FactDev

0.1

Generated by Doxygen 1.8.8

Tue Jan 20 2015 13:50:52

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

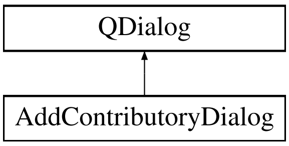
Class Documentation

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

See also

Contributory

3.1.2 Constructor & Destructor Documentation

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

AddContributoryDialog Construct a new windows AddContributory.

Parameters

parent	QWidget

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

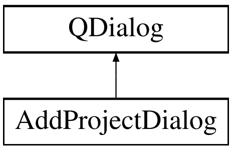
- src/dialogs/addcontributorydialog.h
- src/dialogs/addcontributorydialog.cpp

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

See also

Project

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

id	Project identity
parent	QWidget of the current windows

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

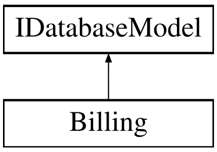
- src/dialogs/addprojectdialog.h
- src/dialogs/addprojectdialog.cpp

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

getId	

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

getDate	the new date of the Billing

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

Billing::setIsBilling. Modify *isBilling* of **Billing**.

Parameters

isBilling	

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

Billing::setNumber. Modify *_number* of Billing.

Parameters

getNumber	the new number of the Billing

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

Billing::setTitle. Modify the title of **Billing**

Parameters

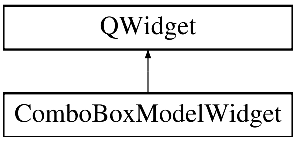
getTitle	Modify the title with *getTitle*

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

- src/models/billing.h
- src/models/billing.cpp

3.4 ComboBoxModelWidget Class Reference

Inheritance diagram for ComboBoxModelWidget:



Public Member Functions

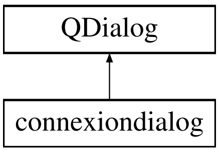
- **ComboBoxModelWidget** (QWidget ∗parent=0)

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

- src/widgets/comboboxmodelwidget.h
- 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:

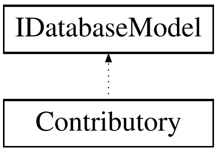
- src/connexiondialog.h
- src/connexiondialog.cpp

3.6 Contributory Class Reference

The Contributory class.

`#include <contributory.h>`

Inheritance diagram for Contributory:



Public Member Functions

- Contributory ()

    Contributory::Contributory Contruct a Contributory.
- void commit ()

    Contributory::commit Update or insert a contributory to the database.
- void hydrat (int id)

    Contributory::hydrat Get data about the Contributory which is specified by the identify id
- void remove ()

Contributory::remove Remove the current Contributory.
- Project getProject () const

  Contributory::getProject Return the project linked to this Contributory.
- void setProject (const Project &id)

  Contributory::setProject Modify the identify id of the Project linke to this Contributory.

Additional Inherited Members

3.6.1 Detailed Description

The Contributory class.

Author

3.6.2 Member Function Documentation

3.6.2.1 Project Contributory::getProject ( ) const

Contributory::getProject Return the project linked to this Contributory.

Returns

Project linked to this Contributory

3.6.2.2 void Contributory::hydrat (int *id*) [virtual]

Contributory::hydrat Get data about the Contributory which is specified by the identify *id*

Parameters

id	Contributory identify

Implements IDatabaseModel.

3.6.2.3 void Contributory::setProject (const Project & *id*)

Contributory::setProject Modify the identify *id* of the Project linke to this Contributory.

Parameters

id	Project Identify

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

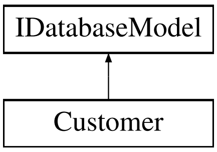
- src/models/contributory.h
- src/models/contributory.cpp

3.7 Customer Class Reference

The Customer class Customer.

`#include <customer.h>`

Inheritance diagram for Customer:



Public Member Functions

- Customer ()

  Customer::Customer Construct a Customer.
- Customer (int id)

  Customer::Customer Constuct a Customer who is specidied 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 differnt 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 Customer.

Author

Antoine de Roquemaurel
Florent Berbie

3.7.2 Constructor & Destructor Documentation

3.7.2.1 Customer::Customer (int *id*)

Customer::Customer Constuct a Customer who is specidied by *id*

Parameters

id	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

id	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

c	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

c	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

address	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

city	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

company	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

country	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

email	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

fax	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

firstname↩ Referent	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

lastname↩ Referent	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

mobilePhone	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

phone	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

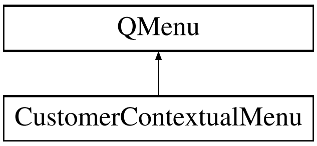
postalCode	New postal code of the customer company

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

- src/models/customer.h
- 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:

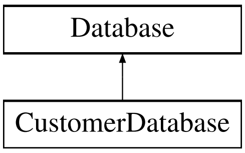
- src/widgets/customercontextualmenu.h
- src/widgets/customercontextualmenu.cpp

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 pId)

  CustomerDatabase::getCustomer get informations about the customer identified by pId
- 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 pId)

  CustomerDatabase::removeCustomer Remove the customer with the id pId
- 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

See also

Database
Customer

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

pId	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

filter	Select only customers who are specified by *filter*

Exceptions

DbException	

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

filter	Select only customers who are specified by *filter*

Exceptions

DbException	

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**

See also

DbException

Returns

Instance of CustomerDatabase

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

CustomerDatabase::removeCustomer Remove the customer with the id *pId*

Parameters

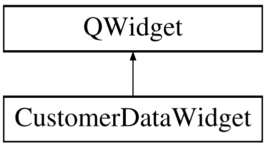
pId	customer id

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

- src/database/customerdatabase.h
- 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:

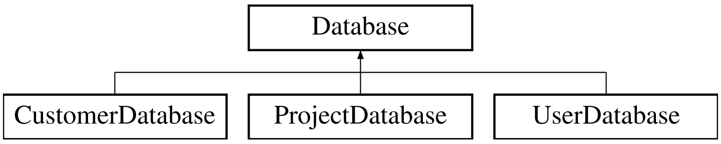
- src/widgets/customerdatawidget.h
- src/widgets/customerdatawidget.cpp

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 occured 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*

Parameters

pNom	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*

Parameters

q	SQL request

Returns

an error message

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

Database::setDatabase Set database.

Parameters

sql	The new database

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

Database::valeur Value of database field.

Parameters

q	Query
champ	Field

Returns

The value

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

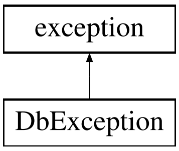
- src/database/database.h
- src/database/database.cpp

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 *fct,* const QString *fctName,* const QString *logError,* float *errorCode*)

DbException::DbException. Construct a DbException.

Parameters

userError	ClassName of error
fctName	Function name
logError	Message error
errorCode	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

parent	

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

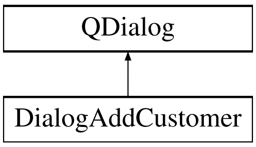
- src/exceptions/dbexception.h
- src/exceptions/dbexception.cpp

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 exits, 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

Cédric 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

id	Customer id
parent	QWidget parent

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

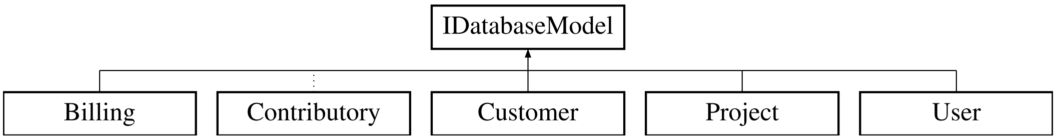
- src/dialogs/dialogaddcustomer.h
- src/dialogs/dialogaddcustomer.cpp

3.14 IDatabaseModel Class Reference

The IDatabaseModel class.

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

Inheritance diagram for IDatabaseModel:



Public Member Functions

- virtual void commit ()=0

  IDatabaseModel::commit Update or insert data into the database.
- virtual void hydrat (int id)=0

  IDatabaseModel::hydrat Get data of the element which is specified by the identify id from the database.
- virtual void remove ()=0

  IDatabaseModel::remove Remove the current element in the database.
- int getId () const

  IDatabaseModel::getId Return the identify of the element of the database.
- void setId (int id)

  IDatabaseModel::setId Replace the current identify by id

Protected Attributes

- int _id

  Element identify.

3.14.1 Detailed Description

The IDatabaseModel class.

Author

Antoine de Roquemaurel

3.14.2 Member Function Documentation

3.14.2.1 int IDatabaseModel::getId ( ) const `[inline]`

IDatabaseModel::getId Return the identify of the element of the database.

Returns

identity

3.14.2.2 virtual void IDatabaseModel::hydrat (int *id*) `[pure virtual]`

IDatabaseModel::hydrat Get data of the element which is specified by the identify *id* from the database.

Parameters

id	

Implemented in Project, Billing, Customer, User, and Contributory.

3.14.2.3 void IDatabaseModel::setId (int *id*) `[inline]`

IDatabaseModel::setId Replace the current identify by *id*

Parameters

id	New identify

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

- 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

type	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

text	

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

logger	Instance of Logger
text	Text to write

Returns

New logger.

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

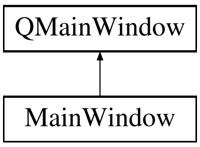
- src/log.h
- src/log.cpp

3.16 MainWindow Class Reference

The MainWindow class Main Window of the software.

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

Inheritance diagram for MainWindow:



Public Slots

- void addCustomer ()

  MainWindow::addCustomer open window to add a new customer.
- void editCustomer ()

  MainWindow::editCustomer open window to modify a customer.

- void removeCustomer ()

  MainWindow::removeCustomer open a popup to confirm the deletion of a customer, if ok remove the customer.

- void openCustomer ()

  MainWindow::openCustomer open a customer and print his informations.

- void editUser ()

  MainWindow::editUser modify the user.

- void search (QString)

  MainWindow::search launch a new search.

- void search ()

  MainWindow:search call search(QString)

- void newProject (void)

  MainWindow::newProject Create a new project for a customer.

- void aboutQt ()

  MainWindow::aboutQt show Qt's details.

- void aboutFact ()

  MainWindow::aboutFact show FACT's details (FACT team)

- void aboutFactDev ()

  MainWindow::aboutFactDev() show FactDev's details (FactDev Software)

- void aboutIcons ()

  MainWindow::aboutIcons() show icons's details.

- void changeCustomer ()

  MainWindow::changeCustomer This method is not used, implements me !

Public Member Functions

- MainWindow (QWidget ∗parent=0)

  MainWindow: Construct a window.

- int getCurrentCustomerId ()

  MainWindow::getCurrentCustomerId get the selected customer.

3.16.1 Detailed Description

The MainWindow class Main Window of the software.

Author

3.16.2 Constructor & Destructor Documentation

3.16.2.1 MainWindow::MainWindow (QWidget ∗ *parent* = `0`) `[explicit]`

MainWindow: Construct a window.

Parameters

parent	

3.16.3 Member Function Documentation

3.16.3.1 void MainWindow::editUser ( ) [slot]

MainWindow::editUser modify the user.

See also

UserDataDialog

3.16.3.2 int MainWindow::getCurrentCustomerId ( )

MainWindow::getCurrentCustomerId get the selected customer.

Returns

id of the selected customer

3.16.3.3 void MainWindow::newProject (void) [slot]

MainWindow::newProject Create a new project for a customer.

See also

AddProjectDialog

3.16.3.4 void MainWindow::openCustomer ( ) [slot]

MainWindow::openCustomer open a customer and print his informations.

See also

CustomerDataWidget

3.16.3.5 void MainWindow::search (QString *text*) [slot]

MainWindow::search launch a new search.

Parameters

QString	text in field

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

- src/mainwindow.h
- src/mainwindow.cpp

3.17 Parameters Class Reference

Static Public Attributes

- static const QString DB_FILENAME = "database.db"

    Database file name.

- static const double VERSION = 0.1

  Application version.

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

- src/parameters.h
- 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:

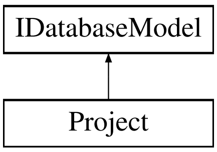
- src/widgets/popup.h
- src/widgets/popup.cpp

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

See also

IDatabaseModel

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

id	

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

id	Project identify

Implements IDatabaseModel.

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

Project::setCustomer Modify the *customer* linked to this project.

Parameters

customer	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

dailyRate	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

description	New project description

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

Project::setName Modify the project *name*

Parameters

name	Project name

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

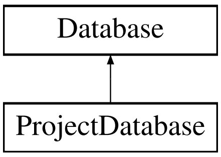
- src/models/project.h
- src/models/project.cpp

3.20 ProjectDatabase Class Reference

The ProjectDatabase class Project table database.

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

Inheritance diagram for ProjectDatabase:



Public Member Functions

- Project ∗ getProject (const int pId)

  ProjectDatabasegetProject Get informations about the project identified by 'pId'.
- 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 pId)

  removeProject Remove the project with the id 'pId'
- int getNbProjects ()

  ProjectDatabase:getNbProjects Return the number of projects existing.
- int getNbProjectsForACustomer (const int pId)

  ProjectDatabase:getNbProjectsForACustomer Return the number of projects existing for an identify customer pId

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 Project table database.

Author

Florent Berbie

See also

Database
Project

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 *pId*)

ProjectDatabase:getNbProjectsForACustomer Return the number of projects existing for an identify customer *pId*

Parameters

pId	Project id

Returns

number of projects

3.20.2.4 **Project** ∗ ProjectDatabase::getProject (const int *pId*)

ProjectDatabasegetProject Get informations about the project identified by 'pId'.

Parameters

pId	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 *pId*)

removeProject Remove the project with the id 'pId'

Parameters

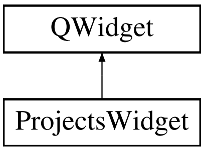
pId	project id

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

- src/database/projectdatabase.h
- src/database/projectdatabase.cpp

3.21 ProjectsWidget Class Reference

Inheritance diagram for ProjectsWidget:



Public Slots

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

Public Member Functions

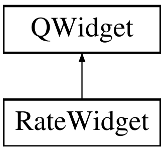
- **ProjectsWidget** (QWidget ∗parent=0)

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

- src/widgets/projectswidget.h
- 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:

- src/widgets/ratewidget.h
- src/widgets/ratewidget.cpp

3.23 Search Class Reference

The Search class.

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

Public Member Functions

- Search ()

  Search::Search Construct a search.
- ∼Search ()

  Search::Search Destruct a search.
- QString getFilter ()

  Search::getFilter Return the search filter.
- bool getSearchInCompanies () const

  Search::getSearchInCompanies Return if we search a company.
- void setSearchInCompanies (bool getSearchInCompanies)

  Search::setSearchInCompanies Modify the filter of search.
- bool getSearchInReferentLastname () const

  Search::getSearchInReferentLastname Return if we search a Last name referent.
- void setSearchInReferentLastname (bool getSearchInReferentLastname)

  Search::setSearchInReferentLastname Modify the filter of search.
- bool getGroupFilter () const

  Search::getGroupFilter Return if the filter is actived.
- void setGroupFilter (bool getGroupFilter)

  Search::setGroupFilter Modify if we active search filter.
- QString getText () const

  Search::getText Return sql portion of filter.
- void setText (const QString &getText)

  Search::setText Modify sql portion.

3.23.1 Detailed Description

The Search class.

Author

Antoine de Roquemaurel

3.23.2 Member Function Documentation

3.23.2.1 QString Search::getFilter ( )

Search::getFilter Return the search filter.

Returns

filter selected (sql portion)

3.23.2.2 bool Search::getGroupFilter ( ) const

Search::getGroupFilter Return if the filter is actived.

Returns

boolean if search filter is actived

3.23.2.3 bool Search::getSearchInCompanies ( ) const

Search::getSearchInCompanies Return if we search a company.

Returns

boolean if we search a company

3.23.2.4 bool Search::getSearchInReferentLastname ( ) const

Search::getSearchInReferentLastname Return if we search a Last name referent.

Returns

boolean if search concerns the last name of referent

3.23.2.5 QString Search::getText ( ) const

Search::getText Return sql portion of filter.

Returns

QString the sql portion

3.23.2.6 void Search::setGroupFilter (bool *getGroupFilter*)

Search::setGroupFilter Modify if we active search filter.

Parameters

getGroupFilter	Get if filter is actived

3.23.2.7 void Search::setSearchInCompanies (bool *getSearchInCompanies*)

Search::setSearchInCompanies Modify the filter of search.

Parameters

getSearchIn↩ Companies	Search in companies is concerned

3.23.2.8 void Search::setSearchInReferentLastname (bool *getSearchInReferentLastname*)

Search::setSearchInReferentLastname Modify the filter of search.

Parameters

getSearchIn↩ Referent↩ Lastname	Search in referent last name is concerned

3.23.2.9 void Search::setText (const QString & *getText*)

Search::setText Modify sql portion.

Parameters

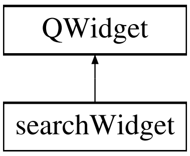
getText	Get sql portion

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

- src/models/search.h
- 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:

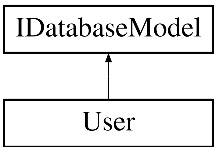
- src/widgets/searchwidget.h
- src/widgets/searchwidget.cpp

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. Contruct 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::setFirstnament 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 addess (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

id	User id

3.25.3 Member Function Documentation

3.25.3.1 QString User::getAddress ( ) const

User::getAddress Return the company addess (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

id	User identify

Implements IDatabaseModel.

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

User::setAddress Modify the user company *address*

Parameters

address	Company address (name and number of street)

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

User::setCity Modify the *city*

Parameters

city	Company city address

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

User::setCompany Modify the user *company* name.

Parameters

company	New user company name

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

User::setEmail Modify the user professional *email*

Parameters

email	The user professional email

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

User::setFirstnament Modify the user *firstname*

Parameters

firstname	New user firstname

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

User::setLastname Modify the user *lastname*

Parameters

lastname	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

mobilePhone	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

noSiret	SIRET number

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

User::setPhone Modify the number of the desktop *phone*

Parameters

phone	Number of the desktop phone

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

User::setPostalCode Modify the postal code *postalCode*

Parameters

postalCode	New postal code

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

User::setTitle Modify the user/company activities *description*

Parameters

title	Short description on activity(ies) of customer company

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

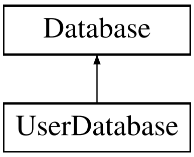
- src/models/user.h
- src/models/user.cpp

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 pId=1)

  getUser Get informations about the user (identified by 'pId')
- 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

See also

Database
User

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

pId	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:

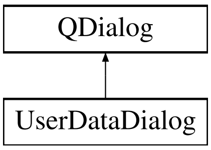
- src/database/userdatabase.h
- src/database/userdatabase.cpp

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 ()

  UserDataDialog::fillFields Fill line edits with the data of the user.
- 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

See also

Project

3.27.2 Constructor & Destructor Documentation

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

UserDataDialog: Construct a window with user data.

Parameters

parent	

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

- src/dialogs/userdatadialog.h
- 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:

- src/utils.h
- src/utils.cpp

Index