]
```

Private methode to set the opposite backlight mode of the actual one.

Since the programm is initiated, determine when the backlight change happen is not required anymore. The program will run constantly after his launch. So after the update time will be constant after the initialisation.

arg: none

return: none

3.2.2.3 __ChangeSettingsListsValues()

```
def Winter20_8INF700.Application.Application.__ChangeSettingsListsValues (
    self ) [private]
```

Private method to set the alarm clock's settings of the interface into the data structures when the user apply the change.

arg: none

return: none

3.2.2.4 __ExportSettingsToCSV()

```
def Winter20_8INF700.Application.Application.__ExportSettingsToCSV (
    self ) [private]
```

Private method to write the new alarm clock's settings into a new csv weather-clock fonctionnality's interface.

arg: none

return: none

3.2.2.5 __FillLists()

```
def Winter20_8INF700.Application.Application.__FillLists (
    self ) [private]
```

Private method to set the lists for the alarm clock by reading the saved settings into a csv.

The csv contains 3 rows of 8 columns.

The first column contains the parameter for the whole week and each other columns are for each day of the week (monday to sunday). First row: Contain the status of the alarm clock, the possible values are "on" or "off" for each column. If the first column of the first row is off, the alarm clock is disabled for the entire week. Second row: This row contains the hour where the alarm will ring (if enabled) for each day of the week. Note that the second and third row of the first column are not used. Third row: This row contains the minute of the hour where the alarm will ring (if enabled) for each day of the week.

arg: none

return: none

3.2.2.6 __ImportAlarmSettingsToGUI()

```
def Winter20_8INF700.Application.Application.__ImportAlarmSettingsToGUI (
    self ) [private]
```

Private method to display the content of the alarm clock' data lists into the alarm clock' interface.

weather-clock fonctionnality's interface.

arg: none

return: none

3.2.2.7 __initialiseSongInfoUpdate()

```
def Winter20_8INF700.Application.Application.__initialiseSongInfoUpdate (
    self ) [private]
```

Private method to do the same task as the __initialiseWeatherDateHourUpdate method but for the Spotify's song informations.

arg: none

return: none

3.2.2.8 __initialiseWeatherDateHourUpdate()

```
def Winter20_8INF700.Application.Application.__initialiseWeatherDateHourUpdate (
    self ) [private]
```

Private method to set the hour, the date and the weather at the beginning of the program.

The timeout.connect method of the QTimer class allow execute a fonction at a regular time interval. The fonction is executed at each timeout of the QTimer. The __initialiseWeatherDateHourUpdate method initialises the timers and set the weather, the hour and the date before the first timeout of the timers.

arg: none

return: none

3.2.2.9 __initiateAlarmTimer()

```
def Winter20_8INF700.Application.Application.__initiateAlarmTimer (
    self ) [private]
```

Private method for initiating the alarm timer useless? arg: none.

return: none

3.2.2.10 __InitiateBacklight()

```
def Winter20_8INF700.Application.Application.__InitiateBacklight (
    self ) [private]
```

Public method to initiate the backlight adjustment.

Between 5:30 and 21:30 the backlight is in day mode (max intensity) and the backlight is in nightmode (very low intensity) the rest of the time. The method determines if the program begins in daymode or in darkmode and initializes the timer to determine when the program will change the backlight mode again.

arg: none

return: none

3.2.2.11 __initiateHourAndWeatherUpdate()

```
def Winter20_8INF700.Application.Application.__initiateHourAndWeatherUpdate (
    self ) [private]
```

This method does the same task as __initialiseSongInfoUpdate and __initialiseWeatherDateHourUpdate but for the weather and clock functionality.

3.2.2.12 __initiateMessengerTimers()

```
def Winter20_8INF700.Application.Application.__initiateMessengerTimers (
    self ) [private]
```

Private method used to initiate two QTimer objects used for calling the methods used to update the notifications.

arg: none

return: none

3.2.2.13 __setActualPrecipIntensity()

```
def Winter20_8INF700.Application.Application.__setActualPrecipIntensity (
    self ) [private]
```

Private method to set the content of the actual precipitation intensity label.

arg: none

return: none

3.2.2.14 __setActualPrecipitationProb()

```
def Winter20_8INF700.Application.Application.__setActualPrecipitationProb (
    self ) [private]
```

Private method to set the content of the actual precipitation probability label.

arg: none

return: none

3.2.2.15 __setActualPrecipType()

```
def Winter20_8INF700.Application.Application.__setActualPrecipType (
    self ) [private]
```

Private method to set the content of the actual precipitation type label.

arg: none

return: none

3.2.2.16 __setActualTemp()

```
def Winter20_8INF700.Application.Application.__setActualTemp (
    self ) [private]
```

Private method to set the content of the actual temperature label.

arg: none

return: none

3.2.2.17 __setActualWindDir()

```
def Winter20_8INF700.Application.Application.__setActualWindDir (
    self ) [private]
```

Private method to set the content of the actual wind direction label.

arg: none

return: none

3.2.2.18 __setActualWindSpeed()

```
def Winter20_8INF700.Application.Application.__setActualWindSpeed (
    self ) [private]
```

Private method to set the content of the actual wind speed label.

arg: none

return: none

3.2.2.19 __setAlarmTimers()

```
def Winter20_8INF700.Application.Application.__setAlarmTimers (
    self ) [private]
```

Public method to set timers required to have a timeout to the time where the alarm clock has to ring.

arg: none

return: none

3.2.2.20 __setAlarmTimerTime()

```
def Winter20_8INF700.Application.Application.__setAlarmTimerTime (
    self ) [private]
```

Private method to determine the number of milisecond before the next alarm.

arg: none

return: none

3.2.2.21 __setDailyMaxFeeledTemp()

```
def Winter20_8INF700.Application.Application.__setDailyMaxFeeledTemp (
    self ) [private]
```

Private method to set the content of the daily maximum feeled temperature label.

arg: none

return: none

3.2.2.22 __setDailyMinFeeledTemp()

```
def Winter20_8INF700.Application.Application.__setDailyMinFeeledTemp (
    self ) [private]
```

Private method to set the content of the daily minimum feeled temperature label.

arg: none

return: none

3.2.2.23 __setFeeledTemp()

```
def Winter20_8INF700.Application.Application.__setFeeledTemp (
    self ) [private]
```

Private method to set the content of the actual feeled temperature label.

arg: none

return: none

3.2.2.24 __setWeatherDesc()

```
def Winter20_8INF700.Application.Application.__setWeatherDesc (
    self ) [private]
```

Private method to set the content of the actual weather description label.

arg: none

return: none

3.2.2.25 __StartMessengerTimer()

```
def Winter20_8INF700.Application.Application.__StartMessengerTimer (
    self ) [private]
```

Private method used to start two QTimer used for calling the methods used to update the notifications.

arg: none

return: none

3.2.2.26 __StopMessengerTimer()

```
def Winter20_8INF700.Application.Application.__StopMessengerTimer (
    self ) [private]
```

Private method used to stop two QTimer used for calling the methods used to update the notifications.

arg: none

return: none

3.2.2.27 __UpdateMessengerNotificationInterface()

```
def Winter20_8INF700.Application.Application.__UpdateMessengerNotificationInterface (
    self ) [private]
```

Private method to update the content of the Messenger notification's interface.

To update the content, the QWidgetList of the MessengerNotifGui class is returned regularly (each ten seconds) and set into the QScrollArea object, __scrollZone.

arg: none

return: none

3.2.2.28 __UpdateNotification()

```
def Winter20_8INF700.Application.Application.__UpdateNotification (
    self ) [private]
```

Private method used by the instance of the **Application** (p. 6) class to update the graphical and data notification structures of the MessengerNotifGui and UnreadMessageFetcher.

arg: none

return: none

3.2.2.29 Alarm()

```
def Winter20_8INF700.Application.Application.Alarm (
    self )
```

Public method to play the alarm ring tone.

arg: none

return: none

3.2.2.30 ApplySettingsChange()

```
def Winter20_8INF700.Application.Application.ApplySettingsChange (
    self )
```

Public method for calling all the private methods required to apply the settings change.

arg: none

return: none

3.2.2.31 DeleteOldCSV()

```
def Winter20_8INF700.Application.Application.DeleteOldCSV (
    self )
```

Public method to delete the csv because the csv module doesn't have an overwrite mode.

So, the file is deleted and recreated with the new content.

arg: none

return: none

3.2.2.32 Interface()

```
def Winter20_8INF700.Application.Application.Interface (
    self )
```

Public method to initiate the graphical user interface.

The graphical user contain two main elements: a a side bar with buttons to navigate between the interfaces of the fonctionnalities of the application and a QStackedWidget. A QStackedWidget object allow to store multiple widgets and with an index system, the the programmer can select wich widget to show. The user interface use a QStackStackedWidget to select wich fonctionnality's interface to show depending on wich button the user touches.

arg: none

return: none

3.2.2.33 setDate()

```
def Winter20_8INF700.Application.Application.setDate (
    self )
```

Private method to set the content of the date Label.

arg: none

return: none

3.2.2.34 setHour()

```
def Winter20_8INF700.Application.Application.setHour (
    self )
```

Private method to set the content of the hour Label.

arg: none

return: none

3.2.2.35 setWeather()

```
def Winter20_8INF700.Application.Application.setWeather (
    self )
```

Public method to set the weather's data by calling all the private methods for setting the weather data used for the weather-clock fonctionnality's interface.

arg: none

return: none

3.2.2.36 StopSpotifyActualisation()

```
def Winter20_8INF700.Application.Application.StopSpotifyActualisation (
    self )
```

Public method to stop the Spotify's songs informations actualisation functionality.

arg: none

return: none

3.2.2.37 StopWeatherAndHourActualisation()

```
def Winter20_8INF700.Application.Application.StopWeatherAndHourActualisation (
    self )
```

Public method to stop weather, hour and date timers.

arg: none

return: none

3.2.2.38 UpdateSpotify()

```
def Winter20_8INF700.Application.Application.UpdateSpotify (
    self )
```

Public method to update the Spotify functionality's interface.

arg: none

return: none

The documentation for this class was generated from the following file:

- Application.py

3.3 Winter20_8INF700.Location.Location Class Reference

Simple class to get the coordinates of the user's city.

Public Member Functions

- **def __init__**(self)
- **def getLatitude**(self)
Public method to return the latitude the user's city Extract the latitude from the self.__location string.
- **def getLongitude**(self)
Public method to return the longitude the user's city Extract the longitude from the self.__location string.

Private Attributes

- `__ipData`
- `__location`
- `__comaPosition`

3.3.1 Detailed Description

Simple class to get the coordinates of the user's city.

3.3.2 Member Function Documentation

3.3.2.1 `getLatitude()`

```
def Winter20_8INF700.Location.Location.getLatitude (
    self )
```

Public method to return the latitude the user's city Extract the latitude from the self.__location string.

arg: none

return: the latitude of the user's city

3.3.2.2 `getLongitude()`

```
def Winter20_8INF700.Location.Location.getLongitude (
    self )
```

Public method to return the longitude the user's city Extract the longitude from the self.__location string.

arg: none

return: the longitude of the user's city

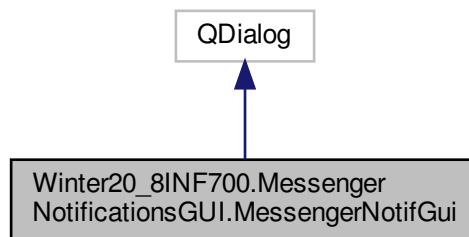
The documentation for this class was generated from the following file:

- `Location.py`

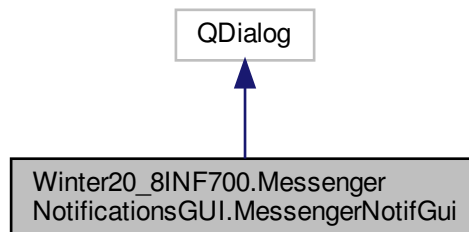
3.4 Winter20_8INF700.MessengerNotificationsGUI.MessengerNotifGui Class Reference

The graphical user interface class of the messenger notification fonctionnality The role of this class is to generate the graphical notifications list for the main interface.

Inheritance diagram for Winter20_8INF700.MessengerNotificationsGUI.MessengerNotifGui:



Collaboration diagram for Winter20_8INF700.MessengerNotificationsGUI.MessengerNotifGui:



Public Member Functions

- def **__init__** (self)
- def **getWidget** (self)

Public method to return notifications widget to the application class' instance for fill the notification's scrollable area by returning all the notification widgets into a single widget list containing all the notifications.
- def **UpdateLists** (self)

Public method used by the application class' instance to actualise the content of the thread list and the notification dictionary list.
- def **UpdateNotificationList** (self)

Public method to update the content of the graphical version of the notification list.
- def **DeleteNotifications** (self)

*Public method used by the application class' instance to empty the graphical and the data notification structures from the **MessengerNotifGui** (p. 21) and UnreadMessageFetcher class instances.*

Public Attributes

- **unreadMessageFetcher**

Private Attributes

- **__mutex**
- **__list**

3.4.1 Detailed Description

The graphical user interface class of the messenger notification fonctionnality The role of this class is to generate the graphical notifications list for the main interface.

3.4.2 Member Function Documentation

3.4.2.1 DeleteNotifications()

```
def Winter20_8INF700.MessengerNotificationsGUI.MessengerNotifGui.DeleteNotifications (
    self )
```

Public method used by the application class' instance to empty the graphical and the data notification structures from the **MessengerNotifGui** (p. 21) and UnreadMessageFetcher class instances.

notification dictionnary list.

arg: none

return: none

3.4.2.2 getWidget()

```
def Winter20_8INF700.MessengerNotificationsGUI.MessengerNotifGui.getWidget (
    self )
```

Public method to return notifications widget to the application class' instance for fill the notification's scrollable area by returning all the notification widgets into a single widget list containing all the notifications.

arg: none

return: __list: a QListWidget (a list for QWidget objects) containing all the notifications.

3.4.2.3 UpdateLists()

```
def Winter20_8INF700.MessengerNotificationsGUI.MessengerNotifGui.UpdateLists (
    self )
```

Public method used by the application class' instance to actualise the content of the thread list and the notification dictionary list.

arg: none

return: none

3.4.2.4 UpdateNotificationList()

```
def Winter20_8INF700.MessengerNotificationsGUI.MessengerNotifGui.UpdateNotificationList (
    self )
```

Public method to update the content of the graphical version of the notification list.

The method uses the size of `__unreadMessageDictionaryList` and the value of `__newUnreadMessageNb` deduce the new unread messages' indexes and loop into the new values add into the `__unreadMessageDictionaryList` to fetch the informations contained into the new dictionaries of the list.

arg: none

return: none

The documentation for this class was generated from the following file:

- MessengerNotificationsGUI.py

3.5 Winter20_8INF700.SpotifyData.Spotify Class Reference

Spotify (p. 23)'s data class for the **Spotify** (p. 23) fonctionnality.

Public Member Functions

- def **__init__** (self)
- def **setAuthToken** (self)
Public method who doesn't need the username and the password as parameters.
- def **setArtists** (self)
Public method to affect the artists' name of the song into the artist attribute of the class.
- def **getAlbum** (self)
Public method to return the album's title for setting the interface's content.
- def **getArtist** (self)
- def **getSongTitle** (self)
- def **getAlbumCoverURL** (self)
- def **setAlbum** (self)
Public method to affect the album's title into the albumTitle attribute of the class.
- def **setSongTitle** (self)
Public method to affect the song's title into the songTitle attribute of the class.
- def **setAlbumCover** (self)
Public method to affect the album cover picture's url into the albumCoverUrl attribute of the class.
- def **CheckSongInfo** (self)
Public method to check if the song's informations change before affecting them into the class' attributes.

Private Member Functions

- `def __setSpotifyAuthToken (self, username, password)`
*Private method to initiate an authorisation token for the **Spotify** (p. 23)'s web player API to the private attribute `spotifyApiAuthorisationToken`.*
- `def __setJSONData (self)`
To set the song's information dictionary into the private attribute `songDict`.
- `def __getAnHeader (self)`
Private method to prepare the request's header.
- `def __getJSONData (self)`
*This function does an HTTP GET request to the **Spotify** (p. 23)'s API to get the JSON dictionary of actually listened song or the last listened song if nothing is playing.*
- `def __FetchArtists (self)`
Fetch the artist's or the artists' name from the song's dictionary.
- `def __FetchAlbum (self)`
Private method to fetch the title of the album from the song dictionary.
- `def __FetchSongTitle (self)`
Private method to fetch the song's title from the song dictionary.
- `def __FetchAlbumCoverURL (self)`
Private method to fetch the album cover picture's URL from the song dictionary.

Private Attributes

- `__songTitle`
- `__artist`
- `__albumTitle`
- `__albumCoverUrl`
- `__requestUrlDict`
- `__songDict`
- `__tokenAutorisationApiSpotify`
- `__requestStatus`
- `__username`
- `__password`

3.5.1 Detailed Description

Spotify (p. 23)'s data class for the **Spotify** (p. 23) fonctionnality.

3.5.2 Member Function Documentation

3.5.2.1 __FetchAlbum()

```
def Winter20_8INF700.SpotifyData.Spotify.__FetchAlbum (
    self ) [private]
```

Private method to fetch the title of the album from the song dictionary.

arg: none

return: albumTitle: The title of the album.

3.5.2.2 __FetchAlbumCoverURL()

```
def Winter20_8INF700.SpotifyData.Spotify.__FetchAlbumCoverURL (
    self ) [private]
```

Private method to fetch the album cover picture's URL from the song dictionary.

arg: none

return: image[1].get('url'): The album cover picture's URL with the following dimensions:300x300.

3.5.2.3 __FetchArtists()

```
def Winter20_8INF700.SpotifyData.Spotify.__FetchArtists (
    self ) [private]
```

Fetch the artist's or the artists' name from the song's dictionary.

arg:

return: artistsName: The name of the artist(s) concatenated into a string (ex: "artistName1, artistName2, artistName3") example: "artistName1, artistName2, artistName3"

3.5.2.4 __FetchSongTitle()

```
def Winter20_8INF700.SpotifyData.Spotify.__FetchSongTitle (
    self ) [private]
```

Private method to fetch the song's title from the song dictionary.

arg: none

return: songTitle: The title of the song.

3.5.2.5 __getAnHeader()

```
def Winter20_8INF700.SpotifyData.Spotify.__getAnHeader (
    self ) [private]
```

Private method to prepare the request's header.

:arg none

:return header: The header for the HTTP request.

Spotify API's tokens always start by "Bearer ", but the module used for generating tokens only return the variable part of a token. The concatenation of the constant part of the token is necessary to generate the header correctly.

3.5.2.6 __getJSONData()

```
def Winter20_8INF700.SpotifyData.Spotify.__getJSONData (
    self ) [private]
```

This function does an HTTP GET request to the **Spotify** (p. 23)'s API to get the JSON dictionary of actually listened song or the last listened song if nothing is playing.

arg: none

return: dict: The JSON dictionary containing the sons's info.

3.5.2.7 __setJSONData()

```
def Winter20_8INF700.SpotifyData.Spotify.__setJSONData (
    self ) [private]
```

To set the song's information dictionary into the private attribute songDict.

arg: none

return: none

3.5.2.8 __setSpotifyAuthToken()

```
def Winter20_8INF700.SpotifyData.Spotify.__setSpotifyAuthToken (
    self,
    username,
    password ) [private]
```

Private method to initiate an authorisation token for the **Spotify** (p. 23)'s web player API to the private attribute spotifyApiAuthorisationToken.

arg: username: the **Spotify** (p. 23)'s username of the user : the **Spotify** (p. 23)'s password of the user

return: none

3.5.2.9 CheckSongInfo()

```
def Winter20_8INF700.SpotifyData.Spotify.CheckSongInfo (
    self )
```

Public method to check if the song's informations change before affecting them into the class' attributes.

Maybe removing it (if I have the time) and emplement directly something into the interface for checking if the URL change to avoid the **Spotify** (p. 23) functionality's interface if the song doesn't changed yet.

arg: none

return: none

3.5.2.10 `getAlbum()`

```
def Winter20_8INF700.SpotifyData.Spotify.getAlbum (
    self )
```

Public method to return the album's title for setting the interface's content.

arg: none

return: albumTitle: The title of the album.

3.5.2.11 `setAlbum()`

```
def Winter20_8INF700.SpotifyData.Spotify.setAlbum (
    self )
```

Public method to affect the album's title into the albumTitle attribute of the class.

arg: none

return: none

3.5.2.12 `setAlbumCover()`

```
def Winter20_8INF700.SpotifyData.Spotify.setAlbumCover (
    self )
```

Public method to affect the album cover picture's url into the albumCoverUrl attribute of the class.

arg: none

return: none

3.5.2.13 `setArtists()`

```
def Winter20_8INF700.SpotifyData.Spotify.setArtists (
    self )
```

Public method to affect the artists' name of the song into the artist attribute of the class.

arg: none

return: none

3.5.2.14 `setAuthToken()`

```
def Winter20_8INF700.SpotifyData.Spotify.setAuthToken (
    self )
```

Public method who doesn't need the username and the password as parameters.

The method sets a new authorisation token by calling the private method `setSpotifyAuthToken`. Each token is valid for a 60 minutes period.

arg: none

return: none

3.5.2.15 setSongTitle()

```
def Winter20_8INF700.SpotifyData.Spotify.setSongTitle (
    self )
```

Public method to affect the song's title into the songTitle attribute of the class.

arg: none

return: none

The documentation for this class was generated from the following file:

- SpotifyData.py

3.6 Winter20_8INF700.MessengerUnreadMessageFetcher.UnreadMessageFetcher Class Reference

The data class for the messenger notification fonctionnality The role of this class is to fetch the notifications' data.

Public Member Functions

- def **__init__** (self, username, password)
- def **EmptyLists** (self)
Public method to empty the list of notifications.
- def **resetNewUnreadMessageNb** (self)
Public method to reset the unread message counter when emptying the notification list.
- def **setLists** (self)
Public method to update the __threadList and __unreadMessageLis attributes.
- def **getTotalUnreadMessageNumber** (self)
Public method to obtain the size of the list Usefull? arg: none.
- def **getNewUnreadMessageNumber** (self)
Public method to get the number of new unread messages fetched during an iteration of the __setUnReadMessage↵ List method.
- def **getAuthorName** (self, dictIndex)
Public method to return the message author's name into the interface.
- def **getConversationPic** (self, dictIndex)
Public method to return the url of the message author's profile picture for loading it into the interface.
- def **getMessage** (self, dictIndex)
Public method to return the message into the interface.

Private Member Functions

- def **__setThreadList** (self)
Private method to affect the list of the ten most recent messenger conversations of the user into the __threadlist attribute.
- def **__setUnreadMessagesList** (self)
Private method for setting the __unreadMessageDictionnaryList attribute by storing the unread messages dictionnaries' into the list.
- def **__getUnReadMessageDict** (self, threadIndex)
Private method to get the unread message dictionary of the selected conversation.
- def **__getaUserName** (self, userId)
Private method to get the name of the user who sent the message in a group conversation.

Private Attributes

- `__newUnreadMessageNb`
- `__unreadMessageDictionaryList`
- `__threadList`
- `__unRodeThread`
- `__client`

3.6.1 Detailed Description

The data class for the messenger notification fonctionnality The role of this class is to fetch the notifications' data.

3.6.2 Member Function Documentation

3.6.2.1 `__getUserName()`

```
def Winter20_8INF700.MessengerUnreadMessageFetcher.UnreadMessageFetcher.__getUserName (
    self,
    userId ) [private]
```

Private method to get the name of the user who sent the message in a group conversation.

Messenger uses the real name of the other user in the name attribut of the dictionary for user to user conversation and group's name for group conversation. To specify who sent the message, this method fetches the profile's informations of the user who sends the message by using is facebook id (each message dictionary returned by the client object by the fbchat module contains the id of the user who sent the message) and extract his name to return it to the `__getUnReadMessageDict` method.

arg: `userId`: The Facebook user's id of the user who sent the message.

return: `name`: The name of the user who sent the message.

3.6.2.2 `__getUnReadMessageDict()`

```
def Winter20_8INF700.MessengerUnreadMessageFetcher.UnreadMessageFetcher.__getUnReadMessageDict
(
    self,
    threadIndex ) [private]
```

Private method to get the unread message dictionary of the selected conversation.

arg: `threadIndex`: the index of the conversation.

return: `unreadMessageDict`: the unread message dictionary of the selected conversation.

3.6.2.3 __setThreadList()

```
def Winter20_8INF700.MessengerUnreadMessageFetcher.UnreadMessageFetcher.__setThreadList (
    self ) [private]
```

Private method to affect the list of the ten most recent messenger conversations of the user into the `__threadlist` attribute.

arg: none

return: none

3.6.2.4 __setUnreadMessagesList()

```
def Winter20_8INF700.MessengerUnreadMessageFetcher.UnreadMessageFetcher.__setUnreadMessages↵
List (
    self ) [private]
```

Private method for setting the `__unreadMessageDictionaryList` attribute by storing the unread messages dictionaries' into the list.

arg: none

return: none

3.6.2.5 EmptyLists()

```
def Winter20_8INF700.MessengerUnreadMessageFetcher.UnreadMessageFetcher.EmptyLists (
    self )
```

Public method to empty the list of notifications.

arg: none

return: none

To avoid to refill the list with already fetched notification, each conversation with at least one unread message the size of the notification list must be superior to zero to empty the list.

3.6.2.6 getAuthorName()

```
def Winter20_8INF700.MessengerUnreadMessageFetcher.UnreadMessageFetcher.getAuthorName (
    self,
    dictIndex )
```

Public method to return the message author's name into the interface.

arg: dictIndex: the index of the selected message dict into the `__unreadMessageDictionaryList` attribute.

return: self.__unreadMessageDictionaryList[dictIndex].get('name'): The message author's name.

3.6.2.7 getConversationPic()

```
def Winter20_8INF700.MessengerUnreadMessageFetcher.UnreadMessageFetcher.getConversationPic (
    self,
    dictIndex )
```

Public method to return the url of the message author's profile picture for loading it into the interface.

arg: dictIndex: the index of the selected message dict into the `__unreadMessageDictionaryList` attribute.

return: `self.__unreadMessageDictionaryList[dictIndex].get('photo')`: The url of the message author's profile picture.

3.6.2.8 getMessage()

```
def Winter20_8INF700.MessengerUnreadMessageFetcher.UnreadMessageFetcher.getMessage (
    self,
    dictIndex )
```

Public method to return the message into the interface.

arg: dictIndex: the index of the selected message dict into the `__unreadMessageDictionaryList` attribute.

return: `self.__unreadMessageDictionaryList[dictIndex].get('message')`: The unread message.

3.6.2.9 getNewUnreadMessageNumber()

```
def Winter20_8INF700.MessengerUnreadMessageFetcher.UnreadMessageFetcher.getNewUnreadMessage↵
Number (
    self )
```

Public method to get the number of new unread messages fetched during an iteration of the `__setUnRead↵MessageList` method.

#Usefull? Must see that at the last refactorings.

arg: none

return: `__newUnReadMessageNb`: The number of new unread messages fetched during an iteration of the `__↵setUnReadMessageList` method.

3.6.2.10 getTotalUnreadMessageNumber()

```
def Winter20_8INF700.MessengerUnreadMessageFetcher.UnreadMessageFetcher.getTotalUnreadMessage↵
Number (
    self )
```

Public method to optain the size of the list Usefull? arg: none.

return: `d`: The size of the `__unreadMessageDictionaryList` attribute.

3.6.2.11 resetNewUnreadMessageNb()

```
def Winter20_8INF700.MessengerUnreadMessageFetcher.UnreadMessageFetcher.resetNewUnreadMessageNb (
    self )
```

Public method to reset the unread message counter when emptying the notification list.

arg: none

return: none

3.6.2.12 setLists()

```
def Winter20_8INF700.MessengerUnreadMessageFetcher.UnreadMessageFetcher.setLists (
    self )
```

Public method to update the __threadList and __unreadMessageLis attributes.

arg: none

return: none

The documentation for this class was generated from the following file:

- MessengerUnreadMessageFetcher.py

3.7 Winter20_8INF700.WeatherClock.Weather Class Reference

Public Member Functions

- def **__init__** (self)
- def **Update** (self)
Public method to refresh the weather dictionnaires.
- def **setDailyForecastData** (self)
Public method to affect the dailyforecast's dictionnary into the dailyForecast dictionnary by extracting it from the weather forecast object.
- def **getTemp** (self)
Public method used to return the actual temp into the interface.
- def **getFeeledTemp** (self)
Public method used to return the actual feeled temp into the interface.
- def **getWeatherDesc** (self)
Public method used to return the actual weather description into the interface.
- def **getPrecipitationProb** (self)
Public method used to return the actual probability of precipitation into the interface.
- def **getPrecipitationIntensity** (self)
Public method used to return the actual precipitation intensity into the interface.
- def **getPrecipitationType** (self)
Public method used to return the actual precipitation type into the interface.
- def **getWindSpeed** (self)
Public method used to return the actual wind speed into the interface.
- def **getWindDir** (self)
Public method used to return the actual wind direction into the interface.
- def **getDailyMaxFeeledTemp** (self)
Public method used to return the daily maximal feeled temperature into the interface.
- def **getDailyMinFeeledTemp** (self)
Public method used to return the daily minimal feeled temperature into the interface.

Private Attributes

- **__positionLongitude**
- **__positionLatitude**
- **__weather**
- **__actualWeatherData**
ctualise forecast object's data with the canadian units.
- **__dailyForecast**

3.7.1 Member Function Documentation

3.7.1.1 getDailyMaxFeeledTemp()

```
def Winter20_8INF700.WeatherClock.Weather.getDailyMaxFeeledTemp (
    self )
```

Public method used to return the daily maximal feeled temperature into the interface.

arg: none

return: actualWeatherData.get('appparentTemperatureMax'): Daily maximal feeled temperature.

3.7.1.2 getDailyMinFeeledTemp()

```
def Winter20_8INF700.WeatherClock.Weather.getDailyMinFeeledTemp (
    self )
```

Public method used to return the daily minimal feeled temperature into the interface.

arg: none

return: actualWeatherData.get('appparentTemperatureMin'): Daily minimal feeled temperature.

3.7.1.3 getFeeledTemp()

```
def Winter20_8INF700.WeatherClock.Weather.getFeeledTemp (
    self )
```

Public method used to return the actual feeled temp into the interface.

arg: none

return: actualWeatherData.get('appparentTemperature'): Actual feeled temperature.

3.7.1.4 getPrecipitationIntensity()

```
def Winter20_8INF700.WeatherClock.Weather.getPrecipitationIntensity (
    self )
```

Public method used to return the actual precipitation intensity into the interface.

arg: none

return: actualWeatherData.get('precipIntensity'): Actual precipitation intensity.

3.7.1.5 getPrecipitationProb()

```
def Winter20_8INF700.WeatherClock.Weather.getPrecipitationProb (
    self )
```

Public method used to return the actual probability of precipitation into the interface.

arg: none

return: actualWeatherData.get('precipProbability'): Actual precipitation probability.

3.7.1.6 getPrecipitationType()

```
def Winter20_8INF700.WeatherClock.Weather.getPrecipitationType (
    self )
```

Public method used to return the actual precipitation type into the interface.

arg: none

return: actualWeatherData.get('precipitationType'): Actual precipitation type.

3.7.1.7 getTemp()

```
def Winter20_8INF700.WeatherClock.Weather.getTemp (
    self )
```

Public method used to return the actual temp into the interface.

arg: none

return: actualWeatherData.get('temperature'): Actual temperature.

3.7.1.8 getWeatherDesc()

```
def Winter20_8INF700.WeatherClock.Weather.getWeatherDesc (
    self )
```

Public method used to return the actual weather description into the interface.

arg: none

return: actualWeatherData.get('summary'): Actual weather's description.

3.7.1.9 getWindDir()

```
def Winter20_8INF700.WeatherClock.Weather.getWindDir (
    self )
```

Public method used to return the actual wind direction into the interface.

arg: none

return: actualWeatherData.get('windBearing'): Actual wind direction.

3.7.1.10 getWindSpeed()

```
def Winter20_8INF700.WeatherClock.Weather.getWindSpeed (
    self )
```

Public method used to return the actual wind speed into the interface.

arg: none

return: actualWeatherData.get('windSpeed'): Actual wind speed.

3.7.1.11 setDailyForecastData()

```
def Winter20_8INF700.WeatherClock.Weather.setDailyForecastData (
    self )
```

Public method to affect the dailyforecast's dictionnary into the dailyForecast dictionnary by extracting it from the weather forecast object.

arg: none

return: none

3.7.1.12 Update()

```
def Winter20_8INF700.WeatherClock.Weather.Update (
    self )
```

Public method to refresh the weather dictionnaires.

A weather object has two dictionnaires: a dictionnary containing the actual weather's data and another one containing the daily forecast's data. The daily forecast data are used to have the daily maximal and the minimal feeled temperature.

arg: none

return: none

3.7.2 Member Data Documentation

3.7.2.1 `__actualWeatherData`

`Winter20_8INF700.WeatherClock.Weather.__actualWeatherData` [private]

ctualise forecast object's data with the canadian units.

The canadian units are those in the Internationnal Units System but `wndSpeed` and `windGust` are expressed in kilometer by hour.

source: <https://pypi.org/project/darkskylib/>

The documentation for this class was generated from the following file:

- `WeatherClock.py`

Index

- __ButtonsAction
 - Winter20_8INF700::Application::Application, 10
- __ChangeBackLightIntensity
 - Winter20_8INF700::Application::Application, 11
- __ChangeSettingsListsValues
 - Winter20_8INF700::Application::Application, 11
- __ExportSettingsToCSV
 - Winter20_8INF700::Application::Application, 11
- __FetchAlbum
 - Winter20_8INF700::SpotifyData::Spotify, 24
- __FetchAlbumCoverURL
 - Winter20_8INF700::SpotifyData::Spotify, 24
- __FetchArtists
 - Winter20_8INF700::SpotifyData::Spotify, 25
- __FetchSongTitle
 - Winter20_8INF700::SpotifyData::Spotify, 25
- __FillLists
 - Winter20_8INF700::Application::Application, 11
- __ImportAlarmSettingsToGUI
 - Winter20_8INF700::Application::Application, 12
- __InitiateBacklight
 - Winter20_8INF700::Application::Application, 13
- __StartMessengerTimer
 - Winter20_8INF700::Application::Application, 16
- __StopMessengerTimer
 - Winter20_8INF700::Application::Application, 16
- __UpdateMessengerNotificationInterface
 - Winter20_8INF700::Application::Application, 16
- __UpdateNotification
 - Winter20_8INF700::Application::Application, 16
- __actualWeatherData
 - Winter20_8INF700::WeatherClock::Weather, 35
- __getAnHeader
 - Winter20_8INF700::SpotifyData::Spotify, 25
- __getJSONData
 - Winter20_8INF700::SpotifyData::Spotify, 25
- __getUnReadMessageDict
 - Winter20_8INF700::MessengerUnreadMessage↔
Fetcher::UnreadMessageFetcher, 29
- __getUserName
 - Winter20_8INF700::MessengerUnreadMessage↔
Fetcher::UnreadMessageFetcher, 29
- __initialiseSongInfoUpdate
 - Winter20_8INF700::Application::Application, 12
- __initialiseWeatherDateHourUpdate
 - Winter20_8INF700::Application::Application, 12
- __initiateAlarmTimer
 - Winter20_8INF700::Application::Application, 12
- __initiateHourAndWeatherUpdate
 - Winter20_8INF700::Application::Application, 13
- __initiateMessengerTimers
 - Winter20_8INF700::Application::Application, 13
- __setActualPrecipIntensity
 - Winter20_8INF700::Application::Application, 13
- __setActualPrecipType
 - Winter20_8INF700::Application::Application, 14
- __setActualPrecipitationProb
 - Winter20_8INF700::Application::Application, 13
- __setActualTemp
 - Winter20_8INF700::Application::Application, 14
- __setActualWindDir
 - Winter20_8INF700::Application::Application, 14
- __setActualWindSpeed
 - Winter20_8INF700::Application::Application, 14
- __setAlarmTimerTime
 - Winter20_8INF700::Application::Application, 15
- __setAlarmTimers
 - Winter20_8INF700::Application::Application, 14
- __setDailyMaxFeeledTemp
 - Winter20_8INF700::Application::Application, 15
- __setDailyMinFeeledTemp
 - Winter20_8INF700::Application::Application, 15
- __setFeeledTemp
 - Winter20_8INF700::Application::Application, 15
- __setJSONData
 - Winter20_8INF700::SpotifyData::Spotify, 26
- __setSpotifyAuthToken
 - Winter20_8INF700::SpotifyData::Spotify, 26
- __setThreadList
 - Winter20_8INF700::MessengerUnreadMessage↔
Fetcher::UnreadMessageFetcher, 29
- __setUnreadMessagesList
 - Winter20_8INF700::MessengerUnreadMessage↔
Fetcher::UnreadMessageFetcher, 30
- __setWeatherDesc
 - Winter20_8INF700::Application::Application, 15
- Alarm
 - Winter20_8INF700::Application::Application, 17
- ApplySettingsChange
 - Winter20_8INF700::Application::Application, 17
- CheckSongInfo
 - Winter20_8INF700::SpotifyData::Spotify, 26
- DeleteNotifications
 - Winter20_8INF700::MessengerNotificationsGUI↔
::MessengerNotifGui, 22
- DeleteOldCSV

- Winter20_8INF700::Application::Application, 17
- EmptyLists
 - Winter20_8INF700::MessengerUnreadMessage↔
Fetcher::UnreadMessageFetcher, 30
- getAlbum
 - Winter20_8INF700::SpotifyData::Spotify, 26
- getAuthorName
 - Winter20_8INF700::MessengerUnreadMessage↔
Fetcher::UnreadMessageFetcher, 30
- getConversationPic
 - Winter20_8INF700::MessengerUnreadMessage↔
Fetcher::UnreadMessageFetcher, 30
- getDailyMaxFeeledTemp
 - Winter20_8INF700::WeatherClock::Weather, 33
- getDailyMinFeeledTemp
 - Winter20_8INF700::WeatherClock::Weather, 33
- getFeeledTemp
 - Winter20_8INF700::WeatherClock::Weather, 33
- getLatitude
 - Winter20_8INF700::Location::Location, 20
- getLongitude
 - Winter20_8INF700::Location::Location, 20
- getMessage
 - Winter20_8INF700::MessengerUnreadMessage↔
Fetcher::UnreadMessageFetcher, 31
- getNewUnreadMessageNumber
 - Winter20_8INF700::MessengerUnreadMessage↔
Fetcher::UnreadMessageFetcher, 31
- getPrecipitationIntensity
 - Winter20_8INF700::WeatherClock::Weather, 33
- getPrecipitationProb
 - Winter20_8INF700::WeatherClock::Weather, 34
- getPrecipitationType
 - Winter20_8INF700::WeatherClock::Weather, 34
- getTemp
 - Winter20_8INF700::WeatherClock::Weather, 34
- getTotalUnreadMessageNumber
 - Winter20_8INF700::MessengerUnreadMessage↔
Fetcher::UnreadMessageFetcher, 31
- getWeatherDesc
 - Winter20_8INF700::WeatherClock::Weather, 34
- getWidget
 - Winter20_8INF700::MessengerNotificationsGUI↔
::MessengerNotifGui, 22
- getWindDir
 - Winter20_8INF700::WeatherClock::Weather, 34
- getWindSpeed
 - Winter20_8INF700::WeatherClock::Weather, 35
- Interface
 - Winter20_8INF700::Application::Application, 17
- resetNewUnreadMessageNb
 - Winter20_8INF700::MessengerUnreadMessage↔
Fetcher::UnreadMessageFetcher, 31
- setAlbum
 - Winter20_8INF700::SpotifyData::Spotify, 27
- setAlbumCover
 - Winter20_8INF700::SpotifyData::Spotify, 27
- setArtists
 - Winter20_8INF700::SpotifyData::Spotify, 27
- setAuthToken
 - Winter20_8INF700::SpotifyData::Spotify, 27
- setDailyForecastData
 - Winter20_8INF700::WeatherClock::Weather, 35
- setDate
 - Winter20_8INF700::Application::Application, 18
- setHour
 - Winter20_8INF700::Application::Application, 18
- setLists
 - Winter20_8INF700::MessengerUnreadMessage↔
Fetcher::UnreadMessageFetcher, 32
- setSongTitle
 - Winter20_8INF700::SpotifyData::Spotify, 27
- setWeather
 - Winter20_8INF700::Application::Application, 18
- StopSpotifyActualisation
 - Winter20_8INF700::Application::Application, 18
- StopWeatherAndHourActualisation
 - Winter20_8INF700::Application::Application, 19
- Update
 - Winter20_8INF700::WeatherClock::Weather, 35
- UpdateLists
 - Winter20_8INF700::MessengerNotificationsGUI↔
::MessengerNotifGui, 22
- UpdateNotificationList
 - Winter20_8INF700::MessengerNotificationsGUI↔
::MessengerNotifGui, 23
- UpdateSpotify
 - Winter20_8INF700::Application::Application, 19
- Winter20_8INF700.Alarm.Alarm, 5
- Winter20_8INF700.Application.Application, 6
- Winter20_8INF700.Location.Location, 19
- Winter20_8INF700.MessengerNotificationsGUI.↔
MessengerNotifGui, 21
- Winter20_8INF700.MessengerUnreadMessage↔
Fetcher.UnreadMessageFetcher, 28
- Winter20_8INF700.SpotifyData.Spotify, 23
- Winter20_8INF700.WeatherClock.Weather, 32
- Winter20_8INF700::Application::Application
 - __ButtonsAction, 10
 - __ChangeBackLightIntensity, 11
 - __ChangeSettingsListsValues, 11
 - __ExportSettingsToCSV, 11
 - __FillLists, 11
 - __ImportAlarmSettingsToGUI, 12
 - __InitiateBacklight, 13
 - __StartMessengerTimer, 16
 - __StopMessengerTimer, 16
 - __UpdateMessengerNotificationInterface, 16
 - __UpdateNotification, 16
 - __initialiseSongInfoUpdate, 12
 - __initialiseWeatherDateHourUpdate, 12

- __initiateAlarmTimer, 12
- __initiateHourAndWeatherUpdate, 13
- __initiateMessengerTimers, 13
- __setActualPrecipIntensity, 13
- __setActualPrecipType, 14
- __setActualPrecipitationProb, 13
- __setActualTemp, 14
- __setActualWindDir, 14
- __setActualWindSpeed, 14
- __setAlarmTimerTime, 15
- __setAlarmTimers, 14
- __setDailyMaxFeeledTemp, 15
- __setDailyMinFeeledTemp, 15
- __setFeeledTemp, 15
- __setWeatherDesc, 15
- Alarm, 17
- ApplySettingsChange, 17
- DeleteOldCSV, 17
- Interface, 17
- setDate, 18
- setHour, 18
- setWeather, 18
- StopSpotifyActualisation, 18
- StopWeatherAndHourActualisation, 19
- UpdateSpotify, 19
- Winter20_8INF700::Location::Location
 - getLatitude, 20
 - getLongitude, 20
- Winter20_8INF700::MessengerNotificationsGUI::←
 - MessengerNotifGui
 - DeleteNotifications, 22
 - getWidget, 22
 - UpdateLists, 22
 - UpdateNotificationList, 23
- Winter20_8INF700::MessengerUnreadMessage←
 - Fetcher::UnreadMessageFetcher
 - __getUnReadMessageDict, 29
 - __getUserName, 29
 - __setThreadList, 29
 - __setUnreadMessagesList, 30
 - EmptyLists, 30
 - getAuthorName, 30
 - getConversationPic, 30
 - getMessage, 31
 - getNewUnreadMessageNumber, 31
 - getTotalUnreadMessageNumber, 31
 - resetNewUnreadMessageNb, 31
 - setLists, 32
- Winter20_8INF700::SpotifyData::Spotify
 - __FetchAlbum, 24
 - __FetchAlbumCoverURL, 24
 - __FetchArtists, 25
 - __FetchSongTitle, 25
 - __getAnHeader, 25
 - __getJSONData, 25
 - __setJSONData, 26
 - __setSpotifyAuthToken, 26
 - CheckSongInfo, 26
 - getAlbum, 26
 - setAlbum, 27
 - setAlbumCover, 27
 - setArtists, 27
 - setAuthToken, 27
 - setSongTitle, 27
- Winter20_8INF700::WeatherClock::Weather
 - __actualWeatherData, 35
 - getDailyMaxFeeledTemp, 33
 - getDailyMinFeeledTemp, 33
 - getFeeledTemp, 33
 - getPrecipitationIntensity, 33
 - getPrecipitationProb, 34
 - getPrecipitationType, 34
 - getTemp, 34
 - getWeatherDesc, 34
 - getWindDir, 34
 - getWindSpeed, 35
 - setDailyForecastData, 35
 - Update, 35