

FactDev

0.1

Generated by Doxygen 1.8.6

Mon Jan 19 2015 14:21:12

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	Class Documentation	5
3.1	AddContributoryDialog Class Reference	5
3.1.1	Detailed Description	5
3.1.2	Constructor & Destructor Documentation	5
3.1.2.1	AddContributoryDialog	5
3.2	AddProjectDialog Class Reference	6
3.2.1	Detailed Description	6
3.2.2	Constructor & Destructor Documentation	6
3.2.2.1	AddProjectDialog	6
3.3	Billing Class Reference	6
3.3.1	Detailed Description	7
3.3.2	Member Function Documentation	7
3.3.2.1	getContributories	7
3.3.2.2	getDate	8
3.3.2.3	getNumber	8
3.3.2.4	getTitle	8
3.3.2.5	hydrat	8
3.3.2.6	isBilling	8
3.3.2.7	setDate	8
3.3.2.8	setIsBilling	8
3.3.2.9	setNumber	9
3.3.2.10	setTitle	9
3.4	ComboBoxModelWidget Class Reference	9
3.5	connexiondialog Class Reference	9
3.6	Contributory Class Reference	10
3.6.1	Detailed Description	10

3.7	Customer Class Reference	10
3.7.1	Detailed Description	12
3.7.2	Constructor & Destructor Documentation	12
3.7.2.1	Customer	12
3.7.3	Member Function Documentation	12
3.7.3.1	getAddress	12
3.7.3.2	getCity	13
3.7.3.3	getCompany	13
3.7.3.4	getCountry	13
3.7.3.5	getEmail	13
3.7.3.6	getFax	13
3.7.3.7	getFirstnameReferent	13
3.7.3.8	getLastnameReferent	13
3.7.3.9	getMobilePhone	14
3.7.3.10	getPhone	14
3.7.3.11	getPostalCode	14
3.7.3.12	hydrat	14
3.7.3.13	operator!=	14
3.7.3.14	operator==	14
3.7.3.15	setAddress	15
3.7.3.16	setCity	15
3.7.3.17	setCompany	15
3.7.3.18	setCountry	15
3.7.3.19	setEmail	15
3.7.3.20	setFax	15
3.7.3.21	setFirstnameReferent	16
3.7.3.22	setLastnameReferent	16
3.7.3.23	setMobilePhone	16
3.7.3.24	setPhone	16
3.7.3.25	setPostalCode	16
3.8	CustomerContextualMenu Class Reference	16
3.9	CustomerDatabase Class Reference	17
3.9.1	Detailed Description	18
3.9.2	Member Function Documentation	18
3.9.2.1	addCustomer	18
3.9.2.2	getCustomer	18
3.9.2.3	getCustomersTable	18
3.9.2.4	getCustomersTree	18
3.9.2.5	getNbCustomers	19
3.9.2.6	instance	19

3.9.2.7	removeCustomer	19
3.10	CustomerDataWidget Class Reference	19
3.11	Database Class Reference	20
3.11.1	Detailed Description	21
3.11.2	Member Function Documentation	21
3.11.2.1	executeFile	21
3.11.2.2	instance	21
3.11.2.3	lastError	21
3.11.2.4	setDatabase	22
3.11.2.5	testCases	22
3.11.2.6	value	22
3.12	DbException Class Reference	22
3.12.1	Detailed Description	23
3.12.2	Constructor & Destructor Documentation	23
3.12.2.1	DbException	23
3.12.3	Member Function Documentation	23
3.12.3.1	popupMessage	23
3.13	DialogAddCustomer Class Reference	24
3.13.1	Detailed Description	24
3.13.2	Constructor & Destructor Documentation	24
3.13.2.1	DialogAddCustomer	24
3.14	IDatabaseModel Class Reference	25
3.14.1	Detailed Description	25
3.15	Log Class Reference	25
3.15.1	Detailed Description	26
3.15.2	Member Function Documentation	26
3.15.2.1	instance	26
3.15.2.2	write	26
3.15.3	Friends And Related Function Documentation	26
3.15.3.1	operator<<	26
3.16	MainWindow Class Reference	27
3.17	Parameters Class Reference	27
3.18	Popup Class Reference	28
3.19	Project Class Reference	28
3.19.1	Detailed Description	29
3.19.2	Constructor & Destructor Documentation	29
3.19.2.1	Project	29
3.19.3	Member Function Documentation	29
3.19.3.1	getCustomer	29
3.19.3.2	getDailyRate	29

3.19.3.3	getDescription	29
3.19.3.4	getName	29
3.19.3.5	hydrat	30
3.19.3.6	setCustomer	31
3.19.3.7	setDailyRate	31
3.19.3.8	setDescription	31
3.19.3.9	setName	31
3.20	ProjectDatabase Class Reference	31
3.20.1	Detailed Description	32
3.20.2	Member Function Documentation	32
3.20.2.1	addProject	32
3.20.2.2	getNbProjects	32
3.20.2.3	getNbProjectsForACustomer	32
3.20.2.4	getProject	33
3.20.2.5	instance	33
3.20.2.6	removeProject	33
3.21	ProjectsWidget Class Reference	33
3.22	RateWidget Class Reference	34
3.23	Search Class Reference	34
3.23.1	Detailed Description	35
3.24	searchWidget Class Reference	35
3.24.1	Member Function Documentation	35
3.24.1.1	getCurrentCustomerId	35
3.25	User Class Reference	36
3.25.1	Detailed Description	37
3.25.2	Constructor & Destructor Documentation	37
3.25.2.1	User	37
3.25.3	Member Function Documentation	37
3.25.3.1	getAddress	37
3.25.3.2	getCity	38
3.25.3.3	getCompany	38
3.25.3.4	getEmail	38
3.25.3.5	getFirstname	38
3.25.3.6	getLastname	38
3.25.3.7	getMobilePhone	38
3.25.3.8	getNoSiret	38
3.25.3.9	getPhone	39
3.25.3.10	getPostalCode	39
3.25.3.11	getTitle	39
3.25.3.12	hydrat	39

3.25.3.13 setAddress	39
3.25.3.14 setCity	39
3.25.3.15 setCompany	39
3.25.3.16 setEmail	40
3.25.3.17 setFirstname	40
3.25.3.18 setLastname	40
3.25.3.19 setMobilePhone	40
3.25.3.20 setNoSiret	40
3.25.3.21 setPhone	40
3.25.3.22 setPostalCode	41
3.25.3.23 setTitle	42
3.26 UserDatabase Class Reference	42
3.26.1 Detailed Description	42
3.26.2 Member Function Documentation	43
3.26.2.1 getUser	43
3.26.2.2 getUserTable	43
3.26.2.3 instance	43
3.27 UserDataDialog Class Reference	43
3.27.1 Detailed Description	44
3.27.2 Constructor & Destructor Documentation	44
3.27.2.1 UserDataDialog	44
3.28 Utils Class Reference	44
Index	45

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Database	??
CustomerDatabase	??
ProjectDatabase	??
UserDatabase	??
exception	
DbException	??
IDatabaseModel	??
Billing	??
Contributory	??
Customer	??
Project	??
User	??
Log	??
Parameters	??
Popup	??
QDialog	
AddContributoryDialog	??
AddProjectDialog	??
connexiondialog	??
DialogAddCustomer	??
UserDataDialog	??
QMainWindow	
MainWindow	??
QMenu	
CustomerContextualMenu	??
QWidget	
ComboBoxModelWidget	??
CustomerDataWidget	??
ProjectsWidget	??
RateWidget	??
searchWidget	??
Search	??
Utils	??

Chapter 2

Class Index

2.1 Class List

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

AddContributoryDialog		
Windows to add a new Contributory	??
AddProjectDialog		
Windows to add a new Project	??
Billing		
: Billing of a Customer	??
ComboBoxModelWidget	??
connexiondialog	??
Contributory		
The Contributory class	??
Customer		
The Customer class	??
CustomerContextualMenu	??
CustomerDatabase		
The CustomerDatabase class Customer table database	??
CustomerDataWidget	??
Database		
The Database class Master class for all database access	??
DbException		
For database exception : queries, db file,	??
DialogAddCustomer		
Window to add or modify a Customer	??
IDatabaseModel		
The IDatabaseModel class	??
Log		
For Simple management of log	??
MainWindow	??
Parameters	??
Popup	??
Project		
: Project linked to a Customer	??
ProjectDatabase		
The ProjectDatabase class	??
ProjectsWidget	??
RateWidget	??
Search		
The Search class	??
searchWidget	??

User	
User of it application	??
UserDatabase	
Access to User data in the the table User of the Database	??
UserDataDialog	
Window to fill user data	??
Utils	??

Chapter 3

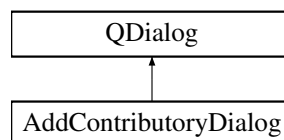
Class Documentation

3.1 AddContributoryDialog Class Reference

The [AddContributoryDialog](#) class Windows to add a new [Contributory](#).

```
#include <addcontributorydialog.h>
```

Inheritance diagram for AddContributoryDialog:



Public Member Functions

- [AddContributoryDialog](#) (QWidget *parent=0)
[AddContributoryDialog](#) Construct a new windows AddContributory.

3.1.1 Detailed Description

The [AddContributoryDialog](#) class Windows to add a new [Contributory](#).

Author

Florent Berbie

3.1.2 Constructor & Destructor Documentation

3.1.2.1 AddContributoryDialog::AddContributoryDialog (QWidget * *parent* = 0) [explicit]

[AddContributoryDialog](#) Construct a new windows AddContributory.

Parameters

<i>parent</i>	QWidget
---------------	---------

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

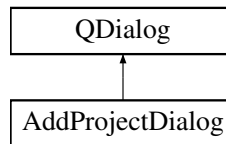
- /home/aroquemaurel/projets/qt/FactDev/src/dialogs/addcontributorydialog.h
- /home/aroquemaurel/projets/qt/FactDev/src/dialogs/addcontributorydialog.cpp

3.2 AddProjectDialog Class Reference

The [AddProjectDialog](#) class Windows to add a new [Project](#).

```
#include <addprojectdialog.h>
```

Inheritance diagram for AddProjectDialog:



Public Member Functions

- [AddProjectDialog](#) (int id=0, QWidget *parent=0)
[AddProjectDialog](#) Construct a windows [AddProjectDialog](#).
- void [accept](#) ()
accept Valid data inputed by user and add these data in [Database](#)
- void [reject](#) ()
reject Cancel the operation and close the windows

3.2.1 Detailed Description

The [AddProjectDialog](#) class Windows to add a new [Project](#).

Author

Florent Berbie

3.2.2 Constructor & Destructor Documentation

3.2.2.1 [AddProjectDialog::AddProjectDialog](#) (int *id* = 0, QWidget * *parent* = 0) [explicit]

[AddProjectDialog](#) Construct a windows [AddProjectDialog](#).

Parameters

<i>id</i>	Project identity
<i>parent</i>	QWidget of the current windows

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

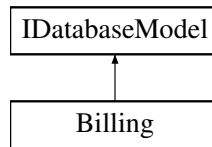
- /home/aroquemaurel/projets/qt/FactDev/src/dialogs/addprojectdialog.h
- /home/aroquemaurel/projets/qt/FactDev/src/dialogs/addprojectdialog.cpp

3.3 Billing Class Reference

The [Billing](#) class : [Billing](#) of a [Customer](#).

```
#include <billing.h>
```

Inheritance diagram for Billing:



Public Member Functions

- **Billing** ()
Billing::Billing. Construct a **Billing**.
- void **commit** ()
Billing::commit. Insert a modification in **Billing** table on the database.
- void **hydrat** (int getId)
Billing::hydrat. Update of the **Billing** which is specified by getId
- void **remove** ()
Billing::remove. Remove a **Billing**.
- QMap< **Project**, QList< **Contributory** > > **getContributories** () const
Billing::getContributories. Return a map of **Contributory** for each **Project** of the **Billing**
- void **setContributories** (const QMap< **Project**, QList< **Contributory** > > &getContributories)
- QString **getTitle** () const
Billing::getTitle. return title of **Billing**
- void **setTitle** (const QString &getTitle)
Billing::setTitle. Modify the title of **Billing**
- int **getNumber** () const
Billing::getNumber. Return number of the **Billing**.
- void **setNumber** (int getNumber)
Billing::setNumber. Modify _number of **Billing**.
- bool **isBilling** () const
Billing::isBilling. Return if it's a billing or a quote.
- void **setIsBilling** (bool isBilling)
Billing::setIsBilling. Modify isBilling of **Billing**.
- QDate **getDate** () const
Billing::getDate. return date of the **Billing**
- void **setDate** (const QDate &getDate)
Billing::setDate. Modify date of the **Billing**

Additional Inherited Members

3.3.1 Detailed Description

The **Billing** class : **Billing** of a **Customer**.

3.3.2 Member Function Documentation

3.3.2.1 QMap< **Project**, QList< **Contributory** > > **Billing::getContributories** () const

Billing::getContributories. Return a map of **Contributory** for each **Project** of the **Billing**

Returns

QMap<**Project**, QList<Contributory>>

3.3.2.2 `QDate Billing::getDate () const`

[Billing::getDate](#). return date of the [Billing](#)

Returns

date of [Billing](#)

3.3.2.3 `int Billing::getNumber () const`

[Billing::getNumber](#). Return number of the [Billing](#).

Returns

`_number` of [Billing](#)

3.3.2.4 `QString Billing::getTitle () const`

[Billing::getTitle](#). return title of [Billing](#)

Returns

title of [Billing](#)

3.3.2.5 `void Billing::hydrat (int getId) [virtual]`

[Billing::hydrat](#). Update of the [Billing](#) which is specified by *getId*

Parameters

<i>getId</i>	
--------------	--

Implements [IDatabaseModel](#).

3.3.2.6 `bool Billing::isBilling () const`

[Billing::isBilling](#). Return if it's a billing or a quote.

Returns

if it's billing or a quote

3.3.2.7 `void Billing::setDate (const QDate & getDate)`

[Billing::setDate](#). Modify *date* of the [Billing](#)

Parameters

<i>getDate</i>	the new date of the Billing
----------------	---

3.3.2.8 `void Billing::setIsBilling (bool isBilling)`

[Billing::setIsBilling](#). Modify *isBilling* of [Billing](#).

Parameters

<i>isBilling</i>	
------------------	--

3.3.2.9 void Billing::setNumber (int *getNumber*)

[Billing::setNumber](#). Modify *_number* of [Billing](#).

Parameters

<i>getNumber</i>	the new number of the Billing
------------------	---

3.3.2.10 void Billing::setTitle (const QString & *getTitle*)

[Billing::setTitle](#). Modify the title of [Billing](#)

Parameters

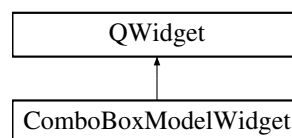
<i>getTitle</i>	Modify the title with <i>getTitle</i>
-----------------	---------------------------------------

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

- /home/aroquemaurel/projets/qt/FactDev/src/models/billing.h
- /home/aroquemaurel/projets/qt/FactDev/src/models/billing.cpp

3.4 ComboBoxModelWidget Class Reference

Inheritance diagram for ComboBoxModelWidget:



Public Member Functions

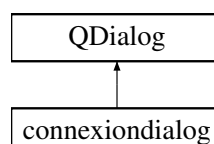
- **ComboBoxModelWidget** (QWidget *parent=0)

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

- /home/aroquemaurel/projets/qt/FactDev/src/widgets/comboboxmodelwidget.h
- /home/aroquemaurel/projets/qt/FactDev/src/widgets/comboboxmodelwidget.cpp

3.5 connexiondialog Class Reference

Inheritance diagram for connexiondialog:



Public Member Functions

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

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

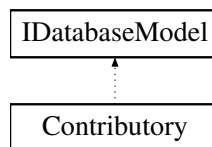
- /home/aroquemaurel/projets/qt/FactDev/src/connexiondialog.h
- /home/aroquemaurel/projets/qt/FactDev/src/connexiondialog.cpp

3.6 Contributory Class Reference

The [Contributory](#) class.

```
#include <contributory.h>
```

Inheritance diagram for Contributory:



Public Member Functions

- void **commit** ()
- void **hydrat** (int getld)
- void **remove** ()
- [Project](#) **getProject** () const
- void **setProject** (const [Project](#) &value)

3.6.1 Detailed Description

The [Contributory](#) class.

Author

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

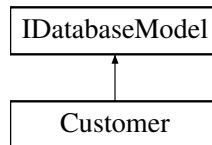
- /home/aroquemaurel/projets/qt/FactDev/src/models/contributory.h
- /home/aroquemaurel/projets/qt/FactDev/src/models/contributory.cpp

3.7 Customer Class Reference

The [Customer](#) class.

```
#include <customer.h>
```

Inheritance diagram for Customer:



Public Member Functions

- `Customer ()`
Customer::Customer Construct a *Customer*.
- `Customer (int id)`
Customer::Customer Construct a *Customer* who is specified by *id*
- `void commit ()`
Customer::commit Update customer data on the database.
- `void hydrat (int id)`
Customer::hydrat Insert into database informations related to the *Customer* who is specified by *id*
- `void remove ()`
Customer::remove Remove the current customer.
- `QString getFirstnameReferent () const`
Customer::getFirstnameReferent Return the firstname of the referent *Customer*.
- `void setFirstnameReferent (const QString &firstnameReferent)`
Customer::setFirstnameReferent Replace the referent firstname of the customer by *firstnameReferent*
- `QString getLastNameReferent () const`
Customer::getLastNameReferent Return the lastname of the referent *Customer*.
- `void setLastNameReferent (const QString &lastnameReferent)`
Customer::setLarstnameReferent Replace the referent lastname of the customer by *lastnameReferent*
- `QString getCompany () const`
Customer::getCompany Return the name of the company customer.
- `void setCompany (const QString &company)`
Customer::setCompany Replace the company name by *company*
- `QString getAddress () const`
Customer::getAddress Return the address (name and street number) of the customer company.
- `void setAddress (const QString &address)`
Customer::setAddress Replace the current company address by the new address
- `QString getPostalCode () const`
Customer::getPostalCode Return the postal code of the customer company.
- `void setPostalCode (const QString &postalCode)`
Customer::setPostalCode Replace the current postal code of the customer company by the new *postalCode*
- `QString getCity () const`
Customer::getCity Return the city of the customer company.
- `void setCity (const QString &city)`
Customer::setCity Replace the current city of the customer company by the new city
- `QString getCountry () const`
Customer::getCountry Return the country of the customer company.
- `void setCountry (const QString &country)`
Customer::setCountry Replace the country of the *Customer* company by *country*
- `QString getEmail () const`
Customer::getEmail Return the email of the customer company.
- `void setEmail (const QString &email)`
Customer::setEmail Replace the email of the *Customer* company by *email*

- QString [getMobilePhone](#) () const
[Customer::getMobilePhone](#) Return the number of the mobile phone of the customer company.
- void [setMobilePhone](#) (const QString &mobilePhone)
[Customer::setMobilePhone](#) Replace the current mobile phone of the customer company by mobilePhone
- QString [getPhone](#) () const
[Customer::getPhone](#) Return the number of desktop phone.
- void [setPhone](#) (const QString &phone)
[Customer::setPhone](#) Replace the current number of desktop phone by phone
- QString [getFax](#) () const
[Customer::getFax](#) Return the fax number.
- void [setFax](#) (const QString &fax)
[Customer::setFax](#) Replace the current fax number by fax
- bool [operator==](#) (const [Customer](#) &c)
[Customer::operator ==](#) Re-define the operator "==" to compare if the current customer is the same to the other [Customer](#) c Return TRUE if both customers are the same, else FALSE.
- bool [operator!=](#) (const [Customer](#) &c)
[Customer::operator !=](#) Re-define the operator "!=" to compare if the current customer is different to the other [Customer](#) c Return TRUE if both customers are different, else FALSE.

Additional Inherited Members

3.7.1 Detailed Description

The [Customer](#) class.

Author

Antoine de Roquemaurel
Florent Berbie

3.7.2 Constructor & Destructor Documentation

3.7.2.1 [Customer::Customer](#) (int *id*)

[Customer::Customer](#) Construct a [Customer](#) who is specified by *id*

Parameters

<i>id</i>	Customer identify
-----------	-----------------------------------

3.7.3 Member Function Documentation

3.7.3.1 QString [Customer::getAddress](#) () const

[Customer::getAddress](#) Return the address (name and street number) of the customer company.

Returns

company address

3.7.3.2 QString Customer::getCity () const

[Customer::getCity](#) Return the city of the customer company.

Returns

city of the customer company

3.7.3.3 QString Customer::getCompany () const

[Customer::getCompany](#) Return the name of the company customer.

Returns

name of the company customer

3.7.3.4 QString Customer::getCountry () const

[Customer::getCountry](#) Return the country of the customer company.

Returns

country of the customer company

3.7.3.5 QString Customer::getEmail () const

[Customer::getEmail](#) Return the email of the customer company.

Returns

email of the customer company

3.7.3.6 QString Customer::getFax () const

[Customer::getFax](#) Return the fax number.

Returns

fax number

3.7.3.7 QString Customer::getFirstnameReferent () const

[Customer::getFirstnameReferent](#) Return the firstname of the referent [Customer](#).

Returns

Referent firstname of the customer

3.7.3.8 QString Customer::getLastnameReferent () const

[Customer::getLastnameReferent](#) Return the lastname of the referent [Customer](#).

Returns

Referent lastname of the customer

3.7.3.9 QString Customer::getMobilePhone () const

[Customer::getMobilePhone](#) Return the number of the mobile phone of the customer company.

Returns

number of mobile phone

3.7.3.10 QString Customer::getPhone () const

[Customer::getPhone](#) Return the number of desktop phone.

Returns

number of desktop phone

3.7.3.11 QString Customer::getPostalCode () const

[Customer::getPostalCode](#) Return the postal code of the customer company.

Returns

postal code of the customer company

3.7.3.12 void Customer::hydrat (int *id*) [virtual]

[Customer::hydrat](#) Insert into database informations related to the [Customer](#) who is specified by *id*

Parameters

<i>id</i>	Customer identify
-----------	-----------------------------------

Implements [IDatabaseModel](#).

3.7.3.13 bool Customer::operator!= (const Customer & *c*)

[Customer::operator ==](#) Re-define the operator "!=" to compare if the current customer is differnt to the other [Customer](#) *c* Return TRUE if both customers are different, else FALSE.

Parameters

<i>c</i>	Customer to compare
----------	-------------------------------------

Returns

boolean

3.7.3.14 bool Customer::operator== (const Customer & *c*)

[Customer::operator ==](#) Re-define the operator "==" to compare if the current customer is the same to the other [Customer](#) *c* Return TRUE if both customers are the same, else FALSE.

Parameters

<i>c</i>	Customer to compare
----------	-------------------------------------

Returns

boolean

3.7.3.15 void Customer::setAddress (const QString & *address*)

[Customer::setAddress](#) Replace the current company address by the new *address*

Parameters

<i>address</i>	New company address
----------------	---------------------

3.7.3.16 void Customer::setCity (const QString & *city*)

[Customer::setCity](#) Replace the current city of the customer company by the new *city*

Parameters

<i>city</i>	New city of the customer company
-------------	----------------------------------

3.7.3.17 void Customer::setCompany (const QString & *company*)

[Customer::setCompany](#) Replace the company name by *company*

Parameters

<i>company</i>	New company name
----------------	------------------

3.7.3.18 void Customer::setCountry (const QString & *country*)

[Customer::setCountry](#) Replace the country of the [Customer](#) company by *country*

Parameters

<i>country</i>	New country of the customer company
----------------	-------------------------------------

3.7.3.19 void Customer::setEmail (const QString & *email*)

[Customer::setEmail](#) Replace the email of the [Customer](#) company by *email*

Parameters

<i>email</i>	New email of the customer company
--------------	-----------------------------------

3.7.3.20 void Customer::setFax (const QString & *fax*)

[Customer::setFax](#) Replace the current fax number by *fax*

Parameters

<i>fax</i>	new fax number
------------	----------------

3.7.3.21 void Customer::setFirstnameReferent (const QString & *firstnameReferent*)

[Customer::setFirstnameReferent](#) Replace the referent firstname of the customer by *firstnameReferent*

Parameters

<i>firstname-Referent</i>	New referent firstname of the Customer
---------------------------	--

3.7.3.22 void Customer::setLastnameReferent (const QString & *lastnameReferent*)

Customer::setLarstnameReferent Replace the referent lastname of the customer by *lastnameReferent*

Parameters

<i>lastname-Referent</i>	New referent lastname of the Customer
--------------------------	---

3.7.3.23 void Customer::setMobilePhone (const QString & *mobilePhone*)

[Customer::setMobilePhone](#) Replace the current mobile phone of the customer company by *mobilePhone*

Parameters

<i>mobilePhone</i>	New number of mobile phone
--------------------	----------------------------

3.7.3.24 void Customer::setPhone (const QString & *phone*)

[Customer::setPhone](#) Replace the current number of desktop phone by *phone*

Parameters

<i>phone</i>	New number of desktop phone
--------------	-----------------------------

3.7.3.25 void Customer::setPostalCode (const QString & *postalCode*)

[Customer::setPostalCode](#) Replace the current postal code of the customer company by the new *postalCode*

Parameters

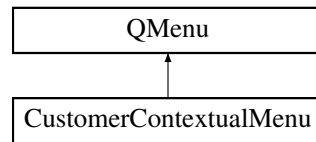
<i>postalCode</i>	New postal code of the customer company
-------------------	---

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

- /home/aroquemaurel/projets/qt/FactDev/src/models/customer.h
- /home/aroquemaurel/projets/qt/FactDev/src/models/customer.cpp

3.8 CustomerContextualMenu Class Reference

Inheritance diagram for CustomerContextualMenu:



Public Member Functions

- **CustomerContextualMenu** (QWidget *w=0)

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

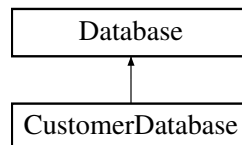
- /home/aroquemaurel/projets/qt/FactDev/src/widgets/customercontextualmenu.h
- /home/aroquemaurel/projets/qt/FactDev/src/widgets/customercontextualmenu.cpp

3.9 CustomerDatabase Class Reference

The **CustomerDatabase** class **Customer** table database.

```
#include <customerdatabase.h>
```

Inheritance diagram for CustomerDatabase:



Public Member Functions

- QStandardItemModel * **getCustomersTable** (QString filter="") throw (DbException*)
CustomerDatabase::getCustomersTable Return an item model of customers for QTableView.
- QStandardItemModel * **getCustomersTree** (QString filter="") throw (DbException*)
CustomerDatabase::getCustomersTree Return an item model of customers for QTree.
- **Customer** * **getCustomer** (const int pld)
CustomerDatabase::getCustomer get informations about the customer identified by pld
- int **addCustomer** (const **Customer** &)
CustomerDatabase::addCustomer Add the customer pCustomer to the database.
- void **updateCustomer** (const **Customer** &)
CustomerDatabase::updateCustomer Update informations about the customer pCustomer
- void **removeCustomer** (const int pld)
CustomerDatabase::removeCustomer Remove the customer with the id pld
- int **getNbCustomers** ()
CustomerDatabase::getNbCustomers Return the number of customers existing.

Static Public Member Functions

- static **CustomerDatabase** * **instance** () throw (DbException*)
CustomerDatabase::getInstance Return an instance of **CustomerDatabase**

Additional Inherited Members

3.9.1 Detailed Description

The [CustomerDatabase](#) class [Customer](#) table database.

Author

Antoine de Roquemaurel

3.9.2 Member Function Documentation

3.9.2.1 `int CustomerDatabase::addCustomer (const Customer & pCustomer)`

[CustomerDatabase::addCustomer](#) Add the customer *pCustomer* to the database.

Returns

customer id

3.9.2.2 `Customer * CustomerDatabase::getCustomer (const int pId)`

[CustomerDatabase::getCustomer](#) get informations about the customer identified by *pId*

Parameters

<i>pId</i>	customer id
------------	-------------

Returns

the [Customer](#)

3.9.2.3 `QStandardItemModel * CustomerDatabase::getCustomersTable (QString filter = " ") throw DbException *`

[CustomerDatabase::getCustomersTable](#) Return an item model of customers for QTableView.

Author

Manantsoa Razanajatovo

Parameters

<i>filter</i>	Select only customers who are specified by <i>filter</i>
---------------	--

Returns

QStandardItemModel an item model for QTableView

3.9.2.4 `QStandardItemModel * CustomerDatabase::getCustomersTree (QString filter = " ") throw DbException *`

[CustomerDatabase::getCustomersTree](#) Return an item model of customers for QTree.

Author

Manantsoa Razanajatovo

Parameters

<i>filter</i>	Select only customers who are specified by <i>filter</i>
---------------	--

Returns

QStandardItemModel an item model for QTableView

3.9.2.5 int CustomerDatabase::getNbCustomers ()

[CustomerDatabase::getNbCustomers](#) Return the number of customers existing.

Returns

number of customers

3.9.2.6 CustomerDatabase * CustomerDatabase::instance () throw DbException *) [static]

[CustomerDatabase::getInstance](#) Return an instance of [CustomerDatabase](#)

Returns

Instance of [CustomerDatabase](#)

3.9.2.7 void CustomerDatabase::removeCustomer (const int *pld*)

[CustomerDatabase::removeCustomer](#) Remove the customer with the id *pld*

Parameters

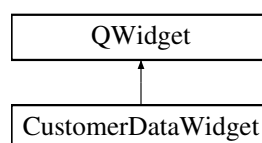
<i>pld</i>	customer id
------------	-------------

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

- /home/aroquemaurel/projets/qt/FactDev/src/database/customerdatabase.h
- /home/aroquemaurel/projets/qt/FactDev/src/database/customerdatabase.cpp

3.10 CustomerDataWidget Class Reference

Inheritance diagram for CustomerDataWidget:



Public Member Functions

- **CustomerDataWidget** (QWidget *parent=0)
- void **printUserData** ()
- void **printInformations** (int id)

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

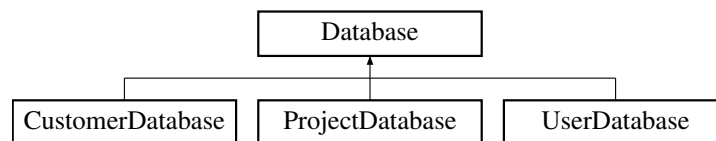
- /home/aroquemaurel/projets/qt/FactDev/src/widgets/customerdatawidget.h
- /home/aroquemaurel/projets/qt/FactDev/src/widgets/customerdatawidget.cpp

3.11 Database Class Reference

The **Database** class Master class for all database access.

```
#include <database.h>
```

Inheritance diagram for Database:



Public Member Functions

- QString **lastError** (const QSqlQuery &q)
Database::lastError Return an error message on the last error occurred during the SQL request q
- void **testCases** ()
Database::testCases Realise a test cases.
- void **cleanDatabase** ()
Database::viderDatabase Clear database.
- void **executeFile** (QString pName)
Database::executerFichier Exeute a specified file named pName
- void **openTransaction** ()
Database::openTransaction Open new transaction.
- void **closeTransaction** ()
Database::closeTransaction Close current transaction.
- void **close** ()
Database::close Close database access.
- void **open** ()
Database::open Open database.
- **~Database** ()
Database::~Database Suppression object, and close database access.
- void **setDatabase** (QSqlDatabase sql)
Database::setDatabase Set database.
- void **createDatabase** ()
Database::creerDatabase Create a new database.

Static Public Member Functions

- static Database * **instance** () throw (DbException*)
Database::getInstance Return an instance of Database.

Protected Member Functions

- [Database](#) () throw (DbException*)
[Database::Database Database](#) is a singleton.
- QVariant [value](#) (const QSqlQuery &q, const QString &champ)
[Database::valeur](#) Value of database field.

Protected Attributes

- QSettings * **_settings**
- QSqlDatabase **mDatabase**
- QList< [Database](#) * > **_instances**

Static Protected Attributes

- static [Database](#) * **_instance** = 0
- static bool **_dbInstance** = 0
- static bool **isOpen** = false

3.11.1 Detailed Description

The [Database](#) class Master class for all database access.

Author

Antoine de Roquemaurel

3.11.2 Member Function Documentation

3.11.2.1 void Database::executeFile (QString *pName*)

[Database::executerFichier](#) Exeute a specified file named *pName*

[Database::executerFichier](#) Exeute a specified file.

Parameters

<i>pNom</i>	File name
-------------	-----------

3.11.2.2 Database * Database::instance () throw DbException * [static]

[Database::getInstance](#) Return an instance of [Database](#).

Returns

Instance of [Database](#)

3.11.2.3 QString Database::lastError (const QSqlQuery &q) [inline]

[Database::lastError](#) Return an error message on the last error occured during the SQL request *q*

[Database::derniereErreur](#) Return text of last error.

Parameters

<i>q</i>	SQL request
----------	-------------

Returns

an error message

Parameters

<i>q</i>	Query
----------	-------

Returns

error message

3.11.2.4 void Database::setDatabase (QSqlDatabase *sql*)

[Database::setDatabase](#) Set database.

Parameters

<i>sql</i>	The new database
------------	------------------

3.11.2.5 void Database::testCases () [inline]

[Database::testCases](#) Realise a test cases.

Database::jeuDEssai Create.

3.11.2.6 QVariant Database::value (const QSqlQuery & *q*, const QString & *champ*) [protected]

Database::valeur Value of database field.

Parameters

<i>q</i>	Query
<i>champ</i>	Field

Returns

The value

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

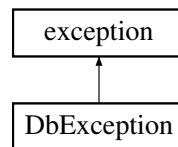
- /home/aroquemaurel/projets/qt/FactDev/src/database/database.h
- /home/aroquemaurel/projets/qt/FactDev/src/database/database.cpp

3.12 DbException Class Reference

The [DbException](#) class for database exception : queries, db file, ...

```
#include <dbexception.h>
```

Inheritance diagram for DbException:



Public Member Functions

- [DbException](#) (const QString fct, const QString fctName, const QString logError, float errorCode)
[DbException::DbException](#). Construct a [DbException](#).
- virtual [~DbException](#) () throw ()
[~DbException](#)
- void [popupMessage](#) (QWidget *parent)
[DbException::popupMessage](#). Display a popup message with the message error.

3.12.1 Detailed Description

The [DbException](#) class for database exception : queries, db file, ...

Author

Antoine de Roquemaurel

3.12.2 Constructor & Destructor Documentation

3.12.2.1 [DbException::DbException](#) (const QString *userError*, const QString *fctName*, const QString *logError*, float *errorCode*)

[DbException::DbException](#). Construct a [DbException](#).

[DbException::DbException](#). Construct an [DbException](#).

Parameters

<i>userError</i>	ClassName of error
<i>fctName</i>	Function name
<i>logError</i>	Message error
<i>errorCode</i>	Code of error

3.12.3 Member Function Documentation

3.12.3.1 void [DbException::popupMessage](#) (QWidget * *parent*)

[DbException::popupMessage](#). Display a popup message with the message error.

Parameters

<i>parent</i>	
---------------	--

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

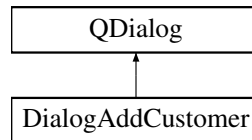
- /home/aroquemaurel/projets/qt/FactDev/src/exceptions/dbexception.h
- /home/aroquemaurel/projets/qt/FactDev/src/exceptions/dbexception.cpp

3.13 DialogAddCustomer Class Reference

The [DialogAddCustomer](#) class Window to add or modify a [Customer](#).

```
#include <dialogaddcustomer.h>
```

Inheritance diagram for DialogAddCustomer:



Public Member Functions

- [DialogAddCustomer](#) (int id=0, QWidget *parent=0)
DialogAddCustomer Construct a window to add/modify a [Customer](#).
- void [fillFields](#) ()
fillFields If the [Customer](#) exists, fill line edits with the data of the current [Customer](#)
- void [accept](#) ()
accept Valid data inputed by user and add these data in [Database](#)
- void [reject](#) ()
reject Cancel the operation and close the windows
- [Customer](#) * [getCustom](#) () const
- void [setCustom](#) ([Customer](#) *getCustom)

3.13.1 Detailed Description

The [DialogAddCustomer](#) class Window to add or modify a [Customer](#).

Author

Cedric Rohaut

3.13.2 Constructor & Destructor Documentation

3.13.2.1 DialogAddCustomer::DialogAddCustomer (int id = 0, QWidget * parent = 0) [explicit]

[DialogAddCustomer](#) Construct a window to add/modify a [Customer](#).

Parameters

<i>id</i>	Customer id
<i>parent</i>	QWidget parent

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

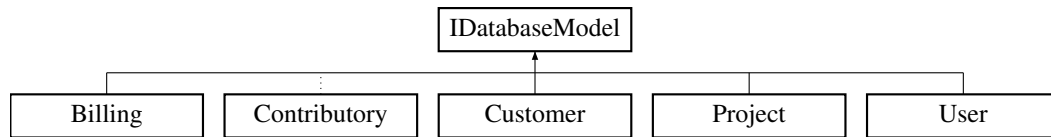
- /home/aroquemaurel/projets/qt/FactDev/src/dialogs/dialogaddcustomer.h
- /home/aroquemaurel/projets/qt/FactDev/src/dialogs/dialogaddcustomer.cpp

3.14 IDatabaseModel Class Reference

The [IDatabaseModel](#) class.

```
#include <idatabasemodel.h>
```

Inheritance diagram for IDatabaseModel:



Public Member Functions

- virtual void **commit** ()=0
- virtual void **hydrat** (int id)=0
- virtual void **remove** ()=0
- int **getId** () const
- void **setId** (int id)

Protected Attributes

- int **_id**

3.14.1 Detailed Description

The [IDatabaseModel](#) class.

Author

Antoine de Roquemaurel

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

- /home/aroquemaurel/projets/qt/FactDev/src/models/idatabasemodel.h

3.15 Log Class Reference

The [Log](#) class for Simple management of log.

```
#include <log.h>
```

Public Member Functions

- [~Log](#) ()
Log::~~Log.
- void [write](#) (const QString text)
Log::write. Write log message in file.
- [Log](#) ()
Log::Log. Log is a singleton.

Static Public Member Functions

- static [Log](#) & [instance](#) (TypeLog type=INFO)
[Log::instance](#). Return the instance of logger.

Friends

- [Log](#) & [operator<<](#) ([Log](#) &logger, const QString &text)
[operator <<](#) for log writing

3.15.1 Detailed Description

The [Log](#) class for Simple management of log.

3.15.2 Member Function Documentation

3.15.2.1 [Log](#) & [Log::instance](#) (TypeLog *type* = INFO) [static]

[Log::instance](#). Return the instance of logger.

Parameters

<i>type</i>	Type of log : WARNING, INFO, ERROR
-------------	------------------------------------

Returns

Instance of logger.

3.15.2.2 void [Log::write](#) (const QString *text*)

[Log::write](#). Write log message in file.

Parameters

<i>text</i>	
-------------	--

3.15.3 Friends And Related Function Documentation

3.15.3.1 [Log](#)& [operator<<](#) ([Log](#) &*logger*, const QString &*text*) [friend]

[operator <<](#) for log writing

Parameters

<i>logger</i>	Instance of Logger
<i>text</i>	Text to write

Returns

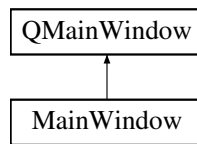
New logger.

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

- /home/aroquemaurel/projets/qt/FactDev/src/log.h
- /home/aroquemaurel/projets/qt/FactDev/src/log.cpp

3.16 MainWindow Class Reference

Inheritance diagram for MainWindow:



Public Slots

- void **addCustomer** ()
- void **editCustomer** ()
- void **removeCustomer** ()
- void **openCustomer** ()
- void **editUser** ()
- void **search** (QString)
- void **search** ()
- void **newProject** (void)
- void **aboutQt** ()
- void **aboutFact** ()
- void **aboutFactDev** ()
- void **aboutIcons** ()
- void **changeCustomer** ()

Public Member Functions

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

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

- /home/aroquemaurel/projets/qt/FactDev/src/mainwindow.h
- /home/aroquemaurel/projets/qt/FactDev/src/mainwindow.cpp

3.17 Parameters Class Reference

Static Public Attributes

- static const QString **DB_FILENAME** = "database.db"
- static const double **VERSION** = 0.1

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

- /home/aroquemaurel/projets/qt/FactDev/src/parameters.h
- /home/aroquemaurel/projets/qt/FactDev/src/parameters.cpp

3.18 Popup Class Reference

Static Public Member Functions

- static void **toImplement** (QString, QWidget *)

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

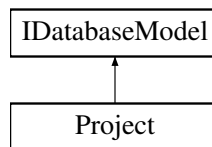
- /home/aroquemaurel/projets/qt/FactDev/src/widgets/popup.h
- /home/aroquemaurel/projets/qt/FactDev/src/widgets/popup.cpp

3.19 Project Class Reference

The [Project](#) class : [Project](#) linked to a [Customer](#).

```
#include <project.h>
```

Inheritance diagram for Project:



Public Member Functions

- [Project](#) ()
Project::Project Construct a [Project](#).
- [Project](#) (int id)
Project::Project Construct a [Project](#) which is specified by an id
- void [commit](#) ()
Project::commit Update project data in the database.
- void [hydrat](#) (int id)
Project::hydrat Insert project data which is specified by id</id> in the database.
- void [remove](#) ()
Project::remove Remove the current project.
- QString [getName](#) () const
Project::getName Return the project name.
- void [setName](#) (const QString &name)
Project::setName Modify the project name
- QString [getDescription](#) () const
Project::getDescription Return a project description.
- void [setDescription](#) (const QString &description)
Project::setDescription Modify the project description
- double [getDailyRate](#) () const
Project::getDailyRate Return the daily rate estimated for this project.
- void [setDailyRate](#) (double dailyRate)
Project::setDailyRate Modify the daily rate dailyRate of the current project.
- [Customer](#) * [getCustomer](#) () const
Project::getCustomer Return the reference to the customer linked to this project.
- void [setCustomer](#) ([Customer](#) *customer)
Project::setCustomer Modify the customer linked to this project.

Additional Inherited Members

3.19.1 Detailed Description

The [Project](#) class : [Project](#) linked to a [Customer](#).

Author

Florent Berbie

3.19.2 Constructor & Destructor Documentation

3.19.2.1 `Project::Project (int id)`

[Project::Project](#) Construct a [Project](#) which is specified by an *id*

Parameters

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

3.19.3 Member Function Documentation

3.19.3.1 `Customer * Project::getCustomer () const`

[Project::getCustomer](#) Return the reference to the customer linked to this project.

Returns

customer linked to this project

3.19.3.2 `double Project::getDailyRate () const`

[Project::getDailyRate](#) Return the daily rate estimated for this project.

Returns

the daily rate linket to the current project

3.19.3.3 `QString Project::getDescription () const`

[Project::getDescription](#) Return a project description.

Returns

project description

3.19.3.4 `QString Project::getName () const`

[Project::getName](#) Return the project name.

Returns

project name

3.19.3.5 void Project::hydrat (int *id*) [virtual]

[Project::hydrat](#) Insert project data which is specified by *id*</id> in the database.

Parameters

<i>id</i>	Project identify
-----------	----------------------------------

Implements [IDatabaseModel](#).

3.19.3.6 void Project::setCustomer (**Customer** * *customer*)

[Project::setCustomer](#) Modify the *customer* linked to this project.

Parameters

<i>customer</i>	New customer associated to this project
-----------------	---

3.19.3.7 void Project::setDailyRate (double *dailyRate*)

[Project::setDailyRate](#) Modify the daily rate *dailyRate* of the current project.

Parameters

<i>dailyRate</i>	New daily rate associated to the current project
------------------	--

3.19.3.8 void Project::setDescription (const QString & *description*)

[Project::setDescription](#) Modify the project *description*

Parameters

<i>description</i>	New project description
--------------------	-------------------------

3.19.3.9 void Project::setName (const QString & *name*)

[Project::setName](#) Modify the project *name*

Parameters

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

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

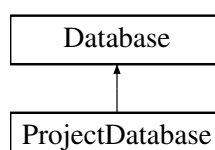
- /home/aroquemaurel/projets/qt/FactDev/src/models/project.h
- /home/aroquemaurel/projets/qt/FactDev/src/models/project.cpp

3.20 ProjectDatabase Class Reference

The [ProjectDatabase](#) class.

```
#include <projectdatabase.h>
```

Inheritance diagram for ProjectDatabase:



Public Member Functions

- [Project](#) * [getProject](#) (const int pld)
ProjectDatabase::getProject Get informations about the project identified by 'pld'.
- int [addProject](#) (const [Project](#) &)
ProjectDatabase::addProject Add the project 'pProject' to the database.
- void [updateProject](#) (const [Project](#) &)
ProjectDatabase::updateProject Update informations about the project.
- void [removeProject](#) (const int pld)
removeProject Remove the project with the id 'pld'
- int [getNbProjects](#) ()
ProjectDatabase::getNbProjects Return the number of projects existing.
- int [getNbProjectsForACustomer](#) (const int pld)
ProjectDatabase::getNbProjectsForACustomer Return the number of projects existing for an identify customer pld

Static Public Member Functions

- static [ProjectDatabase](#) * [instance](#) () throw (DbException*)
ProjectDatabase::getInstance Return an instance of [ProjectDatabase](#).

Additional Inherited Members

3.20.1 Detailed Description

The [ProjectDatabase](#) class.

Author

Florent Berbie

3.20.2 Member Function Documentation

3.20.2.1 int ProjectDatabase::addProject (const [Project](#) & pProject)

[ProjectDatabase](#)::addProject Add the project 'pProject' to the database.

Returns

project id

3.20.2.2 int ProjectDatabase::getNbProjects ()

[ProjectDatabase](#)::getNbProjects Return the number of projects existing.

Returns

number of projects

3.20.2.3 int ProjectDatabase::getNbProjectsForACustomer (const int pld)

[ProjectDatabase](#)::getNbProjectsForACustomer Return the number of projects existing for an identify customer pld

Parameters

<i>pld</i>	Project id
------------	----------------------------

Returns

number of projects

3.20.2.4 **Project * ProjectDatabase::getProject (const int *pld*)**

[ProjectDatabase::getProject](#) Get informations about the project identified by 'pld'.

Parameters

<i>pld</i>	project
------------	---------

Returns

the project

3.20.2.5 **ProjectDatabase * ProjectDatabase::instance () throw DbException *** [static]

[ProjectDatabase::getInstance](#) Return an instance of [ProjectDatabase](#).

Returns

Instance of [ProjectDatabase](#)

3.20.2.6 **void ProjectDatabase::removeProject (const int *pld*)**

[removeProject](#) Remove the project with the id 'pld'

Parameters

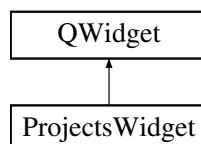
<i>pld</i>	project id
------------	------------

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

- `/home/aroquemaurel/projets/qt/FactDev/src/database/projectdatabase.h`
- `/home/aroquemaurel/projets/qt/FactDev/src/database/projectdatabase.cpp`

3.21 ProjectsWidget Class Reference

Inheritance diagram for ProjectsWidget:



Public Slots

- void **newProject** ()
- void **editSelectedProject** ()
- void **removeSelectedProject** ()

Public Member Functions

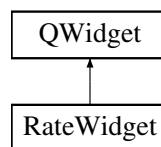
- **ProjectsWidget** (QWidget *parent=0)

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

- /home/aroquemaurel/projets/qt/FactDev/src/widgets/projectswidget.h
- /home/aroquemaurel/projets/qt/FactDev/src/widgets/projectswidget.cpp

3.22 RateWidget Class Reference

Inheritance diagram for RateWidget:



Public Member Functions

- **RateWidget** (QWidget *parent=0)
- void **initRate** ()
- double **getDailyRate** ()
- void **setDailyRate** (double dailyRate)
- double **getHourlyRate** ()
- void **setHourlyRate** (double hourlyRate)

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

- /home/aroquemaurel/projets/qt/FactDev/src/widgets/ratewidget.h
- /home/aroquemaurel/projets/qt/FactDev/src/widgets/ratewidget.cpp

3.23 Search Class Reference

The [Search](#) class.

```
#include <search.h>
```

Public Member Functions

- QString **getFilter** ()
- bool **getSearchInCompanies** () const
- void **setSearchInCompanies** (bool getSearchInCompanies)
- bool **getSearchInReferentLastname** () const
- void **setSearchInReferentLastname** (bool getSearchInReferentLastname)
- bool **getGroupFilter** () const
- void **setGroupFilter** (bool getGroupFilter)
- QString **getText** () const
- void **setText** (const QString &getText)

3.23.1 Detailed Description

The [Search](#) class.

Author

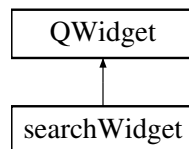
Antoine de Roquemaurel

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

- /home/aroquemaurel/projets/qt/FactDev/src/models/search.h
- /home/aroquemaurel/projets/qt/FactDev/src/models/search.cpp

3.24 searchWidget Class Reference

Inheritance diagram for searchWidget:



Public Slots

- void **search** (QString)

Public Member Functions

- **searchWidget** (QWidget *parent=0)
- int [getCurrentCustomerId](#) ()

getCurrentCustomerId Return the id of the customer selected in the table

3.24.1 Member Function Documentation

3.24.1.1 int searchWidget::getCurrentCustomerId ()

getCurrentCustomerId Return the id of the customer selected in the table

Returns

id of the current customer

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

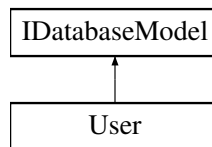
- /home/aroquemaurel/projets/qt/FactDev/src/widgets/searchwidget.h
- /home/aroquemaurel/projets/qt/FactDev/src/widgets/searchwidget.cpp

3.25 User Class Reference

The `User` class `User` of it application.

```
#include <user.h>
```

Inheritance diagram for `User`:



Public Member Functions

- `User ()`
`User::User`. Construct an `User`.
- `User (int id)`
`User::User`. Construct a `User` with the identify `id`
- `void commit ()`
`User::commit` Update user data in `User` table on the database.
- `void hydrat (int id=1)`
`User::hydrat` Get data of the user who is specified by `id` from the database.
- `void remove ()`
`remove` Remove the current `User`
- `QString getFirstname () const`
`User::getFirstname` Return the user firstname.
- `void setFirstname (const QString &firstname)`
`User::setFirstnamet` Modify the user firstname
- `QString getLastName () const`
`User::getLastName` Return the user lastname.
- `void setLastName (const QString &lastname)`
`User::setLastName` Modify the user lastname
- `QString getCompany () const`
`User::getCompany` Return the user company.
- `void setCompany (const QString &company)`
`User::setCompany` Modify the user company name.
- `QString getTitle () const`
`User::getTitle` Return a short description of `User` (company) activity.
- `void setTitle (const QString &title)`
`User::setTitle` Modify the user/company activities description
- `QString getAddress () const`
`User::getAddress` Return the company address (Number and name of street)
- `void setAddress (const QString &address)`
`User::setAddress` Modify the user company address
- `QString getPostalCode () const`
`User::getPostalCode` Return the postal code.
- `void setPostalCode (const QString &postalCode)`
`User::setPostalCode` Modify the postal code `postalCode`
- `QString getCity () const`
`User::getCity` Return the city.

- void [setCity](#) (const QString &city)
[User::setCity](#) Modify the city
- QString [getEmail](#) () const
[User::getEmail](#) Return the user professional email
- void [setEmail](#) (const QString &email)
[User::setEmail](#) Modify the user professional email
- QString [getMobilePhone](#) () const
[User::getMobilePhone](#) Return the number of the professional mobile phone.
- void [setMobilePhone](#) (const QString &mobilePhone)
[User::setMobilePhone](#) Modify the number of the professional user mobile phone mobilePhone
- QString [getPhone](#) () const
[User::getPhone](#) Return the number of the desktop phone.
- void [setPhone](#) (const QString &phone)
[User::setPhone](#) Modify the number of the desktop phone
- int [getNoSiret](#) () const
[User::getNoSiret](#) Return the SIRET number (company registration number)
- void [setNoSiret](#) (int noSiret)
[User::setNoSiret](#) Modify the SIRET number (company registration number) noSiret

Additional Inherited Members

3.25.1 Detailed Description

The [User](#) class [User](#) of it application.

Author

Florent Berbie

3.25.2 Constructor & Destructor Documentation

3.25.2.1 [User::User](#) (int *id*)

[User::User](#). Construct a [User](#) with the identify *id*

Parameters

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

3.25.3 Member Function Documentation

3.25.3.1 QString [User::getAddress](#) () const

[User::getAddress](#) Return the company address (Number and name of street)

Returns

Address company

3.25.3.2 QString User::getCity () const

[User::getCity](#) Return the city.

Returns

city

3.25.3.3 QString User::getCompany () const

[User::getCompany](#) Return the user company.

Returns

New company name

3.25.3.4 QString User::getEmail () const

[User::getEmail](#) Return the user professional *email*

Returns

professional email

3.25.3.5 QString User::getFirstname () const

[User::getFirstname](#) Return the user firstname.

Returns

[User](#) firstname

3.25.3.6 QString User::getLastname () const

[User::getLastname](#) Return the user lastname.

Returns

[User](#) lastname

3.25.3.7 QString User::getMobilePhone () const

[User::getMobilePhone](#) Return the number of the professional mobile phone.

Returns

number of mobile phone

3.25.3.8 int User::getNoSiret () const

[User::getNoSiret](#) Return the SIRET number (company registration number)

Returns

SIRET number

3.25.3.9 QString User::getPhone () const

[User::getPhone](#) Return the number of the desktop phone.

Returns

number of the desktop phone

3.25.3.10 QString User::getPostalCode () const

[User::getPostalCode](#) Return the postal code.

Returns

postal code

3.25.3.11 QString User::getTitle () const

[User::getTitle](#) Return a short description of [User](#) (company) activity.

Returns

a short description of user (company) activity

3.25.3.12 void User::hydrat (int *id* = 1) [virtual]

[User::hydrat](#) Get data of the user who is specified by *id* from the database.

Parameters

<i>id</i>	User identify
-----------	-------------------------------

Implements [IDatabaseModel](#).

3.25.3.13 void User::setAddress (const QString & *address*)

[User::setAddress](#) Modify the user company *address*

Parameters

<i>address</i>	Company address (name and number of street)
----------------	---

3.25.3.14 void User::setCity (const QString & *city*)

[User::setCity](#) Modify the *city*

Parameters

<i>city</i>	Company city address
-------------	----------------------

3.25.3.15 void User::setCompany (const QString & *company*)

[User::setCompany](#) Modify the user *company* name.

Parameters

<i>company</i>	New user company name
----------------	-----------------------

3.25.3.16 void User::setEmail (const QString & *email*)

[User::setEmail](#) Modify the user professional *email*

Parameters

<i>email</i>	The user professional email
--------------	-----------------------------

3.25.3.17 void User::setFirstname (const QString & *firstname*)

[User::setFirstname](#) Modify the user *firstname*

Parameters

<i>firstname</i>	New user firstname
------------------	--------------------

3.25.3.18 void User::setLastname (const QString & *lastname*)

[User::setLastname](#) Modify the user *lastname*

Parameters

<i>lastname</i>	New user lastname
-----------------	-------------------

3.25.3.19 void User::setMobilePhone (const QString & *mobilePhone*)

[User::setMobilePhone](#) Modify the number of the professional user mobile phone *mobilePhone*

Parameters

<i>mobilePhone</i>	Number of the professional mobile phone
--------------------	---

3.25.3.20 void User::setNoSiret (int *noSiret*)

[User::setNoSiret](#) Modify the SIRET number (company registration number) *noSiret*

Parameters

<i>noSiret</i>	SIRET number
----------------	--------------

3.25.3.21 void User::setPhone (const QString & *phone*)

[User::setPhone](#) Modify the number of the desktop *phone*

Parameters

<i>phone</i>	Number of the desktop phone
--------------	-----------------------------

3.25.3.22 void User::setPostalCode (const QString & *postalCode*)

[User::setPostalCode](#) Modify the postal code *postalCode*

Parameters

<i>postalCode</i>	New postal code
-------------------	-----------------

3.25.3.23 void User::setTitle (const QString & title)

[User::setTitle](#) Modify the user/company activities *description*

Parameters

<i>title</i>	Short description on activity(ies) of customer company
--------------	--

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

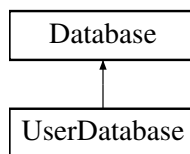
- /home/aroquemaurel/projets/qt/FactDev/src/models/user.h
- /home/aroquemaurel/projets/qt/FactDev/src/models/user.cpp

3.26 UserDatabase Class Reference

The [UserDatabase](#) class Access to [User](#) data in the the table [User](#) of the [Database](#)

```
#include <userdatabase.h>
```

Inheritance diagram for UserDatabase:



Public Member Functions

- QStandardItemModel * [getUserTable](#) () throw (DbException*)
getUserTable Return an item model of [User](#) for QTableView
- [User](#) * [getUser](#) (const int pld=1)
getUser Get informations about the user (identified by 'pld')
- void [updateUser](#) (const [User](#) &)
updateUser Update informations about the user

Static Public Member Functions

- static [UserDatabase](#) * [instance](#) () throw (DbException*)
UserDatabase::getInstance Return an instance of [UserDatabase](#).

Additional Inherited Members

3.26.1 Detailed Description

The [UserDatabase](#) class Access to [User](#) data in the the table [User](#) of the [Database](#)

Author

Florent Berbie

3.26.2 Member Function Documentation

3.26.2.1 `User * UserDatabase::getUser (const int pId = 1)`

`getUser` Get informations about the user (identified by '*pId*')
Parameters

<i>pId</i>	user id (1 default) *
------------	-----------------------

Returns

the user

3.26.2.2 `QStandardItemModel* UserDatabase::getUserTable () throw DbException *`

`getUserTable` Return an item model of [User](#) for QTableView
Returns

QStandardItemModel an item model

3.26.2.3 `UserDatabase * UserDatabase::instance () throw DbException *` [static]

`UserDatabase::getInstance` Return an instance of [UserDatabase](#).
Returns

Instance of [UserDatabase](#)

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

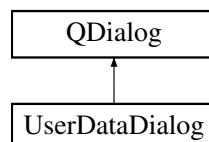
- `/home/aroquemaurel/projets/qt/FactDev/src/database/userdatabase.h`
- `/home/aroquemaurel/projets/qt/FactDev/src/database/userdatabase.cpp`

3.27 UserDataDialog Class Reference

The [UserDataDialog](#) class Window to fill user data.

```
#include <userdatadialog.h>
```

Inheritance diagram for `UserDataDialog`:



Public Member Functions

- [UserDataDialog](#) (QWidget *parent=0)
UserDataDialog: Construct a window with user data.
- void [fillFields](#) ()

- `void accept ()`
accept Valid data inputed by user and add these data in [Database](#)
- `void reject ()`
reject Cancel the operation and close the windows

3.27.1 Detailed Description

The [UserDataDialog](#) class Window to fill user data.

Author

Florent Berbie

3.27.2 Constructor & Destructor Documentation

3.27.2.1 [UserDataDialog::UserDataDialog](#) ([QWidget](#) * *parent* = 0) [explicit]

[UserDataDialog](#): Construct a window with user data.

Parameters

<i>parent</i>	
---------------	--

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

- `/home/aroquemaurel/projets/qt/FactDev/src/dialogs/userdatadialog.h`
- `/home/aroquemaurel/projets/qt/FactDev/src/dialogs/userdatadialog.cpp`

3.28 Utils Class Reference

Static Public Member Functions

- static [QString](#) **firstLetterToUpper** ([QString](#) s)

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

- `/home/aroquemaurel/projets/qt/FactDev/src/Utils.h`
- `/home/aroquemaurel/projets/qt/FactDev/src/Utils.cpp`

Index

- AddContributoryDialog, [5](#)
 - AddContributoryDialog, [5](#)
 - AddContributoryDialog, [5](#)
- addCustomer
 - CustomerDatabase, [18](#)
- addProject
 - ProjectDatabase, [32](#)
- AddProjectDialog, [6](#)
 - AddProjectDialog, [6](#)
 - AddProjectDialog, [6](#)
- Billing, [6](#)
 - getContributories, [7](#)
 - getDate, [7](#)
 - getNumber, [8](#)
 - getTitle, [8](#)
 - hydrat, [8](#)
 - isBilling, [8](#)
 - setDate, [8](#)
 - setIsBilling, [8](#)
 - setNumber, [9](#)
 - setTitle, [9](#)
- ComboBoxModelWidget, [9](#)
- connexiondialog, [9](#)
- Contributory, [10](#)
- Customer, [10](#)
 - Customer, [12](#)
 - getAddress, [12](#)
 - getCity, [12](#)
 - getCompany, [13](#)
 - getCountry, [13](#)
 - getEmail, [13](#)
 - getFax, [13](#)
 - getFirstnameReferent, [13](#)
 - getLastnameReferent, [13](#)
 - getMobilePhone, [13](#)
 - getPhone, [14](#)
 - getPostalCode, [14](#)
 - hydrat, [14](#)
 - operator==, [14](#)
 - setAddress, [15](#)
 - setCity, [15](#)
 - setCompany, [15](#)
 - setCountry, [15](#)
 - setEmail, [15](#)
 - setFax, [15](#)
 - setFirstnameReferent, [16](#)
 - setLastnameReferent, [16](#)
 - setMobilePhone, [16](#)
 - setPhone, [16](#)
 - setPostalCode, [16](#)
- CustomerContextualMenu, [16](#)
- CustomerDataWidget, [19](#)
- CustomerDatabase, [17](#)
 - addCustomer, [18](#)
 - getCustomer, [18](#)
 - getCustomersTable, [18](#)
 - getCustomersTree, [18](#)
 - getNbCustomers, [19](#)
 - instance, [19](#)
 - removeCustomer, [19](#)
- Database, [20](#)
 - executeFile, [21](#)
 - instance, [21](#)
 - lastError, [21](#)
 - setDatabase, [22](#)
 - testCases, [22](#)
 - value, [22](#)
- DbException, [22](#)
 - DbException, [23](#)
 - DbException, [23](#)
 - popupMessage, [23](#)
- DialogAddCustomer, [24](#)
 - DialogAddCustomer, [24](#)
 - DialogAddCustomer, [24](#)
- executeFile
 - Database, [21](#)
- getAddress
 - Customer, [12](#)
 - User, [37](#)
- getCity
 - Customer, [12](#)
 - User, [37](#)
- getCompany
 - Customer, [13](#)
 - User, [38](#)
- getContributories
 - Billing, [7](#)
- getCountry
 - Customer, [13](#)
- getCurrentCustomerId
 - searchWidget, [35](#)
- getCustomer
 - CustomerDatabase, [18](#)
 - Project, [29](#)
- getCustomersTable

- CustomerDatabase, 18
- getCustomersTree
 - CustomerDatabase, 18
- getDailyRate
 - Project, 29
- getDate
 - Billing, 7
- getDescription
 - Project, 29
- getEmail
 - Customer, 13
 - User, 38
- getFax
 - Customer, 13
- getFirstname
 - User, 38
- getFirstnameReferent
 - Customer, 13
- getLastName
 - User, 38
- getLastNameReferent
 - Customer, 13
- getMobilePhone
 - Customer, 13
 - User, 38
- getName
 - Project, 29
- getNbCustomers
 - CustomerDatabase, 19
- getNbProjects
 - ProjectDatabase, 32
- getNbProjectsForACustomer
 - ProjectDatabase, 32
- getNoSiret
 - User, 38
- getNumber
 - Billing, 8
- getPhone
 - Customer, 14
 - User, 38
- getPostalCode
 - Customer, 14
 - User, 39
- getProject
 - ProjectDatabase, 33
- getTitle
 - Billing, 8
 - User, 39
- getUser
 - UserDatabase, 43
- getUserTable
 - UserDatabase, 43
- hydrat
 - Billing, 8
 - Customer, 14
 - Project, 29
 - User, 39
- IDatabaseModel, 25
 - instance
 - CustomerDatabase, 19
 - Database, 21
 - Log, 26
 - ProjectDatabase, 33
 - UserDatabase, 43
- isBilling
 - Billing, 8
- lastError
 - Database, 21
- Log, 25
 - instance, 26
 - operator<<, 26
 - write, 26
- MainWindow, 27
- operator<<
 - Log, 26
- operator==
 - Customer, 14
- Parameters, 27
- Popup, 28
- popupMessage
 - DbException, 23
- Project, 28
 - getCustomer, 29
 - getDailyRate, 29
 - getDescription, 29
 - getName, 29
 - hydrat, 29
 - Project, 29
 - setCustomer, 31
 - setDailyRate, 31
 - setDescription, 31
 - setName, 31
- ProjectDatabase, 31
 - addProject, 32
 - getNbProjects, 32
 - getNbProjectsForACustomer, 32
 - getProject, 33
 - instance, 33
 - removeProject, 33
- ProjectsWidget, 33
- RateWidget, 34
- removeCustomer
 - CustomerDatabase, 19
- removeProject
 - ProjectDatabase, 33
- Search, 34
- searchWidget, 35
 - getCurrentCustomerId, 35
- setAddress
 - Customer, 15
 - User, 39

- setCity
 - Customer, [15](#)
 - User, [39](#)
- setCompany
 - Customer, [15](#)
 - User, [39](#)
- setCountry
 - Customer, [15](#)
- setCustomer
 - Project, [31](#)
- setDailyRate
 - Project, [31](#)
- setDatabase
 - Database, [22](#)
- setDate
 - Billing, [8](#)
- setDescription
 - Project, [31](#)
- setEmail
 - Customer, [15](#)
 - User, [40](#)
- setFax
 - Customer, [15](#)
- setFirstname
 - User, [40](#)
- setFirstnameReferent
 - Customer, [16](#)
- setIsBilling
 - Billing, [8](#)
- setLastname
 - User, [40](#)
- setLastnameReferent
 - Customer, [16](#)
- setMobilePhone
 - Customer, [16](#)
 - User, [40](#)
- setName
 - Project, [31](#)
- setNoSiret
 - User, [40](#)
- setNumber
 - Billing, [9](#)
- setPhone
 - Customer, [16](#)
 - User, [40](#)
- setPostalCode
 - Customer, [16](#)
 - User, [40](#)
- setTitle
 - Billing, [9](#)
 - User, [42](#)
- testCases
 - Database, [22](#)
- User, [36](#)
 - getAddress, [37](#)
 - getCity, [37](#)
 - getCompany, [38](#)
 - getEmail, [38](#)
 - getFirstname, [38](#)
 - getLastname, [38](#)
 - getMobilePhone, [38](#)
 - getNoSiret, [38](#)
 - getPhone, [38](#)
 - getPostalCode, [39](#)
 - getTitle, [39](#)
 - hydrat, [39](#)
 - setAddress, [39](#)
 - setCity, [39](#)
 - setCompany, [39](#)
 - setEmail, [40](#)
 - setFirstname, [40](#)
 - setLastname, [40](#)
 - setMobilePhone, [40](#)
 - setNoSiret, [40](#)
 - setPhone, [40](#)
 - setPostalCode, [40](#)
 - setTitle, [42](#)
 - User, [37](#)
- UserDataDialog, [43](#)
 - UserDataDialog, [44](#)
 - UserDataDialog, [44](#)
- UserDatabase, [42](#)
 - getUser, [43](#)
 - getUserTable, [43](#)
 - instance, [43](#)
- Utils, [44](#)
- value
 - Database, [22](#)
- write
 - Log, [26](#)