

Gestor de Fotografias Premium

2.00

Generated by Doxygen 1.8.14

Contents

1	Module Index	1
1.1	Modules	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	Module Documentation	7
4.1)	7
5	Class Documentation	11
5.1	Album Class Reference	11
5.1.1	Detailed Description	12
5.1.2	Member Function Documentation	12
5.1.2.1	AddAlbum()	12
5.1.2.2	AddFolderName()	12
5.1.2.3	AddId()	12
5.1.2.4	AddName()	13
5.1.2.5	GetDescription()	13
5.1.2.6	GetFolder()	13
5.1.2.7	GetId()	14
5.1.2.8	GetName()	14
5.1.2.9	GetPageType()	14

5.1.2.10	GetPageTypeString()	14
5.2	ourDB::albums Struct Reference	15
5.2.1	Detailed Description	15
5.3	ourDB::association Struct Reference	15
5.3.1	Detailed Description	15
5.4	mainwindow Class Reference	16
5.4.1	Detailed Description	16
5.5	MainWindow Class Reference	16
5.5.1	Member Function Documentation	17
5.5.1.1	SentItemSelectedId	17
5.5.1.2	SentItemSelectedName	17
5.5.1.3	SentPageType	17
5.5.1.4	SentState	18
5.5.1.5	SentStateRootDir	18
5.5.1.6	SlotMoveSelectedPage	18
5.5.1.7	SlotMoveSelectedPhoto	18
5.5.1.8	SlotNewAlbum	19
5.5.1.9	SlotNewPageOther	19
5.5.1.10	SlotNewPageParty	19
5.5.1.11	SlotNewPageThingPeople	20
5.5.1.12	SlotNewPageTrip	20
5.6	ourDB::other Struct Reference	20
5.6.1	Detailed Description	21
5.7	Other Class Reference	21
5.7.1	Detailed Description	21
5.7.2	Member Function Documentation	21
5.7.2.1	AddPageOther()	21
5.7.2.2	GetDateEnd()	22
5.7.2.3	GetDateStart()	22
5.8	ourDB Class Reference	22

5.8.1	Detailed Description	26
5.8.2	Member Function Documentation	26
5.8.2.1	buildDB()	26
5.8.2.2	checkIfAlbumIsEmpty()	26
5.8.2.3	checkIfPagelsEmpty()	27
5.8.2.4	getAlbumDescription()	27
5.8.2.5	getAlbumFolderName()	27
5.8.2.6	getAlbumIdFromPage()	28
5.8.2.7	getAlbumLastID()	28
5.8.2.8	getAlbumName()	28
5.8.2.9	getAlbumPageType()	29
5.8.2.10	getAlbuns()	29
5.8.2.11	getAlbunsFromPageType()	29
5.8.2.12	getAllPhotos()	30
5.8.2.13	getAssociationAlbum()	30
5.8.2.14	getAssociationPage()	31
5.8.2.15	getAssociationPhoto()	31
5.8.2.16	getNumberOfAlbuns()	31
5.8.2.17	getNumberOfAllPhotos()	32
5.8.2.18	getNumberOfPagesFromAlbum()	32
5.8.2.19	getNumberPhotosFromAlbum()	32
5.8.2.20	getNumberPhotosFromPage()	33
5.8.2.21	getOtherPagesToShow()	33
5.8.2.22	getPageDescription()	33
5.8.2.23	getPageFolderName()	34
5.8.2.24	getPageIdFromPhoto()	34
5.8.2.25	getPageLastID()	34
5.8.2.26	getPageName()	35
5.8.2.27	getPages()	35
5.8.2.28	getPagesToShow()	35

5.8.2.29	getPartyPagesToShow()	36
5.8.2.30	getPeople()	36
5.8.2.31	getPhotoDate()	36
5.8.2.32	getPhotoDateFromAlbum()	37
5.8.2.33	getPhotoDateFromPage()	37
5.8.2.34	getPhotoExtension()	37
5.8.2.35	getPhotoHeight()	38
5.8.2.36	getPhotoInformation()	38
5.8.2.37	getPhotoLastID()	38
5.8.2.38	getPhotoName()	39
5.8.2.39	getPhotos()	39
5.8.2.40	getPhotosAssociatedFromPerson()	39
5.8.2.41	getPhotosFromAlbum()	40
5.8.2.42	getPhotosFromDate()	40
5.8.2.43	getPhotosFromPage() [1/2]	40
5.8.2.44	getPhotosFromPage() [2/2]	41
5.8.2.45	getPhotoWidth()	41
5.8.2.46	getTripPagesToShow()	42
5.8.2.47	removeAlbum()	42
5.8.2.48	removePage()	42
5.8.2.49	removePhoto()	43
5.8.2.50	setAlbum()	43
5.8.2.51	setAlbumDescription()	44
5.8.2.52	setAlbumFolderName()	44
5.8.2.53	setAlbumName()	44
5.8.2.54	setAssociation()	45
5.8.2.55	setAssociationAlbum()	45
5.8.2.56	setAssociationPage()	46
5.8.2.57	setMovePage()	46
5.8.2.58	setMovePhoto()	46

5.8.2.59	setPage()	47
5.8.2.60	setPageOther()	47
5.8.2.61	setPageParty()	48
5.8.2.62	setPageTrip()	48
5.8.2.63	setPerson()	49
5.8.2.64	setPhoto()	49
5.9	Page Class Reference	50
5.9.1	Detailed Description	51
5.9.2	Member Function Documentation	51
5.9.2.1	AddFolderName()	51
5.9.2.2	AddId()	51
5.9.2.3	AddName()	51
5.9.2.4	GetDescription()	52
5.9.2.5	GetFolder()	52
5.9.2.6	GetId()	52
5.9.2.7	GetName()	52
5.10	ourDB::pages Struct Reference	53
5.10.1	Detailed Description	53
5.11	Party Class Reference	53
5.11.1	Detailed Description	53
5.11.2	Member Function Documentation	54
5.11.2.1	AddPageParty()	54
5.11.2.2	GetDate()	54
5.11.2.3	GetPartyType()	54
5.12	ourDB::party Struct Reference	55
5.12.1	Detailed Description	55
5.13	ourDB::person Struct Reference	55
5.13.1	Detailed Description	55
5.14	Person Class Reference	55
5.14.1	Detailed Description	56

5.14.2	Member Function Documentation	56
5.14.2.1	AddPerson()	56
5.14.2.2	GetBirthDayDate()	56
5.14.2.3	GetGender()	57
5.14.2.4	GetName()	57
5.14.2.5	GetRelationship()	57
5.15	Photo Class Reference	57
5.15.1	Detailed Description	58
5.15.2	Member Function Documentation	58
5.15.2.1	AddExtension()	58
5.15.2.2	AddId()	58
5.15.2.3	AddName()	59
5.15.2.4	GetExtension()	59
5.15.2.5	GetId()	59
5.15.2.6	GetName()	60
5.16	ourDB::photos Struct Reference	60
5.16.1	Detailed Description	60
5.17	ShowFullScreen Class Reference	60
5.17.1	Member Function Documentation	61
5.17.1.1	keyPressEvent()	61
5.18	showfullscreen Class Reference	61
5.18.1	Detailed Description	61
5.19	State Class Reference	62
5.19.1	Detailed Description	62
5.19.2	Member Function Documentation	62
5.19.2.1	ExistDir()	62
5.19.2.2	GetDir()	63
5.19.2.3	GetDirPathString()	63
5.19.2.4	GetRootDir()	63
5.19.2.5	GetState()	64

5.19.2.6	MakeDir()	64
5.19.2.7	NewDir()	65
5.19.2.8	NextDir()	65
5.19.2.9	NextState()	65
5.19.2.10	PreviousDir()	66
5.19.2.11	PreviousState()	66
5.19.2.12	RmDir()	66
5.19.2.13	RmFile()	66
5.20	ThingPeople Class Reference	67
5.20.1	Detailed Description	67
5.20.2	Member Function Documentation	67
5.20.2.1	AddPageThingPeople()	67
5.21	ourDB::trip Struct Reference	68
5.21.1	Detailed Description	68
5.22	Trip Class Reference	68
5.22.1	Detailed Description	69
5.22.2	Member Function Documentation	69
5.22.2.1	AddPageTrip()	69
5.22.2.2	GetDateEnd()	69
5.22.2.3	GetDateStart()	70
5.23	uiaddalbum Class Reference	70
5.23.1	Detailed Description	70
5.24	UiAddAlbum Class Reference	70
5.24.1	Member Function Documentation	71
5.24.1.1	SentAlbum	71
5.25	uiaddpage Class Reference	71
5.25.1	Detailed Description	71
5.26	UiAddPage Class Reference	71
5.26.1	Member Function Documentation	72
5.26.1.1	SentPage	72

5.26.1.2	SentPageOther	72
5.26.1.3	SentPageParty	73
5.26.1.4	SentPageThingPeople	73
5.26.1.5	SentPageTrip	73
5.26.1.6	SlotGetType	74
5.27	uiaddperson Class Reference	74
5.27.1	Detailed Description	74
5.28	UiAddPerson Class Reference	74
5.29	uiassociate Class Reference	75
5.29.1	Detailed Description	75
5.30	UiAssociate Class Reference	75
5.30.1	Member Function Documentation	75
5.30.1.1	ReceiveIds()	75
5.30.1.2	SlotGetState	76
5.31	uimodify Class Reference	76
5.31.1	Detailed Description	76
5.32	uimodifyalbum Class Reference	76
5.32.1	Detailed Description	76
5.33	UiModifyAlbum Class Reference	77
5.33.1	Member Function Documentation	77
5.33.1.1	AddAlbumId()	77
5.33.1.2	ChangeNameDesc()	77
5.34	UiMove Class Reference	78
5.34.1	Member Function Documentation	78
5.34.1.1	GetItemId()	78
5.34.1.2	SentSelectedItem	79
5.35	UiSlideShow Class Reference	79
5.35.1	Member Function Documentation	80
5.35.1.1	SlotGetAudioFile	80
5.35.1.2	SlotGetPhotosToShow	80
5.35.1.3	SlotGetRootDir	80
5.35.1.4	SlotGetTime	81
5.36	uislideshow Class Reference	81
5.36.1	Detailed Description	81
5.37	uislideshowaddimage Class Reference	81
5.37.1	Detailed Description	81
5.38	UiSlideShowAddImage Class Reference	82
5.38.1	Member Function Documentation	82
5.38.1.1	SentPhotosToSlideShow	82
5.38.1.2	SentRootDir	83
5.38.1.3	SentTime	83
5.38.1.4	SlotGetStateRootDir	83

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

)	7
-------------	---

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Album	11
ourDB::albums	15
ourDB::association	15
mainwindow	16
ourDB::other	20
ourDB	22
Page	50
Other	21
Party	53
ThingPeople	67
Trip	68
ourDB::pages	53
ourDB::party	55
ourDB::person	55
Person	55
Photo	57
ourDB::photos	60
QDialog	
UiAddAlbum	70
UiAddPage	71
UiAddPerson	74
UiAssociate	75
UiModifyAlbum	77
UiMove	78
UiSlideShow	79
UiSlideShowAddImage	82
QLabel	
ShowFullScreen	60
QMainWindow	
MainWindow	16
showfullscreen	61
State	62
ourDB::trip	68
uiaddalbum	70
uiaddpage	71

uiaddperson	74
uiassociate	75
uimodify	76
uimodifyalbum	76
uislideshow	81
uislideshowaddimage	81

Chapter 3

Class Index

3.1 Class List

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

Album	11
ourDB::albuns	
The albuns struct stores the id and the name of albuns returned from queries	15
ourDB::association	
The association struct stores an association between a photo and a person using their ids returned from queries	15
mainwindow	16
MainWindow	16
ourDB::other	
The other struct stores attributes from a struct page, start date and finish date returned from queries	20
Other	
The Other class	21
ourDB	22
Page	50
ourDB::pages	
The pages struct stores the id, name and description of pages returned from queries	53
Party	
The Party class	53
ourDB::party	
The party struct stores attributes from a struct page, date and the type of party returned from queries	55
ourDB::person	
The person struct stores the id and the name of people returned from queries	55
Person	55
Photo	57
ourDB::photos	
The photos struct stores the id, name, date and the resolution (width x height) of photos returned from queries	60
ShowFullScreen	60
showfullscreen	61
State	62
ThingPeople	
The ThingPeople class	67

ourDB::trip	
The trip struct stores attributes from a struct page, start date and finish date returned from queries	68
Trip	
The Trip class	68
uiaddalbum	70
UiAddAlbum	70
uiaddpage	71
UiAddPage	71
uiaddperson	74
UiAddPerson	74
uiassociate	75
UiAssociate	75
uimodify	76
uimodifyalbum	76
UiModifyAlbum	77
UiMove	78
UiSlideShow	79
uislideshow	81
uislideshowaddimage	81
UiSlideShowAddImage	82

Chapter 4

Module Documentation

4.1)

Purely database queries.

Purely database queries.

Represents a virtual pages.

Represents a virtual album.

Purely main code.

Class that belongs to the design part.

The main purpose of this class is to organize queries that our program is supposed to do.

Note

Keep improving!

Author

Author

Bruno Oliveira, Bruno Ferreira, Ricardo Barreto

Version

Revision

1.0

Date

Date

2018/05/15 12:21:47

Contact: uc2014228470@student.uc.pt uc2014201123@student.uc.pt uc2014191164@student.uc.pt ↵

Created on: Wed Mar 21 18:39:37 2018

Id

v1.0 2018/03/21 18:39:37 bv Exp

Note

Keep improving!

Author

Author

Bruno Oliveira, Bruno Ferreira, Ricardo Barreto

Version

Revision

1.0

Date

Date

2018/05/15 12:21:47

Contact: uc2014228470@student.uc.pt uc2014201123@student.uc.pt uc2014191164@student.uc.pt ↵

Created on: Wed Mar 21 18:39:37 2018

Id

v1.0 2018/03/21 18:39:37 bv Exp

The main purpose of this class is to organize information about an album.

Note

.

Author

Author

Bruno Oliveira, Bruno Ferreira, Ricardo Barreto

Version

Revision

1.0

Date

Date

2018/04/13 19:55:13

Contact: uc2014228470@student.uc.pt uc2014201123@student.uc.pt uc2014191164@student.uc.pt ↩
uc.pt

Created on: Thu Apr 13 15:22:19 2018

Id

v1.0 2018/03/21 15:22:19 bv Exp

The main purpose of this class is to organize information about a page.

Note

.

Author

Author

Bruno Oliveira, Bruno Ferreira, Ricardo Barreto

Version

Revision

1.0

Date

Date

2018/04/14 19:20:68

Contact: uc2014228470@student.uc.pt uc2014201123@student.uc.pt uc2014191164@student.uc.pt ↵

Created on: Mon Apr 9 12:27:12 2018

Id

v1.0 2018/03/21 12:27:12 bv Exp

The main purpose of this class is to organize information about a page.

Note

.

Author

Author

Bruno Oliveira, Bruno Ferreira, Ricardo Barreto

Version

Revision

1.0

Date

Date

2018/05/19 16:20:68

Contact: uc2014228470@student.uc.pt uc2014201123@student.uc.pt uc2014191164@student.uc.pt ↵

Created on: Mon Apr 9 12:27:12 2018

Id

v1.0 2018/03/21 12:27:12 bv Exp

Chapter 5

Class Documentation

5.1 Album Class Reference

```
#include <album.h>
```

Public Member Functions

- [Album](#) ()
Constructor of the class [Album](#).
- void [AddAlbum](#) (QString name, QString description, type_page type_of_page)
The method [AddAlbum\(\)](#) adds a new [Album](#) to the class [Album](#).
- void [AddName](#) (QString name)
The method [AddName\(\)](#) adds a new name to the class [Album](#).
- void [AddFolderName](#) (QString folder)
The method [AddFolderName\(\)](#) adds a new folder to the class [Album](#).
- void [AddId](#) (int id)
The method [AddId\(\)](#) adds a new id to the class [Album](#).
- QString [GetName](#) ()
The method [GetName\(\)](#) gets the name of the [Album](#).
- QString [GetDescription](#) ()
The method [GetDescription\(\)](#) gets the description of the [Album](#).
- int [GetPageType](#) ()
The method [GetPageType](#) gets the page type of the [Album](#).
- QString [GetFolder](#) ()
The method [GetFolder](#) gets the folder of the [Album](#).
- int [GetId](#) ()
The method [GetId\(\)](#) gets the id of the [Album](#).

Static Public Member Functions

- static QString [GetPageTypeString](#) (type_page type)
The method [GetPageTypeString](#) gets the page type of the [Album](#).

5.1.1 Detailed Description

(Note, this needs exactly one

5.1.2 Member Function Documentation

5.1.2.1 AddAlbum()

```
void Album::AddAlbum (
    QString name,
    QString description,
    type_page type_of_page )
```

The method [AddAlbum\(\)](#) adds a new [Album](#) to the class [Album](#).

Parameters

<i>name</i>	- stores the name of the Album
<i>description</i>	- stores de description of the Album
<i>type_of_page</i>	- stores the type of page

5.1.2.2 AddFolderName()

```
void Album::AddFolderName (
    QString folder )
```

The method [AddFolderName\(\)](#) adds a new folder to the class [Album](#).

Parameters

<i>folder</i>	- stores the folder
---------------	---------------------

5.1.2.3 AddId()

```
void Album::AddId (
    int id )
```

The method [AddId\(\)](#) adds a new id to the class [Album](#).

Parameters

<i>id</i>	
-----------	--

5.1.2.4 AddName()

```
void Album::AddName (
    QString name )
```

The method [AddName\(\)](#) adds a new name to the class [Album](#).

Parameters

<i>name</i>	- stores the name of the Album
-------------	--

5.1.2.5 GetDescription()

```
QString Album::GetDescription ( )
```

The method [GetDescription\(\)](#) gets the description of the [Album](#).

Returns

Return the description of the [Album](#).

5.1.2.6 GetFolder()

```
QString Album::GetFolder ( )
```

The method [GetFolder](#) gets the folder of the [Album](#).

Returns

Returns the folder of the [Album](#).

5.1.2.7 GetId()

```
int Album::GetId ( )
```

The method [GetId\(\)](#) gets the id of the [Album](#).

Returns

Returns the id of the [Album](#).

5.1.2.8 GetName()

```
QString Album::GetName ( )
```

The method [GetName\(\)](#) gets the name of the [Album](#).

Returns

Return the name of the [Album](#).

5.1.2.9 GetPageType()

```
int Album::GetPageType ( )
```

The method [GetPageType](#) gets the page type of the [Album](#).

Returns

Return the page type of the [Album](#).

5.1.2.10 GetPageTypeString()

```
QString Album::GetPageTypeString (
    type_page type ) [static]
```

The method [GetPageTypeString](#) gets the page type of the [Album](#).

Parameters

<i>type</i>	
-------------	--

Returns

Returns the page type in QString according to the enum type_page.

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

- managementSystem/album.h
- managementSystem/album.cpp

5.2 ourDB::albuns Struct Reference

The albuns struct stores the id and the name of albuns returned from queries.

```
#include <ourDB.h>
```

Public Attributes

- int **id**
- QString **name**

5.2.1 Detailed Description

The albuns struct stores the id and the name of albuns returned from queries.

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

- database/ourDB.h

5.3 ourDB::association Struct Reference

The association struct stores an association between a photo and a person using their ids returned from queries.

```
#include <ourDB.h>
```

Public Attributes

- int **id_photo**
- int **id_person**

5.3.1 Detailed Description

The association struct stores an association between a photo and a person using their ids returned from queries.

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

- database/ourDB.h

5.4 mainwindow Class Reference

```
#include <mainwindow.h>
```

5.4.1 Detailed Description

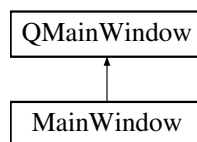
(Note, this needs exactly one

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

- mainwindow.h

5.5 MainWindow Class Reference

Inheritance diagram for MainWindow:



Public Slots

- void [SlotNewAlbum](#) ([Album](#) album_recived)
The method [getPhotosAssociatedFromPerson\(\)](#) seeks for the photos that are associated to a certain person in the database.
- void [SlotNewPageParty](#) ([Party](#) party_recived)
The method [SlotNewPageParty\(\)](#) creates a new [Party Page](#) and add it to the database.
- void [SlotNewPageThingPeople](#) ([ThingPeople](#) thing_people_recived)
The method [SlotNewPageThingPeople\(\)](#) creates a new [Party Page](#) and add it to the database.
- void [SlotNewPageTrip](#) ([Trip](#) trip_recived)
The method [SlotNewPageTrip\(\)](#) creates a new [Party Page](#) and add it to the database.
- void [SlotNewPageOther](#) ([Other](#) other_recived)
The method [SlotNewPageOther\(\)](#) creates a new [Party Page](#) and add it to the database.
- void [SlotMoveSelectedPhoto](#) (QString selected_item, int item_id)
The method [SlotMoveSelectedPhoto\(\)](#) moves a photo to other [Page](#).
- void [SlotMoveSelectedPage](#) (QString selected_item, int item_id)
The method [SlotMoveSelectedPage\(\)](#) moves a [Page](#) to other [Album](#) of the same type.

Signals

- void [SentPageType](#) (int type)
The method [SentPageType\(\)](#) sends the type of page.
- void [SentItemSelectedName](#) (QString name)
The method [SentItemSelectedName\(\)](#) sends the item name.
- void [SentState](#) (int actual_state)
The method [SentState\(\)](#) sends the state.
- void [SentItemSelectedId](#) (int id)
The method [SentItemSelectedId\(\)](#) sends the selected item ID.
- void [SentStateRootDir](#) (QString root_dir)
The method [SentStateRootDir\(\)](#) sends the folder path where the images are saved.

Public Member Functions

- **MainWindow** (QWidget *parent=0)

5.5.1 Member Function Documentation

5.5.1.1 SentItemSelectedId

```
void MainWindow::SentItemSelectedId (  
    int id ) [signal]
```

The method [SentItemSelectedId\(\)](#) sends the selected item ID.

Parameters

<i>id</i>	- stores de ID of the item
-----------	----------------------------

5.5.1.2 SentItemSelectedName

```
void MainWindow::SentItemSelectedName (  
    QString name ) [signal]
```

The method [SentItemSelectedName\(\)](#) sends the item name.

Parameters

<i>name</i>	- stores the name of the item
-------------	-------------------------------

5.5.1.3 SentPageType

```
void MainWindow::SentPageType (  
    int type ) [signal]
```

The method [SentPageType\(\)](#) sends the type of page.

Parameters

<i>type</i>	- stores the type of page
-------------	---------------------------

5.5.1.4 SentState

```
void MainWindow::SentState (
    int actual_state ) [signal]
```

The method [SentState\(\)](#) sends the state.

Parameters

<i>actual_state</i>	- stores the state
---------------------	--------------------

5.5.1.5 SentStateRootDir

```
void MainWindow::SentStateRootDir (
    QString root_dir ) [signal]
```

The method [SentStateRootDir\(\)](#) sends the folder path where the images are saved.

Parameters

<i>root_dir</i>	- stores the root path where the images are saved
-----------------	---

5.5.1.6 SlotMoveSelectedPage

```
void MainWindow::SlotMoveSelectedPage (
    QString selected_item,
    int item_id ) [slot]
```

The method [SlotMoveSelectedPage\(\)](#) moves a [Page](#) to other [Album](#) of the same type.

Parameters

<i>selected_item</i>	- stores the name of the page to move, received from the window form
<i>item_id</i>	- stores the id of the page to be moved

5.5.1.7 SlotMoveSelectedPhoto

```
void MainWindow::SlotMoveSelectedPhoto (
    QString selected_item,
    int item_id ) [slot]
```

The method [SlotMoveSelectedPhoto\(\)](#) moves a photo to other [Page](#).

Parameters

<i>selected_item</i>	- stores the name of the foto to move, received from the window form
<i>item_id</i>	- stores the id of the photo to be moved

5.5.1.8 SlotNewAlbum

```
void MainWindow::SlotNewAlbum (
    Album album_recived ) [slot]
```

The method `getPhotosAssociatedFromPerson()` seeks for the photos that are associated to a certain person in the database.

Parameters

<i>person_id</i>	- stores the id of the desired person
------------------	---------------------------------------

Returns

Returns a `QList` containing all the photos associated to the desired person. The method `SlotNewAlbum()` creates a new `Album` and add it to the database.

Parameters

<i>album_recived</i>	- stores the album information, received from the windows form, and add it to the database
----------------------	--

5.5.1.9 SlotNewPageOther

```
void MainWindow::SlotNewPageOther (
    Other other_received ) [slot]
```

The method `SlotNewPageOther()` creates a new `Party Page` and add it to the database.

Parameters

<i>other_received</i>	- stores the <code>Party</code> information, received from the windows form, and add it to the database
-----------------------	---

5.5.1.10 SlotNewPageParty

```
void MainWindow::SlotNewPageParty (
    Party party_received ) [slot]
```

The method `SlotNewPageParty()` creates a new [Party Page](#) and add it to the database.

Parameters

<code>party_received</code>	- stores the Party information, received from the windows form, and add it to the database
-----------------------------	--

5.5.1.11 SlotNewPageThingPeople

```
void MainWindow::SlotNewPageThingPeople (
    ThingPeople thing_people_received ) [slot]
```

The method `SlotNewPageThingPeople()` creates a new [Party Page](#) and add it to the database.

Parameters

<code>thing_people_received</code>	- stores the Party information, received from the windows form, and add it to the database
------------------------------------	--

5.5.1.12 SlotNewPageTrip

```
void MainWindow::SlotNewPageTrip (
    Trip trip_received ) [slot]
```

The method `SlotNewPageTrip()` creates a new [Party Page](#) and add it to the database.

Parameters

<code>trip_received</code>	- stores the Party information, received from the windows form, and add it to the database
----------------------------	--

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

- mainwindow.h
- mainwindow.cpp

5.6 ourDB::other Struct Reference

The other struct stores attributes from a struct page, start date and finish date returned from queries.

```
#include <ourDB.h>
```

Public Attributes

- [pages](#) page
- QDate **start_date**
- QDate **finish_date**

5.6.1 Detailed Description

The other struct stores attributes from a struct page, start date and finish date returned from queries.

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

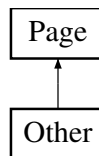
- database/ourDB.h

5.7 Other Class Reference

The [Other](#) class.

```
#include <page.h>
```

Inheritance diagram for Other:



Public Member Functions

- void [AddPageOther](#) (QString description, QDate start_date, QDate end_date)
The method [AddPageOther\(\)](#) adds a new [Other](#) page.
- QDate [GetDateStart](#) ()
The method [GetDateStart\(\)](#) gets the end date of [Other](#).
- QDate [GetDateEnd](#) ()
The method [GetDateEnd\(\)](#) gets the end date of [Other](#).

Additional Inherited Members

5.7.1 Detailed Description

The [Other](#) class.

5.7.2 Member Function Documentation

5.7.2.1 AddPageOther()

```
void Other::AddPageOther (
    QString description,
    QDate start_date,
    QDate end_date )
```

The method [AddPageOther\(\)](#) adds a new [Other](#) page.

Parameters

<i>description</i>	
<i>start_date</i>	
<i>end_date</i>	

5.7.2.2 getDateEnd()

```
QDate Other::getDateEnd ( )
```

The method [getDateEnd\(\)](#) gets the end date of [Other](#).

Returns

Returns the end date of [Other](#).

5.7.2.3 getDateStart()

```
QDate Other::getDateStart ( )
```

The method [getDateStart\(\)](#) gets the end date of [Other](#).

Returns

Returns the start date of [Other](#).

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

- managementSystem/page.h
- managementSystem/page.cpp

5.8 ourDB Class Reference

```
#include <ourDB.h>
```

Classes

- struct [albuns](#)
The *albuns* struct stores the id and the name of albuns returned from queries.
- struct [association](#)
The *association* struct stores an association between a photo and a person using their ids returned from queries.
- struct [other](#)
The *other* struct stores attributes from a struct page, start date and finish date returned from queries.
- struct [pages](#)
The *pages* struct stores the id, name and description of pages returned from queries.
- struct [party](#)
The *party* struct stores attributes from a struct page, date and the type of party returned from queries.
- struct [person](#)
The *person* struct stores the id and the name of people returned from queries.
- struct [photos](#)
The *photos* struct stores the id, name, date and the resolution (width x height) of photos returned from queries.
- struct [trip](#)
The *trip* struct stores attributes from a struct page, start date and finish date returned from queries.

Public Member Functions

- [ourDB](#) ()
Constructor of the class *ourDB*.

Static Public Member Functions

- static bool [buildDB](#) ()
The method *buildDB()* establishes the connection between our software and the database. It verifies if the database is properly created, that is, checks if there all the tables that are required. If there aren't, this method creates them.
- static bool [setAlbum](#) (int n_pages, QString name, QString des, QString folder, int type, QString path)
The method *setAlbum()* adds an album with the following informations to the database.
- static bool [setPage](#) (int al_id, QString name, QString des, QString folder, int n_photos, QString path)
The method *setPage()* adds a page with the following informations to the database.
- static bool [setPageTrip](#) (int pa_id, QDate start_date, QDate finish_date)
The method *setPageTrip()* adds a *Trip* page with the following informations to the database.
- static bool [setPageParty](#) (int pa_id, QDate date, QString type_of_party)
The method *setPageParty()* adds a *Party* page with the following informations to the database.
- static bool [setPageOther](#) (int pa_id, QDate start_date, QDate finish_date)
The method *setPageOther()* adds a *Other* page with the following informations to the database.
- static bool [setPhoto](#) (int pa_id, QString name, QDate date, int size, int width, int height, QString extension)
The method *setPhoto()* adds a photo with the following informations to the database.
- static bool [setPerson](#) (QString name, QString gender, QDate birthday, QString relationship)
The method *setPerson()* adds a person with the following informations to the database.
- static bool [setAssociation](#) (int pho_id, int pe_id)
The method *setAssociation()* adds an association between the a person and a photo to the database.
- static bool [setAlbumName](#) (int id, QString new_name)
The method *setAlbumName()* updates the name of a certain album.
- static bool [setAlbumDescription](#) (int id, QString new_description)
The method *setAlbumDescription()* updates the description of a certain album.
- static bool [setMovePage](#) (int pa_id, int new_al_id)

- The method [setMovePage\(\)](#) updates the parent album of a certain page.*

 - static bool [setMovePhoto](#) (int new_pa_id, int pho_id)
- The method [setMovePhoto\(\)](#) updates the parent page of a certain photo.*

 - static bool [setAssociationAlbum](#) (int al_id, int pe_id)
- The method [setAssociationAlbum\(\)](#) adds an association between a person and an album to the database.*

 - static bool [setAssociationPage](#) (int pa_id, int pe_id)
- The method [setAssociationPage\(\)](#) adds an association between a person and a page to the database.*

 - static bool [setAlbumFolderName](#) (int id, QString new_folder_name)
- The method [setAlbumFolderName\(\)](#) changes the folder name of an album in the database.*

 - static QList< [albuns](#) > [getAlbuns](#) ()
- The method [getAlbuns\(\)](#) seeks all the albuns that exists in the database.*

 - static QList< [pages](#) > [getPages](#) ()
- The method [getPages\(\)](#) seeks all the pages that exists in the database.*

 - static QString [getAlbumName](#) (int id)
- The method [getAlbumName\(\)](#) seeks for the name of a desired album in the database.*

 - static QList< [albuns](#) > [getAlbunsFromPageType](#) (int page_type_id)
- The method [getAlbunsFromPageType\(\)](#) seeks all the albuns that have a certain page type in the database.*

 - static int [getAlbumIdFromPage](#) (int page_id)
- The method [getAlbumIdFromPage\(\)](#) seeks the album id from a desired page in the database.*

 - static QList< [pages](#) > [getPagesToShow](#) (int al_id)
- The method [getPagesToShow\(\)](#) seeks all the pages that exist in the database that have the following album id.*

 - static QList< [party](#) > [getPartyPagesToShow](#) (int al_id)
- The method [getPartyPagesToShow\(\)](#) seeks all the [Party](#) pages that exist in the database that have the following album id.*

 - static QList< [trip](#) > [getTripPagesToShow](#) (int al_id)
- The method [getTripPagesToShow\(\)](#) seeks all the [Trip](#) pages that exist in the database that have the following album id.*

 - static QList< [other](#) > [getOtherPagesToShow](#) (int al_id)
- The method [getOtherPagesToShow\(\)](#) seeks all the [Other](#) pages that exist in the database that have the following album id.*

 - static QList< [photos](#) > [getPhotosFromPage](#) (int pa_id)
- The method [getPhotosFromPage\(\)](#) seeks all the photos that exist in the database that have the following page id.*

 - static QString [getAlbumFolderName](#) (int id)
- The method [getAlbumFolderName\(\)](#) seeks for the folder name of a desired album.*

 - static QString [getAlbumDescription](#) (int id)
- The method [getAlbumDescription\(\)](#) seeks for the description of a desired album.*

 - static QString [getPageName](#) (int id)
- The method [getPageName\(\)](#) seeks for the name of a desired page.*

 - static QString [getPageDescription](#) (int id)
- The method [getPageDescription\(\)](#) seeks for the description of a desired page.*

 - static int [getPageIdFromPhoto](#) (int pho_id)
- The method [getPageIdFromPhoto\(\)](#) seeks for the page id of a desired photo.*

 - static QString [getPageFolderName](#) (int id)
- The method [getPageFolderName\(\)](#) seeks for the folder name of a desired page.*

 - static int [getAlbumPageType](#) (int id)
- The method [getAlbumPageType\(\)](#) seeks for the [Page](#) Type that album has.*

 - static QString [getPhotoName](#) (int id)
- The method [getPhotoName\(\)](#) seeks for the name of a desired photo in the database.*

 - static int [getPhotoLastID](#) ()
- The method [getPhotoLastID\(\)](#) seeks for the last photo id that exists in the database.*

 - static QString [getPhotoExtension](#) (int id)

- The method [getPhotoExtension\(\)](#) seeks for the format of a desired photo.*

 - static QList< [ourDB::person](#) > [getAssociationAlbum](#) (int al_id)

The method [getAssociationAlbum\(\)](#) seeks for the people associated to an album in the database.

 - static QList< [ourDB::person](#) > [getAssociationPage](#) (int pa_id)

The method [getAssociationPage\(\)](#) seeks for the people associated to a page in the database.

 - static QList< [ourDB::person](#) > [getAssociationPhoto](#) (int pho_id)

The method [getAssociationPhoto\(\)](#) seeks for the people associated to a photo in the database.

 - static QList< [photos](#) > [getAllPhotos](#) (bool range, QDate start_date, QDate finish_date, bool order_by_asc)

The method [getAllPhotos\(\)](#) seeks for all the photos existing in the database.

 - static QList< [photos](#) > [getPhotosFromAlbum](#) (int al_id, bool range, QDate start_date, QDate finish_date, bool order_by_asc)

The method [getPhotosFromAlbum\(\)](#) seeks for all the photos that belong to the desired album.

 - static QList< [photos](#) > [getPhotosFromPage](#) (int pa_id, bool range, QDate start_date, QDate finish_date, bool order_by_asc)

The method [getPhotosFromPage\(\)](#) seeks for all the photos that belong to the desired page.

 - static int [getNumberOfAllPhotos](#) ()

The method [getNumberOfAllPhotos\(\)](#) seeks for all the photos existing in the database.

 - static QList< [photos](#) > [getPhotosFromDate](#) (QDate date)

The method [getPhotosFromDate\(\)](#) returns all the photos that contain the desired date in the database.

 - static int [getNumberPhotosFromPage](#) (int pa_id)

The method [getNumberPhotosFromPage\(\)](#) seeks for the number of photos existing in a certain page.

 - static int [getNumberPhotosFromAlbum](#) (int al_id)

The method [getNumberPhotosFromAlbum\(\)](#) seeks for the number of photos existing in a certain album.

 - static QList< [person](#) > [getPeople](#) ()

The method [getPersons\(\)](#) seeks for all the people existing in the database.

 - static QList< [photos](#) > [getPhotos](#) ()

The method [getPhotos\(\)](#) seeks for all the photos existing in the database.

 - static int [getAlbumLastID](#) ()

The method [getAlbumLastID\(\)](#) seeks for the id of the last album in the database.

 - static int [getPageLastID](#) ()

The method [getPageLastID\(\)](#) seeks for the id of the last page in the database.

 - static QList< [ourDB::photos](#) > [getPhotosAssociatedFromPerson](#) (int pe_id)

The method [getPhotosAssociatedFromPerson\(\)](#) seeks for the photos that are associated to a certain person in the database.

 - static [ourDB::photos](#) [getPhotoInformation](#) (int pho_id)

The method [getPhotoInformation\(\)](#) seeks for the information of a desired photo in the database.

 - static int [getNumberOfAlbums](#) ()

The method [getNumberOfAlbums\(\)](#) seeks for the number of albums existing in the database.

 - static int [getNumberOfPagesFromAlbum](#) (int al_id)

The method [getNumberOfPagesFromAlbum\(\)](#) seeks for the number of pages that belongs to an album in the database.

 - static QDate [getPhotoDateFromAlbum](#) (int al_id, bool newest)

The method [getPhotoDateFromAlbum\(\)](#) seeks for the most recent photo from an album or the least recent in the database.

 - static QDate [getPhotoDateFromPage](#) (int pa_id, bool newest)

The method [getPhotoDateFromPage\(\)](#) seeks for the most recent photo from a page or the least recent in the database.

 - static QDate [getPhotoDate](#) (int id)

The method [getPhotoDate\(\)](#) seeks for the date of a desired photo in the database.

 - static QString [getPhotoWidth](#) (int id)

The method [getPhotoWidth\(\)](#) seeks for the width of a desired photo in the database.

 - static QString [getPhotoHeight](#) (int id)

The method `getPhotoHeight()` seeks for the height of a desired photo in the database.

- static bool `removeAlbum` (int id)

The method `removeAlbum()` removes an album from the database with a certain id.

- static bool `removePhoto` (int id)

The method `removePhoto()` removes a photo from the database with a certain id.

- static bool `removePage` (int id)

The method `removePage()` removes a page from the database with a certain id.

- static bool `checkIfAlbumsEmpty` (int al_id)

The method `checkIfAlbumsEmpty()` checks if the album is empty in the database.

- static bool `checkIfPagesEmpty` (int pa_id)

The method `checkIfPagesEmpty()` checks if the page is empty in the database.

5.8.1 Detailed Description

(Note, this needs exactly one

5.8.2 Member Function Documentation

5.8.2.1 buildDB()

```
bool ourDB::buildDB ( ) [static]
```

The method `buildDB()` establishes the connection between our software and the database. It verifies if the database is properly created, that is, checks if there all the tables that are required. If there aren't, this method creates them.

Returns

Returns true if the database was opened successfully and there all the tables required. Otherwise, the method returns false.

5.8.2.2 checkIfAlbumsEmpty()

```
bool ourDB::checkIfAlbumIsEmpty (
    int al_id ) [static]
```

The method `checkIfAlbumsEmpty()` checks if the album is empty in the database.

Parameters

<code>al_id</code>	- stores the id of the desired album
--------------------	--------------------------------------

Returns

Returns true if the album is empty. Otherwise the method returns false.

5.8.2.3 `checkIfPagelsEmpty()`

```
bool ourDB::checkIfPageIsEmpty (
    int pa_id ) [static]
```

The method `checkIfPagelsEmpty()` checks if the page is empty in the database.

Parameters

<code>pa_id</code>	- stores the id of the desired page
--------------------	-------------------------------------

Returns

Returns true if the page is empty. Otherwise the method returns false.

5.8.2.4 `getAlbumDescription()`

```
QString ourDB::getAlbumDescription (
    int id ) [static]
```

The method `getAlbumDescription()` seeks for the description of a desired album.

Parameters

<code>id</code>	- stores the id of the desired album
-----------------	--------------------------------------

Returns

Returns a QString that contains the description.

5.8.2.5 `getAlbumFolderName()`

```
QString ourDB::getAlbumFolderName (
    int id ) [static]
```

The method `getAlbumFolderName()` seeks for the folder name of a desired album.

Parameters

<i>id</i>	- stores the id of the desired album
-----------	--------------------------------------

Returns

Returns a QString that contains the folder name.

5.8.2.6 getAlbumIdFromPage()

```
int ourDB::getAlbumIdFromPage (
    int page_id ) [static]
```

The method [getAlbumIdFromPage\(\)](#) seeks the album id from a desired page in the database.

Parameters

<i>page_id</i>	- stores the id of the desired page
----------------	-------------------------------------

Returns

Returns an integer that represents the album id that contains the page.

5.8.2.7 getAlbumLastID()

```
int ourDB::getAlbumLastID ( ) [static]
```

The method [getAlbumLastID\(\)](#) seeks for the id of the last album in the database.

Returns

Returns an integer containing the id of the last album.

5.8.2.8 getAlbumName()

```
QString ourDB::getAlbumName (
    int id ) [static]
```

The method [getAlbumName\(\)](#) seeks for the name of a desired album in the database.

Parameters

<i>id</i>	- stores the album id
-----------	-----------------------

Returns

Returns a QString that contains the name of the desired album.

5.8.2.9 getAlbumPageType()

```
int ourDB::getAlbumPageType (
    int id ) [static]
```

The method [getAlbumPageType\(\)](#) seeks for the [Page](#) Type that album has.

Parameters

<i>id</i>	- stores the id of the desired album
-----------	--------------------------------------

Returns

Returns an integer that represents the page type.

5.8.2.10 getAlbuns()

```
QList< ourDB::albuns > ourDB::getAlbuns ( ) [static]
```

The method [getAlbuns\(\)](#) seeks all the albuns that exists in the database.

Returns

Returns a QList of albuns struct that contains all the albuns from database.

5.8.2.11 getAlbunsFromPageType()

```
QList< ourDB::albuns > ourDB::getAlbunsFromPageType (
    int page_type_id ) [static]
```

The method [getAlbunsFromPageType\(\)](#) seeks all the albuns that have a certain page type in the database.

Parameters

<i>page_type↔ _id</i>	- stores the id of the page type
---------------------------	----------------------------------

Returns

Returns a QList of all albuns that have the desired page type.

5.8.2.12 getAllPhotos()

```
QList< ourDB::photos > ourDB::getAllPhotos (
    bool range,
    QDate start_date,
    QDate finish_date,
    bool order_by_asc ) [static]
```

The method [getAllPhotos\(\)](#) seeks for all the photos existing in the database.

Parameters

<i>range</i>	- stores true if we want photos within a range, or false if we want all the photos
<i>start_date</i>	- stores the start date of the range
<i>finish_date</i>	- stores the finish date of the range
<i>order_by_asc</i>	- stores true if we want the photos sorted in the ascending order, or false if we want the photos sorted in the descending order

Returns

Returns a QList of all the desired photos.

5.8.2.13 getAssociationAlbum()

```
QList< ourDB::person > ourDB::getAssociationAlbum (
    int al_id ) [static]
```

The method [getAssociationAlbum\(\)](#) seeks for the people associated to an album in the database.

Parameters

<i>al↔ _id</i>	- stores the id of the album
--------------------	------------------------------

Returns

Returns a QList of the people that are associated to the desired album.

5.8.2.14 getAssociationPage()

```
QList< ourDB::person > ourDB::getAssociationPage (
    int pa_id ) [static]
```

The method [getAssociationPage\(\)](#) seeks for the people associated to a page in the database.

Parameters

<i>pa_id</i>	- stores the id of the page
--------------	-----------------------------

Returns

Returns a QList of the people that are associated to the desired page.

5.8.2.15 getAssociationPhoto()

```
QList< ourDB::person > ourDB::getAssociationPhoto (
    int pho_id ) [static]
```

The method [getAssociationPhoto\(\)](#) seeks for the people associated to a photo in the database.

Parameters

<i>pho_id</i>	- stores the id of the photo
---------------	------------------------------

Returns

Returns a QList of the people that are associated to the desired photo.

5.8.2.16 getNumberOfAlbums()

```
int ourDB::getNumberOfAlbums ( ) [static]
```

The method [getNumberOfAlbums\(\)](#) seeks for the number of albums existing in the database.

Returns

Returns an integer that contains the number of albums.

5.8.2.17 `getNumberOfAllPhotos()`

```
int ourDB::getNumberOfAllPhotos ( ) [static]
```

The method `getNumberOfAllPhotos()` seeks for all the photos existing in the database.

Returns

Returns an integer that contains the total number of photos in the database.

5.8.2.18 `getNumberOfPagesFromAlbum()`

```
int ourDB::getNumberOfPagesFromAlbum (
    int al_id ) [static]
```

The method `getNumberOfPagesFromAlbum()` seeks for the number of pages that belongs to an album in the database.

Parameters

$al \leftrightarrow$ _id	- stores the id of the target album
-----------------------------	-------------------------------------

Returns

Returns an integer that contains the number of pages from an album.

5.8.2.19 `getNumberPhotosFromAlbum()`

```
int ourDB::getNumberPhotosFromAlbum (
    int al_id ) [static]
```

The method `getNumberPhotosFromAlbum()` seeks for the number of photos existing in a certain album.

Parameters

$al \leftrightarrow$ _id	- stores the id of the desired album
-----------------------------	--------------------------------------

Returns

Returns an integer that contains all the photos from the desired album.

5.8.2.20 `getNumberPhotosFromPage()`

```
int ourDB::getNumberPhotosFromPage (
    int pa_id ) [static]
```

The method `getNumberPhotosFromPage()` seeks for the number of photos existing in a certain page.

Parameters

<code>pa_id</code>	- stores the id of the desired page
--------------------	-------------------------------------

Returns

Returns an integer that contains all the photos from the desired page.

5.8.2.21 `getOtherPagesToShow()`

```
QList< ourDB::other > ourDB::getOtherPagesToShow (
    int al_id ) [static]
```

The method `getOtherPagesToShow()` seeks all the `Other` pages that exist in the database that have the following album id.

Parameters

<code>al_id</code>	- stores the album id to seek for
--------------------	-----------------------------------

Returns

Returns a QList of other pages struct that contains all the `Other` pages desired to show.

5.8.2.22 `getPageDescription()`

```
QString ourDB::getPageDescription (
    int id ) [static]
```

The method `getPageDescription()` seeks for the description of a desired page.

Parameters

<code>id</code>	- stores the id of the desired page
-----------------	-------------------------------------

Returns

Returns a QString that contains the page description.

5.8.2.23 getPageFolderName()

```
QString ourDB::getPageFolderName (
    int id ) [static]
```

The method [getPageFolderName\(\)](#) seeks for the folder name of a desired page.

Parameters

<i>id</i>	- stores the id of the desired page
-----------	-------------------------------------

Returns

Returns a QString that contains the folder name.

5.8.2.24 getPageIdFromPhoto()

```
int ourDB::getPageIdFromPhoto (
    int pho_id ) [static]
```

The method [getPageIdFromPhoto\(\)](#) seeks for the page id of a desired photo.

Parameters

<i>pho_id</i>	- stores the id of the desired photo
---------------	--------------------------------------

Returns

Returns an integer that represents the id of the page.

5.8.2.25 getPageLastID()

```
int ourDB::getPageLastID ( ) [static]
```

The method [getPageLastID\(\)](#) seeks for the id of the last page in the database.

Returns

Returns an integer containing the id of the last page.

5.8.2.26 getPageName()

```
QString ourDB::getPageName (
    int id ) [static]
```

The method [getPageName\(\)](#) seeks for the name of a desired page.

Parameters

<i>id</i>	- stores the id of the desired page
-----------	-------------------------------------

Returns

Returns a QString that contains the page name.

5.8.2.27 getPages()

```
QList< ourDB::pages > ourDB::getPages ( ) [static]
```

The method [getPages\(\)](#) seeks all the pages that exists in the database.

Returns

Returns a QList of pages struct that contains all the pages from database.

5.8.2.28 getPagesToShow()

```
QList< ourDB::pages > ourDB::getPagesToShow (
    int al_id ) [static]
```

The method [getPagesToShow\(\)](#) seeks all the pages that exist in the database that have the following album id.

Parameters

<i>al↵ _id</i>	- stores the album id to seek for
--------------------	-----------------------------------

Returns

Returns a QList of pages struct that contains all the pages desired to show.

5.8.2.29 getPartyPagesToShow()

```
QList< ourDB::party > ourDB::getPartyPagesToShow (
    int al_id ) [static]
```

The method `getPartyPagesToShow()` seeks all the `Party` pages that exist in the database that have the following album id.

Parameters

<code>al_id</code>	- stores the album id to seek for
--------------------	-----------------------------------

Returns

Returns a QList of party pages struct that contains all the `Party` pages desired to show.

5.8.2.30 getPeople()

```
QList< ourDB::person > ourDB::getPeople ( ) [static]
```

The method `getPersons()` seeks for all the people existing in the database.

Returns

Returns a QList containing the people.

5.8.2.31 getPhotoDate()

```
QDate ourDB::getPhotoDate (
    int id ) [static]
```

The method `getPhotoDate()` seeks for the date of a desired photo in the database.

Parameters

<code>id</code>	- stores the id of the desired photo
-----------------	--------------------------------------

Returns

Returns a QDate that contains the desired date of a photo.

5.8.2.32 getPhotoDateFromAlbum()

```
QDate ourDB::getPhotoDateFromAlbum (
    int al_id,
    bool newest ) [static]
```

The method [getPhotoDateFromAlbum\(\)](#) seeks for the most recent photo from an album or the least recent in the database.

Parameters

<i>al_id</i>	- stores the id for the target album
<i>newest</i>	- stores true if we want the most recent, false if not

Returns

Returns a QDate that contains the desired date of a photo.

5.8.2.33 getPhotoDateFromPage()

```
QDate ourDB::getPhotoDateFromPage (
    int pa_id,
    bool newest ) [static]
```

The method [getPhotoDateFromPage\(\)](#) seeks for the most recent photo from a page or the least recent in the database.

Parameters

<i>pa_id</i>	- stores the id for the target page
<i>newest</i>	- stores true if want the most recent, false if not

Returns

Returns a QDate that contains the desired date of a photo.

5.8.2.34 getPhotoExtension()

```
QString ourDB::getPhotoExtension (
    int id ) [static]
```

The method [getPhotoExtension\(\)](#) seeks for the format of a desired photo.

Parameters

<i>id</i>	- stores the id of the desired photo
-----------	--------------------------------------

Returns

Returns a QString that represents the photo format.

5.8.2.35 getPhotoHeight()

```
QString ourDB::getPhotoHeight (
    int id ) [static]
```

The method [getPhotoHeight\(\)](#) seeks for the height of a desired photo in the database.

Parameters

<i>id</i>	- stores the id of the desired photo
-----------	--------------------------------------

Returns

Returns a QString that contains the height of the photo.

5.8.2.36 getPhotoInformation()

```
ourDB::photos ourDB::getPhotoInformation (
    int pho_id ) [static]
```

The method [getPhotoInformation\(\)](#) seeks for the information of a desired photo in the database.

Parameters

<i>pho_id</i>	- stores the id of the desired photo
---------------	--------------------------------------

Returns

Returns a struct containing all the information from a certain photo.

5.8.2.37 getPhotoLastID()

```
int ourDB::getPhotoLastID ( ) [static]
```

The method [getPhotoLastID\(\)](#) seeks for the last photo id that exists in the database.

Returns

Returns an integer that represents the last id.

5.8.2.38 getPhotoName()

```
QString ourDB::getPhotoName (
    int id ) [static]
```

The method `getPhotoName()` seeks for the name of a desired photo in the database.

Parameters

<i>id</i>	- stores the id of the desired photo
-----------	--------------------------------------

Returns

Returns a QString that contains the name of the photo.

5.8.2.39 getPhotos()

```
QList< ourDB::photos > ourDB::getPhotos ( ) [static]
```

The method `getPhotos()` seeks for all the photos existing in the database.

Returns

Returns a QList containing the photos.

5.8.2.40 getPhotosAssociatedFromPerson()

```
QList< ourDB::photos > ourDB::getPhotosAssociatedFromPerson (
    int pe_id ) [static]
```

The method `getPhotosAssociatedFromPerson()` seeks for the photos that are associated to a certain person in the database.

Parameters

<i>pe_id</i>	- stores the id of the desired person
--------------	---------------------------------------

Returns

Returns a QList containing all the photos associated to the desired person.

5.8.2.41 `getPhotosFromAlbum()`

```
QList< ourDB::photos > ourDB::getPhotosFromAlbum (
    int al_id,
    bool range,
    QDate start_date,
    QDate finish_date,
    bool order_by_asc ) [static]
```

The method `getPhotosFromAlbum()` seeks for all the photos that belong to the desired album.

Parameters

<i>al_id</i>	- stores the id of the desired album
<i>range</i>	- stores true if we want photos within a range, or false if we want all the photos
<i>start_date</i>	- stores the start date of the range
<i>finish_date</i>	- stores the finish date of the range
<i>order_by_asc</i>	- stores true if we want the photos sorted in the ascending order, or false if we want the photos sorted in the descending order

Returns

Returns a QList of all the photos from the desired album.

5.8.2.42 `getPhotosFromDate()`

```
QList< ourDB::photos > ourDB::getPhotosFromDate (
    QDate date ) [static]
```

The method `getPhotosFromDate()` returns all the photos that contain the desired date in the database.

Parameters

<i>date</i>	- stores the desired date
-------------	---------------------------

Returns

Returns a QList of all the photos from the desired date.

5.8.2.43 `getPhotosFromPage()` [1/2]

```
QList< ourDB::photos > ourDB::getPhotosFromPage (
    int pa_id ) [static]
```

The method `getPhotosFromPage()` seeks all the photos that exist in the database that have the following page id.

Parameters

<i>pa_id</i>	- stores the page id to seek for
--------------	----------------------------------

Returns

Returns a QList of photos struct that contains all the photos desired to show.

5.8.2.44 `getPhotosFromPage()` [2/2]

```
QList< ourDB::photos > ourDB::getPhotosFromPage (
    int pa_id,
    bool range,
    QDate start_date,
    QDate finish_date,
    bool order_by_asc ) [static]
```

The method `getPhotosFromPage()` seeks for all the photos that belong to the desired page.

Parameters

<i>pa_id</i>	- stores the id of the desired page
<i>range</i>	- stores true if we want photos within a range, or false if we want all the photos
<i>start_date</i>	- stores the start date of the range
<i>finish_date</i>	- stores the finish date of the range
<i>order_by_asc</i>	- stores true if we want the photos sorted in the ascending order, or false if we want the photos sorted in the descending order

Returns

Returns a QList of all the photos from the desired page.

5.8.2.45 `getPhotoWidth()`

```
QString ourDB::getPhotoWidth (
    int id ) [static]
```

The method `getPhotoWidth()` seeks for the width of a desired photo in the database.

Parameters

<i>id</i>	- stores the id of the desired photo
-----------	--------------------------------------

Returns

Returns a QString that contains the width of the photo.

5.8.2.46 getTripPagesToShow()

```
QList< ourDB::trip > ourDB::getTripPagesToShow (
    int al_id ) [static]
```

The method `getTripPagesToShow()` seeks all the `Trip` pages that exist in the database that have the following album id.

Parameters

<code>al_id</code>	- stores the album id to seek for
--------------------	-----------------------------------

Returns

Returns a QList of trip pages struct that contains all the `Trip` pages desired to show.

5.8.2.47 removeAlbum()

```
bool ourDB::removeAlbum (
    int id ) [static]
```

The method `removeAlbum()` removes an album from the database with a certain id.

Parameters

<code>id</code>	- stores the id of the desired album
-----------------	--------------------------------------

Returns

Returns true if the album was successfully removed. Otherwise the method returns false.

5.8.2.48 removePage()

```
bool ourDB::removePage (
    int id ) [static]
```

The method `removePage()` removes a page from the database with a certain id.

Parameters

<i>id</i>	- stores the id of the desired page
-----------	-------------------------------------

Returns

Returns true if the page was successfully removed. Otherwise the method returns false.

5.8.2.49 removePhoto()

```
bool ourDB::removePhoto (
    int id ) [static]
```

The method [removePhoto\(\)](#) removes a photo from the database with a certain id.

Parameters

<i>id</i>	- stores the id of the desired photo
-----------	--------------------------------------

Returns

Returns true if the photo was successfully removed. Otherwise the method returns false.

5.8.2.50 setAlbum()

```
bool ourDB::setAlbum (
    int n_pages,
    QString name,
    QString des,
    QString folder,
    int type,
    QString path ) [static]
```

The method [setAlbum\(\)](#) adds an album with the following informations to the database.

Parameters

<i>id</i>	- stores the id of the album
<i>n_pages</i>	- stores the number of pages that exist in a certain album
<i>name</i>	- stores the name of the album given by the user
<i>des</i>	- stores the description of the album given by the user
<i>folder</i>	- stores the name of the album after parsing it (name of the folder in user's disk)
<i>type</i>	- stores the type of pages that album must possess
<i>path</i>	- stores the path of album in the user's disk

Returns

Returns true if the album was successfully added. Otherwise the method returns false.

5.8.2.51 `setAlbumDescription()`

```
bool ourDB::setAlbumDescription (
    int id,
    QString new_description ) [static]
```

The method `setAlbumDescription()` updates the description of a certain album.

Parameters

<i>id</i>	- stores the id of the desired album
<i>new_description</i>	- stores the new description of the album

Returns

Returns true if the description was successfully updated. Otherwise the method returns false.

5.8.2.52 `setAlbumFolderName()`

```
bool ourDB::setAlbumFolderName (
    int id,
    QString new_folder_name ) [static]
```

The method `setAlbumFolderName()` changes the folder name of an album in the database.

Parameters

<i>id</i>	- stores the id of the desired album
<i>new_folder_name</i>	- stores the new name for the folder

Returns

Returns true if the name was changed successfully. Otherwise returns false.

5.8.2.53 `setAlbumName()`

```
bool ourDB::setAlbumName (
    int id,
    QString new_name ) [static]
```

The method `setAlbumName()` updates the name of a certain album.

Parameters

<i>id</i>	- stores the id of the desired album
<i>new_name</i>	- stores the new name of the album

Returns

Returns true if the name was successfully updated. Otherwise the method returns false.

5.8.2.54 setAssociation()

```
bool ourDB::setAssociation (
    int pho_id,
    int pe_id ) [static]
```

The method [setAssociation\(\)](#) adds an association between the a person and a photo to the database.

Parameters

<i>pho_id</i>	- stores the id of the photo
<i>pe_id</i>	- stores the id of the person

Returns

Returns true if the association was successfully added. Otherwise the method returns false.

5.8.2.55 setAssociationAlbum()

```
bool ourDB::setAssociationAlbum (
    int al_id,
    int pe_id ) [static]
```

The method [setAssociationAlbum\(\)](#) adds an association between a person and an album to the database.

Parameters

<i>al_id</i>	- stores the id of the album
<i>pe_id</i>	- stores the id of the person

Returns

Returns true if the association was successfully done. Otherwise the method returns false.

5.8.2.56 setAssociationPage()

```
bool ourDB::setAssociationPage (
    int pa_id,
    int pe_id ) [static]
```

The method [setAssociationPage\(\)](#) adds an association between a person and a page to the database.

Parameters

<i>pa_id</i>	- stores the id of the page
<i>pe_id</i>	- stores the id of the person

Returns

Returns true if the association was successfully done. Otherwise the method returns false.

5.8.2.57 setMovePage()

```
bool ourDB::setMovePage (
    int pa_id,
    int new_al_id ) [static]
```

The method [setMovePage\(\)](#) updates the parent album of a certain page.

Parameters

<i>pa_id</i>	- stores the id of the desired page to move
<i>new_al_id</i>	- stores the id of the new parent album

Returns

Returns true if the page was successfully moved. Otherwise the method returns false.

5.8.2.58 setMovePhoto()

```
bool ourDB::setMovePhoto (
    int new_pa_id,
    int pho_id ) [static]
```

The method [setMovePhoto\(\)](#) updates the parent page of a certain photo.

Parameters

<i>new_parent_id</i>	- stores the id of the new parent page
<i>photo_id</i>	- stores the id of the desired photo to move

Returns

Returns true if the photo was successfully moved. Otherwise the method returns false.

5.8.2.59 `setPage()`

```
bool ourDB::setPage (
    int al_id,
    QString name,
    QString des,
    QString folder,
    int n_photos,
    QString path ) [static]
```

The method `setPage()` adds a page with the following informations to the database.

Parameters

<i>al_id</i>	- stores the id of the album where the page is being inserted
<i>id</i>	- stores the id of the page
<i>name</i>	- stores the name of the page given by the user
<i>des</i>	- stores the description of the page given by the user
<i>folder</i>	- stores the name of the page after parsing it (name of the folder in user's disk)
<i>n_photos</i>	- stores the number of photos that the page contains
<i>path</i>	- stores the path of the page in the user's disk

Returns

Returns true if the page was successfully added. Otherwise the method returns false.

5.8.2.60 `setPageOther()`

```
bool ourDB::setPageOther (
    int pa_id,
    QDate start_date,
    QDate finish_date ) [static]
```

The method `setPageOther()` adds a [Other](#) page with the following informations to the database.

Parameters

<i>pa_id</i>	- stores the id of the parent page
<i>start_date</i>	- stores the start date
<i>finish_date</i>	- stores the finish date

Returns

Returns true if the page was successfully added. Otherwise the method returns false.

5.8.2.61 setPageParty()

```
bool ourDB::setPageParty (
    int pa_id,
    QDate date,
    QString type_of_party ) [static]
```

The method [setPageParty\(\)](#) adds a [Party](#) page with the following informations to the database.

Parameters

<i>pa_id</i>	- stores the id of the parent page
<i>date</i>	- stores the date when the party occurred
<i>type_of_party</i>	- stores the kind of party it was

Returns

Returns true if the page was successfully added. Otherwise the method returns false.

5.8.2.62 setPageTrip()

```
bool ourDB::setPageTrip (
    int pa_id,
    QDate start_date,
    QDate finish_date ) [static]
```

The method [setPageTrip\(\)](#) adds a [Trip](#) page with the following informations to the database.

Parameters

<i>pa_id</i>	- stores the id of the parent page
<i>start_date</i>	- stores the start date of the trip
<i>finish_date</i>	- stores the finish date of the trip

Returns

Returns true if the page was successfully added. Otherwise the method returns false.

5.8.2.63 setPerson()

```
bool ourDB::setPerson (
    QString name,
    QString gender,
    QDate birthday,
    QString relationship ) [static]
```

The method [setPerson\(\)](#) adds a person with the following informations to the database.

Parameters

<i>name</i>	- stores the name of the person
<i>gender</i>	- stores the gender of the person
<i>birthday</i>	- stores the birthday date of the person
<i>relationship</i>	- stores the relationship between the person and the user

Returns

Returns true if the person was successfully added. Otherwise the method returns false.

5.8.2.64 setPhoto()

```
bool ourDB::setPhoto (
    int pa_id,
    QString name,
    QDate date,
    int size,
    int width,
    int height,
    QString extension ) [static]
```

The method [setPhoto\(\)](#) adds a photo with the following informations to the database.

Parameters

<i>pa_id</i>	- stores the id of the page where the photo is being inserted
<i>name</i>	- stores the name of the photo
<i>date</i>	- stores the creation date of the photo
<i>size</i>	- stores the size of the photo
<i>width</i>	- stores the original width of the photo
<i>height</i>	- stores the original height of the photo
<i>extension</i>	- stores the photo format

Returns

Returns true if the photo was successfully added. Otherwise the method returns false.

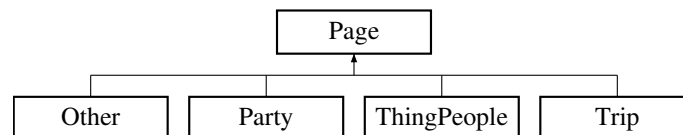
The documentation for this class was generated from the following files:

- database/ourDB.h
- database/ourDB.cpp

5.9 Page Class Reference

```
#include <page.h>
```

Inheritance diagram for Page:

**Public Member Functions**

- [Page](#) ()
Constructor of the class [Page](#).
- void [AddName](#) (QString name)
The method [AddName\(\)](#) add a name to the [Page](#).
- void [AddFolderName](#) (QString folder)
The method [AddFolderName\(\)](#) adds a folder name to the [Page](#).
- void [AddId](#) (int id)
The method [AddId\(\)](#) adds a id to the [Page](#).
- void **AddDescription** (QString description)
- QString [GetName](#) ()
The method [GetName\(\)](#) gets the name of the [Page](#).
- QString [GetDescription](#) ()
The method [GetDescription\(\)](#) gets the description of the [Page](#).
- QString [GetFolder](#) ()
The method [GetFolder\(\)](#) gets the name of the [Page](#) folder.
- int [GetId](#) ()
The method [GetId\(\)](#) gets the id of the [Page](#).

Protected Attributes

- QString [description_](#)
[description_](#) - stores the discription of the [Page](#).

5.9.1 Detailed Description

(Note, this needs exactly one

5.9.2 Member Function Documentation

5.9.2.1 AddFolderName()

```
void Page::AddFolderName (
    QString folder )
```

The method [AddFolderName\(\)](#) adds a folder name to the [Page](#).

Parameters

<i>folder</i>	
---------------	--

5.9.2.2 AddId()

```
void Page::AddId (
    int id )
```

The method [AddId\(\)](#) adds a id to the [Page](#).

Parameters

<i>id</i>	
-----------	--

5.9.2.3 AddName()

```
void Page::AddName (
    QString name )
```

The method [AddName\(\)](#) add a name to the [Page](#).

Parameters

<i>name</i>	
-------------	--

5.9.2.4 GetDescription()

```
QString Page::GetDescription ( )
```

The method [GetDescription\(\)](#) gets the description of the [Page](#).

Returns

Returns the description of the [Page](#).

5.9.2.5 GetFolder()

```
QString Page::GetFolder ( )
```

The method [GetFolder\(\)](#) gets the name of the [Page](#) folder.

Returns

Returns the folder of the [Page](#).

5.9.2.6 GetId()

```
int Page::GetId ( )
```

The method [GetId\(\)](#) gets the id of the [Page](#).

Returns

Returns the id of the [Page](#).

5.9.2.7 GetName()

```
QString Page::GetName ( )
```

The method [GetName\(\)](#) gets the name of the [Page](#).

Returns

Returns the name of the page.

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

- managementSystem/page.h
- managementSystem/page.cpp

5.10 ourDB::pages Struct Reference

The pages struct stores the id, name and description of pages returned from queries.

```
#include <ourDB.h>
```

Public Attributes

- int **id**
- QString **name**
- QString **description**

5.10.1 Detailed Description

The pages struct stores the id, name and description of pages returned from queries.

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

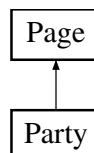
- database/ourDB.h

5.11 Party Class Reference

The [Party](#) class.

```
#include <page.h>
```

Inheritance diagram for Party:



Public Member Functions

- void [AddPageParty](#) (QString description, QString type_of_party, QDate date)
The method [AddPageParty\(\)](#) adds a new party page.
- QString [GetPartyType](#) ()
The method [GetPartyType\(\)](#) gets the type of party.
- QDate [GetDate](#) ()
The method [GetDate\(\)](#) gets the date of the [Party](#).

Additional Inherited Members

5.11.1 Detailed Description

The [Party](#) class.

5.11.2 Member Function Documentation

5.11.2.1 AddPageParty()

```
void Party::AddPageParty (
    QString description,
    QString type_of_party,
    QDate date )
```

The method [AddPageParty\(\)](#) adds a new party page.

Parameters

<i>description</i>	
<i>type_of_party</i>	
<i>date</i>	

5.11.2.2 GetDate()

```
QDate Party::GetDate ( )
```

The method [GetDate\(\)](#) gets the date of the [Party](#).

Returns

Returns the date of the [Party](#).

5.11.2.3 GetPartyType()

```
QString Party::GetPartyType ( )
```

The method [GetPartyType\(\)](#) gets the type of party.

Returns

Return the type of party.

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

- managementSystem/page.h
- managementSystem/page.cpp

5.12 ourDB::party Struct Reference

The party struct stores attributes from a struct page, date and the type of party returned from queries.

```
#include <ourDB.h>
```

Public Attributes

- [pages](#) **page**
- QDate **date**
- QString **type_of_party**

5.12.1 Detailed Description

The party struct stores attributes from a struct page, date and the type of party returned from queries.

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

- database/ourDB.h

5.13 ourDB::person Struct Reference

The person struct stores the id and the name of people returned from queries.

```
#include <ourDB.h>
```

Public Attributes

- int **id**
- QString **name**

5.13.1 Detailed Description

The person struct stores the id and the name of people returned from queries.

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

- database/ourDB.h

5.14 Person Class Reference

```
#include <person.h>
```

Public Member Functions

- [Person](#) ()
Constructor of the class [Person](#).
- void [AddPerson](#) (QString name, QString gender, QDate birthdat_date, QString relationship)
The method [AddPerson\(\)](#) adds a new person storing the values in this class.
- QString [GetName](#) ()
The method [GetName\(\)](#) returns the name of the [Person](#).
- QString [GetGender](#) ()
The method [GetGender\(\)](#) returns the gender of the Gender.
- QString [GetRelationship](#) ()
The method [GetRelationship\(\)](#) returns the relationship of the [Person](#).
- QDate [GetBirthDayDate](#) ()
The method [GetBirthDayDate\(\)](#) returns the birthday date of the [Person](#).

5.14.1 Detailed Description

(Note, this needs exactly one

5.14.2 Member Function Documentation

5.14.2.1 AddPerson()

```
void Person::AddPerson (
    QString name,
    QString gender,
    QDate birthdat_date,
    QString relationship )
```

The method [AddPerson\(\)](#) adds a new person storing the values in this class.

Parameters

<i>name</i>	- stores the name of the person
<i>gender-</i>	stores the gender of the person
<i>birthdat_date</i>	- stores the birthday date of the person
<i>relationship</i>	- stores the relationship of the person

5.14.2.2 GetBirthDayDate()

```
QDate Person::GetBirthDayDate ( )
```

The method [GetBirthDayDate\(\)](#) returns the birthday date of the [Person](#).

Returns

Returns a QDate with the birthday date

5.14.2.3 GetGender()

```
QString Person::GetGender ( )
```

The method [GetGender\(\)](#) returns the gender of the Gender.

Returns

Returns a QString with the Gender

5.14.2.4 GetName()

```
QString Person::GetName ( )
```

The method [GetName\(\)](#) returns the name of the [Person](#).

Returns

Returns a QString with the name of the [Person](#)

5.14.2.5 GetRelationship()

```
QString Person::GetRelationship ( )
```

The method [GetRelationship\(\)](#) returns the relationship of the [Person](#).

Returns

Returns a QString with the Relationship

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

- managementSystem/person.h
- managementSystem/person.cpp

5.15 Photo Class Reference

```
#include <photo.h>
```

Public Member Functions

- [Photo](#) ()
Constructor of the class [Photo](#).
- void [AddName](#) (QString name)
The method [AddName\(\)](#) adds a new name to the class [Photo](#).
- void [AddId](#) (int id)
The method [AddId\(\)](#) adds a new id to the class [Photo](#).
- void [AddExtension](#) (QString extension)
The method [AddExtension\(\)](#) adds an extension to the class [Photo](#).
- QString [GetName](#) ()
The method [GetName\(\)](#) seeks for the name of the photo.
- int [GetId](#) ()
The method [GetId\(\)](#) seeks for the id of the photo.
- QString [GetExtension](#) ()
The method [GetExtension\(\)](#) seeks for the extension of the photo.

5.15.1 Detailed Description

(Note, this needs exactly one

5.15.2 Member Function Documentation

5.15.2.1 AddExtension()

```
void Photo::AddExtension (
    QString extension )
```

The method [AddExtension\(\)](#) adds an extension to the class [Photo](#).

Parameters

<i>extension</i>	- stores the extension of the photo
------------------	-------------------------------------

5.15.2.2 AddId()

```
void Photo::AddId (
    int id )
```

The method [AddId\(\)](#) adds a new id to the class [Photo](#).

Parameters

<i>id</i>	- stores the id of the photo
-----------	------------------------------

5.15.2.3 AddName()

```
void Photo::AddName (
    QString name )
```

The method [AddName\(\)](#) adds a new name to the class [Photo](#).

Parameters

<i>name</i>	- stores the name of the photo
-------------	--------------------------------

5.15.2.4 GetExtension()

```
QString Photo::GetExtension ( )
```

The method [GetExtension\(\)](#) seeks for the extension of the photo.

Returns

Returns a QString that contains the extension of the photo.

5.15.2.5 GetId()

```
int Photo::GetId ( )
```

The method [GetId\(\)](#) seeks for the id of the photo.

Returns

Returns an integer that represents the id of the photo.

5.15.2.6 GetName()

```
QString Photo::GetName ( )
```

The method [GetName\(\)](#) seeks for the name of the photo.

Returns

Returns a QString that contains the name of the photo.

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

- managementSystem/photo.h
- managementSystem/photo.cpp

5.16 ourDB::photos Struct Reference

The photos struct stores the id, name, date and the resolution (width x height) of photos returned from queries.

```
#include <ourDB.h>
```

Public Attributes

- int **id**
- QString **name**
- QDate **date**
- int **width**
- int **height**
- QString **extension**

5.16.1 Detailed Description

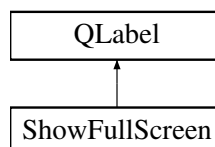
The photos struct stores the id, name, date and the resolution (width x height) of photos returned from queries.

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

- database/ourDB.h

5.17 ShowFullScreen Class Reference

Inheritance diagram for ShowFullScreen:



Public Member Functions

- [ShowFullScreen](#) ()

Constructor of the class showfullscreen.

Protected Member Functions

- virtual void [keyPressEvent](#) (QKeyEvent *ev)

The method [keyPressEvent\(\)](#) waits for a key that corresponds to an event.

5.17.1 Member Function Documentation

5.17.1.1 keyPressEvent()

```
void ShowFullScreen::keyPressEvent (
    QKeyEvent * ev ) [protected], [virtual]
```

The method [keyPressEvent\(\)](#) waits for a key that corresponds to an event.

Parameters

<code>ev</code>	- stores the key pressed
-----------------	--------------------------

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

- design/showfullscreen.h
- design/showfullscreen.cpp

5.18 showfullscreen Class Reference

```
#include <showfullscreen.h>
```

5.18.1 Detailed Description

(Note, this needs exactly one

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

- design/showfullscreen.h

5.19 State Class Reference

```
#include <state.h>
```

Public Member Functions

- [State](#) ()
Constructor of the class [State](#).
- void [Start](#) ()
The method [Start\(\)](#) starts with the initial state.
- bool [NextState](#) ()
The method [NextState\(\)](#) goes to the next state.
- bool [PreviousState](#) ()
The method [PreviousState\(\)](#) goes to the previous state.
- int [GetState](#) ()
The method [GetState\(\)](#) returns the current state.
- bool [NewDir](#) (QString dir)
The method [NewDir\(\)](#) adds or check if exist the system storage folder.
- void [RootDir](#) ()
The method [RootDir\(\)](#) puts the root dir in the dir_ attribute.
- void [MakeDir](#) (QString dir)
The method [MakeDir\(\)](#) creates a new directory.
- bool [NextDir](#) (QString dir)
The method [NextDir\(\)](#) goes to the next directory.
- bool [PreviousDir](#) ()
The method [PreviousDir\(\)](#) goes to the previous directory.
- QString [GetDir](#) ()
The method [GetDir\(\)](#) returns the current directory.
- QString [GetDirPathString](#) ()
The method [GetDirPathString\(\)](#) returns the current directory in short mode.
- QString [GetRootDir](#) ()
The method [GetRootDir\(\)](#) returns the directory of the system storage folder.
- bool [ExistDir](#) (QString item_name)
The method [ExistDir\(\)](#) checks if a directory exists.
- bool [RmDir](#) (QString folder_name)
RmDir.
- bool [RmFile](#) (QString file_name)
RmFile.

5.19.1 Detailed Description

(Note, this needs exactly one

5.19.2 Member Function Documentation

5.19.2.1 ExistDir()

```
bool State::ExistDir (
    QString item_name )
```

The method [ExistDir\(\)](#) checks if a directory exists.

Parameters

<i>The</i>	attribute item_name stores the name of directory to check
------------	---

Returns

Returns true if successeful, else return false

5.19.2.2 GetDir()

```
QString State::GetDir ( )
```

The method [GetDir\(\)](#) returns the current directory.

Returns

Returns a QString with the current directory

5.19.2.3 GetDirPathString()

```
QString State::GetDirPathString ( )
```

The method [GetDirPathString\(\)](#) returns the current directory in short mode.

Returns

Returns a QString with the current directory

5.19.2.4 GetRootDir()

```
QString State::GetRootDir ( )
```

The method [GetRootDir\(\)](#) returns the directory of the system storage folder.

Returns

Returns a QString with the system storage folder path

5.19.2.5 `GetState()`

```
int State::GetState ( )
```

The method [GetState\(\)](#) returns the current state.

Returns

Returns a int with the current state

5.19.2.6 `MakeDir()`

```
void State::MakeDir (
    QString dir )
```

The method [MakeDir\(\)](#) creates a new directory.

Parameters

<i>The</i>	attribute dir stores the name of the new directory
------------	--

5.19.2.7 NewDir()

```
bool State::NewDir (
    QString dir )
```

The method [NewDir\(\)](#) adds or check if exist the system storage folder.

Parameters

<i>The</i>	attribute dir stores the name of the system storage folder
------------	--

Returns

Returns true if successeful, else return false

5.19.2.8 NextDir()

```
bool State::NextDir (
    QString dir )
```

The method [NextDir\(\)](#) goes to the next directory.

Parameters

<i>The</i>	attribute dir stores the new directory to go
------------	--

Returns

Returns true if successeful, else return false

5.19.2.9 NextState()

```
bool State::NextState ( )
```

The method [NextState\(\)](#) goes to the next state.

Returns

Returns true if success, else return false

5.19.2.10 PreviousDir()

```
bool State::PreviousDir ( )
```

The method [PreviousDir\(\)](#) goes to the previous directory.

Returns

Returns true if successeful, else return false

5.19.2.11 PreviousState()

```
bool State::PreviousState ( )
```

The method [PreviousState\(\)](#) goes to the previous state.

Returns

Returns true if success, else return false

5.19.2.12 RmDir()

```
bool State::RmDir (
    QString folder_name )
```

RmDir.

Parameters

<i>folder_name</i>	
--------------------	--

Returns

Returns true if successeful, else return false

5.19.2.13 RmFile()

```
bool State::RmFile (
    QString file_name )
```

RmFile.

Parameters

<code>file_name</code>	
------------------------	--

Returns

Returns true if successeful, else return false

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

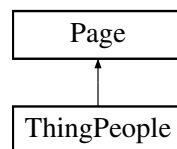
- `managementSystem/state.h`
- `managementSystem/state.cpp`

5.20 ThingPeople Class Reference

The [ThingPeople](#) class.

```
#include <page.h>
```

Inheritance diagram for ThingPeople:



Public Member Functions

- void [AddPageThingPeople](#) (QString description)
The method [AddPageThingPeople\(\)](#) adds a new thing people page.

Additional Inherited Members

5.20.1 Detailed Description

The [ThingPeople](#) class.

5.20.2 Member Function Documentation

5.20.2.1 AddPageThingPeople()

```
void ThingPeople::AddPageThingPeople (
    QString description )
```

The method [AddPageThingPeople\(\)](#) adds a new thing people page.

Parameters

<i>description</i>	
--------------------	--

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

- managementSystem/page.h
- managementSystem/page.cpp

5.21 ourDB::trip Struct Reference

The trip struct stores attributes from a struct page, start date and finish date returned from queries.

```
#include <ourDB.h>
```

Public Attributes

- [pages](#) **page**
- QDate **start_date**
- QDate **finish_date**

5.21.1 Detailed Description

The trip struct stores attributes from a struct page, start date and finish date returned from queries.

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

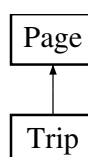
- database/ourDB.h

5.22 Trip Class Reference

The [Trip](#) class.

```
#include <page.h>
```

Inheritance diagram for Trip:



Public Member Functions

- void [AddPageTrip](#) (QString description, QDate start_date, QDate end_date)
The method [AddPageTrip\(\)](#) adds a new [Trip](#) page.
- QDate [GetDateStart](#) ()
The method [GetDateStart\(\)](#) gets the start date of the [Trip](#).
- QDate [GetDateEnd](#) ()
The method [GetDateEnd\(\)](#) gets the end date of the [Trip](#).

Additional Inherited Members

5.22.1 Detailed Description

The [Trip](#) class.

5.22.2 Member Function Documentation

5.22.2.1 AddPageTrip()

```
void Trip::AddPageTrip (  
    QString description,  
    QDate start_date,  
    QDate end_date )
```

The method [AddPageTrip\(\)](#) adds a new [Trip](#) page.

Parameters

<i>description</i>	
<i>start_date</i>	
<i>end_date</i>	

5.22.2.2 GetDateEnd()

```
QDate Trip::GetDateEnd ( )
```

The method [GetDateEnd\(\)](#) gets the end date of the [Trip](#).

Returns

Returns the end date of the [Trip](#).

5.22.2.3 `GetDateStart()`

```
QDate Trip::GetDateStart ( )
```

The method `GetDateStart()` gets the start date of the `Trip`.

Returns

Returns the start date of the `Trip`.

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

- `managementSystem/page.h`
- `managementSystem/page.cpp`

5.23 `uiaddalbum` Class Reference

```
#include <uiaddalbum.h>
```

5.23.1 Detailed Description

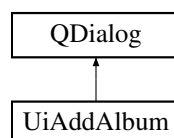
(Note, this needs exactly one

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

- `design/uiaddalbum.h`

5.24 `UiAddAlbum` Class Reference

Inheritance diagram for `UiAddAlbum`:



Signals

- void `SentAlbum` (`Album` album)
The method `SentAlbum()` sends an album.

Public Member Functions

- **`UiAddAlbum`** (`QWidget *parent=0`)

Public Attributes

- QPushButton * **pushButton**

5.24.1 Member Function Documentation

5.24.1.1 SentAlbum

```
void UiAddAlbum::SentAlbum (
    Album album ) [signal]
```

The method [SentAlbum\(\)](#) sends an album.

Parameters

<i>album</i>	- stores a struct that represents an album
--------------	--

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

- design/uiaddalbum.h
- design/uiaddalbum.cpp

5.25 uiaddpage Class Reference

```
#include <uiaddpage.h>
```

5.25.1 Detailed Description

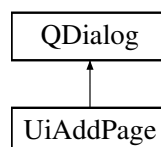
(Note, this needs exactly one

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

- design/uiaddpage.h

5.26 UiAddPage Class Reference

Inheritance diagram for UiAddPage:



Public Slots

- void [SlotGetType](#) (int type)
The method [SlotGetType\(\)](#) seeks for the slot type.

Signals

- void [SentPage](#) ([Page](#) page_send)
The method [SentPage\(\)](#) represents a signal that means a sent page.
- void [SentPageParty](#) ([Party](#) party_send)
The method [SentPageParty\(\)](#) represents a signal that means a sent party page.
- void [SentPageThingPeople](#) ([ThingPeople](#) thing_people_send)
The method [SentPageThingPeople\(\)](#) represents a signal that means a sent thing/people page.
- void [SentPageTrip](#) ([Trip](#) trip_send)
The method [SentPageTrip\(\)](#) represents a signal that means a sent trip page.
- void [SentPageOther](#) ([Other](#) other_send)
The method [SentPageOther\(\)](#) represents a signal that means a sent other page.

Public Member Functions

- **UiAddPage** (QWidget *parent=0)
- void [AcceptAndClose](#) ()
The method [AcceptAndClose](#) represents an action of accepting and closing the window.

5.26.1 Member Function Documentation

5.26.1.1 SentPage

```
void UiAddPage::SentPage (
    Page page_send ) [signal]
```

The method [SentPage\(\)](#) represents a signal that means a sent page.

Parameters

page_send	- stores a struct that represents the page
---------------------------	--

5.26.1.2 SentPageOther

```
void UiAddPage::SentPageOther (
    Other other_send ) [signal]
```

The method [SentPageOther\(\)](#) represents a signal that means a sent other page.

Parameters

<i>other_send</i>	- stores a struct that represents a other page
-------------------	--

5.26.1.3 SentPageParty

```
void UiAddPage::SentPageParty (  
    Party party_send ) [signal]
```

The method [SentPageParty\(\)](#) represents a signal that means a sent party page.

Parameters

<i>party_send</i>	- stores a struct that represents a party page
-------------------	--

5.26.1.4 SentPageThingPeople

```
void UiAddPage::SentPageThingPeople (  
    ThingPeople thing_people_send ) [signal]
```

The method [SentPageThingPeople\(\)](#) represents a signal that means a sent thing/people page.

Parameters

<i>thing_people_send</i>	- stores a struct that represents a thing/people page
--------------------------	---

5.26.1.5 SentPageTrip

```
void UiAddPage::SentPageTrip (  
    Trip trip_send ) [signal]
```

The method [SentPageTrip\(\)](#) represents a signal that means a sent trip page.

Parameters

<i>trip_send</i>	- stores a struct that represents a trip page
------------------	---

5.26.1.6 SlotGetType

```
void UiAddPage::SlotGetType (
    int type ) [slot]
```

The method [SlotGetType\(\)](#) seeks for the slot type.

Parameters

<i>type</i>	- stores the slot type
-------------	------------------------

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

- design/uiaddpage.h
- design/uiaddpage.cpp

5.27 uiaddperson Class Reference

```
#include <uiaddperson.h>
```

5.27.1 Detailed Description

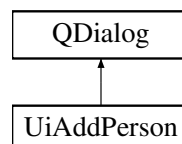
(Note, this needs exactly one

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

- design/uiaddperson.h

5.28 UiAddPerson Class Reference

Inheritance diagram for UiAddPerson:



Public Member Functions

- **UiAddPerson** (QWidget *parent=0)

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

- design/uiaddperson.h
- design/uiaddperson.cpp

5.29 uiassociate Class Reference

```
#include <uiassociate.h>
```

5.29.1 Detailed Description

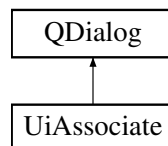
(Note, this needs exactly one

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

- design/uiassociate.h

5.30 UiAssociate Class Reference

Inheritance diagram for UiAssociate:



Public Slots

- void [SlotGetState](#) (int actual_state)
The method [SlotGetState\(\)](#) seeks for the current state.

Public Member Functions

- **UiAssociate** (QWidget *parent=0)
- void [AddItems](#) ()
The method [AddItems\(\)](#) adds itens.
- void [ReceiveIds](#) (QList< int > ids)
The method [ReceiveIds\(\)](#) receives ids.

5.30.1 Member Function Documentation

5.30.1.1 ReceiveIds()

```
void UiAssociate::ReceiveIds (  
    QList< int > ids )
```

The method [ReceiveIds\(\)](#) receives ids.

Parameters

<i>ids</i>	- stores a list that contains the desired ids
------------	---

5.30.1.2 SlotGetState

```
void UiAssociate::SlotGetState (
    int actual_state ) [slot]
```

The method [SlotGetState\(\)](#) seeks for the current state.

Parameters

<i>actual_state</i>	- stores the current state
---------------------	----------------------------

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

- design/uiassociate.h
- design/uiassociate.cpp

5.31 uimodify Class Reference

```
#include <uimove.h>
```

5.31.1 Detailed Description

(Note, this needs exactly one

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

- design/uimove.h

5.32 uimodifyalbum Class Reference

```
#include <uimodifyalbum.h>
```

5.32.1 Detailed Description

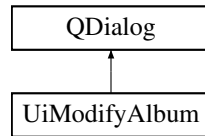
(Note, this needs exactly one

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

- design/uimodifyalbum.h

5.33 UiModifyAlbum Class Reference

Inheritance diagram for UiModifyAlbum:



Public Member Functions

- **UiModifyAlbum** (QWidget *parent=0)
- void [ChangeNameDesc](#) (bool name)
The method [ChangeNameDesc](#) changes the album's name or description depending on what the user wants.
- void [AddAlbumId](#) (int id)
The method [AddAlbumId\(\)](#) adds an id to an album.

5.33.1 Member Function Documentation

5.33.1.1 AddAlbumId()

```
void UiModifyAlbum::AddAlbumId (
    int id )
```

The method [AddAlbumId\(\)](#) adds an id to an album.

Parameters

<i>id</i>	- stores the desired id
-----------	-------------------------

5.33.1.2 ChangeNameDesc()

```
void UiModifyAlbum::ChangeNameDesc (
    bool name )
```

The method [ChangeNameDesc](#) changes the album's name or description depending on what the user wants.

Parameters

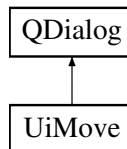
<i>name</i>	- stores true if the user wants to change the album's name. Stores false if he wants to change the description
-------------	--

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

- design/uimodifyalbum.h
- design/uimodifyalbum.cpp

5.34 UiMove Class Reference

Inheritance diagram for UiMove:



Signals

- void [SentSelectedItem](#) (QString selected_item, int item_id)
The method [SentSelectedItem\(\)](#) sens the selected item id and name.

Public Member Functions

- **UiMove** (QWidget *parent=0)
- void [MovePage](#) ()
The method [MovePage\(\)](#) moves a page.
- void [MovePhoto](#) ()
The method [MovePhoto\(\)](#) moves a photo.
- void [AcceptAndClose](#) ()
The method [AcceptAndClose\(\)](#) represents an action of accepting and closing the window.
- void [GetItemId](#) (int item_id)
The method [GetItemId\(\)](#) seeks for id of the item.

5.34.1 Member Function Documentation

5.34.1.1 GetItemId()

```
void UiMove::GetItemId (  
    int item_id )
```

The method [GetItemId\(\)](#) seeks for id of the item.

Parameters

<i>item</i> ↔ _id	- stores the item's id
----------------------	------------------------

5.34.1.2 SentSelectedItem

```
void UiMove::SentSelectedItem (
    QString selected_item,
    int item_id ) [signal]
```

The method [SentSelectedItem\(\)](#) sens the selected item id and name.

Parameters

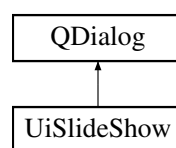
<i>selected_item</i>	- stores the selected item
<i>item_id</i>	- stores the item's id

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

- design/uimove.h
- design/uimove.cpp

5.35 UiSlideShow Class Reference

Inheritance diagram for UiSlideShow:



Public Slots

- void [SlotGetPhotosToShow](#) (QList< [ourDB::photos](#) > photos)
The method [SlotGetPhotosToShow\(\)](#) seeks for the photos to show to the user.
- void [SlotUpdatePhoto](#) ()
The method [SlotUpdatePhoto](#) updates a photo.
- void [SlotGetRootDir](#) (QString root_dir)
The method [SlotGetRootDir\(\)](#) seeks for the root directory.
- void [SlotGetTime](#) (int time)
The method [SlotGetTime\(\)](#) seeks for the time of the audio.
- void [SlotGetAudioFile](#) (QString audio_file)
The method [SlotGetAudioFile\(\)](#) seeks for the file that contains the audio.

Public Member Functions

- **UiSlideShow** (QWidget *parent=0)
- void [PlayAudio](#) ()
The method [PlayAudio\(\)](#) is responsible to play audio for the slideshow.
- void [ShowImage](#) ()
The method [ShowImage\(\)](#) is responsible to show images for the slideshow.

Protected Member Functions

- virtual void **resizeEvent** (QResizeEvent *)
- virtual void **keyPressEvent** (QKeyEvent *e)

5.35.1 Member Function Documentation

5.35.1.1 SlotGetAudioFile

```
void UiSlideShow::SlotGetAudioFile (
    QString audio_file ) [slot]
```

The method [SlotGetAudioFile\(\)](#) seeks for the file that contains the audio.

Parameters

<i>audio_file</i>	- stores the name of the audio file
-------------------	-------------------------------------

5.35.1.2 SlotGetPhotosToShow

```
void UiSlideShow::SlotGetPhotosToShow (
    QList< ourDB::photos > photos ) [slot]
```

The method [SlotGetPhotosToShow\(\)](#) seeks for the photos to show to the user.

Parameters

<i>photos</i>	- stores a list of structs that represent photos
---------------	--

5.35.1.3 SlotGetRootDir

```
void UiSlideShow::SlotGetRootDir (
    QString root_dir ) [slot]
```

The method [SlotGetRootDir\(\)](#) seeks for the root directory.

Parameters

<i>root_dir</i>	- stores the root directory
-----------------	-----------------------------

5.35.1.4 SlotGetTime

```
void UiSlideShow::SlotGetTime (
    int time ) [slot]
```

The method [SlotGetTime\(\)](#) seeks for the time of the audio.

Parameters

<i>time</i>	- stores the time
-------------	-------------------

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

- design/uislideshow.h
- design/uislideshow.cpp

5.36 uislideshow Class Reference

```
#include <uislideshow.h>
```

5.36.1 Detailed Description

(Note, this needs exactly one

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

- design/uislideshow.h

5.37 uislideshowaddimage Class Reference

```
#include <uislideshowaddimage.h>
```

5.37.1 Detailed Description

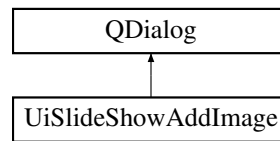
(Note, this needs exactly one

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

- design/uislideshowaddimage.h

5.38 UiSlideShowAddImage Class Reference

Inheritance diagram for UiSlideShowAddImage:



Public Slots

- void [SlotGetStateRootDir](#) (QString root_dir)
The method [SlotGetStateRootDir\(\)](#) seeks for the directory of the root state.

Signals

- void [SentPhotosToSlideShow](#) (QList< [ourDB::photos](#) > photos)
The method [SentPhotosToSlideShow\(\)](#) sends the photos to the slideshow.
- void [SentRootDir](#) (QString root_dir)
The method [SentRootDir](#) sends the root directory.
- void [SentTime](#) (int time)
The method [SentTime\(\)](#) sends the remaining time.
- void [SentAudioFile](#) (QString)
The method [SentAudioFile\(\)](#) sends the name of the audio file.

Public Member Functions

- **UiSlideShowAddImage** (QWidget *parent=0)
- void [AddAlbuns](#) ()
The method [AddAlbuns\(\)](#) adds albuns to the slideshow.

5.38.1 Member Function Documentation

5.38.1.1 SentPhotosToSlideShow

```
void UiSlideShowAddImage::SentPhotosToSlideShow (
    QList< ourDB::photos > photos ) [signal]
```

The method [SentPhotosToSlideShow\(\)](#) sends the photos to the slideshow.

Parameters

<i>photos</i>	- represents a list of photo structs
---------------	--------------------------------------

5.38.1.2 SentRootDir

```
void UiSlideShowAddImage::SentRootDir (
    QString root_dir ) [signal]
```

The method `SentRootDir` sends the root directory.

Parameters

<i>root_dir</i>	- stores the root directory
-----------------	-----------------------------

5.38.1.3 SentTime

```
void UiSlideShowAddImage::SentTime (
    int time ) [signal]
```

The method `SentTime()` sends the remaining time.

Parameters

<i>time</i>	- stores the time
-------------	-------------------

5.38.1.4 SlotGetStateRootDir

```
void UiSlideShowAddImage::SlotGetStateRootDir (
    QString root_dir ) [slot]
```

The method `SlotGetStateRootDir()` seeks for the directory of the root state.

Parameters

<i>root_dir</i>	
-----------------	--

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

- design/uilideshowaddimage.h
- design/uilideshowaddimage.cpp

Index

), [7](#)

AddAlbum

Album, [12](#)

AddAlbumId

UiModifyAlbum, [77](#)

AddExtension

Photo, [58](#)

AddFolderName

Album, [12](#)

Page, [51](#)

AddId

Album, [12](#)

Page, [51](#)

Photo, [58](#)

AddName

Album, [13](#)

Page, [51](#)

Photo, [59](#)

AddPageOther

Other, [21](#)

AddPageParty

Party, [54](#)

AddPageThingPeople

ThingPeople, [67](#)

AddPageTrip

Trip, [69](#)

AddPerson

Person, [56](#)

Album, [11](#)

AddAlbum, [12](#)

AddFolderName, [12](#)

AddId, [12](#)

AddName, [13](#)

GetDescription, [13](#)

GetFolder, [13](#)

GetId, [13](#)

GetName, [14](#)

GetPageType, [14](#)

GetPageTypeString, [14](#)

buildDB

ourDB, [26](#)

ChangeNameDesc

UiModifyAlbum, [77](#)

checkIfAlbumIsEmpty

ourDB, [26](#)

checkIfPageIsEmpty

ourDB, [27](#)

ExistDir

State, [62](#)

getAlbumDescription

ourDB, [27](#)

getAlbumFolderName

ourDB, [27](#)

getAlbumIdFromPage

ourDB, [28](#)

getAlbumLastId

ourDB, [28](#)

getAlbumName

ourDB, [28](#)

getAlbumPageType

ourDB, [29](#)

getAlbums

ourDB, [29](#)

getAlbumsFromPageType

ourDB, [29](#)

getAllPhotos

ourDB, [30](#)

getAssociationAlbum

ourDB, [30](#)

getAssociationPage

ourDB, [31](#)

getAssociationPhoto

ourDB, [31](#)

GetBirthDayDate

Person, [56](#)

GetDate

Party, [54](#)

GetDateEnd

Other, [22](#)

Trip, [69](#)

GetDateStart

Other, [22](#)

Trip, [69](#)

GetDescription

Album, [13](#)

Page, [51](#)

GetDir

State, [63](#)

GetDirPathString

State, [63](#)

GetExtension

Photo, [59](#)

GetFolder

Album, [13](#)

Page, [52](#)

GetGender

- Person, [57](#)
- GetId
 - Album, [13](#)
 - Page, [52](#)
 - Photo, [59](#)
- GetItemId
 - UiMove, [78](#)
- GetName
 - Album, [14](#)
 - Page, [52](#)
 - Person, [57](#)
 - Photo, [59](#)
- getNumberOfAlbums
 - ourDB, [31](#)
- getNumberOfAllPhotos
 - ourDB, [31](#)
- getNumberOfPagesFromAlbum
 - ourDB, [32](#)
- getNumberPhotosFromAlbum
 - ourDB, [32](#)
- getNumberPhotosFromPage
 - ourDB, [32](#)
- getOtherPagesToShow
 - ourDB, [33](#)
- getPageDescription
 - ourDB, [33](#)
- getPageFolderName
 - ourDB, [34](#)
- getPageIdFromPhoto
 - ourDB, [34](#)
- getPageLastId
 - ourDB, [34](#)
- getPageName
 - ourDB, [34](#)
- GetPageType
 - Album, [14](#)
- GetPageTypeString
 - Album, [14](#)
- getPages
 - ourDB, [35](#)
- getPagesToShow
 - ourDB, [35](#)
- getPartyPagesToShow
 - ourDB, [35](#)
- GetPartyType
 - Party, [54](#)
- getPeople
 - ourDB, [36](#)
- getPhotoDate
 - ourDB, [36](#)
- getPhotoDateFromAlbum
 - ourDB, [36](#)
- getPhotoDateFromPage
 - ourDB, [37](#)
- getPhotoExtension
 - ourDB, [37](#)
- getPhotoHeight
 - ourDB, [38](#)
- getPhotoInformation
 - ourDB, [38](#)
- getPhotoLastId
 - ourDB, [38](#)
- getPhotoName
 - ourDB, [38](#)
- getPhotoWidth
 - ourDB, [41](#)
- getPhotos
 - ourDB, [39](#)
- getPhotosAssociatedFromPerson
 - ourDB, [39](#)
- getPhotosFromAlbum
 - ourDB, [39](#)
- getPhotosFromDate
 - ourDB, [40](#)
- getPhotosFromPage
 - ourDB, [40](#), [41](#)
- GetRelationship
 - Person, [57](#)
- GetRootDir
 - State, [63](#)
- GetState
 - State, [63](#)
- getTripPagesToShow
 - ourDB, [42](#)
- keyPressEvent
 - ShowFullScreen, [61](#)
- MainWindow, [16](#)
 - SentItemSelectedId, [17](#)
 - SentItemSelectedName, [17](#)
 - SentPageType, [17](#)
 - SentState, [17](#)
 - SentStateRootDir, [18](#)
 - SlotMoveSelectedPage, [18](#)
 - SlotMoveSelectedPhoto, [18](#)
 - SlotNewAlbum, [19](#)
 - SlotNewPageOther, [19](#)
 - SlotNewPageParty, [19](#)
 - SlotNewPageThingPeople, [20](#)
 - SlotNewPageTrip, [20](#)
- mainwindow, [16](#)
- MakeDir
 - State, [64](#)
- NewDir
 - State, [65](#)
- NextDir
 - State, [65](#)
- NextState
 - State, [65](#)
- Other, [21](#)
 - AddPageOther, [21](#)
 - GetDateEnd, [22](#)
 - GetDateStart, [22](#)
- ourDB::albums, [15](#)

- ourDB::association, 15
- ourDB::other, 20
- ourDB::pages, 53
- ourDB::party, 55
- ourDB::person, 55
- ourDB::photos, 60
- ourDB::trip, 68
- ourDB, 22
 - buildDB, 26
 - checkIfAlbumIsEmpty, 26
 - checkIfPagelsEmpty, 27
 - getAlbumDescription, 27
 - getAlbumFolderName, 27
 - getAlbumIdFromPage, 28
 - getAlbumLastID, 28
 - getAlbumName, 28
 - getAlbumPageType, 29
 - getAlbums, 29
 - getAlbumsFromPageType, 29
 - getAllPhotos, 30
 - getAssociationAlbum, 30
 - getAssociationPage, 31
 - getAssociationPhoto, 31
 - getNumberOfAlbums, 31
 - getNumberOfAllPhotos, 31
 - getNumberOfPagesFromAlbum, 32
 - getNumberPhotosFromAlbum, 32
 - getNumberPhotosFromPage, 32
 - getOtherPagesToShow, 33
 - getPageDescription, 33
 - getPageFolderName, 34
 - getPageldFromPhoto, 34
 - getPageLastID, 34
 - getPageName, 34
 - getPages, 35
 - getPagesToShow, 35
 - getPartyPagesToShow, 35
 - getPeople, 36
 - getPhotoDate, 36
 - getPhotoDateFromAlbum, 36
 - getPhotoDateFromPage, 37
 - getPhotoExtension, 37
 - getPhotoHeight, 38
 - getPhotoInformation, 38
 - getPhotoLastID, 38
 - getPhotoName, 38
 - getPhotoWidth, 41
 - getPhotos, 39
 - getPhotosAssociatedFromPerson, 39
 - getPhotosFromAlbum, 39
 - getPhotosFromDate, 40
 - getPhotosFromPage, 40, 41
 - getTripPagesToShow, 42
 - removeAlbum, 42
 - removePage, 42
 - removePhoto, 43
 - setAlbum, 43
 - setAlbumDescription, 44
 - setAlbumFolderName, 44
 - setAlbumName, 44
 - setAssociation, 45
 - setAssociationAlbum, 45
 - setAssociationPage, 46
 - setMovePage, 46
 - setMovePhoto, 46
 - setPage, 47
 - setPageOther, 47
 - setPageParty, 48
 - setPageTrip, 48
 - setPerson, 49
 - setPhoto, 49
- Page, 50
 - AddFolderName, 51
 - AddId, 51
 - AddName, 51
 - GetDescription, 51
 - GetFolder, 52
 - GetId, 52
 - GetName, 52
- Party, 53
 - AddPageParty, 54
 - GetDate, 54
 - GetPartyType, 54
- Person, 55
 - AddPerson, 56
 - GetBirthDayDate, 56
 - GetGender, 57
 - GetName, 57
 - GetRelationship, 57
- Photo, 57
 - AddExtension, 58
 - AddId, 58
 - AddName, 59
 - GetExtension, 59
 - GetId, 59
 - GetName, 59
- PreviousDir
 - State, 65
- PreviousState
 - State, 66
- ReceiveIds
 - UiAssociate, 75
- removeAlbum
 - ourDB, 42
- removePage
 - ourDB, 42
- removePhoto
 - ourDB, 43
- Rmdir
 - State, 66
- RmFile
 - State, 66
- SentAlbum
 - UiAddAlbum, 71

- SentItemSelectedId
 - MainWindow, [17](#)
- SentItemSelectedName
 - MainWindow, [17](#)
- SentPage
 - UiAddPage, [72](#)
- SentPageOther
 - UiAddPage, [72](#)
- SentPageParty
 - UiAddPage, [73](#)
- SentPageThingPeople
 - UiAddPage, [73](#)
- SentPageTrip
 - UiAddPage, [73](#)
- SentPageType
 - MainWindow, [17](#)
- SentPhotosToSlideShow
 - UiSlideShowAddImage, [82](#)
- SentRootDir
 - UiSlideShowAddImage, [83](#)
- SentSelectedItem
 - UiMove, [79](#)
- SentState
 - MainWindow, [17](#)
- SentStateRootDir
 - MainWindow, [18](#)
- SentTime
 - UiSlideShowAddImage, [83](#)
- setAlbum
 - ourDB, [43](#)
- setAlbumDescription
 - ourDB, [44](#)
- setAlbumFolderName
 - ourDB, [44](#)
- setAlbumName
 - ourDB, [44](#)
- setAssociation
 - ourDB, [45](#)
- setAssociationAlbum
 - ourDB, [45](#)
- setAssociationPage
 - ourDB, [46](#)
- setMovePage
 - ourDB, [46](#)
- setMovePhoto
 - ourDB, [46](#)
- setPage
 - ourDB, [47](#)
- setPageOther
 - ourDB, [47](#)
- setPageParty
 - ourDB, [48](#)
- setPageTrip
 - ourDB, [48](#)
- setPerson
 - ourDB, [49](#)
- setPhoto
 - ourDB, [49](#)
- ShowFullScreen, [60](#)
 - keyPressEvent, [61](#)
- showfullscreen, [61](#)
- SlotGetAudioFile
 - UiSlideShow, [80](#)
- SlotGetPhotosToShow
 - UiSlideShow, [80](#)
- SlotGetRootDir
 - UiSlideShow, [80](#)
- SlotGetState
 - UiAssociate, [76](#)
- SlotGetStateRootDir
 - UiSlideShowAddImage, [83](#)
- SlotGetTime
 - UiSlideShow, [81](#)
- SlotGetType
 - UiAddPage, [73](#)
- SlotMoveSelectedPage
 - MainWindow, [18](#)
- SlotMoveSelectedPhoto
 - MainWindow, [18](#)
- SlotNewAlbum
 - MainWindow, [19](#)
- SlotNewPageOther
 - MainWindow, [19](#)
- SlotNewPageParty
 - MainWindow, [19](#)
- SlotNewPageThingPeople
 - MainWindow, [20](#)
- SlotNewPageTrip
 - MainWindow, [20](#)
- State, [62](#)
 - ExistDir, [62](#)
 - GetDir, [63](#)
 - GetDirPathString, [63](#)
 - GetRootDir, [63](#)
 - GetState, [63](#)
 - MakeDir, [64](#)
 - NewDir, [65](#)
 - NextDir, [65](#)
 - NextState, [65](#)
 - PreviousDir, [65](#)
 - PreviousState, [66](#)
 - RmDir, [66](#)
 - RmFile, [66](#)
- ThingPeople, [67](#)
 - AddPageThingPeople, [67](#)
- Trip, [68](#)
 - AddPageTrip, [69](#)
 - GetDateEnd, [69](#)
 - GetDateStart, [69](#)
- UiAddAlbum, [70](#)
 - SentAlbum, [71](#)
- UiAddPage, [71](#)
 - SentPage, [72](#)
 - SentPageOther, [72](#)
 - SentPageParty, [73](#)

- SentPageThingPeople, [73](#)
 - SentPageTrip, [73](#)
 - SlotGetType, [73](#)
- UiAddPerson, [74](#)
- UiAssociate, [75](#)
 - Receivelds, [75](#)
 - SlotGetState, [76](#)
- UiModifyAlbum, [77](#)
 - AddAlbumId, [77](#)
 - ChangeNameDesc, [77](#)
- UiMove, [78](#)
 - GetItemId, [78](#)
 - SentSelectedItem, [79](#)
- UiSlideShow, [79](#)
 - SlotGetAudioFile, [80](#)
 - SlotGetPhotosToShow, [80](#)
 - SlotGetRootDir, [80](#)
 - SlotGetTime, [81](#)
- UiSlideShowAddImage, [82](#)
 - SentPhotosToSlideShow, [82](#)
 - SentRootDir, [83](#)
 - SentTime, [83](#)
 - SlotGetStateRootDir, [83](#)
- uiaddalbum, [70](#)
- uiaddpage, [71](#)
- uiaddperson, [74](#)
- uiassociate, [75](#)
- uimodify, [76](#)
- uimodifyalbum, [76](#)
- uislideshow, [81](#)
- uislideshowaddimage, [81](#)