FactDev

0.1

Generated by Doxygen 1.8.6

Mon Mar 2 2015 08:31:03

Contents

1 FactDev documentation 1

2 README 3

3 QTestRunner 5

4 Namespace Index 7

4.1 Namespace List . . . 7

5 Hierarchical Index 9

5.1 Class Hierarchy . . . 9

6 Class Index 13

6.1 Class List . . . 13

7 Namespace Documentation 17

7.1 Databases Namespace Reference . . . 17

7.1.1 Detailed Description . . . 17

7.2 Gui Namespace Reference . . . 17

7.2.1 Detailed Description . . . 18

7.3 Gui::Widgets Namespace Reference . . . 18

7.3.1 Detailed Description . . . 18

7.4 Models Namespace Reference . . . 18

7.4.1 Detailed Description . . . 19

8 Class Documentation 21

8.1 Gui::Dialogs::AddProjectDialog Class Reference . . . 21

8.1.1 Detailed Description . . . 21

8.1.2 Constructor & Destructor Documentation . . . 22

8.1.2.1 AddProjectDialog . . . 22

8.1.2.2 AddProjectDialog . . . 22

8.2 Gui::Dialogs::AddQuoteDialog Class Reference . . . 22

8.2.1 Detailed Description . . . 23

8.2.2 Constructor & Destructor Documentation . . . 23

8.2.2.1 AddQuoteDialog . . . 23
8.2.3 Member Function Documentation . . . 23
8.2.3.1 getNumber . . . 23
8.3 Models::Billing Class Reference . . . 23
8.3.1 Detailed Description . . . 25
8.3.2 Constructor & Destructor Documentation . . . 25
8.3.2.1 Billing . . . 25
8.3.3 Member Function Documentation . . . 25
8.3.3.1 addContributory . . . 25
8.3.3.2 getContributories . . . 25
8.3.3.3 getDataMap . . . 25
8.3.3.4 getDate . . . 26
8.3.3.5 getDescription . . . 26
8.3.3.6 getNumber . . . 26
8.3.3.7 getTitle . . . 26
8.3.3.8 hydrat . . . 26
8.3.3.9 isBilling . . . 26
8.3.3.10 operator!= . . . 26
8.3.3.11 operator< . . . 27
8.3.3.12 operator== . . . 27
8.3.3.13 setDate . . . 27
8.3.3.14 setDescription . . . 27
8.3.3.15 setIsBilling . . . 27
8.3.3.16 setNumber . . . 28
8.3.3.17 setTitle . . . 28
8.4 Databases::BillingDatabase Class Reference . . . 28
8.4.1 Detailed Description . . . 29
8.4.2 Member Function Documentation . . . 29
8.4.2.1 addBilling . . . 29
8.4.2.2 addBillingProject . . . 29
8.4.2.3 getAllBillingsOfProject . . . 30
8.4.2.4 getBilling . . . 30
8.4.2.5 getBilling . . . 30
8.4.2.6 getBillingsTable . . . 30
8.4.2.7 getMaxBillingNumber . . . 31
8.4.2.8 getMaxQuoteNumber . . . 31
8.4.2.9 instance . . . 31
8.4.2.10 removeBilling . . . 31
8.4.2.11 removeBillingProject . . . 31
8.5 BillingDatabaseTest Class Reference . . . 32

8.6 BillingModelTest Class Reference . . . 32
8.7 Gui::Widgets::WdgModels::BillingsTableModel Class Reference . . . 32
8.7.1 Detailed Description . . . 33
8.7.2 Constructor & Destructor Documentation . . . 33
8.7.2.1 BillingsTableModel . . . 33
8.7.3 Member Function Documentation . . . 33
8.7.3.1 append . . . 33
8.7.3.2 columnCount . . . 34
8.7.3.3 count . . . 34
8.7.3.4 data . . . 34
8.7.3.5 flags . . . 34
8.7.3.6 getBillings . . . 34
8.7.3.7 headerData . . . 34
8.7.3.8 remove . . . 35
8.7.3.9 rowCount . . . 35
8.7.3.10 setData . . . 35
8.8 Gui::Widgets::CheckFields::CheckCity Class Reference . . . 35
8.8.1 Detailed Description . . . 36
8.8.2 Constructor & Destructor Documentation . . . 36
8.8.2.1 CheckCity . . . 36
8.9 Gui::Widgets::CheckFields::CheckCountry Class Reference . . . 36
8.9.1 Detailed Description . . . 37
8.9.2 Constructor & Destructor Documentation . . . 37
8.9.2.1 CheckCountry . . . 37
8.10 Gui::Widgets::CheckFields::CheckEmail Class Reference . . . 37
8.10.1 Detailed Description . . . 38
8.10.2 Constructor & Destructor Documentation . . . 38
8.10.2.1 CheckEmail . . . 38
8.10.3 Member Function Documentation . . . 38
8.10.3.1 check . . . 38
8.11 Gui::Widgets::CheckFields::CheckFieldsLetters Class Reference . . . 39
8.11.1 Detailed Description . . . 39
8.11.2 Constructor & Destructor Documentation . . . 39
8.11.2.1 CheckFieldsLetters . . . 39
8.11.3 Member Function Documentation . . . 39
8.11.3.1 check . . . 39
8.12 Gui::Widgets::CheckFields::CheckName Class Reference . . . 40
8.12.1 Detailed Description . . . 40
8.12.2 Constructor & Destructor Documentation . . . 40
8.12.2.1 CheckName . . . 40

8.13 Gui::Widgets::CheckFields::CheckPhone Class Reference . . . 41
8.13.1 Detailed Description . . . 41
8.13.2 Constructor & Destructor Documentation . . . 41
8.13.2.1 CheckPhone . . . 41
8.13.3 Member Function Documentation . . . 42
8.13.3.1 check . . . 42
8.13.3.2 getCountry . . . 42
8.13.3.3 setCountry . . . 42
8.14 Gui::Widgets::CheckFields::CheckPostalCode Class Reference . . . 42
8.14.1 Detailed Description . . . 43
8.14.2 Constructor & Destructor Documentation . . . 43
8.14.2.1 CheckPostalCode . . . 43
8.14.3 Member Function Documentation . . . 43
8.14.3.1 check . . . 43
8.14.3.2 getCountry . . . 44
8.14.3.3 setCountry . . . 44
8.15 Gui::Widgets::CheckFields::CheckQLineEdit Class Reference . . . 44
8.15.1 Detailed Description . . . 45
8.15.2 Constructor & Destructor Documentation . . . 45
8.15.2.1 CheckQLineEdit . . . 45
8.15.3 Member Function Documentation . . . 45
8.15.3.1 getBtnValid . . . 45
8.15.3.2 isValid . . . 45
8.15.3.3 setBtnValid . . . 45
8.16 Gui::Widgets::CheckFields::CheckSiretNumber Class Reference . . . 46
8.16.1 Detailed Description . . . 46
8.16.2 Constructor & Destructor Documentation . . . 46
8.16.2.1 CheckSiretNumber . . . 46
8.16.3 Member Function Documentation . . . 46
8.16.3.1 check . . . 46
8.17 Gui::Widgets::CheckFields::CheckUntilField Class Reference . . . 47
8.17.1 Detailed Description . . . 47
8.17.2 Constructor & Destructor Documentation . . . 47
8.17.2.1 CheckUntilField . . . 47
8.17.3 Member Function Documentation . . . 47
8.17.3.1 check . . . 47
8.18 Gui::Widgets::CheckFields::CheckValidField Class Reference . . . 48
8.18.1 Detailed Description . . . 48
8.18.2 Constructor & Destructor Documentation . . . 48
8.18.2.1 CheckValidField . . . 48

8.18.3 Member Function Documentation . . . 49
8.18.3.1 check . . . 49
8.19 Gui::Widgets::Delegates::ComboBoxDelegate Class Reference . . . 49
8.20 Gui::Widgets::ComboBoxModelWidget Class Reference . . . 50
8.20.1 Detailed Description . . . 50
8.21 Mustache::Context Class Reference . . . 50
8.21.1 Detailed Description . . . 51
8.21.2 Constructor & Destructor Documentation . . . 51
8.21.2.1 Context . . . 51
8.21.3 Member Function Documentation . . . 51
8.21.3.1 canEval . . . 51
8.21.3.2 eval . . . 51
8.21.3.3 isFalse . . . 51
8.21.3.4 listCount . . . 51
8.21.3.5 partialResolver . . . 51
8.21.3.6 partialValue . . . 51
8.21.3.7 pop . . . 52
8.21.3.8 push . . . 52
8.21.3.9 stringValue . . . 52
8.22 ContributoriesDatabaseTest Class Reference . . . 52
8.23 Models::ContributoriesList Class Reference . . . 52
8.24 Gui::Widgets::WdgModels::ContributoriesTableModel Class Reference . . . 53
8.24.1 Detailed Description . . . 54
8.24.2 Constructor & Destructor Documentation . . . 54
8.24.2.1 ContributoriesTableModel . . . 54
8.24.3 Member Function Documentation . . . 54
8.24.3.1 append . . . 54
8.24.3.2 columnCount . . . 54
8.24.3.3 count . . . 55
8.24.3.4 data . . . 55
8.24.3.5 flags . . . 55
8.24.3.6 getContributories . . . 55
8.24.3.7 headerData . . . 55
8.24.3.8 remove . . . 56
8.24.3.9 rowCount . . . 57
8.24.3.10 setData . . . 57
8.25 Gui::Widgets::ContributoriesWidget Class Reference . . . 57
8.25.1 Member Function Documentation . . . 58
8.25.1.1 add . . . 58
8.25.1.2 getContributories . . . 58

8.26 Models::Contributory Class Reference . . . 58
8.26.1 Detailed Description . . . 59
8.26.2 Constructor & Destructor Documentation . . . 59
8.26.2.1 Contributory . . . 59
8.26.3 Member Function Documentation . . . 60
8.26.3.1 getDataMap . . . 60
8.26.3.2 getDescription . . . 60
8.26.3.3 getNbHours . . . 60
8.26.3.4 getProject . . . 60
8.26.3.5 hydrat . . . 60
8.26.3.6 operator!= . . . 60
8.26.3.7 operator== . . . 61
8.26.3.8 setDescription . . . 61
8.26.3.9 setNbHours . . . 61
8.26.3.10 setProject . . . 61
8.27 Databases::ContributoryDatabase Class Reference . . . 61
8.27.1 Detailed Description . . . 62
8.27.2 Member Function Documentation . . . 62
8.27.2.1 addContributory . . . 62
8.27.2.2 getContributory . . . 62
8.27.2.3 getContributory . . . 63
8.27.2.4 instance . . . 63
8.27.2.5 removeContributory . . . 63
8.28 ContributoryListTest Class Reference . . . 63
8.29 ContributoryModelTest Class Reference . . . 64
8.30 CounterContext Class Reference . . . 64
8.30.1 Member Function Documentation . . . 64
8.30.1.1 canEval . . . 64
8.30.1.2 canEval . . . 65
8.30.1.3 eval . . . 65
8.30.1.4 eval . . . 65
8.30.1.5 stringValue . . . 65
8.30.1.6 stringValue . . . 65
8.31 Models::Customer Class Reference . . . 65
8.31.1 Detailed Description . . . 67
8.31.2 Constructor & Destructor Documentation . . . 67
8.31.2.1 Customer . . . 67
8.31.3 Member Function Documentation . . . 68
8.31.3.1 getAddress . . . 68
8.31.3.2 getCity . . . 68

8.31.3.3 getCompany . . . 68
8.31.3.4 getCountry . . . 68
8.31.3.5 getDataMap . . . 68
8.31.3.6 getEmail . . . 68
8.31.3.7 getFax . . . 69
8.31.3.8 getFirstnameReferent . . . 69
8.31.3.9 getLastnameReferent . . . 69
8.31.3.10 getMobilePhone . . . 69
8.31.3.11 getNameFolder . . . 69
8.31.3.12 getPath . . . 69
8.31.3.13 getPhone . . . 69
8.31.3.14 getPostalCode . . . 70
8.31.3.15 hydrat . . . 70
8.31.3.16 operator!= . . . 70
8.31.3.17 operator== . . . 70
8.31.3.18 setAddress . . . 70
8.31.3.19 setCity . . . 70
8.31.3.20 setCompany . . . 71
8.31.3.21 setCountry . . . 71
8.31.3.22 setEmail . . . 71
8.31.3.23 setFax . . . 71
8.31.3.24 setFirstnameReferent . . . 71
8.31.3.25 setLastnameReferent . . . 71
8.31.3.26 setMobilePhone . . . 72
8.31.3.27 setPhone . . . 73
8.31.3.28 setPostalCode . . . 73
8.32 Gui::Widgets::CustomerContextualMenu Class Reference . . . 73
8.32.1 Detailed Description . . . 73
8.32.2 Constructor & Destructor Documentation . . . 74
8.32.2.1 CustomerContextualMenu . . . 74
8.33 Databases::CustomerDatabase Class Reference . . . 74
8.33.1 Detailed Description . . . 75
8.33.2 Member Function Documentation . . . 75
8.33.2.1 addCustomer . . . 75
8.33.2.2 getCustomer . . . 75
8.33.2.3 getCustomer . . . 75
8.33.2.4 getCustomersTable . . . 76
8.33.2.5 getItemBillQuote . . . 76
8.33.2.6 getItemCustomer . . . 76
8.33.2.7 getItemProject . . . 76

8.33.2.8 getItemRoot . . . 77
8.33.2.9 getNbCustomers . . . 77
8.33.2.10 getTree . . . 77
8.33.2.11 instance . . . 77
8.33.2.12 removeCustomer . . . 77
8.33.2.13 updateCustomer . . . 78
8.34 CustomerDatabaseTest Class Reference . . . 78
8.35 Gui::Widgets::CustomerDataWidget Class Reference . . . 78
8.35.1 Detailed Description . . . 79
8.35.2 Member Function Documentation . . . 79
8.35.2.1 printInformations . . . 79
8.36 CustomerModelTest Class Reference . . . 79
8.37 Gui::Widgets::WdgModels::CustomersTableModel Class Reference . . . 79
8.37.1 Detailed Description . . . 80
8.37.2 Constructor & Destructor Documentation . . . 80
8.37.2.1 CustomersTableModel . . . 80
8.37.3 Member Function Documentation . . . 80
8.37.3.1 append . . . 80
8.37.3.2 columnCount . . . 81
8.37.3.3 count . . . 81
8.37.3.4 data . . . 81
8.37.3.5 flags . . . 81
8.37.3.6 getCustomers . . . 81
8.37.3.7 headerData . . . 82
8.37.3.8 remove . . . 83
8.37.3.9 rowCount . . . 83
8.37.3.10 setData . . . 83
8.38 Databases::Database Class Reference . . . 83
8.38.1 Detailed Description . . . 85
8.38.2 Member Function Documentation . . . 85
8.38.2.1 executeFile . . . 85
8.38.2.2 instance . . . 85
8.38.2.3 lastError . . . 85
8.38.2.4 setDatabase . . . 85
8.38.2.5 value . . . 86
8.39 Exceptions::DbException Class Reference . . . 86
8.39.1 Detailed Description . . . 86
8.39.2 Constructor & Destructor Documentation . . . 87
8.39.2.1 DbException . . . 87
8.39.3 Member Function Documentation . . . 88

8.39.3.1 popupMessage . . . 88
8.40 Gui::Dialogs::DialogAddCustomer Class Reference . . . 88
8.40.1 Detailed Description . . . 89
8.40.2 Constructor & Destructor Documentation . . . 89
8.40.2.1 DialogAddCustomer . . . 89
8.41 Gui::Widgets::Delegates::DoubleSpinBoxDelegate Class Reference . . . 89
8.42 Exceptions::FileException Class Reference . . . 89
8.42.1 Detailed Description . . . 90
8.42.2 Constructor & Destructor Documentation . . . 90
8.42.2.1 FileException . . . 90
8.42.3 Member Function Documentation . . . 90
8.42.3.1 popupMessage . . . 90
8.43 Generation Class Reference . . . 90
8.44 Generator Class Reference . . . 91
8.45 Utils::HierarchicalSystem Class Reference . . . 91
8.45.1 Detailed Description . . . 91
8.45.2 Member Function Documentation . . . 92
8.45.2.1 addBillingToProject . . . 92
8.45.2.2 addProjectToCustomer . . . 92
8.45.2.3 getCustomers . . . 92
8.45.2.4 getProjects . . . 92
8.46 Gui::Widgets::CheckFields::ICheckField Class Reference . . . 92
8.46.1 Detailed Description . . . 93
8.46.2 Member Function Documentation . . . 93
8.46.2.1 check . . . 93
8.47 Models::IModel Class Reference . . . 93
8.47.1 Detailed Description . . . 94
8.47.2 Member Function Documentation . . . 94
8.47.2.1 getDataMap . . . 94
8.47.2.2 getId . . . 94
8.47.2.3 hydrat . . . 94
8.47.2.4 isToRemoved . . . 95
8.47.2.5 setId . . . 95
8.47.2.6 setToRemoved . . . 95
8.48 Utils::ItemType Class Reference . . . 95
8.48.1 Detailed Description . . . 96
8.48.2 Constructor & Destructor Documentation . . . 96
8.48.2.1 ItemType . . . 96
8.48.3 Member Function Documentation . . . 96
8.48.3.1 getModel . . . 96

8.48.3.2 getName . . . 96
8.48.3.3 getType . . . 96
8.48.3.4 setName . . . 97
8.48.3.5 setType . . . 97
8.49 ItemTypeTest Class Reference . . . 97
8.50 Utils::Log Class Reference . . . 97
8.50.1 Detailed Description . . . 98
8.50.2 Member Function Documentation . . . 98
8.50.2.1 instance . . . 98
8.50.2.2 write . . . 98
8.50.3 Friends And Related Function Documentation . . . 98
8.50.3.1 operator<< . . . 98
8.51 Gui::MainWindow Class Reference . . . 99
8.51.1 Detailed Description . . . 100
8.51.2 Constructor & Destructor Documentation . . . 100
8.51.2.1 MainWindow . . . 100
8.51.3 Member Function Documentation . . . 100
8.51.3.1 addBill . . . 100
8.51.3.2 addDoc . . . 100
8.51.3.3 addQuote . . . 101
8.51.3.4 editUser . . . 101
8.51.3.5 getCurrentCustomerId . . . 101
8.51.3.6 getCurrentCustomerName . . . 101
8.51.3.7 getCurrentProjectId . . . 101
8.51.3.8 getCurrentProjectName . . . 101
8.51.3.9 getCurrentQuoteId . . . 102
8.51.3.10 newProject . . . 102
8.51.3.11 rootTree . . . 102
8.51.3.12 search . . . 102
8.51.3.13 treeLevel . . . 102
8.52 Gui::Dialogs::MessageBox Class Reference . . . 102
8.52.1 Detailed Description . . . 103
8.52.2 Constructor & Destructor Documentation . . . 103
8.52.2.1 MessageBox . . . 103
8.52.3 Member Function Documentation . . . 103
8.52.3.1 setImage . . . 103
8.52.3.2 setText . . . 104
8.53 Parameters Class Reference . . . 104
8.53.1 Detailed Description . . . 104
8.53.2 Member Data Documentation . . . 104

8.53.2.1 DB_FILENAME . . . 104
8.53.2.2 VERSION . . . 104
8.54 Mustache::PartialFileLoader Class Reference . . . 105
8.54.1 Detailed Description . . . 105
8.54.2 Member Function Documentation . . . 105
8.54.2.1 getPartial . . . 105
8.55 Mustache::PartialMap Class Reference . . . 105
8.55.1 Detailed Description . . . 106
8.55.2 Member Function Documentation . . . 106
8.55.2.1 getPartial . . . 106
8.56 Mustache::PartialResolver Class Reference . . . 106
8.56.1 Detailed Description . . . 106
8.56.2 Member Function Documentation . . . 106
8.56.2.1 getPartial . . . 106
8.57 Gui::Widgets::Popup Class Reference . . . 107
8.57.1 Detailed Description . . . 107
8.58 Models::Project Class Reference . . . 107
8.58.1 Detailed Description . . . 108
8.58.2 Constructor & Destructor Documentation . . . 109
8.58.2.1 Project . . . 109
8.58.3 Member Function Documentation . . . 110
8.58.3.1 getBeginDate . . . 110
8.58.3.2 getCustomer . . . 110
8.58.3.3 getDailyRate . . . 110
8.58.3.4 getDataMap . . . 110
8.58.3.5 getDescription . . . 110
8.58.3.6 getEndDate . . . 110
8.58.3.7 getName . . . 111
8.58.3.8 hydrat . . . 111
8.58.3.9 operator!= . . . 111
8.58.3.10 operator< . . . 111
8.58.3.11 operator== . . . 111
8.58.3.12 setBeginDate . . . 111
8.58.3.13 setCustomer . . . 112
8.58.3.14 setDailyRate . . . 112
8.58.3.15 setDescription . . . 112
8.58.3.16 setEndDate . . . 112
8.58.3.17 setName . . . 112
8.59 Gui::Widgets::Delegates::ProjectComboDelegate Class Reference . . . 112
8.60 Gui::Widgets::WdgModels::ProjectContributoriesTableModel Class Reference . . . 113

8.60.1 Constructor & Destructor Documentation . . . 114
8.60.1.1 ProjectContributoriesTableModel . . . 114
8.60.2 Member Function Documentation . . . 114
8.60.2.1 columnCount . . . 114
8.60.2.2 data . . . 114
8.60.2.3 flags . . . 114
8.60.2.4 headerData . . . 115
8.60.2.5 rowCount . . . 115
8.60.2.6 setData . . . 115
8.61 Databases::ProjectDatabase Class Reference . . . 115
8.61.1 Detailed Description . . . 116
8.61.2 Member Function Documentation . . . 117
8.61.2.1 addProject . . . 117
8.61.2.2 getAllProjects . . . 117
8.61.2.3 getNbProjects . . . 117
8.61.2.4 getNbProjectsForACustomer . . . 117
8.61.2.5 getProject . . . 117
8.61.2.6 getProject . . . 117
8.61.2.7 getProjectsOfCustomer . . . 118
8.61.2.8 getProjectsTable . . . 118
8.61.2.9 instance . . . 118
8.61.2.10 removeProject . . . 118
8.61.2.11 updateProject . . . 118
8.62 ProjectDatabaseTest Class Reference . . . 119
8.63 ProjectModelTest Class Reference . . . 119
8.64 Gui::Widgets::WdgModels::ProjectsTableModel Class Reference . . . 119
8.64.1 Detailed Description . . . 120
8.64.2 Constructor & Destructor Documentation . . . 120
8.64.2.1 ProjectsTableModel . . . 120
8.64.3 Member Function Documentation . . . 121
8.64.3.1 append . . . 121
8.64.3.2 columnCount . . . 121
8.64.3.3 count . . . 121
8.64.3.4 data . . . 121
8.64.3.5 flags . . . 121
8.64.3.6 getProjects . . . 122
8.64.3.7 headerData . . . 122
8.64.3.8 remove . . . 122
8.64.3.9 rowCount . . . 122
8.64.3.10 setData . . . 122

8.65 Gui::Widgets::ProjectsWidget Class Reference . . . 123
8.65.1 Detailed Description . . . 123
8.65.2 Constructor & Destructor Documentation . . . 123
8.65.2.1 ProjectsWidget . . . 123
8.65.3 Member Function Documentation . . . 124
8.65.3.1 updateBtn . . . 124
8.66 Mustache::QtVariantContext Class Reference . . . 124
8.66.1 Detailed Description . . . 124
8.66.2 Member Typedef Documentation . . . 125
8.66.2.1 fn_t . . . 125
8.66.3 Member Function Documentation . . . 125
8.66.3.1 canEval . . . 125
8.66.3.2 eval . . . 125
8.66.3.3 isFalse . . . 125
8.66.3.4 listCount . . . 125
8.66.3.5 pop . . . 125
8.66.3.6 push . . . 126
8.66.3.7 stringValue . . . 126
8.67 Models::Rate Class Reference . . . 126
8.68 Databases::RateDatabase Class Reference . . . 126
8.68.1 Member Function Documentation . . . 127
8.68.1.1 instance . . . 127
8.69 RateModelTest Class Reference . . . 127
8.70 Gui::Widgets::RateWidget Class Reference . . . 127
8.70.1 Detailed Description . . . 128
8.70.2 Constructor & Destructor Documentation . . . 128
8.70.2.1 RateWidget . . . 128
8.70.3 Member Function Documentation . . . 128
8.70.3.1 getDailyRate . . . 128
8.70.3.2 getHourlyRate . . . 129
8.70.3.3 setDailyRate . . . 129
8.70.3.4 setHourlyRate . . . 129
8.70.3.5 setWidgetDailyRateValue . . . 129
8.70.3.6 setWidgetHourlyRateValue . . . 129
8.71 Mustache::Renderer Class Reference . . . 129
8.71.1 Detailed Description . . . 130
8.71.2 Member Function Documentation . . . 130
8.71.2.1 error . . . 130
8.71.2.2 errorPartial . . . 130
8.71.2.3 errorPos . . . 130

8.71.2.4 render . . . 130
8.71.2.5 setTagMarkers . . . 130
8.72 Models::Search Class Reference . . . 130
8.72.1 Detailed Description . . . 132
8.72.2 Member Function Documentation . . . 132
8.72.2.1 filterOnBillsOrQuotes . . . 132
8.72.2.2 filterOnCompany . . . 132
8.72.2.3 filterOnContributories . . . 132
8.72.2.4 filterOnCustomersWithoutProject . . . 132
8.72.2.5 filterOnNumberElements . . . 132
8.72.2.6 filterOnProjects . . . 133
8.72.2.7 filterOnReferentLastname . . . 133
8.72.2.8 filterOnVarcharElements . . . 133
8.72.2.9 getFilter . . . 133
8.72.2.10 getGroupFilter . . . 133
8.72.2.11 getSearchInBillsQuotes . . . 133
8.72.2.12 getSearchInCompanies . . . 134
8.72.2.13 getSearchInProjects . . . 134
8.72.2.14 getSearchInReferentLastname . . . 134
8.72.2.15 getText . . . 134
8.72.2.16 searchInContributories . . . 134
8.72.2.17 setGroupFilter . . . 134
8.72.2.18 setSearchInBillsQuotes . . . 135
8.72.2.19 setSearchInCompanies . . . 135
8.72.2.20 setSearchInContributories . . . 135
8.72.2.21 setSearchInProjects . . . 135
8.72.2.22 setSearchInReferentLastname . . . 135
8.72.2.23 setText . . . 135
8.73 searchTest Class Reference . . . 136
8.74 Gui::Widgets::searchWidget Class Reference . . . 136
8.74.1 Detailed Description . . . 137
8.74.2 Constructor & Destructor Documentation . . . 137
8.74.2.1 searchWidget . . . 137
8.74.3 Member Function Documentation . . . 137
8.74.3.1 getCurrentCustomerId . . . 137
8.74.3.2 isCustomerSelected . . . 137
8.74.3.3 search . . . 137
8.75 Utils::String Class Reference . . . 138
8.75.1 Detailed Description . . . 138
8.75.2 Member Function Documentation . . . 138

8.75.2.1 firstLetterToUpper . . . 138
8.76 StringTest Class Reference . . . 138
8.77 Mustache::Tag Struct Reference . . . 139
8.77.1 Detailed Description . . . 139
8.77.2 Member Enumeration Documentation . . . 139
8.77.2.1 Type . . . 139
8.78 TestAdder< T > Class Template Reference . . . 139
8.79 testadder Class Reference . . . 140
8.80 TestMustache Class Reference . . . 140
8.81 TestRunner Class Reference . . . 140
8.82 Gui::Widgets::Delegates::UnitComboDelegate Class Reference . . . 140
8.83 Models::User Class Reference . . . 141
8.83.1 Detailed Description . . . 143
8.83.2 Constructor & Destructor Documentation . . . 143
8.83.2.1 User . . . 143
8.83.3 Member Function Documentation . . . 143
8.83.3.1 getAddress . . . 143
8.83.3.2 getCity . . . 143
8.83.3.3 getCompany . . . 143
8.83.3.4 getDataMap . . . 144
8.83.3.5 getEmail . . . 144
8.83.3.6 getFirstname . . . 144
8.83.3.7 getLastname . . . 144
8.83.3.8 getMobilePhone . . . 144
8.83.3.9 getNoSiret . . . 144
8.83.3.10 getPhone . . . 145
8.83.3.11 getPostalCode . . . 145
8.83.3.12 getTitle . . . 145
8.83.3.13 getWorkspaceName . . . 145
8.83.3.14 getWorkspacePath . . . 145
8.83.3.15 hydrat . . . 145
8.83.3.16 operator!= . . . 145
8.83.3.17 operator== . . . 146
8.83.3.18 setAddress . . . 146
8.83.3.19 setCity . . . 146
8.83.3.20 setCompany . . . 146
8.83.3.21 setEmail . . . 146
8.83.3.22 setFirstname . . . 147
8.83.3.23 setLastname . . . 147
8.83.3.24 setMobilePhone . . . 147

8.83.3.25 setNoSiret . . . 147
8.83.3.26 setPhone . . . 147
8.83.3.27 setPostalCode . . . 147
8.83.3.28 setTitle . . . 147
8.83.3.29 setWorkspaceName . . . 148
8.83.3.30 setWorkspacePath . . . 148
8.84 Databases::UserDatabase Class Reference . . . 148
8.84.1 Detailed Description . . . 149
8.84.2 Member Function Documentation . . . 149
8.84.2.1 getUser . . . 149
8.84.2.2 instance . . . 149
8.85 UserDatabaseTest Class Reference . . . 149
8.86 Gui::Dialogs::UserDataDialog Class Reference . . . 150
8.86.1 Detailed Description . . . 150
8.86.2 Constructor & Destructor Documentation . . . 150
8.86.2.1 UserDataDialog . . . 150
8.87 UserModelTest Class Reference . . . 151

Index **152**

Chapter 1

FactDev documentation

This website contains the documentation of FactDev, a billings and quotes software.

Team

FACT team is a team for Universitiy Project. Members of this team are :

- Florent Berbie (KraTuX31)
- Antoine de Roquemaurel (aroquemaurel)
- Cédric Rohaut (Oxynos)
- Manantsoa Andriamihary Razanajatovo (manantsoa)

For more informations, you can go to http://fact-team.github.io.

What is FactDev ?

FactDev is a software for quotes and billings developped by FACT team for university project in Toulouse III University – Paul Sabatier.

This software is developped with C++ and Qt framework, and there is differents features :

- Customers database
- Differents projects for customers
- Quotes
- Billings

Documentation

You can access to the documentation generated by Doxygen here :

- HTML Documentation
- PDF Documentation

Installation and using

- User Manual

Chapter 2

README

`![Build Status](https://travis-ci.org/robertknight/qt-mustache.svg?branch=master)`

Qt Mustache

qt-mustache is a simple library for rendering `Mustache templates`.

Example Usage

“‘cpp #include "mustache.h"

QVariantHash contact; contact["name"] = "John Smith"; contact["email"] = "john.smith@gmail.com";

QString contactTemplate = "{{name}} <a href=\"mailto:{{email}}">{{email}}";

Mustache::Renderer renderer; Mustache::QtVariantContext context(contact);

QTextStream output(stdout); output << renderer.render(contactTemplate, &context); “‘

Outputs: `<b>John Smith</b> <a href="mailto:john.smith\@gmail.com">john.-smith@gmail.com</a>`

For further examples, see the tests in `test_mustache.cpp`

Building

- To build the tests, run `qmake` followed by `make`
- To use qt-mustache in your project, just add the `mustache.h` and `mustache.cpp` files to your project.

License

qt-mustache is licensed under the BSD license.

Dependencies

qt-mustache depends on the QtCore library. It is compatible with Qt 4 and Qt 5.

Usage

Syntax

qt-mustache uses the standard Mustache syntax. See the Mustache manual for details.

Data Sources

qt-mustache expands Mustache tags using values from a Mustache::Context. Mustache::QtVariantContext is a simple context implementation which wraps a QVariantHash or QVariantMap. If you want to render a template using a custom data source, you can either create a QVariantHash which mirrors the data source or you can re-implement Mustache::Context.

Partials

When a {{>partial}} Mustache tag is encountered, qt-mustache will attempt to load the partial using a Mustache::PartialResolver provided by the context. Mustache::PartialMap is a simple resolver which takes a QHash<QString,QString> map of partial names to values and looks up partials in that map. Mustache::PartialFileLoader is another simple resolver which fetches partials from <partial name>.mustache files in a specified directory.

You can re-implement the Mustache::PartialResolver interface if you want to load partials from a custom source (eg. a database).

Error Handling

If an error occurs when rendering a template, Mustache::Renderer::errorPosition() is set to non-negative value and template rendering stops. If the error occurs whilst rendering a partial template, errorPartial() contains the name of the partial.

Lambdas

The Mustache manual provides a mechanism to customize rendering of template sections by setting the value for a tag to a callable object (eg. a lambda in Ruby or Javascript), which takes the unrendered block of text for a template section and renders it itself. qt-mustache supports this via the Context::canEval() and Context::eval() methods.

Chapter 3

QTestRunner

A repo for Increase QTest productivity

Thanks to https://marcoarena.wordpress.com/2012/06/23/increase-your-qtest-productivity/
!

> The QTestLib framework is a tool for unit testing Qt based applications and libraries. I find it precious and simple to use, though it lacks some important features, supported, for example, by GTest. I'm not talking about mocking – for this you generally need an out-and-out framework (like GMock) – instead, I'm referring to simple things like fast deploying. Suppose you have written a test class like this:

This repo contains only the code in this article.

Chapter 4

Namespace Index

4.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Databases
    Class for Database, contains queries . . . . . ??
Gui . . . . . ??
Gui::Widgets
    Contains Widgets classes . . . . . ??
Models
    Models classes . . . . . ??

Chapter 5

Hierarchical Index

5.1 Class Hierarchy

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

- Mustache::Context . . . ??
  - Mustache::QtVariantContext . . . ??
    - CounterContext . . . ??
    - CounterContext . . . ??
- ContributoryListTest . . . ??
- Databases::Database . . . ??
  - Databases::BillingDatabase . . . ??
  - Databases::ContributoryDatabase . . . ??
  - Databases::CustomerDatabase . . . ??
  - Databases::ProjectDatabase . . . ??
  - Databases::RateDatabase . . . ??
  - Databases::UserDatabase . . . ??
- exception
  - Exceptions::DbException . . . ??
- Exceptions::FileException . . . ??
- Generator . . . ??
- Utils::HierarchicalSystem . . . ??
- Gui::Widgets::CheckFields::ICheckField . . . ??
  - Gui::Widgets::CheckFields::CheckQLineEdit . . . ??
    - Gui::Widgets::CheckFields::CheckEmail . . . ??
    - Gui::Widgets::CheckFields::CheckSiretNumber . . . ??
    - Gui::Widgets::CheckFields::CheckUntilField . . . ??
      - Gui::Widgets::CheckFields::CheckFieldsLetters . . . ??
        - Gui::Widgets::CheckFields::CheckCity . . . ??
        - Gui::Widgets::CheckFields::CheckCountry . . . ??
        - Gui::Widgets::CheckFields::CheckName . . . ??
      - Gui::Widgets::CheckFields::CheckPhone . . . ??
      - Gui::Widgets::CheckFields::CheckPostalCode . . . ??
    - Gui::Widgets::CheckFields::CheckValidField . . . ??
- Models::IModel . . . ??
  - Models::Billing . . . ??
  - Models::Contributory . . . ??
  - Models::Customer . . . ??
  - Models::Project . . . ??
  - Models::User . . . ??
- Utils::ItemType . . . ??

Utils::Log . . . ??
Parameters . . . ??
Mustache::PartialResolver . . . ??
  Mustache::PartialFileLoader . . . ??
  Mustache::PartialMap . . . ??
Gui::Widgets::Popup . . . ??
QAbstractTableModel
  Gui::Widgets::WdgModels::BillingsTableModel . . . ??
  Gui::Widgets::WdgModels::ContributoriesTableModel . . . ??
  Gui::Widgets::WdgModels::CustomersTableModel . . . ??
  Gui::Widgets::WdgModels::ProjectContributoriesTableModel . . . ??
  Gui::Widgets::WdgModels::ProjectsTableModel . . . ??
QDialog
  Gui::Dialogs::AddProjectDialog . . . ??
  Gui::Dialogs::AddQuoteDialog . . . ??
  Gui::Dialogs::DialogAddCustomer . . . ??
  Gui::Dialogs::MessageBox . . . ??
  Gui::Dialogs::UserDataDialog . . . ??
QItemDelegate
  Gui::Widgets::Delegates::ComboBoxDelegate . . . ??
    Gui::Widgets::Delegates::ProjectComboDelegate . . . ??
    Gui::Widgets::Delegates::UnitComboDelegate . . . ??
  Gui::Widgets::Delegates::DoubleSpinBoxDelegate . . . ??
QLineEdit
  Gui::Widgets::CheckFields::CheckQLineEdit . . . ??
QMainWindow
  Gui::MainWindow . . . ??
QMap
  Models::ContributoriesList . . . ??
QMenu
  Gui::Widgets::CustomerContextualMenu . . . ??
QObject
  BillingDatabaseTest . . . ??
  BillingModelTest . . . ??
  ContributoriesDatabaseTest . . . ??
  ContributoryModelTest . . . ??
  CustomerDatabaseTest . . . ??
  CustomerModelTest . . . ??
  Generation . . . ??
  ItemTypeTest . . . ??
  ProjectDatabaseTest . . . ??
  ProjectModelTest . . . ??
  RateModelTest . . . ??
  searchTest . . . ??
  StringTest . . . ??
  TestMustache . . . ??
  TestMustache . . . ??
  UserDatabaseTest . . . ??
  UserModelTest . . . ??
QWidget
  Gui::Widgets::ComboBoxModelWidget . . . ??
  Gui::Widgets::ContributoriesWidget . . . ??
  Gui::Widgets::CustomerDataWidget . . . ??
  Gui::Widgets::ProjectsWidget . . . ??
  Gui::Widgets::RateWidget . . . ??
  Gui::Widgets::searchWidget . . . ??
Models::Rate . . . ??
Mustache::Renderer . . . ??

Models::Search . . . . . ??
Utils::String . . . . . ??
Mustache::Tag . . . . . ??
TestAdder< T > . . . . . ??
testadder . . . . . ??
TestRunner . . . . . ??

Chapter 6

Class Index

6.1 Class List

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

Gui::Dialogs::AddProjectDialog
: Windows to add a new Project . . . ??
Gui::Dialogs::AddQuoteDialog
: Window to add or modify a Quote . . . ??
Models::Billing
: Billing or Quote of a Customer . . . ??
Databases::BillingDatabase
: The **BillingDatabase** class Billing (or Quote) table database . . . ??
BillingDatabaseTest . . . ??
BillingModelTest . . . ??
Gui::Widgets::WdgModels::BillingsTableModel
: For a Billing table . . . ??
Gui::Widgets::CheckFields::CheckCity
: Line Edit of City with a check icon . . . ??
Gui::Widgets::CheckFields::CheckCountry
: CheckCountry::CheckCountry Line Edit of country with a check icon . . . ??
Gui::Widgets::CheckFields::CheckEmail
: Line Edit of email with a check icon . . . ??
Gui::Widgets::CheckFields::CheckFieldsLetters
: Field with only letters (no numbers) . . . ??
Gui::Widgets::CheckFields::CheckName
: Line edit of name with a check icon . . . ??
Gui::Widgets::CheckFields::CheckPhone
: Line Edit of Phone number with a check icon . . . ??
Gui::Widgets::CheckFields::CheckPostalCode
: Line Edit of postal code with a check icon . . . ??
Gui::Widgets::CheckFields::CheckQLineEdit
: LineEdit custom with a check of text inputed . . . ??
Gui::Widgets::CheckFields::CheckSiretNumber
: Line Edit with a check icon . . . ??
Gui::Widgets::CheckFields::CheckUntilField
: The CheckUntilField class . . . ??
Gui::Widgets::CheckFields::CheckValidField
: Check field not required . . . ??
Gui::Widgets::Delegates::ComboBoxDelegate . . . ??
Gui::Widgets::ComboBoxModelWidget
: The ComboBoxModelWidget class . . . ??
Mustache::Context . . . ??

ContributoriesDatabaseTest . . . ??
Models::ContributoriesList . . . ??
Gui::Widgets::WdgModels::ContributoriesTableModel
For a custom table for contributories widget . . . ??
Gui::Widgets::ContributoriesWidget . . . ??
Models::Contributory
The Contributory class . . . ??
Databases::ContributoryDatabase
The **ContributoryDatabase** class Contributory (or Quote) table database . . . ??
ContributoryListTest . . . ??
ContributoryModelTest . . . ??
CounterContext . . . ??
Models::Customer
Customer . . . ??
Gui::Widgets::CustomerContextualMenu
Display contextual menu on a customer . . . ??
Databases::CustomerDatabase
The **CustomerDatabase** class Customer table database . . . ??
CustomerDatabaseTest . . . ??
Gui::Widgets::CustomerDataWidget
Class for display info of a customer . . . ??
CustomerModelTest . . . ??
Gui::Widgets::WdgModels::CustomersTableModel
For a customer table . . . ??
Databases::Database
The **Database** class Master class for all database access . . . ??
Exceptions::DbException
For database exception : queries, db file, ... ??
Gui::Dialogs::DialogAddCustomer
Window to add or modify a Customer . . . ??
Gui::Widgets::Delegates::DoubleSpinBoxDelegate . . . ??
Exceptions::FileException
For file/acess file exception . . . ??
Generation . . . ??
Generator . . . ??
Utils::HierarchicalSystem
Create class which contains hierarchical system of FactDev . . . ??
Gui::Widgets::CheckFields::ICheckField
Interface to check fields validity . . . ??
Models::IModel
The IModel class . . . ??
Utils::ItemType
Item type model . . . ??
ItemTypeTest . . . ??
Utils::Log
For Simple management of log . . . ??
Gui::MainWindow
Main Window of the software . . . ??
Gui::Dialogs::MessageBox
Information window with message . . . ??
Parameters
Class for simple user parameters . . . ??
Mustache::PartialFileLoader . . . ??
Mustache::PartialMap . . . ??
Mustache::PartialResolver . . . ??
Gui::Widgets::Popup
Class for display popup quickly . . . ??

Models::Project
: Project linked to a Customer . . . ??
Gui::Widgets::Delegates::ProjectComboDelegate . . . ??
Gui::Widgets::WdgModels::ProjectContributoriesTableModel . . . ??
Databases::ProjectDatabase
Project table database . . . ??
ProjectDatabaseTest . . . ??
ProjectModelTest . . . ??
Gui::Widgets::WdgModels::ProjectsTableModel
For a Project table . . . ??
Gui::Widgets::ProjectsWidget
Actions on Project . . . ??
Mustache::QtVariantContext . . . ??
Models::Rate . . . ??
Databases::RateDatabase . . . ??
RateModelTest . . . ??
Gui::Widgets::RateWidget
Class for display Rate . . . ??
Mustache::Renderer . . . ??
Models::Search
The Search class . . . ??
searchTest . . . ??
Gui::Widgets::searchWidget
Class for search in database . . . ??
Utils::String
The Utils class . . . ??
StringTest . . . ??
Mustache::Tag . . . ??
TestAdder< T > . . . ??
testadder . . . ??
TestMustache . . . ??
TestRunner . . . ??
Gui::Widgets::Delegates::UnitComboDelegate . . . ??
Models::User
User of it application . . . ??
Databases::UserDatabase
Access to User data in the the table User of the **Database** . . . ??
UserDatabaseTest . . . ??
Gui::Dialogs::UserDataDialog
Window to fill user data . . . ??
UserModelTest . . . ??

Chapter 7

Namespace Documentation

7.1 Databases Namespace Reference

Class for Database, contains queries.

Classes

- class BillingDatabase

  *The **BillingDatabase** class Billing (or Quote) table database.*
- class ContributoryDatabase

  *The **ContributoryDatabase** class Contributory (or Quote) table database.*
- class CustomerDatabase

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

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

  The ProjectDatabase class Project table database.
- class RateDatabase
- class UserDatabase

  *The UserDatabase class Access to User data in the the table User of the **Database***

7.1.1 Detailed Description

Class for Database, contains queries.

7.2 Gui Namespace Reference

Namespaces

- Widgets

  Contains Widgets classes.

Classes

- class MainWindow

  The MainWindow class Main Window of the software.

7.2.1 Detailed Description

Classes for Graphical User Interface

7.3 Gui::Widgets Namespace Reference

Contains Widgets classes.

Classes

- class ComboBoxModelWidget

  The ComboBoxModelWidget class.
- class ContributoriesWidget
- class CustomerContextualMenu

  Display contextual menu on a customer.
- class CustomerDataWidget

  Class for display info of a customer.
- class Popup

  Class for display popup quickly.
- class ProjectsWidget

  The ProjectsWidget class Actions on Project.
- class RateWidget

  Class for display Rate.
- class searchWidget

  Class for search in database.

7.3.1 Detailed Description

Contains Widgets classes.

7.4 Models Namespace Reference

Models classes.

Classes

- class Billing

  The Billing class : Billing or Quote of a Customer.
- class ContributoriesList
- class Contributory

  The Contributory class.
- class Customer

  The Customer class Customer.
- class IModel

  The IModel class.
- class Project

  The Project class : Project linked to a Customer.
- class Rate

- class Search

  The Search class.
- class User

  *The User class **User** of it application.*

7.4.1 Detailed Description

Models classes.

Chapter 8

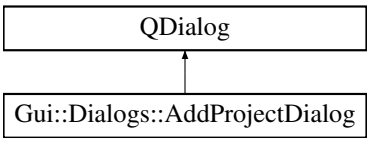
Class Documentation

8.1 Gui::Dialogs::AddProjectDialog Class Reference

The AddProjectDialog class Windows to add a new Project.

`#include <addprojectdialog.h>`

Inheritance diagram for Gui::Dialogs::AddProjectDialog:



Public Slots

- void checkFields ()

  AddProjectDialog::checkFields Check if fields are valid.

Public Member Functions

- AddProjectDialog (int id=0, QWidget ∗parent=0)

  AddProjectDialog::AddProjectDialog Construct a windows AddProjectDialog.
- AddProjectDialog (int noRowCustomer, int idProject=0, QWidget ∗parent=0)

  AddProjectDialogAddProjectDialog Construct a windows according an idCustomer and, optionnaly, an idProject
- void accept ()

  AddProjectDialog::accept Valid data inputed by user and add these data in Database.
- void reject ()

  AddProjectDialog::reject Cancel the operation and close the windows.
- void fillFields ()

  AddProjectDialog::fillFields Fill the differents fields of the current windows according the Project data existing As a project requires to be linked to a Customer, the Customer selection part may be disable.

8.1.1 Detailed Description

The AddProjectDialog class Windows to add a new Project.

Author

Florent Berbie

See Also

Project

8.1.2 Constructor & Destructor Documentation

8.1.2.1 Gui::Dialogs::AddProjectDialog::AddProjectDialog (int *id* = 0, QWidget * *parent* = 0) [explicit]

AddProjectDialog::AddProjectDialog Construct a windows AddProjectDialog.

Parameters

id	Project identity
parent	QWidget of the current windows

8.1.2.2 Gui::Dialogs::AddProjectDialog::AddProjectDialog (int *noRowCustomer*, int *idProject* = 0, QWidget * *parent* = 0) [explicit]

AddProjectDialogAddProjectDialog Construct a windows according an *idCustomer* and, optionnaly, an *idProject*

Parameters

noRowCustomer	Row number of the Customer
idProject	Project identify
parent	QWidget of the current windows

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

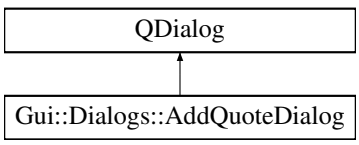
- /home/travis/build/FACT-Team/FactDev/src/gui/dialogs/addprojectdialog.h
- /home/travis/build/FACT-Team/FactDev/src/gui/dialogs/addprojectdialog.cpp

8.2 Gui::Dialogs::AddQuoteDialog Class Reference

The AddQuoteDialog class Window to add or modify a Quote.

`#include <addquotedialog.h>`

Inheritance diagram for Gui::Dialogs::AddQuoteDialog:



Public Slots

- void **updateBtn** (void)

Public Member Functions

- AddQuoteDialog (bool isBilling, int idCustomer=0, int id=0, QWidget ∗parent=0)

  AddQuoteDialog::AddQuoteDialog Construct a windows AddQuoteDialog.
- void fillFields ()

  AddQuoteDialog::Fill line edits with the data of the quote.
- int getNumber ()

  AddQuoteDialog::getNumber return the number of bill or quote.
- void accept ()

  AddQuoteDialog::accept Valid data inputed by user and add these data in Database.
- void reject ()

  AddQuoteDialog::reject Cancel the operation and close the windows.

8.2.1 Detailed Description

The AddQuoteDialog class Window to add or modify a Quote.

Author

Cédric Rohaut

8.2.2 Constructor & Destructor Documentation

8.2.2.1 Gui::Dialogs::AddQuoteDialog::AddQuoteDialog (bool *isBilling*, int *idCustomer* = 0, int *id* = 0, QWidget ∗ *parent* = 0) [explicit]

AddQuoteDialog::AddQuoteDialog Construct a windows AddQuoteDialog.

Parameters

parent	QWidget of the current windows

8.2.3 Member Function Documentation

8.2.3.1 int Gui::Dialogs::AddQuoteDialog::getNumber ( )

AddQuoteDialog::getNumber return the number of bill or quote.

Returns

int

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

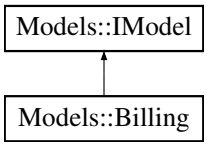
- /home/travis/build/FACT-Team/FactDev/src/gui/dialogs/addquotedialog.h
- /home/travis/build/FACT-Team/FactDev/src/gui/dialogs/addquotedialog.cpp

8.3 Models::Billing Class Reference

The Billing class : Billing or Quote of a Customer.

`#include <billing.h>`

Inheritance diagram for Models::Billing:



Public Member Functions

- Billing ()

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

  Billing::Billing. Construct a Billing or quote.
- ~Billing ()

  destruct a billing object
- 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.
- QVariantHash getDataMap ()

  getDataMap Get all data of model with a HashMap key/value
- void generateTex ()

  generateTex Generate a .tex file for the billing
- ContributoriesList & getContributories ()

  *Billing::getContributories. Return a map of **Contributory** for each **Project** of the **Billing***
- void addContributory (Contributory &c)

  addContributories Add a new contributory for project p
- double **getSumRate** ()
- double **getSumQuantity** ()
- QString getTitle () const

  *Billing::getTitle. return title of **Billing***
- void setTitle (const QString &getTitle)

  *Billing::setTitle. Modify the title of **Billing***
- QString getDescription () const

  *Billing::getDescription. return description of **Billing***
- void setDescription (const QString &getDescription)

  *Billing::setDescription. Modify the description 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***

- bool operator== (const Billing &b)

  Billing::operator == define the operator "==" to compare two billings and to see if they are the same.
- bool operator!= (const Billing &b)

  *Billing::operator != defines the operator "!=" to compare two **Billing** and to see if they are different.*
- void **setContributories** (const ContributoriesList &contributories)
- bool operator< (const Billing &b) const

  *Billing::operator < defines the operator "< to compare two **Billing** and to see if the fisrt is anterior to the second.*

Additional Inherited Members

8.3.1 Detailed Description

The Billing class : Billing or Quote of a Customer.

Author

Cédric Rohaut for the quote part

8.3.2 Constructor & Destructor Documentation

8.3.2.1 Models::Billing::Billing (int *id*)

Billing::Billing. Construct a Billing or quote.

Parameters

int	id

8.3.3 Member Function Documentation

8.3.3.1 void Models::Billing::addContributory (Contributory & *c*)

addContributories Add a new contributory for project p

Parameters

p	The Project who contain Contributory
c	The new Contributory

8.3.3.2 ContributoriesList & Models::Billing::getContributories ( )

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

Returns

QMap<Project, QList<Contributory>>

8.3.3.3 QVariantHash Models::Billing::getDataMap ( ) [virtual]

getDataMap Get all data of model with a HashMap key/value

Returns

Model's data

Implements Models::IModel.

8.3.3.4 QDate Models::Billing::getDate ( ) const

Billing::getDate. return date of the **Billing**

Returns

date of Billing

8.3.3.5 QString Models::Billing::getDescription ( ) const

Billing::getDescription. return description of **Billing**

Returns

description of Billing

8.3.3.6 int Models::Billing::getNumber ( ) const

Billing::getNumber. Return number of the **Billing**.

Returns

_number of Billing

8.3.3.7 QString Models::Billing::getTitle ( ) const

Billing::getTitle. return title of **Billing**

Returns

title of Billing

8.3.3.8 void Models::Billing::hydrat (int *getId*) [virtual]

Billing::hydrat. Update of the Billing which is specified by *getId*

Parameters

getId	

Implements Models::IModel.

8.3.3.9 bool Models::Billing::isBilling ( ) const

Billing::isBilling. Return if it's a billing or a quote.

Returns

if it's billing or a quote

8.3.3.10 bool Models::Billing::operator!= (const Billing & *b*)

Billing::operator != defines the operator "!=" to compare two **Billing** and to see if they are different.

Parameters

b	the **Billing** to compare with the current **Billing**

Returns

true if the **Billing** are different else false

8.3.3.11 bool Models::Billing::operator< (const Billing & *b*) const

Billing::operator < defines the operator "< to compare two **Billing** and to see if the fisrt is anterior to the second.

Parameters

b	the **Billing** to compare with the current **Billing**

Returns

true if the **Billing** are different else false

8.3.3.12 bool Models::Billing::operator== (const Billing & *b*)

Billing::operator == define the operator "==" to compare two billings and to see if they are the same.

Parameters

b	the **Billing** to compare with the current **Billing**

Returns

true if they are the same billings else false

8.3.3.13 void Models::Billing::setDate (const QDate & *getDate*)

Billing::setDate. Modify *date* of the **Billing**

Parameters

getDate	the new date of the Billing

8.3.3.14 void Models::Billing::setDescription (const QString & *getDescription*)

Billing::setDescription. Modify the description of **Billing**

Parameters

getDescription	Modify the description with *getDescription*

8.3.3.15 void Models::Billing::setIsBilling (bool *isBilling*)

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

Parameters

isBilling	

8.3.3.16 void Models::Billing::setNumber (int *getNumber*)

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

Parameters

getNumber	the new number of the Billing

8.3.3.17 void Models::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:

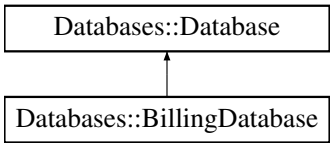
- /home/travis/build/FACT-Team/FactDev/src/models/billing.h
- /home/travis/build/FACT-Team/FactDev/src/models/billing.cpp

8.4 Databases::BillingDatabase Class Reference

The **BillingDatabase** class Billing (or Quote) table database.

`#include <billingdatabase.h>`

Inheritance diagram for Databases::BillingDatabase:



Public Member Functions

- Models::Billing * getBilling (const int pId)

  BillingDatabase::getCustomer get informations about the billing identified by pId
- WdgModels::BillingsTableModel * getBillingsTable (const int idProject) throw (DbException*)

  BillingDatabase::getBillingsTable Return an item model of billings for QTableView.
- int addBilling (const Models::Billing &)

  BillingDatabase::addBilling Add the billing pBilling to the database.
- void updateBilling (const Models::Billing &)

  BillingDatabase::updateCustomer Update informations about the billing pCustomer
- void removeBilling (const int pId)

  BillingDatabase::removeCustomer Remove the billing with the id pId
- void addBillingProject (const int idProject, const int idBilling, const int idContributory)

  BillingDatabase::addBillingProject Link a project, a billing and a contributory in the table BillingProject.
- void removeBillingProject (const int idProject, const int idBilling, const int idContributory)

BillingDatabase::removeBillingProject remove a link between a project, a billing and a contributory in the table BillingProject.

- int getMaxBillingNumber ()

  getMaxBillingNumber Get the last number of a billing

- int getMaxQuoteNumber ()

  getMaxQuoteNumber Get the last number of a quote

- QSharedPointer< Models::Billing > getBilling (QSqlQuery &q)

  BillingDatabase::getBilling Add the element of the q request and return their.

- QMap< Project *, Billing * > getAllBillingsOfProject ()

  BillingDatabase::getAllBillingsOfProject Return a map with the project id as key linked to the billing.

Static Public Member Functions

- static BillingDatabase * instance () throw (DbException*)

  BillingDatabase::getInstance Return an instance of ***BillingDatabase***

Additional Inherited Members

8.4.1 Detailed Description

The **BillingDatabase** class Billing (or Quote) table database.

Author

Cédric Rohaut

See Also

Database
Billing/Quote

8.4.2 Member Function Documentation

8.4.2.1 int Databases::BillingDatabase::addBilling (const Models::Billing & *pBilling*)

BillingDatabase::addBilling Add the billing *pBilling* to the database.

Returns

billing id

8.4.2.2 void Databases::BillingDatabase::addBillingProject (const int *idProject*, const int *idBilling*, const int *idContributory*)

BillingDatabase::addBillingProject Link a project, a billing and a contributory in the table BillingProject.

Parameters

idProject	Project id
idBilling	Billing id

idContributory	Contributory id

8.4.2.3 QMap< Project *, Billing * > Databases::BillingDatabase::getAllBillingsOfProject ( )

BillingDatabase::getAllBillingsOfProject Return a map with the project id as key linked to the billing.

Returns

Map with projects and Billing

8.4.2.4 Models::Billing * Databases::BillingDatabase::getBilling (const int *pId*)

BillingDatabase::getCustomer get informations about the billing identified by *pId*

Parameters

pId	billing id

Returns

the Billing

8.4.2.5 QSharedPointer< Billing > Databases::BillingDatabase::getBilling (QSqlQuery & *q*)

BillingDatabase::getBilling Add the element of the *q* request and return their.

Parameters

q	SQL request

Returns

a billing formed according to QSharedPointer

8.4.2.6 WdgModels::BillingsTableModel * Databases::BillingDatabase::getBillingsTable (const int *idProject*) throw DbException *)

BillingDatabase::getBillingsTable Return an item model of billings for QTableView.

Parameters

pId	the project id of the billings returned

Exceptions

DbException	

Returns

QStandardItemModel an item model for QTableView

8.4.2.7 int Databases::BillingDatabase::getMaxBillingNumber ( )

getMaxBillingNumber Get the last number of a billing

Returns

The max number

8.4.2.8 int Databases::BillingDatabase::getMaxQuoteNumber ( )

getMaxQuoteNumber Get the last number of a quote

Returns

The last number

8.4.2.9 BillingDatabase * Databases::BillingDatabase::instance ( ) throw DbException *) [static]

BillingDatabase::getInstance Return an instance of **BillingDatabase**

See Also

DbException

Returns

Instance of BillingDatabase

8.4.2.10 void Databases::BillingDatabase::removeBilling (const int *pId*)

BillingDatabase::removeCustomer Remove the billing with the id *pId*

Parameters

pId	billing id

8.4.2.11 void Databases::BillingDatabase::removeBillingProject (const int *idProject*, const int *idBilling*, const int *idContributory*)

BillingDatabase::removeBillingProject remove a link between a project, a billing and a contributory in the table BillingProject.

Parameters

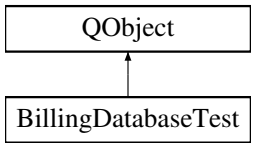
idProject	Project id
idBilling	Billing id
idContributory	Contributory id

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

- /home/travis/build/FACT-Team/FactDev/src/database/billingdatabase.h
- /home/travis/build/FACT-Team/FactDev/src/database/billingdatabase.cpp

8.5 BillingDatabaseTest Class Reference

Inheritance diagram for BillingDatabaseTest:

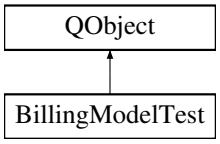


The documentation for this class was generated from the following files:

- /home/travis/build/FACT-Team/FactDev/tests/database/billingdatabasetest.h
- /home/travis/build/FACT-Team/FactDev/tests/database/billingdatabasetest.cpp

8.6 BillingModelTest Class Reference

Inheritance diagram for BillingModelTest:



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

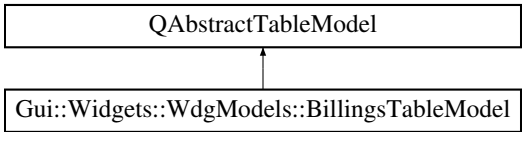
- /home/travis/build/FACT-Team/FactDev/tests/models/billingmodeltest.h
- /home/travis/build/FACT-Team/FactDev/tests/models/billingmodeltest.cpp

8.7 Gui::Widgets::WdgModels::BillingsTableModel Class Reference

The BillingsTableModel class for a Billing table.

`#include <billingstablemodel.h>`

Inheritance diagram for Gui::Widgets::WdgModels::BillingsTableModel:



Public Member Functions

- BillingsTableModel ()

  BillingsTableModel::BillingsTableModel Construct a BillingsTableModel.
- int rowCount (const QModelIndex &) const

  BillingsTableModel::rowCount Number of billings row.

- int columnCount (const QModelIndex &) const

  BillingsTableModel::columnCount Number of column of a Billing.
- QVariant data (const QModelIndex &index, int role=Qt::DisplayRole) const

  BillingsTableModel::data Obtains data of a specify cell.
- QVariant headerData (int section, Qt::Orientation orientation, int role=Qt::DisplayRole) const

  BillingsTableModel::headerData Obtains header title of table.
- bool setData (const QModelIndex &index, const QVariant &value, int role=Qt::EditRole)

  BillingsTableModel::setData Change data of a cell.
- void append (const Billing &billing)

  BillingsTableModel::append Add a new line in table.
- void remove (const int i)

  BillingsTableModel::remove Remove a line.
- Qt::ItemFlags flags (const QModelIndex &index) const

  BillingsTableModel::flags Differents table flags.
- int count ()

  BillingsTableModel::count Number of billings in table.
- QList< Billing > getBillings () const

  BillingsTableModel::getbillings Return the list of billings.

8.7.1 Detailed Description

The BillingsTableModel class for a Billing table.

Author

Florent Berbie

See Also

Billing

8.7.2 Constructor & Destructor Documentation

8.7.2.1 Gui::Widgets::WdgModels::BillingsTableModel::BillingsTableModel ( )

BillingsTableModel::BillingsTableModel Construct a BillingsTableModel.

Parameters

parent	Parent widget

8.7.3 Member Function Documentation

8.7.3.1 void Gui::Widgets::WdgModels::BillingsTableModel::append (const Billing & *billing*)

BillingsTableModel::append Add a new line in table.

Parameters

Billing	The new Billing

8.7.3.2 int Gui::Widgets::WdgModels::BillingsTableModel::columnCount (const QModelIndex &) const

BillingsTableModel::columnCount Number of column of a Billing.

Returns

The number of column

8.7.3.3 int Gui::Widgets::WdgModels::BillingsTableModel::count ( )

BillingsTableModel::count Number of billings in table.

Returns

The number of billings

8.7.3.4 QVariant Gui::Widgets::WdgModels::BillingsTableModel::data (const QModelIndex & *index*, int *role* = `Qt::DisplayRole`) const

BillingsTableModel::data Obtains data of a specify cell.

Parameters

index	The cell who we want data
role	The role of set

Returns

The data of cell

8.7.3.5 Qt::ItemFlags Gui::Widgets::WdgModels::BillingsTableModel::flags (const QModelIndex & *index*) const

BillingsTableModel::flags Differents table flags.

Parameters

index	The cell who we want to know flags

Returns

Flags

8.7.3.6 QList< Billing > Gui::Widgets::WdgModels::BillingsTableModel::getBillings ( ) const

BillingsTableModel::getbillings Return the list of billings.

Returns

list of billings

8.7.3.7 QVariant Gui::Widgets::WdgModels::BillingsTableModel::headerData (int *section*, Qt::Orientation *orientation*, int *role* = `Qt::DisplayRole`) const

BillingsTableModel::headerData Obtains header title of table.

Parameters

section	The number of column
orientation	The table orientation
role	

Returns

The Title header of column

8.7.3.8 void Gui::Widgets::WdgModels::BillingsTableModel::remove (const int *i*)

BillingsTableModel::remove Remove a line.

Parameters

i	The number of line to remove

8.7.3.9 int Gui::Widgets::WdgModels::BillingsTableModel::rowCount (const QModelIndex &) const

BillingsTableModel::rowCount Number of billings row.

Returns

The number of billings

8.7.3.10 bool Gui::Widgets::WdgModels::BillingsTableModel::setData (const QModelIndex & *index*, const QVariant & *value*, int *role* = `Qt::EditRole`)

BillingsTableModel::setData Change data of a cell.

Parameters

index	The cell to change data
value	The new value
role	The role of cell

Returns

True if we could edit

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

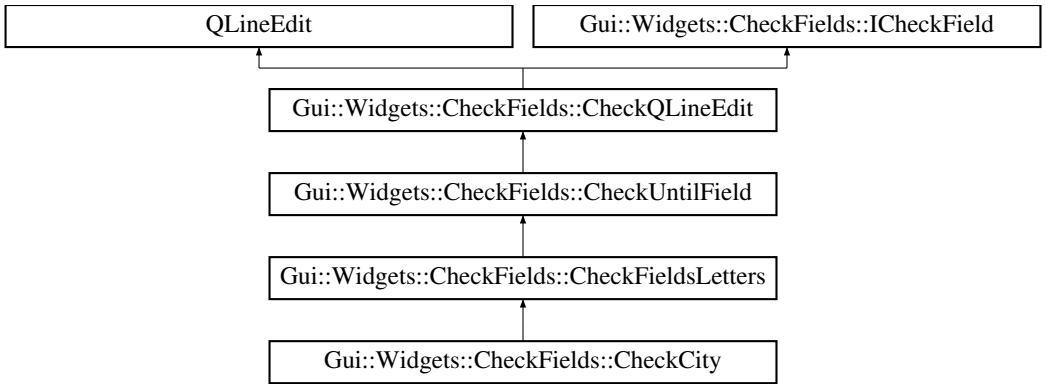
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/widgetsmodels/billingstablemodel.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/widgetsmodels/billingstablemodel.cpp

8.8 Gui::Widgets::CheckFields::CheckCity Class Reference

The CheckCity class Line Edit of City with a check icon.

`#include <checkcity.h>`

Inheritance diagram for Gui::Widgets::CheckFields::CheckCity:



Public Member Functions

- CheckCity (QWidget *w=0, QPushButton *btn=0)

  CheckCity::CheckCity Construct a CheckCity.

Additional Inherited Members

8.8.1 Detailed Description

The CheckCity class Line Edit of City with a check icon.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 Gui::Widgets::CheckFields::CheckCity::CheckCity (QWidget * *w* = 0, QPushButton * *btn* = 0)

CheckCity::CheckCity Construct a CheckCity.

Parameters

w	QWidget linked to **CheckCity**

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

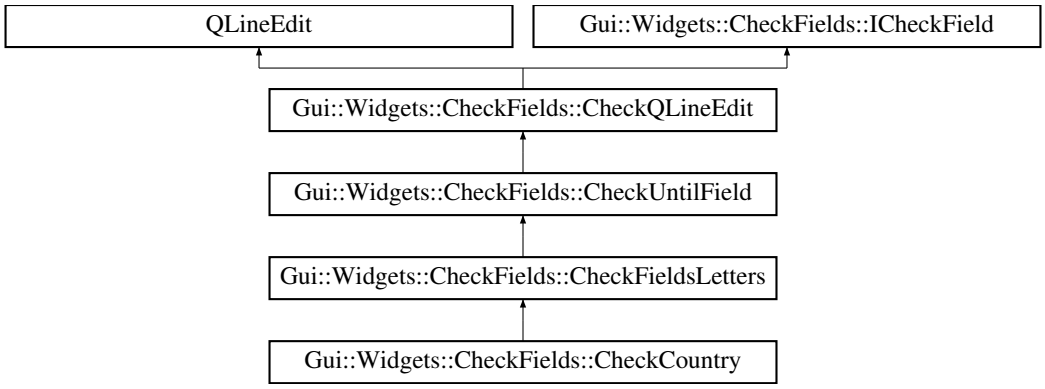
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkcity.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkcity.cpp

8.9 Gui::Widgets::CheckFields::CheckCountry Class Reference

CheckCountry::CheckCountry Line Edit of country with a check icon.

`#include <checkcountry.h>`

Inheritance diagram for Gui::Widgets::CheckFields::CheckCountry:



Public Member Functions

- CheckCountry (QWidget ∗w=0, QPushButton ∗btn=0)

  CheckEmail::CheckCountry Construct a CheckCountry.

Additional Inherited Members

8.9.1 Detailed Description

CheckCountry::CheckCountry Line Edit of country with a check icon.

8.9.2 Constructor & Destructor Documentation

8.9.2.1 Gui::Widgets::CheckFields::CheckCountry::CheckCountry (QWidget ∗ *w* = 0, QPushButton ∗ *btn* = 0)

CheckEmail::CheckCountry Construct a CheckCountry.

Parameters

w	QWidget linked to **CheckCountry**

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

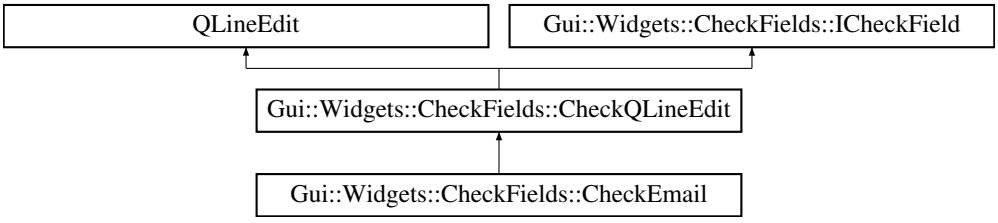
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkcountry.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkcountry.cpp

8.10 Gui::Widgets::CheckFields::CheckEmail Class Reference

The CheckEmail class Line Edit of email with a check icon.

`#include <checkemail.h>`

Inheritance diagram for Gui::Widgets::CheckFields::CheckEmail:



Public Member Functions

- CheckEmail (QWidget *w=0, QPushButton *btn=0)

  CheckEmail::CheckEmail Construct a CheckMail.

- bool check (const QString text)

  CheckEmail::check Check if the field email is valid. To be valid, an email address should be under this form: `me@me.-xx` *An email address need:*

Additional Inherited Members

8.10.1 Detailed Description

The CheckEmail class Line Edit of email with a check icon.

8.10.2 Constructor & Destructor Documentation

8.10.2.1 Gui::Widgets::CheckFields::CheckEmail::CheckEmail (QWidget * *w* = `0`, QPushButton * *btn* = `0`)

CheckEmail::CheckEmail Construct a CheckMail.

Parameters

w	QWidget linked to **CheckEmail**

8.10.3 Member Function Documentation

8.10.3.1 bool Gui::Widgets::CheckFields::CheckEmail::check (const QString *text*) `[virtual]`

CheckEmail::check Check if the field email is valid. To be valid, an email address should be under this form: `me@me.xx` An email address need:

- 1 character [A-Z] or [a-z] minimum before the character </i>
- *the character '@'*
- *1 character [A-Z] or [a-z] after the character</i>minimum and before the character .*
- *1 character [A-Z] or [a-z] minimum afer the character . Return TRUE if email address is valid, else FALSE*

  Parameters

text	

  Returns

  boolean

Implements Gui::Widgets::CheckFields::ICheckField.

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

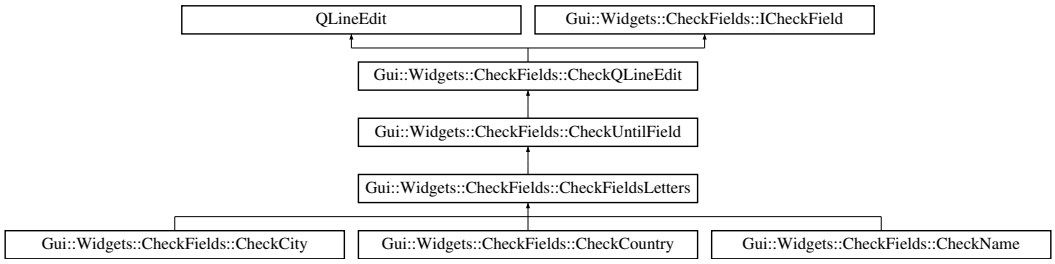
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkemail.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkemail.cpp

8.11 Gui::Widgets::CheckFields::CheckFieldsLetters Class Reference

The CheckFieldsLetters class Field with only letters (no numbers)

`#include <checkfieldsletters.h>`

Inheritance diagram for Gui::Widgets::CheckFields::CheckFieldsLetters:



Public Member Functions

- CheckFieldsLetters (QWidget ∗w=0, QPushButton ∗btn=0)

  CheckFieldsLetters::CheckFieldsLetters Construct a CheckFieldsLetters.
- bool check (QString text)

  CheckName::check Check if the field contains only letters.

Additional Inherited Members

8.11.1 Detailed Description

The CheckFieldsLetters class Field with only letters (no numbers)

8.11.2 Constructor & Destructor Documentation

8.11.2.1 Gui::Widgets::CheckFields::CheckFieldsLetters::CheckFieldsLetters (QWidget ∗ *w* = `0`, QPushButton ∗ *btn* = `0`)

CheckFieldsLetters::CheckFieldsLetters Construct a CheckFieldsLetters.

Parameters

w	QWidget linked to **CheckFieldsLetters**

8.11.3 Member Function Documentation

8.11.3.1 bool Gui::Widgets::CheckFields::CheckFieldsLetters::check (QString *text*) `[virtual]`

CheckName::check Check if the field contains only letters.

Parameters

text	

Returns

boolean

Implements Gui::Widgets::CheckFields::ICheckField.

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

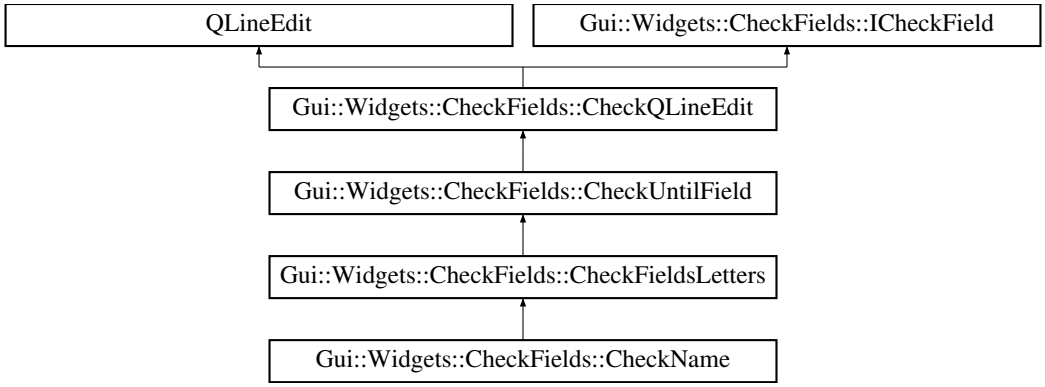
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkfieldsletters.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkfieldsletters.cpp

8.12 Gui::Widgets::CheckFields::CheckName Class Reference

The CheckName class Line edit of name with a check icon.

`#include <checkname.h>`

Inheritance diagram for Gui::Widgets::CheckFields::CheckName:



Public Member Functions

- CheckName (QWidget ∗w=0, QPushButton ∗btn=0)
  CheckName::CheckName Construct a CheckName.

Additional Inherited Members

8.12.1 Detailed Description

The CheckName class Line edit of name with a check icon.

8.12.2 Constructor & Destructor Documentation

8.12.2.1 Gui::Widgets::CheckFields::CheckName::CheckName (QWidget ∗ *w* = 0, QPushButton ∗ *btn* = 0)

CheckName::CheckName Construct a CheckName.

Parameters

w	QWidget linked to **CheckName**

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

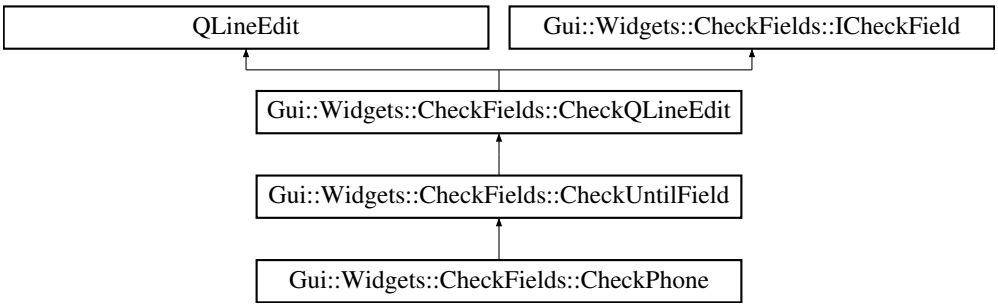
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkname.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkname.cpp

8.13 Gui::Widgets::CheckFields::CheckPhone Class Reference

The CheckPhone class Line Edit of Phone number with a check icon.

`#include <checkphone.h>`

Inheritance diagram for Gui::Widgets::CheckFields::CheckPhone:



Public Member Functions

- CheckPhone (QWidget ∗w=0, QPushButton ∗btn=0)

  CheckPhone::CheckPhone Construct a CheckPhone.
- bool check (QString text)

  CheckPhone::check Check if the field is valid. To be valid, a name should be composed of a character.
- QString getCountry () const

  CheckPhone::getCountry Return the country linked to current field.
- void setCountry (const QString &country)

  CheckPhone::setCountry Modify the country linked to field.

Additional Inherited Members

8.13.1 Detailed Description

The CheckPhone class Line Edit of Phone number with a check icon.

8.13.2 Constructor & Destructor Documentation

8.13.2.1 Gui::Widgets::CheckFields::CheckPhone::CheckPhone (QWidget ∗ *w* = 0, QPushButton ∗ *btn* = 0)

CheckPhone::CheckPhone Construct a CheckPhone.

Parameters

w	QWidget linked to **CheckPhone**

8.13.3 Member Function Documentation

8.13.3.1 bool Gui::Widgets::CheckFields::CheckPhone::check (QString *text*) [virtual]

CheckPhone::check Check if the field is valid. To be valid, a name should be composed of a character.

Parameters

text	

Returns

boolean

Implements Gui::Widgets::CheckFields::ICheckField.

8.13.3.2 QString Gui::Widgets::CheckFields::CheckPhone::getCountry ( ) const

CheckPhone::getCountry Return the country linked to current field.

Returns

8.13.3.3 void Gui::Widgets::CheckFields::CheckPhone::setCountry (const QString & *country*)

CheckPhone::setCountry Modify the *country* linked to field.

Parameters

country	New country

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

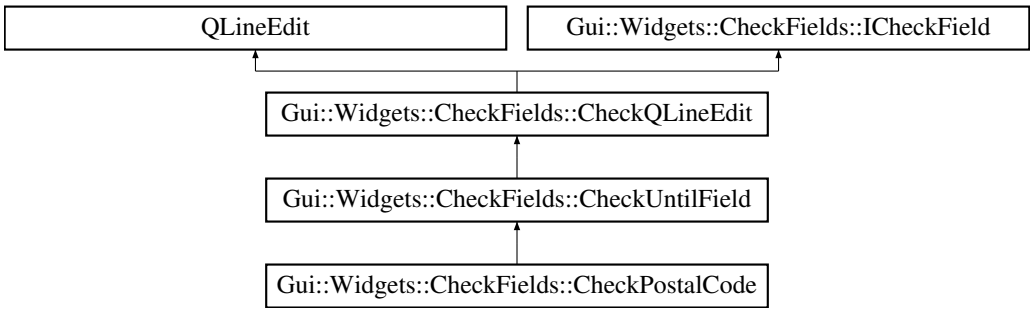
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkphone.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkphone.cpp

8.14 Gui::Widgets::CheckFields::CheckPostalCode Class Reference

The CheckPostalCode class Line Edit of postal code with a check icon.

`#include <checkpostalcode.h>`

Inheritance diagram for Gui::Widgets::CheckFields::CheckPostalCode:



Public Member Functions

- CheckPostalCode (QWidget *w=0, QPushButton *btn=0)

  CheckPostalCode::CheckPostalCode Construct a CheckPostalCode.
- bool check (QString text)

  CheckPostalCode::check Check if the field is valid. To be valid, a name should be composed of a character.
- QString getCountry () const

  CheckPostalCode::getCountry Return the country linked to current field.
- void setCountry (const QString &country)

  CheckPostalCode::setCountry Modify the country linked to field.

Additional Inherited Members

8.14.1 Detailed Description

The CheckPostalCode class Line Edit of postal code with a check icon.

8.14.2 Constructor & Destructor Documentation

8.14.2.1 Gui::Widgets::CheckFields::CheckPostalCode::CheckPostalCode (QWidget * *w* = 0, QPushButton * *btn* = 0)

CheckPostalCode::CheckPostalCode Construct a CheckPostalCode.

Parameters

w	QWidget linked to **CheckPostalCode**

8.14.3 Member Function Documentation

8.14.3.1 bool Gui::Widgets::CheckFields::CheckPostalCode::check (QString *text*) [virtual]

CheckPostalCode::check Check if the field is valid. To be valid, a name should be composed of a character.

Parameters

text	

Returns

boolean

Implements Gui::Widgets::CheckFields::ICheckField.

8.14.3.2 QString Gui::Widgets::CheckFields::CheckPostalCode::getCountry ( ) const

CheckPostalCode::getCountry Return the country linked to current field.

Returns

country

8.14.3.3 void Gui::Widgets::CheckFields::CheckPostalCode::setCountry (const QString & *country*)

CheckPostalCode::setCountry Modify the *country* linked to field.

Parameters

country	New country

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

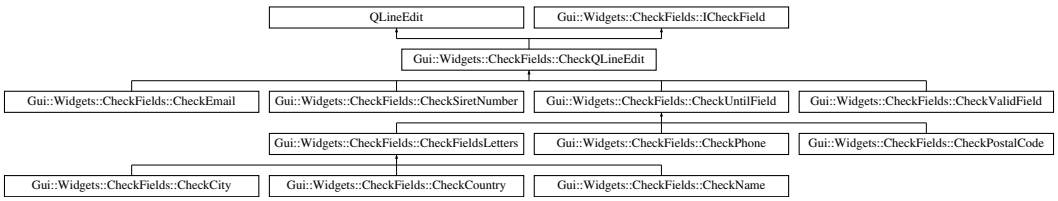
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkpostalcode.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkpostalcode.cpp

8.15 Gui::Widgets::CheckFields::CheckQLineEdit Class Reference

The CheckQLineEdit class LineEdit custom with a check of text inputed.

`#include <checkqlineedit.h>`

Inheritance diagram for Gui::Widgets::CheckFields::CheckQLineEdit:



Public Slots

- void fieldTextChanged (const QString &text)

  CheckQLineEdit::fieldTextChanged For each new characater inputed or removed, displays an icon to show if the field is valid or not.

Public Member Functions

- CheckQLineEdit (QWidget ∗parent=0, QPushButton ∗btn=0)

  CheckQLineEdit::CheckQLineEdit Construct a CheckQLineEdit.
- void displayCheckValidFieldIcon ()

  CheckQLineEdit::displayCheckValidFieldIcon Display a valid icon into the field.
- void displayCheckNoValidFieldIcon ()

  CheckQLineEdit::displayCheckNoValidFieldIcon Display a "no valid" icon into the field.
- QPushButton ∗ getBtnValid () const

  CheckQLineEdit::getBtnValid.
- void setBtnValid (QPushButton ∗getBtnValid)

CheckQLineEdit::setBtnValid.

- bool isValid ()

  isValid Return true if the current field if valid

8.15.1 Detailed Description

The CheckQLineEdit class LineEdit custom with a check of text inputed.

8.15.2 Constructor & Destructor Documentation

8.15.2.1 Gui::Widgets::CheckFields::CheckQLineEdit::CheckQLineEdit (QWidget * *parent* = 0, QPushButton * *btn* = 0) [explicit]

CheckQLineEdit::CheckQLineEdit Construct a CheckQLineEdit.

Parameters

parent	

8.15.3 Member Function Documentation

8.15.3.1 QPushButton * Gui::Widgets::CheckFields::CheckQLineEdit::getBtnValid ( ) const

CheckQLineEdit::getBtnValid.

Returns

a

8.15.3.2 bool Gui::Widgets::CheckFields::CheckQLineEdit::isValid ( )

isValid Return true if the current field if valid

Returns

boolean

8.15.3.3 void Gui::Widgets::CheckFields::CheckQLineEdit::setBtnValid (QPushButton * *getBtnValid*)

CheckQLineEdit::setBtnValid.

Parameters

getBtnValid	

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

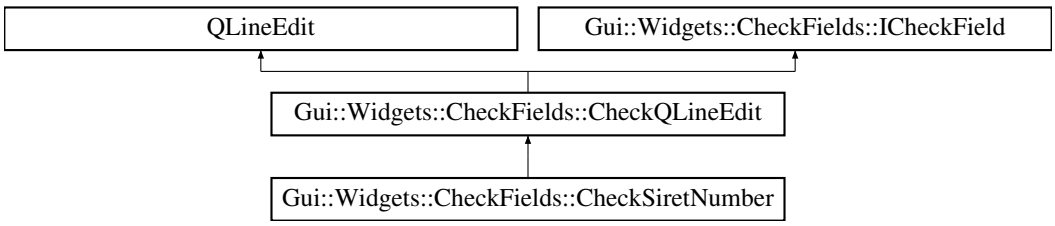
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkqlineedit.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkqlineedit.cpp

8.16 Gui::Widgets::CheckFields::CheckSiretNumber Class Reference

The CheckSiretNumber class Line Edit with a check icon.

`#include <checksiretnumber.h>`

Inheritance diagram for Gui::Widgets::CheckFields::CheckSiretNumber:



Public Member Functions

- CheckSiretNumber (QWidget ∗w=0, QPushButton ∗btn=0)

  CheckSiretNumber::CheckSiretNumber Construct a CheckSiretNumber.
- bool check (QString text)

  CheckSiretNumber::check Check if the field noSiret is valid. To be valid, a SIRET number should be composed of numbers.

Additional Inherited Members

8.16.1 Detailed Description

The CheckSiretNumber class Line Edit with a check icon.

8.16.2 Constructor & Destructor Documentation

8.16.2.1 Gui::Widgets::CheckFields::CheckSiretNumber::CheckSiretNumber (QWidget ∗ *w* = 0, QPushButton ∗ *btn* = 0)

CheckSiretNumber::CheckSiretNumber Construct a CheckSiretNumber.

Parameters

w	QWidget linked to **CheckSiretNumber**

8.16.3 Member Function Documentation

8.16.3.1 bool Gui::Widgets::CheckFields::CheckSiretNumber::check (QString *text*) [virtual]

CheckSiretNumber::check Check if the field noSiret is valid. To be valid, a SIRET number should be composed of numbers.

Parameters

text	

Returns

boolean

Implements Gui::Widgets::CheckFields::ICheckField.

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

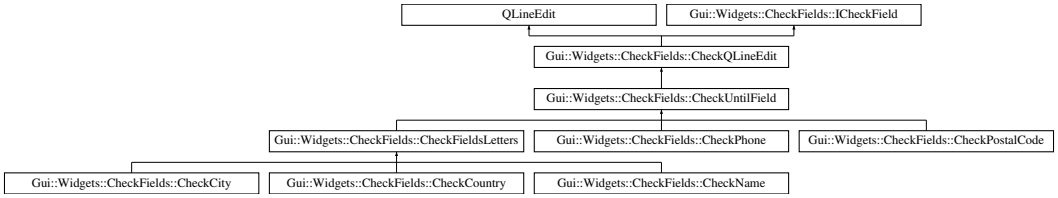
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checksiretnumber.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checksiretnumber.cpp

8.17 Gui::Widgets::CheckFields::CheckUntilField Class Reference

The CheckUntilField class.

`#include <checkuntilfield.h>`

Inheritance diagram for Gui::Widgets::CheckFields::CheckUntilField:



Public Member Functions

- CheckUntilField (QWidget *w=0, QPushButton *btn=0)

  CheckUntilField::CheckUntilField Construct a CheckUntilField.
- bool check (QString text)

  CheckUntilField::check Check if the field is valid. To be valid, a name should be composed of a character.

Additional Inherited Members

8.17.1 Detailed Description

The CheckUntilField class.

8.17.2 Constructor & Destructor Documentation

8.17.2.1 Gui::Widgets::CheckFields::CheckUntilField::CheckUntilField (QWidget * *w* = `0`, QPushButton * *btn* = `0`)

CheckUntilField::CheckUntilField Construct a CheckUntilField.

Parameters

w	QWidget linked to **CheckUntilField**

8.17.3 Member Function Documentation

8.17.3.1 bool Gui::Widgets::CheckFields::CheckUntilField::check (QString *text*) `[virtual]`

CheckUntilField::check Check if the field is valid. To be valid, a name should be composed of a character.

Parameters

text	

Returns

boolean

Implements Gui::Widgets::CheckFields::ICheckField.

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

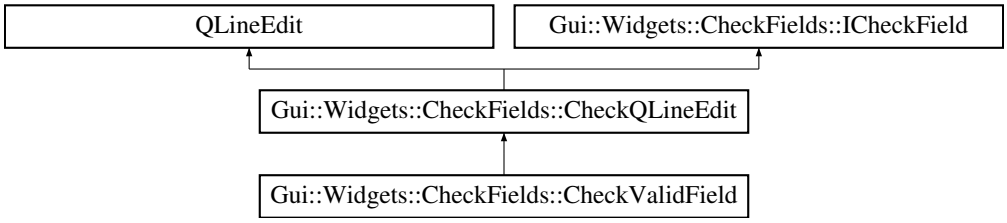
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkuntilfield.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkuntilfield.cpp

8.18 Gui::Widgets::CheckFields::CheckValidField Class Reference

The CheckValidField class Check field not required.

`#include <checkvalidfield.h>`

Inheritance diagram for Gui::Widgets::CheckFields::CheckValidField:



Public Member Functions

- CheckValidField (QWidget ∗w=0, QPushButton ∗btn=0)
  CheckValidField::CheckValidField.
- bool check (QString text)
  CheckValidField::check Return TRUE : the field is not required.

Additional Inherited Members

8.18.1 Detailed Description

The CheckValidField class Check field not required.

8.18.2 Constructor & Destructor Documentation

8.18.2.1 Gui::Widgets::CheckFields::CheckValidField::CheckValidField (QWidget ∗ *w* = 0, QPushButton ∗ *btn* = 0)

CheckValidField::CheckValidField.

Parameters

w	QWidget linked to **CheckValidField**

8.18.3 Member Function Documentation

8.18.3.1 bool Gui::Widgets::CheckFields::CheckValidField::check (QString *text*) [virtual]

CheckValidField::check Return TRUE : the field is not required.

Parameters

text	

Returns

boolean

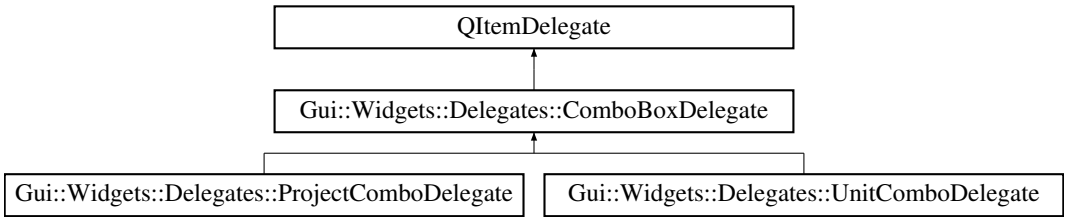
Implements Gui::Widgets::CheckFields::ICheckField.

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

- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkvalidfield.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkvalidfield.cpp

8.19 Gui::Widgets::Delegates::ComboBoxDelegate Class Reference

Inheritance diagram for Gui::Widgets::Delegates::ComboBoxDelegate:



Public Member Functions

- **ComboBoxDelegate** (QObject ∗parent=0)
- virtual QWidget ∗ **createEditor** (QWidget ∗parent, const QStyleOptionViewItem &option, const QModelIndex &index) const =0
- void **paint** (QPainter ∗painter, const QStyleOptionViewItem &option, const QModelIndex &index) const =0
- void **setEditorData** (QWidget ∗editor, const QModelIndex &index) const
- void **setModelData** (QWidget ∗editor, QAbstractItemModel ∗model, const QModelIndex &index) const
- void **updateEditorGeometry** (QWidget ∗editor, const QStyleOptionViewItem &option, const QModelIndex &index) const

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

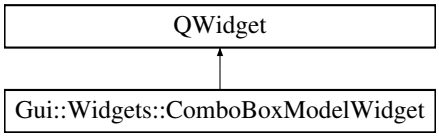
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/delegates/comboboxdelegate.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/delegates/comboboxdelegate.cpp

8.20 Gui::Widgets::ComboBoxModelWidget Class Reference

The ComboBoxModelWidget class.

`#include <comboboxmodelwidget.h>`

Inheritance diagram for Gui::Widgets::ComboBoxModelWidget:



Public Member Functions

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

8.20.1 Detailed Description

The ComboBoxModelWidget class.

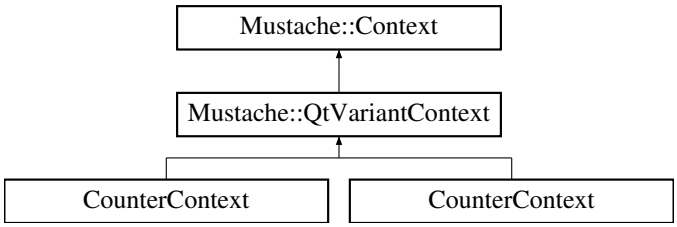
The documentation for this class was generated from the following files:

- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/comboboxmodelwidget.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/comboboxmodelwidget.cpp

8.21 Mustache::Context Class Reference

`#include <mustache.h>`

Inheritance diagram for Mustache::Context:



Public Member Functions

- Context (PartialResolver *resolver=0)
- virtual QString stringValue (const QString &key) const =0
- virtual bool isFalse (const QString &key) const =0
- virtual int listCount (const QString &key) const =0
- virtual void push (const QString &key, int index=-1)=0
- virtual void pop ()=0
- QString partialValue (const QString &key) const
- PartialResolver * partialResolver () const
- virtual bool canEval (const QString &key) const
- virtual QString eval (const QString &key, const QString &_template, Renderer *renderer)

8.21.1 Detailed Description

Context is an interface that Mustache::Renderer::render() uses to fetch substitutions for template tags.

8.21.2 Constructor & Destructor Documentation

8.21.2.1 Context::Context (PartialResolver * *resolver* = 0) [explicit]

Create a context. `resolver` is used to fetch the expansions for any {{>partial}} tags which appear in a template.

8.21.3 Member Function Documentation

8.21.3.1 bool Context::canEval (const QString & *key*) const [virtual]

Returns true if eval() should be used to render section tags using `key`. If canEval() returns true for a key, the renderer will pass the literal, unrendered block of text for the section to eval() and replace the section with the result.

canEval() and eval() are equivalents for callable objects (eg. lambdas) in other Mustache implementations.

The default implementation always returns false.

Reimplemented in CounterContext, CounterContext, and Mustache::QtVariantContext.

8.21.3.2 QString Context::eval (const QString & *key*, const QString & *_template*, Renderer * *renderer*) [virtual]

Callback used to render a template section with the given `key`. `renderer` will substitute the original section tag with the result of eval().

The default implementation returns an empty string.

Reimplemented in CounterContext, CounterContext, and Mustache::QtVariantContext.

8.21.3.3 virtual bool Mustache::Context::isFalse (const QString & *key*) const [pure virtual]

Returns true if the value for `key` is 'false' or an empty list. 'False' values typically include empty strings, the boolean value false etc.

When processing a section Mustache tag, the section is not rendered if the key is false, or for an inverted section tag, the section is only rendered if the key is false.

Implemented in Mustache::QtVariantContext.

8.21.3.4 virtual int Mustache::Context::listCount (const QString & *key*) const [pure virtual]

Returns the number of items in the list value for `key` or 0 if the value for `key` is not a list.

Implemented in Mustache::QtVariantContext.

8.21.3.5 PartialResolver * Context::partialResolver ( ) const

Returns the partial resolver passed to the constructor.

8.21.3.6 QString Context::partialValue (const QString & *key*) const

Returns the partial template for a given `key`.

8.21.3.7 virtual void Mustache::Context::pop ( ) [pure virtual]

Exit the current context.

Implemented in Mustache::QtVariantContext.

8.21.3.8 virtual void Mustache::Context::push (const QString & *key*, int *index* = -1) [pure virtual]

Set the current context to the value for `key`. If index is >= 0, set the current context to the `index'th` value in the list value for `key`.

Implemented in Mustache::QtVariantContext.

8.21.3.9 virtual QString Mustache::Context::stringValue (const QString & *key*) const [pure virtual]

Returns a string representation of the value for `key` in the current context. This is used to replace a Mustache value tag.

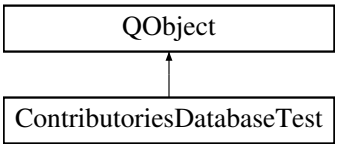
Implemented in CounterContext, CounterContext, and Mustache::QtVariantContext.

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

- /home/travis/build/FACT-Team/FactDev/src/libs/qt-mustache/src/mustache.h
- /home/travis/build/FACT-Team/FactDev/src/libs/qt-mustache/src/mustache.cpp

8.22 ContributoriesDatabaseTest Class Reference

Inheritance diagram for ContributoriesDatabaseTest:

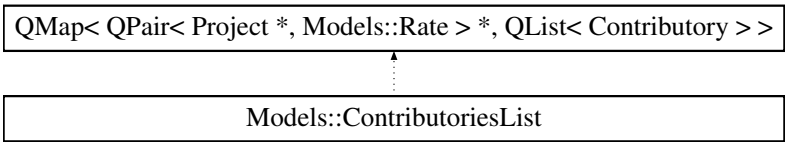


The documentation for this class was generated from the following files:

- /home/travis/build/FACT-Team/FactDev/tests/database/contributoriesdatabasetest.h
- /home/travis/build/FACT-Team/FactDev/tests/database/contributoriesdatabasetest.cpp

8.23 Models::ContributoriesList Class Reference

Inheritance diagram for Models::ContributoriesList:



Public Member Functions

- virtual void commit ()

  commit Update or insert data into the database
- void **addContributory** (Models::Contributory &contributory)
- void **addProject** (Project *p, Models::Rate rate)
- QList< Contributory > & **getContributories** (Project *p)
- int **getIdBilling** () const
- void **setIdBilling** (int getIdBilling)
- bool **getInsert** () const
- void **setInsert** (bool getInsert)
- int **getNbProjects** ()
- double **getSumRate** ()
- double **getSumQuantity** ()
- QSharedPointer< Customer > **getCustomer** ()
- QList< Project * > **getProjects** ()
- QList< Contributory > * **getAllContributories** ()
- Models::Rate **getRate** (Models::Project *project)
- QVariantList **getDataMap** ()

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

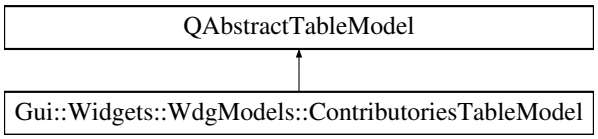
- /home/travis/build/FACT-Team/FactDev/src/models/contributorieslist.h
- /home/travis/build/FACT-Team/FactDev/src/models/contributorieslist.cpp

8.24 Gui::Widgets::WdgModels::ContributoriesTableModel Class Reference

The ContributoriesTableModel class for a custom table for contributories widget.

`#include <contributoriestablemodel.h>`

Inheritance diagram for Gui::Widgets::WdgModels::ContributoriesTableModel:



Public Member Functions

- ContributoriesTableModel (QObject *parent=0)

  ContributoriesTableModel Construct a ContributoriesTableModel.
- int rowCount (const QModelIndex &) const

  rowCount Number of contributories row
- int columnCount (const QModelIndex &) const

  columnCount Number of column of a contributory
- QVariant data (const QModelIndex &index, int role) const

  data Obtains data of a specify cell
- QVariant headerData (int section, Qt::Orientation orientation, int role) const

  headerData Obtains header title of table
- bool setData (const QModelIndex &index, const QVariant &value, int role=Qt::EditRole)

  setData Change data of a cell

- void append (const Contributory &contributory)

  append Add a new line in table
- void remove (const int i)

  remove Remove a line
- Qt::ItemFlags flags (const QModelIndex &index) const

  flags Differents table flags
- QList< Contributory > getContributories ()

  getContributories Get all contributories of table
- int count ()

  count Number of contributories in table
- double **getSumQuantity** () const
- void **clear** ()

8.24.1 Detailed Description

The ContributoriesTableModel class for a custom table for contributories widget.

Author

Antoine de Roquemaurel

See Also

Contributory

8.24.2 Constructor & Destructor Documentation

8.24.2.1 Gui::Widgets::WdgModels::ContributoriesTableModel::ContributoriesTableModel (QObject * *parent* = 0)

ContributoriesTableModel Construct a ContributoriesTableModel.

Parameters

parent	Parent widget

8.24.3 Member Function Documentation

8.24.3.1 void Gui::Widgets::WdgModels::ContributoriesTableModel::append (const Contributory & *contributory*)

append Add a new line in table

Parameters

contributory	The new contributory

8.24.3.2 int Gui::Widgets::WdgModels::ContributoriesTableModel::columnCount (const QModelIndex &) const

columnCount Number of column of a contributory

Returns

The number of column

8.24.3.3 int Gui::Widgets::WdgModels::ContributoriesTableModel::count ( )

count Number of contributories in table

Returns

The number of contributories

8.24.3.4 QVariant Gui::Widgets::WdgModels::ContributoriesTableModel::data (const QModelIndex & *index,* int *role*) const

data Obtains data of a specify cell

Parameters

index	The cell who we want data
role	The role of set

Returns

The data of cell

8.24.3.5 Qt::ItemFlags Gui::Widgets::WdgModels::ContributoriesTableModel::flags (const QModelIndex & *index*) const

flags Differents table flags

Parameters

index	The cell who we want to know flags

Returns

Flags

8.24.3.6 QList< Contributory > Gui::Widgets::WdgModels::ContributoriesTableModel::getContributories ( )

getContributories Get all contributories of table

Returns

The contributory list

8.24.3.7 QVariant Gui::Widgets::WdgModels::ContributoriesTableModel::headerData (int *section,* Qt::Orientation *orientation,* int *role*) const

headerData Obtains header title of table

Parameters

section	The number of column
orientation	The table orientation
role	

Returns

The Title header of column

8.24.3.8 void Gui::Widgets::WdgModels::ContributoriesTableModel::remove (const int *i*)

remove Remove a line

Parameters

i	The number of line to remove

8.24.3.9 int Gui::Widgets::WdgModels::ContributoriesTableModel::rowCount (const QModelIndex &) const

rowCount Number of contributories row

Returns

The number of contributories

8.24.3.10 bool Gui::Widgets::WdgModels::ContributoriesTableModel::setData (const QModelIndex & *index*, const QVariant & *value*, int *role* = `Qt::EditRole`)

setData Change data of a cell

Parameters

index	The cell to change data
value	The new value
role	THe role of cell

Returns

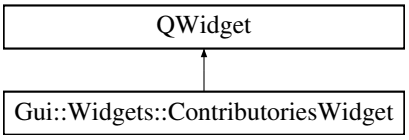
True if we could edit

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

- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/widgetsmodels/contributoriestablemodel.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/widgetsmodels/contributoriestablemodel.cpp

8.25 Gui::Widgets::ContributoriesWidget Class Reference

Inheritance diagram for Gui::Widgets::ContributoriesWidget:



Public Slots

- void add (void)

  add Add a new empty contributory
- void remove (void)

  remove Remove the current contributory
- void **addProject** (QPair< Project *, Rate > *p=0)
- void **removeProject** (void)
- void **changeProject** (void)
- void **editing** (void)
- void **updateUi** (void)
- void **updatePrice** (void)

Signals

- void **contributoryChanged** ()

Public Member Functions

- **ContributoriesWidget** (QSharedPointer< Customer > c, QWidget *parent=0)
- ContributoriesList * getContributories () const

  getContributories Get contributories List
- int **count** ()
- void add (ContributoriesList &list)

  add Add the Contributory c in the model

8.25.1 Member Function Documentation

8.25.1.1 void Gui::Widgets::ContributoriesWidget::add (ContributoriesList & *list*)

add Add the Contributory *c* in the model

Parameters

c	the **Contributory**

8.25.1.2 ContributoriesList * Gui::Widgets::ContributoriesWidget::getContributories ( ) const

getContributories Get contributories List

Returns

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

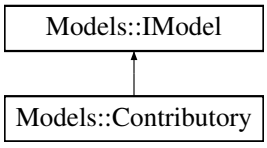
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/contributorieswidget.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/contributorieswidget.cpp

8.26 Models::Contributory Class Reference

The Contributory class.

`#include <contributory.h>`

Inheritance diagram for Models::Contributory:



Public Member Functions

- Contributory ()

  Contributory::Contributory Contruct a Contributory.
- Contributory (int id)

  Contributory::Contributory Contruct a Contributory and get data in database.
- ~Contributory ()

  Destroy an contributory object.
- 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.
- QVariantHash getDataMap ()

  getDataMap Get all data of model with a HashMap key/value
- Project * getProject () const

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

  Contributory::setProject Modify the identify id of the Project linked to this Contributory.
- double getNbHours () const

  getNbHours Number of work hour of a contributory
- void setNbHours (double value)

  setNbHours Change nbHours
- QString getDescription () const

  getDescription Description of a contributory
- void setDescription (const QString &getDescription)

  setDescription Change the contributory description
- bool operator== (const Contributory &c)

  *operator == define the operator "==" to compare two **Contributory***
- bool operator!= (const Contributory &c)

  *operator != define the operator "!=" to compare two **Contributory***

Additional Inherited Members

8.26.1 Detailed Description

The Contributory class.

Author

8.26.2 Constructor & Destructor Documentation

8.26.2.1 Models::Contributory::Contributory (int *id*)

Contributory::Contributory Contruct a Contributory and get data in database.

Parameters

id	Contributory's id

8.26.3 Member Function Documentation

8.26.3.1 QVariantHash Models::Contributory::getDataMap ( ) [virtual]

getDataMap Get all data of model with a HashMap key/value

Returns

Model's data

Implements Models::IModel.

8.26.3.2 QString Models::Contributory::getDescription ( ) const

getDescription Description of a contributory

Returns

The description

8.26.3.3 double Models::Contributory::getNbHours ( ) const

getNbHours Number of work hour of a contributory

Returns

Then number of hours

8.26.3.4 Project * Models::Contributory::getProject ( ) const

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

Returns

Project linked to this Contributory

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

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

Parameters

id	Contributory identify

Implements Models::IModel.

8.26.3.6 bool Models::Contributory::operator!= (const Contributory & *c*)

operator != define the operator "!=" to compare two **Contributory**

Parameters

c	the **Contributory** to compare with the current **Contributory**

Returns

true if the **Contributory** are different else false

8.26.3.7 bool Models::Contributory::operator== (const Contributory & *c*)

operator == define the operator "==" to compare two **Contributory**

Parameters

c	the **Contributory** to compare with the current **Contributory**

Returns

true if the **Contributory** are equals else false

8.26.3.8 void Models::Contributory::setDescription (const QString & *getDescription*)

setDescription Change the contributory description

Parameters

getDescription	The new description

8.26.3.9 void Models::Contributory::setNbHours (double *value*)

setNbHours Change nbHours

Parameters

value	The new value of nbHours

8.26.3.10 void Models::Contributory::setProject (Project * *id*)

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

Parameters

id	Project Identify

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

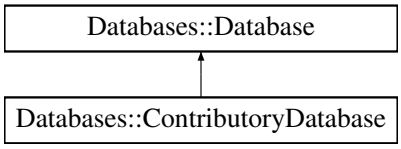
- /home/travis/build/FACT-Team/FactDev/src/models/contributory.h
- /home/travis/build/FACT-Team/FactDev/src/models/contributory.cpp

8.27 Databases::ContributoryDatabase Class Reference

The **ContributoryDatabase** class Contributory (or Quote) table database.

`#include <contributorydatabase.h>`

Inheritance diagram for Databases::ContributoryDatabase:



Public Member Functions

- Models::Contributory * getContributory (const int idContributory)

  ContributoryDatabase::getCustomer get informations about the Contributory identified by pId
- Models::ContributoriesList **getContributoriesByBilling** (const int idBilling)
- int addContributory (const Models::Contributory &)

  ContributoryDatabase::addContributory Add the Contributory pContributory to the database.
- void updateContributory (const Models::Contributory &)

  ContributoryDatabase::updateCustomer Update informations about the Contributory pCustomer
- void removeContributory (const int pId)

  ContributoryDatabase::removeCustomer Remove the Contributory with the id pId
- Models::Contributory * getContributory (QSqlQuery &q)

  getContributory Obtain a contributory without new query

Static Public Member Functions

- static ContributoryDatabase * instance () throw (DbException*)

  *ContributoryDatabase::getInstance Return an instance of **ContributoryDatabase***

Additional Inherited Members

8.27.1 Detailed Description

The **ContributoryDatabase** class Contributory (or Quote) table database.

Author

Cédric Rohaut

See Also

Database
Contributory/Quote

8.27.2 Member Function Documentation

8.27.2.1 int Databases::ContributoryDatabase::addContributory (const Models::Contributory & *pContributory*)

ContributoryDatabase::addContributory Add the Contributory *pContributory* to the database.

Returns

Contributory id

8.27.2.2 Models::Contributory * Databases::ContributoryDatabase::getContributory (const int *idContributory*)

ContributoryDatabase::getCustomer get informations about the Contributory identified by *pId*

Parameters

idContributory	Contributory id

Returns

the Contributory

8.27.2.3 Models::Contributory * Databases::ContributoryDatabase::getContributory (QSqlQuery & *q*)

getContributory Obtain a contributory without new query

Parameters

q	The query to use

Returns

The contributory linked to q

8.27.2.4 ContributoryDatabase * Databases::ContributoryDatabase::instance ( ) throw DbException *) [static]

ContributoryDatabase::getInstance Return an instance of **ContributoryDatabase**

See Also

DbException

Returns

Instance of ContributoryDatabase

8.27.2.5 void Databases::ContributoryDatabase::removeContributory (const int *pId*)

ContributoryDatabase::removeCustomer Remove the Contributory with the id *pId*

Parameters

pId	Contributory id

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

- /home/travis/build/FACT-Team/FactDev/src/database/contributorydatabase.h
- /home/travis/build/FACT-Team/FactDev/src/database/contributorydatabase.cpp

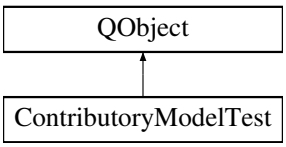
8.28 ContributoryListTest Class Reference

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

- /home/travis/build/FACT-Team/FactDev/tests/models/contributorylisttest.h
- /home/travis/build/FACT-Team/FactDev/tests/models/contributorylisttest.cpp

8.29 ContributoryModelTest Class Reference

Inheritance diagram for ContributoryModelTest:

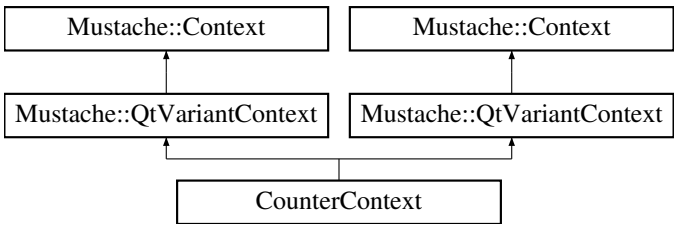


The documentation for this class was generated from the following files:

- /home/travis/build/FACT-Team/FactDev/tests/models/contributorymodeltest.h
- /home/travis/build/FACT-Team/FactDev/tests/models/contributorymodeltest.cpp

8.30 CounterContext Class Reference

Inheritance diagram for CounterContext:



Public Member Functions

- **CounterContext** (const QVariantHash &map)
- virtual bool canEval (const QString &key) const
- virtual QString eval (const QString &key, const QString &_template, Mustache::Renderer *renderer)
- virtual QString stringValue (const QString &key) const
- **CounterContext** (const QVariantHash &map)
- virtual bool canEval (const QString &key) const
- virtual QString eval (const QString &key, const QString &_template, Mustache::Renderer *renderer)
- virtual QString stringValue (const QString &key) const

Public Attributes

- int **counter**

Additional Inherited Members

8.30.1 Member Function Documentation

8.30.1.1 virtual bool CounterContext::canEval (const QString & *key*) const [inline], [virtual]

Returns true if eval() should be used to render section tags using `key`. If canEval() returns true for a key, the renderer will pass the literal, unrendered block of text for the section to eval() and replace the section with the result.

canEval() and eval() are equivalents for callable objects (eg. lambdas) in other Mustache implementations.

The default implementation always returns false.

Reimplemented from Mustache::QtVariantContext.

8.30.1.2 virtual bool CounterContext::canEval (const QString & *key*) const [inline], [virtual]

Returns true if eval() should be used to render section tags using `key`. If canEval() returns true for a key, the renderer will pass the literal, unrendered block of text for the section to eval() and replace the section with the result.

canEval() and eval() are equivalents for callable objects (eg. lambdas) in other Mustache implementations.

The default implementation always returns false.

Reimplemented from Mustache::QtVariantContext.

8.30.1.3 virtual QString CounterContext::eval (const QString & *key*, const QString & *_template*, Mustache::Renderer * *renderer*) [inline], [virtual]

Callback used to render a template section with the given `key`. `renderer` will substitute the original section tag with the result of eval().

The default implementation returns an empty string.

Reimplemented from Mustache::QtVariantContext.

8.30.1.4 virtual QString CounterContext::eval (const QString & *key*, const QString & *_template*, Mustache::Renderer * *renderer*) [inline], [virtual]

Callback used to render a template section with the given `key`. `renderer` will substitute the original section tag with the result of eval().

The default implementation returns an empty string.

Reimplemented from Mustache::QtVariantContext.

8.30.1.5 virtual QString CounterContext::stringValue (const QString & *key*) const [inline], [virtual]

Returns a string representation of the value for `key` in the current context. This is used to replace a Mustache value tag.

Reimplemented from Mustache::QtVariantContext.

8.30.1.6 virtual QString CounterContext::stringValue (const QString & *key*) const [inline], [virtual]

Returns a string representation of the value for `key` in the current context. This is used to replace a Mustache value tag.

Reimplemented from Mustache::QtVariantContext.

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

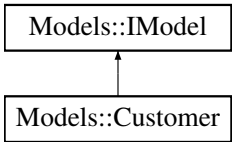
- /home/travis/build/FACT-Team/FactDev/src/libs/qt-mustache/tests/test_mustache.cpp

8.31 Models::Customer Class Reference

The Customer class Customer.

`#include <customer.h>`

Inheritance diagram for Models::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.
- QVariantHash getDataMap ()

  getDataMap Get all data of model with a HashMap key/value
- 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

- QString getPath () const

  Customer::getPath Return the path of the workspace for the current Customer.

- QString getNameFolder () const

  Customer::getNameFolder Return the name of the current Customer's folder in the workspace.

- 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

8.31.1 Detailed Description

The Customer class Customer.

Author

Antoine de Roquemaurel
Florent Berbie

8.31.2 Constructor & Destructor Documentation

8.31.2.1 Models::Customer::Customer (int *id*)

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

Parameters

id	Customer identify

8.31.3 Member Function Documentation

8.31.3.1 QString Models::Customer::getAddress ( ) const

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

Returns

company address

8.31.3.2 QString Models::Customer::getCity ( ) const

Customer::getCity Return the city of the customer company.

Returns

city of the customer company

8.31.3.3 QString Models::Customer::getCompany ( ) const

Customer::getCompany Return the name of the company customer.

Returns

name of the company customer

8.31.3.4 QString Models::Customer::getCountry ( ) const

Customer::getCountry Return the country of the customer company.

Returns

country of the customer company

8.31.3.5 QVariantHash Models::Customer::getDataMap ( ) [virtual]

getDataMap Get all data of model with a HashMap key/value

Returns

Model's data

Implements Models::IModel.

8.31.3.6 QString Models::Customer::getEmail ( ) const

Customer::getEmail Return the email of the customer company.

Returns

email of the customer company

8.31.3.7 QString Models::Customer::getFax ( ) const

Customer::getFax Return the fax number.

Returns

fax number

8.31.3.8 QString Models::Customer::getFirstnameReferent ( ) const

Customer::getFirstnameReferent Return the firstname of the referent Customer.

Returns

Referent firstname of the customer

8.31.3.9 QString Models::Customer::getLastnameReferent ( ) const

Customer::getLastnameReferent Return the lastname of the referent Customer.

Returns

Referent lastname of the customer

8.31.3.10 QString Models::Customer::getMobilePhone ( ) const

Customer::getMobilePhone Return the number of the mobile phone of the customer company.

Returns

number of mobile phone

8.31.3.11 QString Models::Customer::getNameFolder ( ) const

Customer::getNameFolder Return the name of the current Customer's folder in the workspace.

Returns

name of the Customer's folder

8.31.3.12 QString Models::Customer::getPath ( ) const

Customer::getPath Return the path of the workspace for the current Customer.

Returns

workspace path

8.31.3.13 QString Models::Customer::getPhone ( ) const

Customer::getPhone Return the number of desktop phone.

Returns

number of desktop phone

8.31.3.14 QString Models::Customer::getPostalCode ( ) const

Customer::getPostalCode Return the postal code of the customer company.

Returns

postal code of the customer company

8.31.3.15 void Models::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 Models::IModel.

8.31.3.16 bool Models::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

8.31.3.17 bool Models::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

8.31.3.18 void Models::Customer::setAddress (const QString & *address*)

Customer::setAddress Replace the current company address by the new *address*

Parameters

address	New company address

8.31.3.19 void Models::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

8.31.3.20 void Models::Customer::setCompany (const QString & *company*)

Customer::setCompany Replace the company name by *company*

Parameters

company	New company name

8.31.3.21 void Models::Customer::setCountry (const QString & *country*)

Customer::setCountry Replace the country of the Customer company by *country*

Parameters

country	New country of the customer company

8.31.3.22 void Models::Customer::setEmail (const QString & *email*)

Customer::setEmail Replace the email of the Customer company by *email*

Parameters

email	New email of the customer company

8.31.3.23 void Models::Customer::setFax (const QString & *fax*)

Customer::setFax Replace the current fax number by *fax*

Parameters

fax	new fax number

8.31.3.24 void Models::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

8.31.3.25 void Models::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

8.31.3.26 void Models::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

8.31.3.27 void Models::Customer::setPhone (const QString & *phone*)

Customer::setPhone Replace the current number of desktop phone by *phone*

Parameters

phone	New number of desktop phone

8.31.3.28 void Models::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:

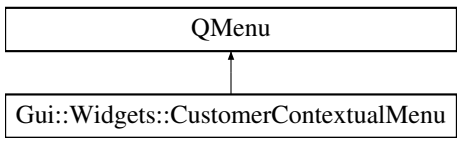
- /home/travis/build/FACT-Team/FactDev/src/models/customer.h
- /home/travis/build/FACT-Team/FactDev/src/models/customer.cpp

8.32 Gui::Widgets::CustomerContextualMenu Class Reference

Display contextual menu on a customer.

`#include <customercontextualmenu.h>`

Inheritance diagram for Gui::Widgets::CustomerContextualMenu:



Public Member Functions

- CustomerContextualMenu (QWidget *w=0)

  CustomerContextualMenu Construct a new contextual menu.
- ~CustomerContextualMenu ()

  Destruct the contextual menu.

8.32.1 Detailed Description

Display contextual menu on a customer.

Author

Antoine de Roquemaurel

8.32.2 Constructor & Destructor Documentation

8.32.2.1 Gui::Widgets::CustomerContextualMenu::CustomerContextualMenu (QWidget * w = 0)

CustomerContextualMenu Construct a new contextual menu.

Parameters

w	Parent

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

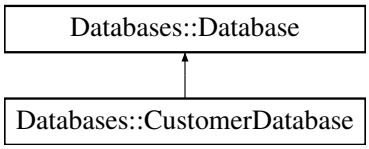
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/customercontextualmenu.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/customercontextualmenu.cpp

8.33 Databases::CustomerDatabase Class Reference

The **CustomerDatabase** class Customer table database.

`#include <customerdatabase.h>`

Inheritance diagram for Databases::CustomerDatabase:



Public Member Functions

- WdgModels::CustomersTableModel * getCustomersTable (QString filter="") throw (DbException*)

  CustomerDatabase::getCustomersTable Return an item model of customers for QTableView.
- QStandardItemModel * getTree (QString filter="") throw (DbException*)

  CustomerDatabase::getTree Return an item model of customers for QTree.
- QSharedPointer< Models::Customer > getCustomer (const int pId)

  CustomerDatabase::getCustomer get informations about the customer identified by pId
- int addCustomer (const Models::Customer &)

  CustomerDatabase::addCustomer Add the customer pCustomer to the database.
- void updateCustomer (const Models::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.
- QStandardItem * getItemRoot ()

CustomerDatabase::getItemRoot Return the first item for the QStandardItemModel.

- QStandardItem * getItemCustomer (QSqlQuery q1)

  CustomerDatabase::getItemCustomer Return the customer item for the QStandardItemModel.
- QStandardItem * getItemProject (QSqlQuery q2)

  CustomerDatabase::getItemProject Return the project item for the QStandardItemModel.
- QStandardItem * getItemBillQuote (QSqlQuery q3)

  CustomerDatabase::getItemCustomer Return the bill/quote item for the QStandardItemModel.
- QSharedPointer< Models::Customer > getCustomer (QSqlQuery &q)

  CustomerDatabase::getCustomer Add the element of the q request and return their.
- void updateCustomer (QSqlQuery &q, Customer &pCustomer)

  CustomerDatabase::updateCustomer Update customer data according to the request q

Static Public Member Functions

- static CustomerDatabase * instance () throw (DbException*)

  *CustomerDatabase::instance Return an instance of **CustomerDatabase***

Additional Inherited Members

8.33.1 Detailed Description

The **CustomerDatabase** class Customer table database.

Author

Antoine de Roquemaurel
Manantsoa Razanajatovo
Florent Berbie
Cédric Rohaut

See Also

Database
Customer

8.33.2 Member Function Documentation

8.33.2.1 int Databases::CustomerDatabase::addCustomer (const Models::Customer & *pCustomer*)

CustomerDatabase::addCustomer Add the customer *pCustomer* to the database.

Returns

customer id

8.33.2.2 QSharedPointer< Models::Customer > Databases::CustomerDatabase::getCustomer (const int *pId*)

CustomerDatabase::getCustomer get informations about the customer identified by *pId*

Parameters

pId	customer id

Returns

the Customer

8.33.2.3 QSharedPointer< Models::Customer > Databases::CustomerDatabase::getCustomer (QSqlQuery & *q*)

CustomerDatabase::getCustomer Add the element of the *q* request and return their.

Parameters

q	SQL request

Returns

a customer formed according to QSharedPointer

8.33.2.4 WdgModels::CustomersTableModel * Databases::CustomerDatabase::getCustomersTable (QString *filter* = " ") throw DbException *)

CustomerDatabase::getCustomersTable Return an item model of customers for QTableView.

Parameters

filter	Select only customers who are specified by *filter*

Exceptions

DbException	

Returns

QStandardItemModel an item model for QTableView

8.33.2.5 QStandardItem * Databases::CustomerDatabase::getItemBillQuote (QSqlQuery *q3*)

CustomerDatabase::getItemCustomer Return the bill/quote item for the QStandardItemModel.

Parameters

q3	the first row of the sql query for bills/quotes

Returns

QStandardItem an item for QTree (level/depth 3)

8.33.2.6 QStandardItem * Databases::CustomerDatabase::getItemCustomer (QSqlQuery *q1*)

CustomerDatabase::getItemCustomer Return the customer item for the QStandardItemModel.

Parameters

q1	the row of the sql query for customers

Returns

QStandardItem an item for QTree (level/depth 1)

8.33.2.7 QStandardItem * Databases::CustomerDatabase::getItemProject (QSqlQuery *q2*)

CustomerDatabase::getItemProject Return the project item for the QStandardItemModel.

Parameters

q2	the row of the sql query for projects

Returns

QStandardItem an item for QTree (level/depth 2)

8.33.2.8 QStandardItem * Databases::CustomerDatabase::getItemRoot ( )

CustomerDatabase::getItemRoot Return the first item for the QStandardItemModel.

Returns

QStandardItem an item for QTree (level/depth 0)

8.33.2.9 int Databases::CustomerDatabase::getNbCustomers ( )

CustomerDatabase::getNbCustomers Return the number of customers existing.

Returns

number of customers

8.33.2.10 QStandardItemModel * Databases::CustomerDatabase::getTree (QString *filter* = " ") throw DbException *)

CustomerDatabase::getTree Return an item model of customers for QTree.

Parameters

filter	Select only customers who are specified by *filter*

Exceptions

DbException	

Returns

QStandardItemModel an item model for QTreeView

8.33.2.11 CustomerDatabase * Databases::CustomerDatabase::instance ( ) throw DbException *) [static]

CustomerDatabase::instance Return an instance of **CustomerDatabase**

See Also

DbException

Returns

Instance of CustomerDatabase

8.33.2.12 void Databases::CustomerDatabase::removeCustomer (const int *pId*)

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

Parameters

pId	customer id

8.33.2.13 void Databases::CustomerDatabase::updateCustomer (QSqlQuery & *q*, Customer & *pCustomer*)

CustomerDatabase::updateCustomer Update customer data according to the request *q*

Parameters

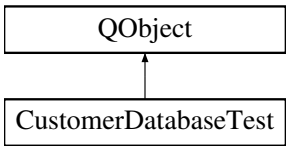
q	SQL request

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

- /home/travis/build/FACT-Team/FactDev/src/database/customerdatabase.h
- /home/travis/build/FACT-Team/FactDev/src/database/customerdatabase.cpp

8.34 CustomerDatabaseTest Class Reference

Inheritance diagram for CustomerDatabaseTest:



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

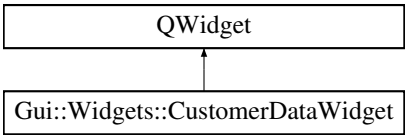
- /home/travis/build/FACT-Team/FactDev/tests/database/customerdatabasetest.h
- /home/travis/build/FACT-Team/FactDev/tests/database/customerdatabasetest.cpp

8.35 Gui::Widgets::CustomerDataWidget Class Reference

Class for display info of a customer.

`#include <customerdatawidget.h>`

Inheritance diagram for Gui::Widgets::CustomerDataWidget:



Public Member Functions

- **CustomerDataWidget** (QWidget ∗parent=0)
- void printUserData ()

  printUserData Print Data of current user
- void printInformations (int id)

  printInformations Print Data of customer id

8.35.1 Detailed Description

Class for display info of a customer.

Author

Cédric Rohaut

8.35.2 Member Function Documentation

8.35.2.1 void Gui::Widgets::CustomerDataWidget::printInformations (int *id*)

printInformations Print Data of customer id

Parameters

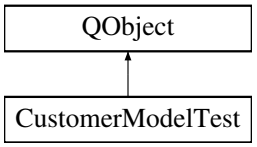
id	of customer to print

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

- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/customerdatawidget.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/customerdatawidget.cpp

8.36 CustomerModelTest Class Reference

Inheritance diagram for CustomerModelTest:



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

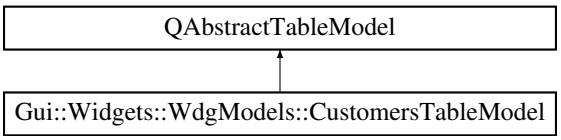
- /home/travis/build/FACT-Team/FactDev/tests/models/customermodeltest.h
- /home/travis/build/FACT-Team/FactDev/tests/models/customermodeltest.cpp

8.37 Gui::Widgets::WdgModels::CustomersTableModel Class Reference

The CustomersTableModel class for a customer table.

`#include <customerstablemodel.h>`

Inheritance diagram for Gui::Widgets::WdgModels::CustomersTableModel:



Public Member Functions

- CustomersTableModel (QObject *parent=0)

  CustomersTableModel::CustomersTableModel Construct a CustomersTableModel.
- int rowCount (const QModelIndex &) const

  CustomersTableModel::rowCount Number of customers row.
- int columnCount (const QModelIndex &) const

  CustomersTableModel::columnCount Number of column of a customer.
- QVariant data (const QModelIndex &index, int role=Qt::DisplayRole) const

  CustomersTableModel::data Obtains data of a specify cell.
- QVariant headerData (int section, Qt::Orientation orientation, int role=Qt::DisplayRole) const

  CustomersTableModel::headerData Obtains header title of table.
- bool setData (const QModelIndex &index, const QVariant &value, int role=Qt::EditRole)

  CustomersTableModel::setData Change data of a cell.
- void append (const Customer &customer)

  CustomersTableModel::append Add a new line in table.
- void remove (const int i)

  CustomersTableModel::remove Remove a line.
- Qt::ItemFlags flags (const QModelIndex &index) const

  CustomersTableModel::flags Differents table flags.
- int count ()

  CustomersTableModel::count Number of customers in table.
- QList< Customer > getCustomers () const

  CustomersTableModel::getCustomers Return the list of customers.

8.37.1 Detailed Description

The CustomersTableModel class for a customer table.

Author

Florent Berbie

See Also

Customer

8.37.2 Constructor & Destructor Documentation

8.37.2.1 Gui::Widgets::WdgModels::CustomersTableModel::CustomersTableModel (QObject * *parent* = 0)

CustomersTableModel::CustomersTableModel Construct a CustomersTableModel.

Parameters

parent	Parent widget

8.37.3 Member Function Documentation

8.37.3.1 void Gui::Widgets::WdgModels::CustomersTableModel::append (const Customer & *customer*)

CustomersTableModel::append Add a new line in table.

Parameters

Customer	The new customer

8.37.3.2 int Gui::Widgets::WdgModels::CustomersTableModel::columnCount (const QModelIndex &) const

CustomersTableModel::columnCount Number of column of a customer.

Returns

The number of column

8.37.3.3 int Gui::Widgets::WdgModels::CustomersTableModel::count ( )

CustomersTableModel::count Number of customers in table.

Returns

The number of customers

8.37.3.4 QVariant Gui::Widgets::WdgModels::CustomersTableModel::data (const QModelIndex & *index*, int *role* = `Qt::DisplayRole`) const

CustomersTableModel::data Obtains data of a specify cell.

Parameters

index	The cell who we want data
role	The role of set

Returns

The data of cell

8.37.3.5 Qt::ItemFlags Gui::Widgets::WdgModels::CustomersTableModel::flags (const QModelIndex & *index*) const

CustomersTableModel::flags Differents table flags.

Parameters

index	The cell who we want to know flags

Returns

Flags

8.37.3.6 QList< Customer > Gui::Widgets::WdgModels::CustomersTableModel::getCustomers ( ) const

CustomersTableModel::getCustomers Return the list of customers.

Returns

list of Customers

8.37.3.7 QVariant Gui::Widgets::WdgModels::CustomersTableModel::headerData (int *section*, Qt::Orientation *orientation*, int *role* = `Qt::DisplayRole`) const

CustomersTableModel::headerData Obtains header title of table.

Parameters

section	The number of column
orientation	The table orientation
role	

Returns

The Title header of column

8.37.3.8 void Gui::Widgets::WdgModels::CustomersTableModel::remove (const int *i*)

CustomersTableModel::remove Remove a line.

Parameters

i	The number of line to remove

8.37.3.9 int Gui::Widgets::WdgModels::CustomersTableModel::rowCount (const QModelIndex &) const

CustomersTableModel::rowCount Number of customers row.

Returns

The number of customers

8.37.3.10 bool Gui::Widgets::WdgModels::CustomersTableModel::setData (const QModelIndex & *index*, const QVariant & *value*, int *role* = `Qt::EditRole`)

CustomersTableModel::setData Change data of a cell.

Parameters

index	The cell to change data
value	The new value
role	The role of cell

Returns

True if we could edit

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

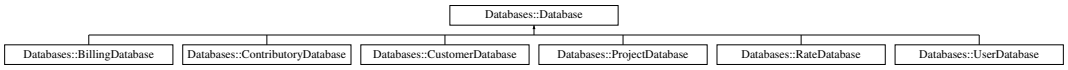
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/widgetsmodels/customerstablemodel.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/widgetsmodels/customerstablemodel.cpp

8.38 Databases::Database Class Reference

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

`#include <database.h>`

Inheritance diagram for Databases::Database:



Public Member Functions

- QString lastError (const QSqlQuery &q) const

  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::createDatabase Create a new database.
- void updateBillingNumber ()

  Database::updateBillingNumber Update the billing number.

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) const

  Database::valeur Value of database field.

Protected Attributes

- QSettings * _settings
  settings
- QSqlDatabase mDatabase
  contains Database
- QList< Database * > _instances
  List of instances.

Static Protected Attributes

- static Database * _instance = 0
  Instance.
- static bool _dbInstance = 0
  an instance of db is open
- static bool isOpen = false
  Database is open.

8.38.1 Detailed Description

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

Author

Antoine de Roquemaurel

8.38.2 Member Function Documentation

8.38.2.1 void Databases::Database::executeFile (QString *pName*)

Database::executerFichier Exeute a specified file named *pName*

Parameters

pNom	File name

8.38.2.2 Database * Databases::Database::instance ( ) throw DbException *) [static]

Database::getInstance Return an instance of Database.

Returns

Instance of Database

8.38.2.3 QString Databases::Database::lastError (const QSqlQuery & *q*) const [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

8.38.2.4 void Databases::Database::setDatabase (QSqlDatabase *sql*)

Database::setDatabase Set database.

Parameters

sql	The new database

8.38.2.5 QVariant Databases::Database::value (const QSqlQuery & *q*, const QString & *champ*) const [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:

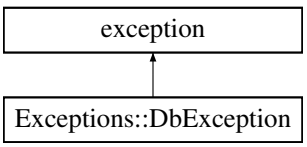
- /home/travis/build/FACT-Team/FactDev/src/database/database.h
- /home/travis/build/FACT-Team/FactDev/src/database/database.cpp

8.39 Exceptions::DbException Class Reference

The DbException class for database exception : queries, db file, ...

`#include <dbexception.h>`

Inheritance diagram for Exceptions::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.

8.39.1 Detailed Description

The DbException class for database exception : queries, db file, ...

Author

Antoine de Roquemaurel

8.39.2 Constructor & Destructor Documentation

8.39.2.1 Exceptions::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

8.39.3 Member Function Documentation

8.39.3.1 void Exceptions::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:

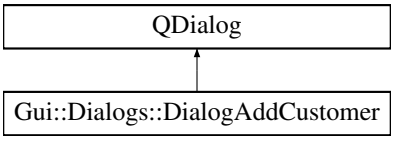
- /home/travis/build/FACT-Team/FactDev/src/exceptions/dbexception.h
- /home/travis/build/FACT-Team/FactDev/src/exceptions/dbexception.cpp

8.40 Gui::Dialogs::DialogAddCustomer Class Reference

The DialogAddCustomer class Window to add or modify a Customer.

`#include <dialogaddcustomer.h>`

Inheritance diagram for Gui::Dialogs::DialogAddCustomer:



Public Slots

- void checkFields ()

  setFieldsComponents

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

8.40.1 Detailed Description

The DialogAddCustomer class Window to add or modify a Customer.

Author

Cédric Rohaut

8.40.2 Constructor & Destructor Documentation

8.40.2.1 Gui::Dialogs::DialogAddCustomer::DialogAddCustomer (int *id* = 0, QWidget * *parent* = 0) [explicit]

DialogAddCustomer Construct a window to add/modify a Customer.

Parameters

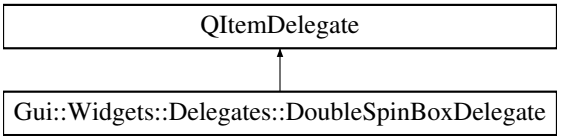
id	Customer id
parent	QWidget parent

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

- /home/travis/build/FACT-Team/FactDev/src/gui/dialogs/dialogaddcustomer.h
- /home/travis/build/FACT-Team/FactDev/src/gui/dialogs/dialogaddcustomer.cpp

8.41 Gui::Widgets::Delegates::DoubleSpinBoxDelegate Class Reference

Inheritance diagram for Gui::Widgets::Delegates::DoubleSpinBoxDelegate:



Public Member Functions

- **DoubleSpinBoxDelegate** (QObject *parent=0)
- QWidget * **createEditor** (QWidget *parent, const QStyleOptionViewItem &option, const QModelIndex &index) const
- void **setEditorData** (QWidget *editor, const QModelIndex &index) const
- void **setModelData** (QWidget *editor, QAbstractItemModel *model, const QModelIndex &index) const
- void **updateEditorGeometry** (QWidget *editor, const QStyleOptionViewItem &option, const QModelIndex &index) const

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

- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/delegates/doublespinboxdelegate.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/delegates/doublespinboxdelegate.cpp

8.42 Exceptions::FileException Class Reference

The FileException class for file/acess file exception.

`#include <fileexception.h>`

Public Member Functions

- FileException (const QString userError, const QString fctName, const QString logError, float errorCode)

  FileException::FileException. Construct a FileException.
- void popupMessage (QWidget ∗parent)

  FileException::popupMessage. Display a popup message with the message error.

8.42.1 Detailed Description

The FileException class for file/acess file exception.

Author

Florent Berbie

8.42.2 Constructor & Destructor Documentation

8.42.2.1 Exceptions::FileException::FileException (const QString *userError,* const QString *fctName,* const QString *logError,* float *errorCode*)

FileException::FileException. Construct a FileException.

Parameters

userError	ClassName of error
fctName	Function name
logError	Message error
errorCode	Code of error

8.42.3 Member Function Documentation

8.42.3.1 void Exceptions::FileException::popupMessage (QWidget ∗ *parent*)

FileException::popupMessage. Display a popup message with the message error.

Parameters

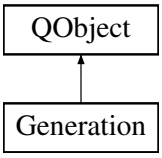
parent	

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

- /home/travis/build/FACT-Team/FactDev/src/exceptions/fileexception.h
- /home/travis/build/FACT-Team/FactDev/src/exceptions/fileexception.cpp

8.43 Generation Class Reference

Inheritance diagram for Generation:



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

- /home/travis/build/FACT-Team/FactDev/tests/generation.h
- /home/travis/build/FACT-Team/FactDev/tests/generation.cpp

8.44 Generator Class Reference

Public Member Functions

- **Generator** (QString tpl)
- void **generate** (QVariantHash data, QString path)

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

- /home/travis/build/FACT-Team/FactDev/src/generator.h
- /home/travis/build/FACT-Team/FactDev/src/generator.cpp

8.45 Utils::HierarchicalSystem Class Reference

The HierarchicalSystem class Create class which contains hierarchical system of FactDev.

`#include <hierarchicalsystem.h>`

Public Member Functions

- HierarchicalSystem ()

  HierarchicalSystem::HierarchicalSystem Construct a HierarchicalSystem.
- void getAllProjects ()

  HierarchicalSystem::getAllProjects Get all projects and add each project to Customer linked.
- void getAllBillings ()

  HierarchicalSystem::getAllBillings Get all billings and add each billing to Project linked.
- void updateData ()

  HierarchicalSystem::updateData Update data on Customers, Projects, Billings.
- void addProjectToCustomer (Project ∗p, Customer c)

  HierarchicalSystem::addProjectToCustomer Add the Project p to the Customer c
- void addBillingToProject (Billing ∗b, Project ∗p)

  HierarchicalSystem::addBillingToProject Add the Billing b to the Project p
- QMap< Project ∗, Customer > getCustomers () const

  HierarchicalSystem::getCustomers Return all customers and these projects linked.
- QMap< Billing ∗, Project ∗ > getProjects () const

  HierarchicalSystem::getProjects Return all projects and these billing linked.

8.45.1 Detailed Description

The HierarchicalSystem class Create class which contains hierarchical system of FactDev.

Author

Florent Berbie

See Also

Customer
Project
Billing

8.45.2 Member Function Documentation

8.45.2.1 void Utils::HierarchicalSystem::addBillingToProject (Billing * *b*, Project * *p*)

HierarchicalSystem::addBillingToProject Add the Billing *b* to the Project p

Parameters

b	Billing
p	Project

8.45.2.2 void Utils::HierarchicalSystem::addProjectToCustomer (Project * *p*, Customer *c*)

HierarchicalSystem::addProjectToCustomer Add the Project *p* to the Customer *c*

Parameters

p	Project
c	Customer

8.45.2.3 QMap< Project *, Customer > Utils::HierarchicalSystem::getCustomers ( ) const

HierarchicalSystem::getCustomers Return all customers and these projects linked.

Returns

Projects linked to Customers

8.45.2.4 QMap< Billing *, Project * > Utils::HierarchicalSystem::getProjects (void) const

HierarchicalSystem::getProjects Return all projects and these billing linked.

Returns

Billing linked to Projects

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

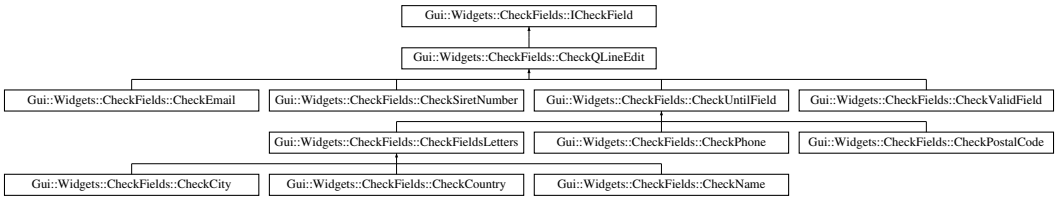
- /home/travis/build/FACT-Team/FactDev/src/utils/hierarchicalsystem.h
- /home/travis/build/FACT-Team/FactDev/src/utils/hierarchicalsystem.cpp

8.46 Gui::Widgets::CheckFields::ICheckField Class Reference

The ICheckField class Interface to check fields validity.

`#include <icheckfield.h>`

Inheritance diagram for Gui::Widgets::CheckFields::ICheckField:



Public Member Functions

- virtual bool check (QString text)=0

  check Check if the field (line edit) is valid Return TRUE if the field is valid, else FALSE

8.46.1 Detailed Description

The ICheckField class Interface to check fields validity.

8.46.2 Member Function Documentation

8.46.2.1 virtual bool Gui::Widgets::CheckFields::ICheckField::check (QString *text*) [pure virtual]

check Check if the field (line edit) is valid Return TRUE if the field is valid, else FALSE

Returns

boolean

Implemented in Gui::Widgets::CheckFields::CheckEmail, Gui::Widgets::CheckFields::CheckPhone, Gui::Widgets::CheckFields::CheckPostalCode, Gui::Widgets::CheckFields::CheckSiretNumber, Gui::Widgets::CheckFields::CheckUntilField, Gui::Widgets::CheckFields::CheckFieldsLetters, and Gui::Widgets::CheckFields::CheckValidField.

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

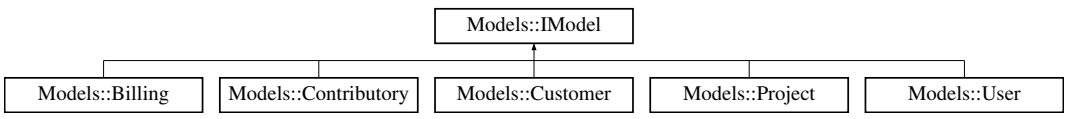
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/icheckfield.h

8.47 Models::IModel Class Reference

The IModel class.

`#include <imodel.h>`

Inheritance diagram for Models::IModel:



Public Member Functions

- virtual ~IModel ()

  ~IModel Remove an instance of IModel
- virtual void commit ()=0

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

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

  IModel::remove Remove the current element in the database.
- virtual QVariantHash getDataMap ()=0

  getDataMap Get all data of model with a HashMap key/value
- int getId () const

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

  IModel::setId Replace the current identify by id
- bool isToRemoved () const

  toRemoved return if object must be removed.
- void setToRemoved (bool toRemoved)

  setToRemoved Change the flag for removed object

Protected Attributes

- int _id

  Element identify.
- bool _toRemoved

  Flag to know if the object must be removed.

8.47.1 Detailed Description

The IModel class.

Author

Antoine de Roquemaurel

8.47.2 Member Function Documentation

8.47.2.1 virtual QVariantHash Models::IModel::getDataMap ( ) [pure virtual]

getDataMap Get all data of model with a HashMap key/value

Returns

Model's data

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

8.47.2.2 int Models::IModel::getId ( ) const [inline]

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

Returns

identity

8.47.2.3 virtual void Models::IModel::hydrat (int *id*) [pure virtual]

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

Parameters

id	

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

8.47.2.4 bool Models::IModel::isToRemoved ( ) const [inline]

toRemoved return if object must be removed.

Returns

8.47.2.5 void Models::IModel::setId (int *id*) [inline]

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

Parameters

id	New identify

8.47.2.6 void Models::IModel::setToRemoved (bool *toRemoved*) [inline]

setToRemoved Change the flag for removed object

Parameters

toRemoved	The new flag

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

- /home/travis/build/FACT-Team/FactDev/src/models/imodel.h

8.48 Utils::ItemType Class Reference

The ItemType class Item type model.

`#include <itemtype.h>`

Public Member Functions

- ItemType (int type, QString name)

  ItemType::ItemType Construct an Item type.
- QString getName () const

  ItemType::getName Get item name.
- Models::IModel * getModel (int id)

  *ItemType::getModel Get the databasemodel of the **ItemType** according to this identity id*
- void setName (const QString &name)

  ItemType::setName Modify the item name.
- int getType () const

ItemType::getType Get the type of the current item.

- void setType (int type)

  ItemType::setType Modify the type of the current item.

Static Public Attributes

- static const int CUSTOMER = 0

  constant value assigned to Customer
- static const int PROJECT = 1

  constant value assigned to Project
- static const int BILLING = 2

  constant value assigned to Billing
- static const int QUOTE = 3

  constant value assigned to Quote

8.48.1 Detailed Description

The ItemType class Item type model.

8.48.2 Constructor & Destructor Documentation

8.48.2.1 Utils::ItemType::ItemType (int *type*, QString *name*)

ItemType::ItemType Construct an Item type.

Parameters

type	Type of the item
name	Name of the item

8.48.3 Member Function Documentation

8.48.3.1 Models::IModel * Utils::ItemType::getModel (int *id*)

ItemType::getModel Get the databasemodel of the **ItemType** according to this identity *id*

Parameters

id	Item type identity

Returns

database model

8.48.3.2 QString Utils::ItemType::getName ( ) const

ItemType::getName Get item name.

Returns

item name

8.48.3.3 int Utils::ItemType::getType ( ) const

ItemType::getType Get the type of the current item.

Returns

type of the current item

8.48.3.4 void Utils::ItemType::setName (const QString & *name*)

ItemType::setName Modify the item name.

Parameters

name	New Item name

8.48.3.5 void Utils::ItemType::setType (int *type*)

ItemType::setType Modify the type of the current item.

Parameters

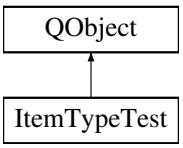
type	New item type

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

- /home/travis/build/FACT-Team/FactDev/src/utils/itemtype.h
- /home/travis/build/FACT-Team/FactDev/src/utils/itemtype.cpp

8.49 ItemTypeTest Class Reference

Inheritance diagram for ItemTypeTest:



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

- /home/travis/build/FACT-Team/FactDev/tests/utils/itemtypetest.h
- /home/travis/build/FACT-Team/FactDev/tests/utils/itemtypetest.cpp

8.50 Utils::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

8.50.1 Detailed Description

The Log class for Simple management of log.

8.50.2 Member Function Documentation

8.50.2.1 Log & Utils::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.

8.50.2.2 void Utils::Log::write (const QString *text*)

Log::write. Write log message in file.

Parameters

text	

8.50.3 Friends And Related Function Documentation

8.50.3.1 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:

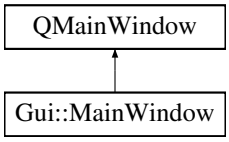
- /home/travis/build/FACT-Team/FactDev/src/utils/log.h
- /home/travis/build/FACT-Team/FactDev/src/utils/log.cpp

8.51 Gui::MainWindow Class Reference

The MainWindow class Main Window of the software.

`#include <mainwindow.h>`

Inheritance diagram for Gui::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 addQuote ()

  MainWindow::addQuote open window to add a new quote.
- void addBill ()

  MainWindow::addBill open window to add a new bill.
- void editUser ()

  MainWindow::editUser modify the user.
- void search (QString s)

  MainWindow::search launch a new search.
- void search ()

  MainWindow:search call search(QString)
- void newProject ()

  MainWindow::newProject Create a new project for a customer.
- void removeProject (void)

  MainWindow::removeProject Remove a project for a customer.
- void editProject (void)

  MainWindow::editProject Modify the customer project.
- 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 updateButtons (void)

  updateButton Update all button to disable or enabled its
- void editDoc ()

  MainWindow::editDoc Edit the quote or bill of the project.
- void removeDoc ()

  MainWindow::removeDoc Remove the quote or bill of the project.

Public Member Functions

- MainWindow (QWidget ∗parent=0)

  MainWindow: Construct a window.
- int getCurrentCustomerId ()

  MainWindow::getCurrentCustomerId get the selected customer.
- int getCurrentProjectId ()

  MainWindow::getCurrentProjectId get the selected project id.
- int getCurrentQuoteId ()

  MainWindow::getCurrentQuoteId get the selected quote id.
- QString getCurrentCustomerName ()

  MainWindow::getCurrentCustomerName get the selected customer name in the customers' table.
- QString getCurrentProjectName ()

  MainWindow::getCurrentProjectName get the selected project name in the table of projects.
- int treeLevel ()

  MainWindow::treeLevel return the level of the node selected in the tree.
- QModelIndex rootTree ()

  MainWindow::rootTree return the root of the tree "Tous les clients".
- void addDoc (bool isBilling)

  MainWindow::addDoc open window to add a new document.
- void **demo** ()

8.51.1 Detailed Description

The MainWindow class Main Window of the software.

Author

Everybody

8.51.2 Constructor & Destructor Documentation

8.51.2.1 Gui::MainWindow::MainWindow (QWidget ∗ *parent* = 0) [explicit]

MainWindow: Construct a window.

Parameters

parent	

8.51.3 Member Function Documentation

8.51.3.1 void Gui::MainWindow::addBill ( ) [slot]

MainWindow::addBill open window to add a new bill.

See Also

AddQuoteDialog

8.51.3.2 void Gui::MainWindow::addDoc (bool *isBilling*)

MainWindow::addDoc open window to add a new document.

Parameters

bool	quote or bill

See Also

addBill addQuote

8.51.3.3 void Gui::MainWindow::addQuote ( ) [slot]

MainWindow::addQuote open window to add a new quote.

See Also

AddQuoteDialog

8.51.3.4 void Gui::MainWindow::editUser ( ) [slot]

MainWindow::editUser modify the user.

See Also

UserDataDialog

8.51.3.5 int Gui::MainWindow::getCurrentCustomerId ( )

MainWindow::getCurrentCustomerId get the selected customer.

Returns

id of the selected customer

8.51.3.6 QString Gui::MainWindow::getCurrentCustomerName ( )

MainWindow::getCurrentCustomerName get the selected customer name in the customers' table.

Returns

name of the selected customer

8.51.3.7 int Gui::MainWindow::getCurrentProjectId ( )

MainWindow::getCurrentProjectId get the selected project id.

Returns

id of the selected project

8.51.3.8 QString Gui::MainWindow::getCurrentProjectName ( )

MainWindow::getCurrentProjectName get the selected project name in the table of projects.

Returns

name of the selected project

8.51.3.9 int Gui::MainWindow::getCurrentQuoteId ( )

MainWindow::getCurrentQuoteId get the selected quote id.

Returns

id of the selected quote

8.51.3.10 void Gui::MainWindow::newProject ( ) [slot]

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

See Also

AddProjectDialog

8.51.3.11 QModelIndex Gui::MainWindow::rootTree ( )

MainWindow::rootTree return the root of the tree "Tous les clients".

Returns

QModelIndex

8.51.3.12 void Gui::MainWindow::search (QString *s*) [slot]

MainWindow::search launch a new search.

Parameters

s	text in field

8.51.3.13 int Gui::MainWindow::treeLevel ( )

MainWindow::treeLevel return the level of the node selected in the tree.

Returns

integer, depth of the item in tree

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

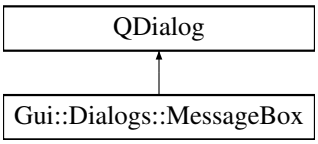
- /home/travis/build/FACT-Team/FactDev/src/mainwindow.h
- /home/travis/build/FACT-Team/FactDev/src/mainwindow.cpp

8.52 Gui::Dialogs::MessageBox Class Reference

The MessageBox class Information window with message.

`#include <messagebox.h>`

Inheritance diagram for Gui::Dialogs::MessageBox:



Public Member Functions

- MessageBox (QWidget *parent=0)
  *MessageBox::MessageBox Construt a **MessageBox***
- void aboutFact ()
  MessageBox::aboutFact Defines FACT team information.
- void aboutFactDev ()
  MessageBox::aboutFactDev Defines FactDev software information.
- void aboutIcons ()
  MessageBox::aboutIcons Defines icons theme information.
- void setImage (QString img, int width=128, int height=128)
  MessageBox::setImage Add the icon img to the current window.
- void setText (QString txt)
  MessageBox::setText Add the text txt to the current window.

Static Public Member Functions

- static void showAboutFact ()
  MessageBox::showAboutFact Shows window about FACT team.
- static void showAboutFactDev ()
  MessageBox::showAboutFactDev Shows window about FactDev software.
- static void showAboutIcons ()
  MessageBox::showAboutIcons Shows about icons theme of FactDev software.

8.52.1 Detailed Description

The MessageBox class Information window with message.

Author

Florent Berbie

8.52.2 Constructor & Destructor Documentation

8.52.2.1 Gui::Dialogs::MessageBox::MessageBox (QWidget * *parent* = 0) [explicit]

MessageBox::MessageBox Construt a **MessageBox**

Parameters

parent	

8.52.3 Member Function Documentation

8.52.3.1 void Gui::Dialogs::MessageBox::setImage (QString *img*, int *width* = 128, int *height* = 128)

MessageBox::setImage Add the icon *img* to the current window.

Parameters

img	Icon
width	Icon width (default: 128)
height	Icon height (default: 128)

8.52.3.2 void Gui::Dialogs::MessageBox::setText (QString *txt*)

MessageBox::setText Add the text *txt* to the current window.

Parameters

txt	Text inside the current window

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

- /home/travis/build/FACT-Team/FactDev/src/gui/dialogs/messagebox.h
- /home/travis/build/FACT-Team/FactDev/src/gui/dialogs/messagebox.cpp

8.53 Parameters Class Reference

The Parameters class Class for simple user parameters.

`#include <parameters.h>`

Static Public Attributes

- static const QString DB_FILENAME = "database.db"

  DB_FILENAME The database sqlite file name.
- static const double VERSION = 0.2

  VERSION Version number of software.

8.53.1 Detailed Description

The Parameters class Class for simple user parameters.

Author

Antoine de Roquemaurel

8.53.2 Member Data Documentation

8.53.2.1 const QString Parameters::DB_FILENAME = "database.db" [static]

DB_FILENAME The database sqlite file name.

Database file name

8.53.2.2 const double Parameters::VERSION = 0.2 [static]

VERSION Version number of software.

Application version

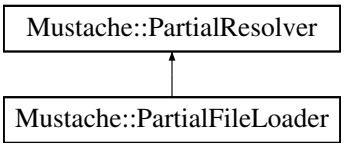
The documentation for this class was generated from the following files:

- /home/travis/build/FACT-Team/FactDev/src/parameters.h
- /home/travis/build/FACT-Team/FactDev/src/parameters.cpp

8.54 Mustache::PartialFileLoader Class Reference

`#include <mustache.h>`

Inheritance diagram for Mustache::PartialFileLoader:



Public Member Functions

- **PartialFileLoader** (const QString &basePath)
- virtual QString getPartial (const QString &name)

8.54.1 Detailed Description

A partial fetcher when loads templates from '<name>.mustache' files in a given directory.

Once a partial has been loaded, it is cached for future use.

8.54.2 Member Function Documentation

8.54.2.1 QString PartialFileLoader::getPartial (const QString & *name*) [virtual]

Returns the partial template with a given `name`.

Implements Mustache::PartialResolver.

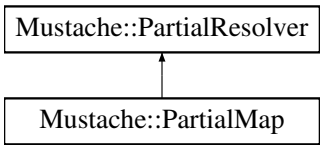
The documentation for this class was generated from the following files:

- /home/travis/build/FACT-Team/FactDev/src/libs/qt-mustache/src/mustache.h
- /home/travis/build/FACT-Team/FactDev/src/libs/qt-mustache/src/mustache.cpp

8.55 Mustache::PartialMap Class Reference

`#include <mustache.h>`

Inheritance diagram for Mustache::PartialMap:



Public Member Functions

- **PartialMap** (const QHash< QString, QString > &partials)
- virtual QString getPartial (const QString &name)

8.55.1 Detailed Description

A simple partial fetcher which returns templates from a map of (partial name -> template)

8.55.2 Member Function Documentation

8.55.2.1 QString PartialMap::getPartial (const QString & *name*) [virtual]

Returns the partial template with a given `name`.

Implements Mustache::PartialResolver.

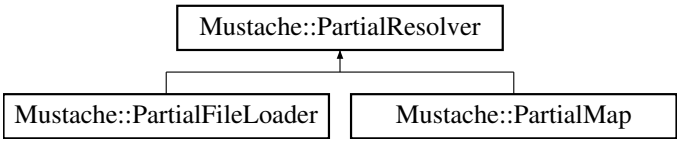
The documentation for this class was generated from the following files:

- /home/travis/build/FACT-Team/FactDev/src/libs/qt-mustache/src/mustache.h
- /home/travis/build/FACT-Team/FactDev/src/libs/qt-mustache/src/mustache.cpp

8.56 Mustache::PartialResolver Class Reference

`#include <mustache.h>`

Inheritance diagram for Mustache::PartialResolver:



Public Member Functions

- virtual QString getPartial (const QString &name)=0

8.56.1 Detailed Description

Interface for fetching template partials.

8.56.2 Member Function Documentation

8.56.2.1 virtual QString Mustache::PartialResolver::getPartial (const QString & *name*) [pure virtual]

Returns the partial template with a given `name`.

Implemented in Mustache::PartialFileLoader, and Mustache::PartialMap.

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

- /home/travis/build/FACT-Team/FactDev/src/libs/qt-mustache/src/mustache.h

8.57 Gui::Widgets::Popup Class Reference

Class for display popup quickly.

`#include <popup.h>`

Static Public Member Functions

- static void toImplement (QString, QWidget *)

  Popup::toImplement Method to display a critical message : feature is not implemented now.

8.57.1 Detailed Description

Class for display popup quickly.

Author

Antoine de Roquemaurel

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

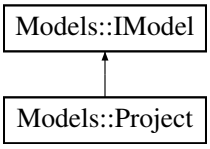
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/popup.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/popup.cpp

8.58 Models::Project Class Reference

The Project class : Project linked to a Customer.

`#include <project.h>`

Inheritance diagram for Models::Project:



Public Member Functions

- Project ()

  Project::Project Construct a Project.
- Project (QString name)

  Project::project Construct a project with a name.
- Project (int id)

  Project::Project Construct a Project which is specified by an id
- virtual ~Project ()

  ~Project Desctruct project object
- void commit ()

  Project::commit Update project data in the database.
- void hydrat (int id)

  Project::hydrat Insert project data which is specified by id in the database.
- void remove ()

  Project::remove Remove the current project.
- QVariantHash getDataMap ()

  getDataMap Get all data of model with a HashMap key/value
- 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
- QDate getBeginDate () const

  *Project::getBeginDate return the date of creation of the **Project***
- void setBeginDate (QDate beginDate)

  *Project::setBeginDate Modify beginDate of a **Project***
- QDate getEndDate () const

  *Project::getEndDate Return the endDate of the **Project***
- void setEndDate (QDate endDate)

  *Project::setEndDate Modify endDate of **Project***
- 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.
- QSharedPointer< Customer > getCustomer () const

  Project::getCustomer Return the reference to the customer linked to this project.
- void setCustomer (QSharedPointer< Customer > customer)

  Project::setCustomer Modify the customer linked to this project.
- bool operator== (const Project &p)

  *Project::operator == Re-define the operator "==" to compare if the current project is the same to the other **Project** p Return TRUE if both projects are the same, else FALSE.*

- bool operator< (const Project &p) const

  *Project::operator < defines the operator "< to compare two **Project** and to see if the fisrt is anterior to the second.*
- bool operator!= (const Project &p)

  *Project::operator == Re-define the operator "!=" to compare if the current project is differnt to the other **Project** p Return TRUE if both projects are different, else FALSE.*

Additional Inherited Members

8.58.1 Detailed Description

The Project class : Project linked to a Customer.

Author

Florent Berbie

See Also

IModel

8.58.2 Constructor & Destructor Documentation

8.58.2.1 Models::Project::Project (int *id*)

Project::Project Construct a Project which is specified by an *id*

Parameters

id	

8.58.3 Member Function Documentation

8.58.3.1 QDate Models::Project::getBeginDate ( ) const

Project::getBeginDate return the date of creation of the **Project**

Returns

the begin date of the Project

8.58.3.2 QSharedPointer< Customer > Models::Project::getCustomer ( ) const

Project::getCustomer Return the reference to the customer linked to this project.

Returns

customer linked to this project

8.58.3.3 double Models::Project::getDailyRate ( ) const

Project::getDailyRate Return the daily rate estimated for this project.

Returns

the daily rate linket to the current project

8.58.3.4 QVariantHash Models::Project::getDataMap ( ) [virtual]

getDataMap Get all data of model with a HashMap key/value

Returns

Model's data

Implements Models::IModel.

8.58.3.5 QString Models::Project::getDescription ( ) const

Project::getDescription Return a project description.

Returns

project description

8.58.3.6 QDate Models::Project::getEndDate ( ) const

Project::getEndDate Return the *endDate* of the **Project**

Returns

the end date of the project

8.58.3.7 QString Models::Project::getName ( ) const

Project::getName Return the project name.

Returns

project name

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

Project::hydrat Insert project data which is specified by *id* in the database.

Parameters

id	Project identify

Implements Models::IModel.

8.58.3.9 bool Models::Project::operator!= (const Project & *p*)

Project::operator == Re-define the operator "!=" to compare if the current project is differnt to the other **Project** *p* Return TRUE if both projects are different, else FALSE.

Parameters

c	Project to compare

Returns

boolean

8.58.3.10 bool Models::Project::operator< (const Project & *p*) const

Project::operator < defines the operator "< to compare two **Project** and to see if the fisrt is anterior to the second.

Parameters

b	the **Project** to compare with the current **Project**

Returns

true if the **Project** are different else false

8.58.3.11 bool Models::Project::operator== (const Project & *p*)

Project::operator == Re-define the operator "==" to compare if the current project is the same to the other **Project** *p* Return TRUE if both projects are the same, else FALSE.

Parameters

c	Project to compare

Returns

boolean

8.58.3.12 void Models::Project::setBeginDate (QDate *beginDate*)

Project::setBeginDate Modify *beginDate* of a **Project**

Parameters

beginDate	the new date of creation of the project

8.58.3.13 void Models::Project::setCustomer (QSharedPointer< Customer > *customer*)

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

Parameters

customer	New customer associated to this project

8.58.3.14 void Models::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

8.58.3.15 void Models::Project::setDescription (const QString & *description*)

Project::setDescription Modify the project *description*

Parameters

description	New project description

8.58.3.16 void Models::Project::setEndDate (QDate *endDate*)

Project::setEndDate Modify *endDate* of **Project**

Parameters

endDate	the new end date of the project

8.58.3.17 void Models::Project::setName (const QString & *name*)

Project::setName Modify the project *name*

Parameters

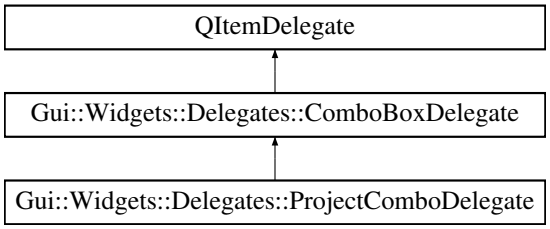
name	Project name

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

- /home/travis/build/FACT-Team/FactDev/src/models/project.h
- /home/travis/build/FACT-Team/FactDev/src/models/project.cpp

8.59 Gui::Widgets::Delegates::ProjectComboDelegate Class Reference

Inheritance diagram for Gui::Widgets::Delegates::ProjectComboDelegate:



Public Member Functions

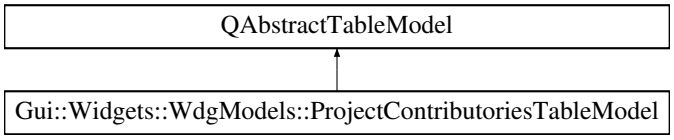
- **ProjectComboDelegate** (QSharedPointer< Models::Customer > c, QObject *parent=0)
- QWidget * **createEditor** (QWidget *parent, const QStyleOptionViewItem &option, const QModelIndex &index) const
- void **paint** (QPainter *painter, const QStyleOptionViewItem &option, const QModelIndex &index) const
- void **removeInCombo** (QList< int > &)
- QMap< int, Models::Project > **getProjects** () const
- bool **getLocked** () const
- void **setLocked** (bool getLlocked)

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

- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/delegates/projectcombodelegate.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/delegates/projectcombodelegate.cpp

8.60 Gui::Widgets::WdgModels::ProjectContributoriesTableModel Class Reference

Inheritance diagram for Gui::Widgets::WdgModels::ProjectContributoriesTableModel:



Public Member Functions

- ProjectContributoriesTableModel (QObject *parent=0)

  ContributoriesTableModel Construct a ContributoriesTableModel.
- int rowCount (const QModelIndex &) const

  rowCount Number of contributories row
- int columnCount (const QModelIndex &) const

  columnCount Number of column of a contributory
- QVariant data (const QModelIndex &index, int role) const

  data Obtains data of a specify cell
- QVariant headerData (int section, Qt::Orientation orientation, int role) const

  headerData Obtains header title of table
- bool setData (const QModelIndex &index, const QVariant &value, int role=Qt::EditRole)

  setData Change data of a cell
- Qt::ItemFlags flags (const QModelIndex &index) const

  flags Differents table flags
- void **append** (QPair< Models::Project *, Models::Rate > p)
- void **append** ()
- bool **allProjectsChose** ()
- QList< int > & **getSelectedProjects** ()
- void **remove** (int index)
- QPair< Models::Project *, Models::Rate > **getProject** (const int row)
- QList< QPair< Models::Project *, Models::Rate > > **getProjects** (void)

8.60.1 Constructor & Destructor Documentation

8.60.1.1 Gui::Widgets::WdgModels::ProjectContributoriesTableModel::ProjectContributoriesTableModel (QObject * *parent* = 0)

ContributoriesTableModel Construct a ContributoriesTableModel.

Parameters

parent	Parent widget

8.60.2 Member Function Documentation

8.60.2.1 int Gui::Widgets::WdgModels::ProjectContributoriesTableModel::columnCount (const QModelIndex &) const

columnCount Number of column of a contributory

Returns

The number of column

8.60.2.2 QVariant Gui::Widgets::WdgModels::ProjectContributoriesTableModel::data (const QModelIndex & *index*, int *role*) const

data Obtains data of a specify cell

Parameters

index	The cell who we want data
role	The role of set

Returns

The data of cell

8.60.2.3 Qt::ItemFlags Gui::Widgets::WdgModels::ProjectContributoriesTableModel::flags (const QModelIndex & *index*) const

flags Differents table flags

Parameters

index	The cell who we want to know flags

Returns

Flags

8.60.2.4 QVariant Gui::Widgets::WdgModels::ProjectContributoriesTableModel::headerData (int *section,* Qt::Orientation *orientation,* int *role*) const

headerData Obtains header title of table

Parameters

section	The number of column
orientation	The table orientation
role	

Returns

The Title header of column

8.60.2.5 int Gui::Widgets::WdgModels::ProjectContributoriesTableModel::rowCount (const QModelIndex &) const

rowCount Number of contributories row

Returns

The number of contributories

8.60.2.6 bool Gui::Widgets::WdgModels::ProjectContributoriesTableModel::setData (const QModelIndex & *index,* const QVariant & *value,* int *role* = `Qt::EditRole`)

setData Change data of a cell

Parameters

index	The cell to change data
value	The new value
role	THe role of cell

Returns

True if we could edit

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

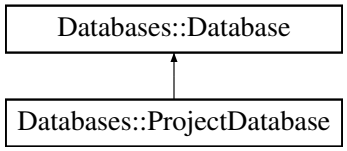
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/widgetsmodels/projectcontributoriestablemodel.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/widgetsmodels/projectcontributoriestablemodel.cpp

8.61 Databases::ProjectDatabase Class Reference

The ProjectDatabase class Project table database.

`#include <projectdatabase.h>`

Inheritance diagram for Databases::ProjectDatabase:



Public Member Functions

- Models::Project * getProject (const int pId)

  ProjectDatabase::getProject Get informations about the project identified by 'pId'.
- Models::Project * getProject (QSqlQuery &q)

  ProjectDatabase::getProject.
- int addProject (const Models::Project &)

  ProjectDatabase:addProject Add the project 'pProject' to the database.
- void updateProject (const Models::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
- QMap< int, Models::Project > getProjectsOfCustomer (QSharedPointer< Models::Customer > c)

  getProjectsOfCustomer Return all projects of a customer
- WdgModels::ProjectsTableModel * getProjectsTable (const int pId) throw (DbException*)

  getProjectsTable Return all projects of a customer in QStandardItemModel
- QSharedPointer< Project > updateProject (QSqlQuery &q)

  getProject Obtain a project without new query
- QList< Project * > getAllProjects ()

  Project::getAllProjects Return the list of projects.

Static Public Member Functions

- static ProjectDatabase * instance () throw (DbException*)

  ProjectDatabase::getInstance Return an instance of ProjectDatabase.

Additional Inherited Members

8.61.1 Detailed Description

The ProjectDatabase class Project table database.

Author

Florent Berbie

See Also

Database
Project

8.61.2 Member Function Documentation

8.61.2.1 int Databases::ProjectDatabase::addProject (const Models::Project & *pProject*)

ProjectDatabase:addProject Add the project 'pProject' to the database.

Returns

project id

8.61.2.2 QList< Project * > Databases::ProjectDatabase::getAllProjects ( )

Project::getAllProjects Return the list of projects.

Returns

Projects list

8.61.2.3 int Databases::ProjectDatabase::getNbProjects ( )

ProjectDatabase:getNbProjects Return the number of projects existing.

Returns

number of projects

8.61.2.4 int Databases::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

8.61.2.5 Models::Project * Databases::ProjectDatabase::getProject (const int *pId*)

ProjectDatabase::getProject Get informations about the project identified by 'pId'.

Parameters

pId	project

Returns

the project

8.61.2.6 Models::Project * Databases::ProjectDatabase::getProject (QSqlQuery & *q*)

ProjectDatabase::getProject.

Parameters

q	

Returns

8.61.2.7 QMap< int, Models::Project > Databases::ProjectDatabase::getProjectsOfCustomer (QSharedPointer< Models::Customer > *c*)

getProjectsOfCustomer Return all projects of a customer

Parameters

c	The customer

Returns

All projects of c with id in key

8.61.2.8 WdgModels::ProjectsTableModel * Databases::ProjectDatabase::getProjectsTable (const int *pId*) throw DbException *)

getProjectsTable Return all projects of a customer in QStandardItemModel

Parameters

filter	Select only projects who are specified by *filter*

Returns

QStandardItemModel an item model for QTableView

8.61.2.9 ProjectDatabase * Databases::ProjectDatabase::instance ( ) throw DbException *) [static]

ProjectDatabase::getInstance Return an instance of ProjectDatabase.

Returns

Instance of ProjectDatabase

8.61.2.10 void Databases::ProjectDatabase::removeProject (const int *pId*)

removeProject Remove the project with the id 'pId'

Parameters

pId	project id

8.61.2.11 QSharedPointer< Models::Project > Databases::ProjectDatabase::updateProject (QSqlQuery & *q*)

getProject Obtain a project without new query

Parameters

q	The query to use

Returns

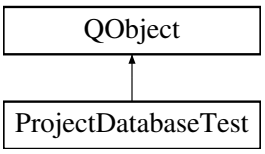
The project linked to q

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

- /home/travis/build/FACT-Team/FactDev/src/database/projectdatabase.h
- /home/travis/build/FACT-Team/FactDev/src/database/projectdatabase.cpp

8.62 ProjectDatabaseTest Class Reference

Inheritance diagram for ProjectDatabaseTest:

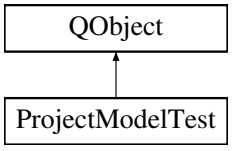


The documentation for this class was generated from the following files:

- /home/travis/build/FACT-Team/FactDev/tests/database/projectdatabasetest.h
- /home/travis/build/FACT-Team/FactDev/tests/database/projectdatabasetest.cpp

8.63 ProjectModelTest Class Reference

Inheritance diagram for ProjectModelTest:



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

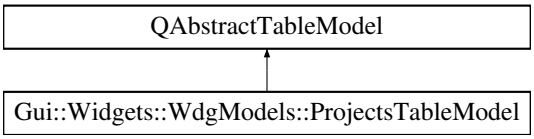
- /home/travis/build/FACT-Team/FactDev/tests/models/projectmodeltest.h
- /home/travis/build/FACT-Team/FactDev/tests/models/projectmodeltest.cpp

8.64 Gui::Widgets::WdgModels::ProjectsTableModel Class Reference

The ProjectsTableModel class for a Project table.

`#include <projectstablemodel.h>`

Inheritance diagram for Gui::Widgets::WdgModels::ProjectsTableModel:



Public Member Functions

- ProjectsTableModel ()

  ProjectsTableModel::ProjectsTableModel Construct a ProjectsTableModel.
- int rowCount (const QModelIndex &) const

  ProjectsTableModel::rowCount Number of projects row.
- int columnCount (const QModelIndex &) const

  ProjectsTableModel::columnCount Number of column of a Project.
- QVariant data (const QModelIndex &index, int role=Qt::DisplayRole) const

  ProjectsTableModel::data Obtains data of a specify cell.
- QVariant headerData (int section, Qt::Orientation orientation, int role=Qt::DisplayRole) const

  ProjectsTableModel::headerData Obtains header title of table.
- bool setData (const QModelIndex &index, const QVariant &value, int role=Qt::EditRole)

  ProjectsTableModel::setData Change data of a cell.
- void append (const Project &project)

  ProjectsTableModel::append Add a new line in table.
- void remove (const int i)

  ProjectsTableModel::remove Remove a line.
- Qt::ItemFlags flags (const QModelIndex &index) const

  ProjectsTableModel::flags Differents table flags.
- int count ()

  ProjectsTableModel::count Number of projects in table.
- QList< Project > getProjects () const

  ProjectsTableModel::getProjects Return the list of projects.

8.64.1 Detailed Description

The ProjectsTableModel class for a Project table.

Author

Florent Berbie

See Also

Project

8.64.2 Constructor & Destructor Documentation

8.64.2.1 Gui::Widgets::WdgModels::ProjectsTableModel::ProjectsTableModel ( )

ProjectsTableModel::ProjectsTableModel Construct a ProjectsTableModel.

Parameters

parent	Parent widget

8.64.3 Member Function Documentation

8.64.3.1 void Gui::Widgets::WdgModels::ProjectsTableModel::append (const Project & *project*)

ProjectsTableModel::append Add a new line in table.

Parameters

Project	The new Project

8.64.3.2 int Gui::Widgets::WdgModels::ProjectsTableModel::columnCount (const QModelIndex &) const

ProjectsTableModel::columnCount Number of column of a Project.

Returns

The number of column

8.64.3.3 int Gui::Widgets::WdgModels::ProjectsTableModel::count ( )

ProjectsTableModel::count Number of projects in table.

Returns

The number of projects

8.64.3.4 QVariant Gui::Widgets::WdgModels::ProjectsTableModel::data (const QModelIndex & *index*, int *role* = `Qt::DisplayRole`) const

ProjectsTableModel::data Obtains data of a specify cell.

Parameters

index	The cell who we want data
role	The role of set

Returns

The data of cell

8.64.3.5 Qt::ItemFlags Gui::Widgets::WdgModels::ProjectsTableModel::flags (const QModelIndex & *index*) const

ProjectsTableModel::flags Differents table flags.

Parameters

index	The cell who we want to know flags

Returns

Flags

8.64.3.6 QList< Project > Gui::Widgets::WdgModels::ProjectsTableModel::getProjects (void) const

ProjectsTableModel::getProjects Return the list of projects.

Returns

list of projects

8.64.3.7 QVariant Gui::Widgets::WdgModels::ProjectsTableModel::headerData (int *section*, Qt::Orientation *orientation*, int *role* = `Qt::DisplayRole`) const

ProjectsTableModel::headerData Obtains header title of table.

Parameters

section	The number of column
orientation	The table orientation
role	

Returns

The Title header of column

8.64.3.8 void Gui::Widgets::WdgModels::ProjectsTableModel::remove (const int *i*)

ProjectsTableModel::remove Remove a line.

Parameters

i	The number of line to remove

8.64.3.9 int Gui::Widgets::WdgModels::ProjectsTableModel::rowCount (const QModelIndex &) const

ProjectsTableModel::rowCount Number of projects row.

Returns

The number of projects

8.64.3.10 bool Gui::Widgets::WdgModels::ProjectsTableModel::setData (const QModelIndex & *index*, const QVariant & *value*, int *role* = `Qt::EditRole`)

ProjectsTableModel::setData Change data of a cell.

Parameters

index	The cell to change data
value	The new value
role	The role of cell

Returns

True if we could edit

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

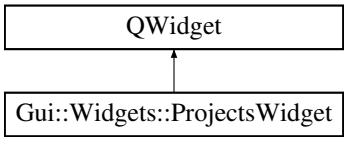
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/widgetsmodels/projectstablemodel.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/widgetsmodels/projectstablemodel.cpp

8.65 Gui::Widgets::ProjectsWidget Class Reference

The ProjectsWidget class Actions on Project.

`#include <projectswidget.h>`

Inheritance diagram for Gui::Widgets::ProjectsWidget:



Public Slots

- void newProject ()

  ProjectsWidget::newProject Event which sends a signal to add a new project.
- void editSelectedProject ()

  ProjectsWidget::editSelectedProject Event which sends a signal to edit the project selected.
- void removeSelectedProject ()

  ProjectsWidget::removeSelectedProject Event which sends a signal to remove the project selected.
- void updateBtn (bool b)

  ProjectsWidget::updateBtn Update the toolbar in tblProjects.

Signals

- void addProject ()

  ProjectsWidget::addProject Add a new project to the current Customer.
- void editProject ()

  ProjectsWidget::editProject Edit the current Customer selected.
- void removeProject ()

  ProjectsWidget::removeProject Remove the current Customer selected.

Public Member Functions

- ProjectsWidget (QWidget ∗parent=0)

  ProjectsWidget::ProjectsWidget Construct a ProjectsWidget.

8.65.1 Detailed Description

The ProjectsWidget class Actions on Project.

Author

Florent Berbie

8.65.2 Constructor & Destructor Documentation

8.65.2.1 Gui::Widgets::ProjectsWidget::ProjectsWidget (QWidget ∗ *parent* = 0) [explicit]

ProjectsWidget::ProjectsWidget Construct a ProjectsWidget.

Parameters

parent	

8.65.3 Member Function Documentation

8.65.3.1 void Gui::Widgets::ProjectsWidget::updateBtn (bool *b*) [slot]

ProjectsWidget::updateBtn Update the toolbar in tblProjects.

Parameters

boolean	if a row is selected

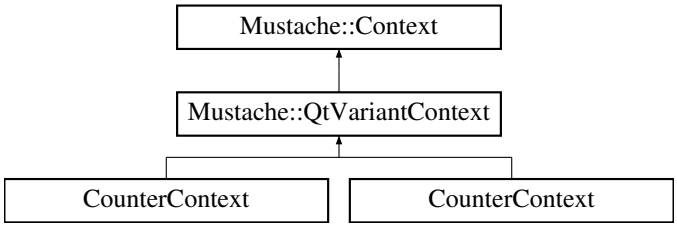
The documentation for this class was generated from the following files:

- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/projectswidget.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/projectswidget.cpp

8.66 Mustache::QtVariantContext Class Reference

`#include <mustache.h>`

Inheritance diagram for Mustache::QtVariantContext:



Public Types

- typedef QString(∗ fn_t)(const QString &, Mustache::Renderer ∗, Mustache::Context ∗)

Public Member Functions

- **QtVariantContext** (const QVariant &root, PartialResolver ∗resolver=0)
- virtual QString stringValue (const QString &key) const
- virtual bool isFalse (const QString &key) const
- virtual int listCount (const QString &key) const
- virtual void push (const QString &key, int index=-1)
- virtual void pop ()
- virtual bool canEval (const QString &key) const
- virtual QString eval (const QString &key, const QString &_template, Mustache::Renderer ∗renderer)

8.66.1 Detailed Description

A context implementation which wraps a QVariantHash or QVariantMap.

8.66.2 Member Typedef Documentation

8.66.2.1 typedef QString(* Mustache::QtVariantContext::fn_t)(const QString &, Mustache::Renderer *, Mustache::Context *)

Construct a QtVariantContext which wraps a dictionary in a QVariantHash or a QVariantMap.

8.66.3 Member Function Documentation

8.66.3.1 bool QtVariantContext::canEval (const QString & *key*) const [virtual]

Returns true if eval() should be used to render section tags using `key`. If canEval() returns true for a key, the renderer will pass the literal, unrendered block of text for the section to eval() and replace the section with the result.

canEval() and eval() are equivalents for callable objects (eg. lambdas) in other Mustache implementations.

The default implementation always returns false.

Reimplemented from Mustache::Context.

Reimplemented in CounterContext, and CounterContext.

8.66.3.2 QString QtVariantContext::eval (const QString & *key,* const QString & *_template,* Mustache::Renderer * *renderer*) [virtual]

Callback used to render a template section with the given `key`. `renderer` will substitute the original section tag with the result of eval().

The default implementation returns an empty string.

Reimplemented from Mustache::Context.

Reimplemented in CounterContext, and CounterContext.

8.66.3.3 bool QtVariantContext::isFalse (const QString & *key*) const [virtual]

Returns true if the value for `key` is 'false' or an empty list. 'False' values typically include empty strings, the boolean value false etc.

When processing a section Mustache tag, the section is not rendered if the key is false, or for an inverted section tag, the section is only rendered if the key is false.

Implements Mustache::Context.

8.66.3.4 int QtVariantContext::listCount (const QString & *key*) const [virtual]

Returns the number of items in the list value for `key` or 0 if the value for `key` is not a list.

Implements Mustache::Context.

8.66.3.5 void QtVariantContext::pop ( ) [virtual]

Exit the current context.

Implements Mustache::Context.

8.66.3.6 void QtVariantContext::push (const QString & *key*, int *index* = -1) [virtual]

Set the current context to the value for `key`. If index is >= 0, set the current context to the `index'th` value in the list value for `key`.

Implements Mustache::Context.

8.66.3.7 QString QtVariantContext::stringValue (const QString & *key*) const [virtual]

Returns a string representation of the value for `key` in the current context. This is used to replace a Mustache value tag.

Implements Mustache::Context.

Reimplemented in CounterContext, and CounterContext.

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

- /home/travis/build/FACT-Team/FactDev/src/libs/qt-mustache/src/mustache.h
- /home/travis/build/FACT-Team/FactDev/src/libs/qt-mustache/src/mustache.cpp

8.67 Models::Rate Class Reference

Public Member Functions

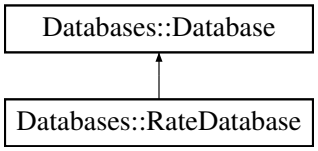
- **Rate** (double hourly)
- **Rate** (int idBilling, int idProject)
- void **setDailyRate** (const double &)
- void **setHourlyRate** (const double &)
- double **getDailyRate** () const
- double **getHourlyRate** () const
- int **getNbDailyHours** () const
- double **getLegalRate** () const

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

- /home/travis/build/FACT-Team/FactDev/src/models/rate.h
- /home/travis/build/FACT-Team/FactDev/src/models/rate.cpp

8.68 Databases::RateDatabase Class Reference

Inheritance diagram for Databases::RateDatabase:



Public Member Functions

- void **addRateProject** (int idProject, int idBilling, double hourlyRate)
- double **getRate** (const int idBilling, const int idProject) const
- void **updateRateProject** (int idProject, int idBilling, double hourlyRate)

Static Public Member Functions

- static RateDatabase * instance () throw (DbException*)

  RateDatabase::getInstance Return an instance of RateDatabase.

Additional Inherited Members

8.68.1 Member Function Documentation

8.68.1.1 RateDatabase * Databases::RateDatabase::instance ( ) throw DbException *) [static]

RateDatabase::getInstance Return an instance of RateDatabase.

Returns

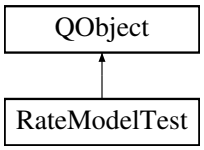
Instance of RateDatabase

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

- /home/travis/build/FACT-Team/FactDev/src/database/ratedatabase.h
- /home/travis/build/FACT-Team/FactDev/src/database/ratedatabase.cpp

8.69 RateModelTest Class Reference

Inheritance diagram for RateModelTest:



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

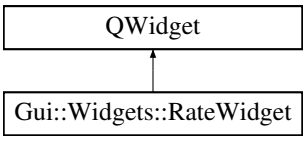
- /home/travis/build/FACT-Team/FactDev/tests/models/ratemodeltest.h
- /home/travis/build/FACT-Team/FactDev/tests/models/ratemodeltest.cpp

8.70 Gui::Widgets::RateWidget Class Reference

Class for display Rate.

`#include <ratewidget.h>`

Inheritance diagram for Gui::Widgets::RateWidget:



Public Slots

- void setDailyRate ()

  setDailyRate Set a new value for the daily rate
- void setHourlyRate ()

  setHourlyRate Set a new value for the hourly rate

Public Member Functions

- RateWidget (QWidget *parent=0)

  RateWidget::RateWidget Construct a rate widget.
- void initRate ()

  RateWidget::initRate Initialize the rate.
- void setWidgetDailyRateValue (double value)

  RateWidget::setWidgetDailyRateValue Modify the value of the daily rate spin box component.
- double getDailyRate ()

  RateWidget::getDailyRate Get the daily rate.
- double getHourlyRate ()

  RateWidget::getHourlyRate Get the hourly rate.
- void setWidgetHourlyRateValue (double value)

  RateWidget::setWidgetHourlyRateValue Modify the value of the hourly rate spin box component.
- void updateConversionRate ()

  updateConversionRate Update daily rate or hourly rate

8.70.1 Detailed Description

Class for display Rate.

Author

Florent Berbie

8.70.2 Constructor & Destructor Documentation

8.70.2.1 Gui::Widgets::RateWidget::RateWidget (QWidget * *parent* = 0) [explicit]

RateWidget::RateWidget Construct a rate widget.

Parameters

parent	The QWidget parent

8.70.3 Member Function Documentation

8.70.3.1 double Gui::Widgets::RateWidget::getDailyRate ( )

RateWidget::getDailyRate Get the daily rate.

Returns

The daily rate

8.70.3.2 double Gui::Widgets::RateWidget::getHourlyRate ( )

RateWidget::getHourlyRate Get the hourly rate.

Returns

The hourly rate

8.70.3.3 void Gui::Widgets::RateWidget::setDailyRate ( ) [slot]

setDailyRate Set a new value for the daily rate

Parameters

dailyRate	The new daily rate

8.70.3.4 void Gui::Widgets::RateWidget::setHourlyRate ( ) [slot]

setHourlyRate Set a new value for the hourly rate

Parameters

hourlyRate	The new hourly rate

8.70.3.5 void Gui::Widgets::RateWidget::setWidgetDailyRateValue (double *value*)

RateWidget::setWidgetDailyRateValue Modify the *value* of the daily rate spin box component.

Parameters

value	New Value

8.70.3.6 void Gui::Widgets::RateWidget::setWidgetHourlyRateValue (double *value*)

RateWidget::setWidgetHourlyRateValue Modify the *value* of the hourly rate spin box component.

Parameters

value	New value

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

- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/ratewidget.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/ratewidget.cpp

8.71 Mustache::Renderer Class Reference

`#include <mustache.h>`

Public Member Functions

- QString render (const QString &_template, Context *context)
- QString error () const
- int errorPos () const

- QString errorPartial () const
- void setTagMarkers (const QString &startMarker, const QString &endMarker)

8.71.1 Detailed Description

Renders Mustache templates, replacing mustache tags with values from a provided context.

8.71.2 Member Function Documentation

8.71.2.1 QString Renderer::error ( ) const

Returns a message describing the last error encountered by the previous render() call.

8.71.2.2 QString Renderer::errorPartial ( ) const

Returns the name of the partial where the error occurred, or an empty string if the error occurred in the main template.

8.71.2.3 int Renderer::errorPos ( ) const

Returns the position in the template where the last error occurred when rendering the template or -1 if no error occurred.

If the error occurred in a partial template, the returned position is the offset in the partial template.

8.71.2.4 QString Renderer::render (const QString & *_template,* Context * *context*)

Render a Mustache template, using `context` to fetch the values used to replace Mustache tags.

8.71.2.5 void Renderer::setTagMarkers (const QString & *startMarker,* const QString & *endMarker*)

Sets the default tag start and end markers. This can be overridden within a template.

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

- /home/travis/build/FACT-Team/FactDev/src/libs/qt-mustache/src/mustache.h
- /home/travis/build/FACT-Team/FactDev/src/libs/qt-mustache/src/mustache.cpp

8.72 Models::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.

- void filterOnVarcharElements (QString &filter, const QStringList list, QString element)

  Search::filterOnVarcharElements Search, for each word of the list from the filter, if it corresponds to the element in the database.
- void filterOnNumberElements (QString &filter, const QStringList list, QString element)

  Search::filterOnNumberElements Search, for each number of the list from the filter, if it corresponds to the element in the database.
- void filterOnCompany (QString &filter, const QStringList list)

  Search::filterOnCompany Search all companies contained in the list of the filter
- void filterOnReferentLastname (QString &filter, const QStringList list)

  Search::filterOnReferentLastname Search all referents last name contained in the list of the filter
- void filterOnProjects (QString &filter, const QStringList list)

  Search::filterOnProjects Search all projects contained in the list of the filter
- void filterOnContributories (QString &filter, const QStringList list)

  Search::filterOnContributories Search all contributories contained in the list of the filter
- void filterOnBillsOrQuotes (QString &filter, const QStringList list)

  Search::filterOnBillsOrQuotes Search all bills or quotes which are contained in the list of the filter
- void filterOnCustomersWithoutProject (QString &filter, const QStringList list)

  Search::filterOnCustomersWithoutProject Search all customers which are contained in the list of the filter
- bool getSearchInCompanies () const

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

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

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

  Search::setSearchInReferentLastname Modify the filter of referents last name search.
- bool getSearchInProjects () const

  Search::getSearchInProjects Return TRUE if it exists one or more projets in database.
- void setSearchInProjects (bool searchInProjects)

  Search::setSearchInProjects Modify the filter of projects search.
- bool searchInContributories () const

  Search::getSearchInContributories Return TRUE if it exists one or more projets in database.
- void setSearchInContributories (bool searchInContributories)

  Search::setSearchInContributories Modify the filter of contributories search.
- bool getSearchInBillsQuotes () const

  Search::searchInBillsQuotes Return TRUE if it exists one or more bills or quotes in database.
- void setSearchInBillsQuotes (bool searchInBillsQuotes)

  Search::setSearchInBillsQuotes Modify the filter of bills and quotes 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.

8.72.1 Detailed Description

The Search class.

Author

Antoine de Roquemaurel
Florent Berbie

8.72.2 Member Function Documentation

8.72.2.1 void Models::Search::filterOnBillsOrQuotes (QString & *filter*, const QStringList *list*)

Search::filterOnBillsOrQuotes Search all bills or quotes which are contained in the *list* of the *filter*

Parameters

filter	Text inputed in searchbar
list	List of bills or quotes

8.72.2.2 void Models::Search::filterOnCompany (QString & *filter*, const QStringList *list*)

Search::filterOnCompany Search all companies contained in the *list* of the *filter*

Parameters

filter	Text inputed in searchbar
list	List of companies

8.72.2.3 void Models::Search::filterOnContributories (QString & *filter*, const QStringList *list*)

Search::filterOnContributories Search all contributories contained in the *list* of the *filter*

Parameters

filter	Text inputed in searchbar
list	List of contributories

8.72.2.4 void Models::Search::filterOnCustomersWithoutProject (QString & *filter*, const QStringList *list*)

Search::filterOnCustomersWithoutProject Search all customers which are contained in the *list* of the *filter*

Parameters

filter	Text inputed in searchbar
list	List of bills or quotes

8.72.2.5 void Models::Search::filterOnNumberElements (QString & *filter*, const QStringList *list*, QString *element*)

Search::filterOnNumberElements Search, for each number of the *list* from the *filter*, if it corresponds to the *element* in the database.

Parameters

filter	Text inputed in the searchbar
list	List of numbers from the *filter*
element	Attribute name into the database

8.72.2.6 void Models::Search::filterOnProjects (QString & *filter,* const QStringList *list*)

Search::filterOnProjects Search all projects contained in the *list* of the *filter*

Parameters

filter	Text inputed in searchbar
list	List of projects

8.72.2.7 void Models::Search::filterOnReferentLastname (QString & *filter,* const QStringList *list*)

Search::filterOnReferentLastname Search all referents last name contained in the *list* of the *filter*

Parameters

filter	Text inputed in searchbar
list	List of referent last name

8.72.2.8 void Models::Search::filterOnVarcharElements (QString & *filter,* const QStringList *list,* QString *element*)

Search::filterOnVarcharElements Search, for each word of the *list* from the *filter*, if it corresponds to the *element* in the database.

Parameters

filter	Text inputed in the searchbar
list	List of words from the *filter*
element	Attribute name into the database

8.72.2.9 QString Models::Search::getFilter ( )

Search::getFilter Return the search filter.

Returns

filter selected (sql portion)

8.72.2.10 bool Models::Search::getGroupFilter ( ) const

Search::getGroupFilter Return if the filter is actived.

Returns

boolean if search filter is actived

8.72.2.11 bool Models::Search::getSearchInBillsQuotes ( ) const

Search::searchInBillsQuotes Return TRUE if it exists one or more bills or quotes in database.

Returns

boolean if bills or quotes are existing

8.72.2.12 bool Models::Search::getSearchInCompanies ( ) const

Search::getSearchInCompanies Return if we search a company.

Returns

boolean if we search a company

8.72.2.13 bool Models::Search::getSearchInProjects ( ) const

Search::getSearchInProjects Return TRUE if it exists one or more projets in database.

Returns

boolean if project are existing

8.72.2.14 bool Models::Search::getSearchInReferentLastname ( ) const

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

Returns

boolean if search concerns the last name of referent

8.72.2.15 QString Models::Search::getText ( ) const

Search::getText Return sql portion of filter.

Returns

QString the sql portion

8.72.2.16 bool Models::Search::searchInContributories ( ) const

Search::getSearchInContributories Return TRUE if it exists one or more projets in database.

Returns

boolean if contributories are existing

8.72.2.17 void Models::Search::setGroupFilter (bool *getGroupFilter*)

Search::setGroupFilter Modify if we active search filter.

Parameters

getGroupFilter	Get if filter is actived

8.72.2.18 void Models::Search::setSearchInBillsQuotes (bool *searchInBillsQuotes*)

Search::setSearchInBillsQuotes Modify the filter of bills and quotes search.

Parameters

searchInBills-Quotes	Search in bills or quotes which are concerned

8.72.2.19 void Models::Search::setSearchInCompanies (bool *searchInCompanies*)

Search::setSearchInCompanies Modify the filter of companies search.

Parameters

getSearchIn-Companies	Search in companies is concerned

8.72.2.20 void Models::Search::setSearchInContributories (bool *searchInContributories*)

Search::setSearchInContributories Modify the filter of contributories search.

Parameters

searchIn-Contributories	Search in contributories which are concerned

8.72.2.21 void Models::Search::setSearchInProjects (bool *searchInProjects*)

Search::setSearchInProjects Modify the filter of projects search.

Parameters

searchInProjects	Search in projects which are concerned

8.72.2.22 void Models::Search::setSearchInReferentLastname (bool *searchInReferentLastname*)

Search::setSearchInReferentLastname Modify the filter of referents last name search.

Parameters

searchIn-Referent-Lastname	Search in referents last name which are concerned

8.72.2.23 void Models::Search::setText (const QString & *getText*)

Search::setText Modify sql portion.

Parameters

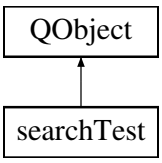
getText	Get sql portion

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

- /home/travis/build/FACT-Team/FactDev/src/models/search.h
- /home/travis/build/FACT-Team/FactDev/src/models/search.cpp

8.73 searchTest Class Reference

Inheritance diagram for searchTest:



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

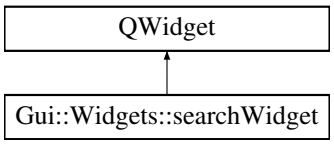
- /home/travis/build/FACT-Team/FactDev/tests/models/searchtest.h
- /home/travis/build/FACT-Team/FactDev/tests/models/searchtest.cpp

8.74 Gui::Widgets::searchWidget Class Reference

Class for search in database.

`#include <searchwidget.h>`

Inheritance diagram for Gui::Widgets::searchWidget:



Public Slots

- void search (QString toSearch)

  searchWidget::launch a search
- void getCustomerData ()

  searchWidget::getCustomerData Return data on the customer selected in the QTableView and display this data (Firstname, Lastname, Company)

Signals

- void **selectCustomer** ()

Public Member Functions

- searchWidget (QWidget *parent=0)

  searchWidget::searchWidget Construct a search widget
- int getCurrentCustomerId ()

  searchWidget::getCurrentCustomerId Return the id of the customer selected in the table
- bool isCustomerSelected () const

  isCustomerSelected Return TRUE if a customer is selected, else FALSE
- void **selectCustomer** (int id)

8.74.1 Detailed Description

Class for search in database.

Author

Antoine de Roquemaurel

8.74.2 Constructor & Destructor Documentation

8.74.2.1 Gui::Widgets::searchWidget::searchWidget (QWidget * *parent* = 0) [explicit]

searchWidget::searchWidget Construct a search widget

Parameters

parent	The QWidget parent

8.74.3 Member Function Documentation

8.74.3.1 int Gui::Widgets::searchWidget::getCurrentCustomerId ( )

searchWidget::getCurrentCustomerId Return the id of the customer selected in the table

Returns

id of the current customer

8.74.3.2 bool Gui::Widgets::searchWidget::isCustomerSelected ( ) const

isCustomerSelected Return TRUE if a customer is selected, else FALSE

Returns

boolean

8.74.3.3 void Gui::Widgets::searchWidget::search (QString *toSearch*) [slot]

searchWidget::launch a search

Parameters

toSearch	The value to search

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

- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/searchwidget.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/searchwidget.cpp

8.75 Utils::String Class Reference

The Utils class.

`#include <string.h>`

Static Public Member Functions

- static QString firstLetterToUpper (QString s)
  firstLetterToUpper Put the first letter of a string in capslock

8.75.1 Detailed Description

The Utils class.

Author

Antoine de Roquemaurel

8.75.2 Member Function Documentation

8.75.2.1 QString Utils::String::firstLetterToUpper (QString *s*) [static]

firstLetterToUpper Put the first letter of a string in capslock

Parameters

s	The string to display

Returns

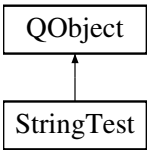
The new string with caps

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

- /home/travis/build/FACT-Team/FactDev/src/utils/string.h
- /home/travis/build/FACT-Team/FactDev/src/utils/string.cpp

8.76 StringTest Class Reference

Inheritance diagram for StringTest:



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

- /home/travis/build/FACT-Team/FactDev/tests/utils/stringtest.h
- /home/travis/build/FACT-Team/FactDev/tests/utils/stringtest.cpp

8.77 Mustache::Tag Struct Reference

`#include <mustache.h>`

Public Types

- enum Type {
  Null, **Value**, SectionStart, InvertedSectionStart,
  SectionEnd, Partial, Comment, SetDelimiter }
- enum **EscapeMode** { **Escape**, **Unescape**, **Raw** }

Public Attributes

- Type **type**
- QString **key**
- int **start**
- int **end**
- EscapeMode **escapeMode**

8.77.1 Detailed Description

Holds properties of a tag in a mustache template.

8.77.2 Member Enumeration Documentation

8.77.2.1 enum Mustache::Tag::Type

Enumerator

SectionStart A {{key}} or {{{key}}} tag.

InvertedSectionStart A {{#section}} tag.

SectionEnd An {{^inverted-section}} tag.

Partial A {{/section}} tag.

Comment A {{^partial}} tag.

SetDelimiter A {{! comment }} tag. A {{=<% %>=}} tag

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

- /home/travis/build/FACT-Team/FactDev/src/libs/qt-mustache/src/mustache.h

8.78 TestAdder< T > Class Template Reference

Public Member Functions

- **TestAdder** (const QString &name)

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

- /home/travis/build/FACT-Team/FactDev/tests/QTestRunner/testadder.cpp

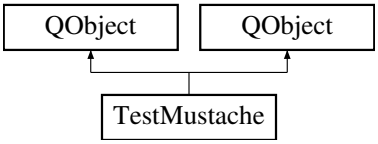
8.79 testadder Class Reference

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

- /home/travis/build/FACT-Team/FactDev/tests/QTestRunner/testadder.h

8.80 TestMustache Class Reference

Inheritance diagram for TestMustache:



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

- /home/travis/build/FACT-Team/FactDev/src/libs/qt-mustache/tests/test_mustache.h
- /home/travis/build/FACT-Team/FactDev/src/libs/qt-mustache/tests/test_mustache.cpp

8.81 TestRunner Class Reference

Public Member Functions

- template<typename T >
  char **RegisterTest** (QString name)
- int **RunAll** (int argc, char *argv[])

Static Public Member Functions

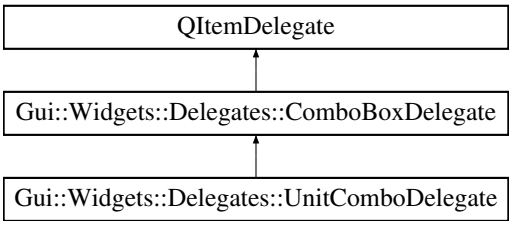
- static TestRunner & **Instance** ()

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

- /home/travis/build/FACT-Team/FactDev/tests/QTestRunner/testrunner.h
- /home/travis/build/FACT-Team/FactDev/tests/QTestRunner/testrunner.cpp

8.82 Gui::Widgets::Delegates::UnitComboDelegate Class Reference

Inheritance diagram for Gui::Widgets::Delegates::UnitComboDelegate:



Public Member Functions

- **UnitComboDelegate** (QObject *parent=0)
- QWidget * **createEditor** (QWidget *parent, const QStyleOptionViewItem &option, const QModelIndex &index) const
- void **paint** (QPainter *painter, const QStyleOptionViewItem &option, const QModelIndex &index) const

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

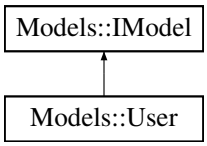
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/delegates/unitcombodelegate.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/delegates/unitcombodelegate.cpp

8.83 Models::User Class Reference

The User class **User** of it application.

`#include <user.h>`

Inheritance diagram for Models::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
- QVariantHash getDataMap ()

  getDataMap Get all data of model with a HashMap key/value
- 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
- QString getNoSiret () const

  User::getNoSiret Return the SIRET number (company registration number)
- void setNoSiret (const QString &noSiret)

  User::setNoSiret Modify the SIRET number (company registration number) noSiret
- QString getWorkspaceName () const

  User::getWorkspaceName Return the name of the workspace user.
- void setWorkspaceName (const QString &workspaceName)

  User::setWorkspaceName Change the current workspace name by the new workspaceName
- QString getWorkspacePath () const

  User::getWorkspacePath Return the path of the workspace user.
- void setWorkspacePath (const QString &workspacePath)

  User::setWorkspacePath Change the current workspace path by the new workspacePath
- bool operator== (const User &u)

*User::operator == Re-define the operator "==" to compare if the current User is the same to the other **User** c Return TRUE if both Users are the same, else FALSE.*

- bool operator!= (const User &u)

  *User::operator == Re-define the operator "!=" to compare if the current User is differnt to the other **User** c Return TRUE if both Users are different, else FALSE.*

Additional Inherited Members

8.83.1 Detailed Description

The User class **User** of it application.

Author

Florent Berbie

8.83.2 Constructor & Destructor Documentation

8.83.2.1 Models::User::User (int *id*)

User::User. Construct a User with the identify *id*

Parameters

id	User id

8.83.3 Member Function Documentation

8.83.3.1 QString Models::User::getAddress ( ) const

User::getAddress Return the company addess (Number and name of street)

Returns

Address company

8.83.3.2 QString Models::User::getCity ( ) const

User::getCity Return the city.

Returns

city

8.83.3.3 QString Models::User::getCompany ( ) const

User::getCompany Return the user company.

Returns

New company name

8.83.3.4 QVariantHash Models::User::getDataMap ( ) [virtual]

getDataMap Get all data of model with a HashMap key/value

Returns

Model's data

Implements Models::IModel.

8.83.3.5 QString Models::User::getEmail ( ) const

User::getEmail Return the user professional *email*

Returns

professional email

8.83.3.6 QString Models::User::getFirstname ( ) const

User::getFirstname Return the user firstname.

Returns

User firstname

8.83.3.7 QString Models::User::getLastname ( ) const

User::getLastname Return the user lastname.

Returns

User lastname

8.83.3.8 QString Models::User::getMobilePhone ( ) const

User::getMobilePhone Return the number of the professional mobile phone.

Returns

number of mobile phone

8.83.3.9 QString Models::User::getNoSiret ( ) const

User::getNoSiret Return the SIRET number (company registration number)

Returns

SIRET number

8.83.3.10 QString Models::User::getPhone ( ) const

User::getPhone Return the number of the desktop phone.

Returns

number of the desktop phone

8.83.3.11 QString Models::User::getPostalCode ( ) const

User::getPostalCode Return the postal code.

Returns

postal code

8.83.3.12 QString Models::User::getTitle ( ) const

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

Returns

a short description of user (company) activity

8.83.3.13 QString Models::User::getWorkspaceName ( ) const

User::getWorkspaceName Return the name of the workspace user.

Returns

workspace name

8.83.3.14 QString Models::User::getWorkspacePath ( ) const

User::getWorkspacePath Return the path of the workspace user.

Returns

workspace path

8.83.3.15 void Models::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 Models::IModel.

8.83.3.16 bool Models::User::operator!= (const User & *u*)

User::operator == Re-define the operator "!=" to compare if the current User is differnt to the other **User** *c* Return TRUE if both Users are different, else FALSE.

Parameters

u	User to compare

Returns

boolean

8.83.3.17 bool Models::User::operator== (const User & *u*)

User::operator == Re-define the operator "==" to compare if the current User is the same to the other **User** *c* Return TRUE if both Users are the same, else FALSE.

Parameters

u	User to compare

Returns

boolean

8.83.3.18 void Models::User::setAddress (const QString & *address*)

User::setAddress Modify the user company *address*

Parameters

address	Company address (name and number of street)

8.83.3.19 void Models::User::setCity (const QString & *city*)

User::setCity Modify the *city*

Parameters

city	Company city address

8.83.3.20 void Models::User::setCompany (const QString & *company*)

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

Parameters

company	New user company name

8.83.3.21 void Models::User::setEmail (const QString & *email*)

User::setEmail Modify the user professional *email*

Parameters

email	The user professional email

8.83.3.22 void Models::User::setFirstname (const QString & *firstname*)

User::setFirstnament Modify the user *firstname*

Parameters

firstname	New user firstname

8.83.3.23 void Models::User::setLastname (const QString & *lastname*)

User::setLastname Modify the user *lastname*

Parameters

lastname	New user lastname

8.83.3.24 void Models::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

8.83.3.25 void Models::User::setNoSiret (const QString & *noSiret*)

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

Parameters

noSiret	SIRET number

8.83.3.26 void Models::User::setPhone (const QString & *phone*)

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

Parameters

phone	Number of the desktop phone

8.83.3.27 void Models::User::setPostalCode (const QString & *postalCode*)

User::setPostalCode Modify the postal code *postalCode*

Parameters

postalCode	New postal code

8.83.3.28 void Models::User::setTitle (const QString & *title*)

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

Parameters

title	Short description on activity(ies) of User company

8.83.3.29 void Models::User::setWorkspaceName (const QString & *workspaceName*)

User::setWorkspaceName Change the current workspace name by the new *workspaceName*

Parameters

workspaceName	

8.83.3.30 void Models::User::setWorkspacePath (const QString & *workspacePath*)

User::setWorkspacePath Change the current workspace path by the new *workspacePath*

Parameters

workspacePath	

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

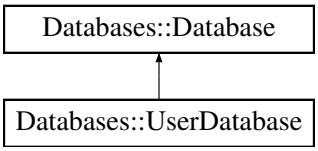
- /home/travis/build/FACT-Team/FactDev/src/models/user.h
- /home/travis/build/FACT-Team/FactDev/src/models/user.cpp

8.84 Databases::UserDatabase Class Reference

The UserDatabase class Access to User data in the the table User of the **Database**

`#include <userdatabase.h>`

Inheritance diagram for Databases::UserDatabase:



Public Member Functions

- Models::User * getUser (const int pId=1)

  getUser Get informations about the user (identified by 'pId')
- void updateUser (const Models::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

8.84.1 Detailed Description

The UserDatabase class Access to User data in the the table User of the **Database**

Author

Florent Berbie

See Also

Database
User

8.84.2 Member Function Documentation

8.84.2.1 Models::User * Databases::UserDatabase::getUser (const int *pId* = 1)

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

Parameters

pId	user id (1 default) *

Returns

the user

8.84.2.2 UserDatabase * Databases::UserDatabase::instance ( ) throw DbException *) [static]

UserDatabase::getInstance Return an instance of UserDatabase.

Returns

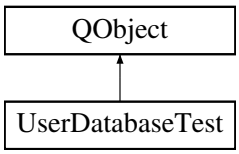
Instance of UserDatabase

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

- /home/travis/build/FACT-Team/FactDev/src/database/userdatabase.h
- /home/travis/build/FACT-Team/FactDev/src/database/userdatabase.cpp

8.85 UserDatabaseTest Class Reference

Inheritance diagram for UserDatabaseTest:



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

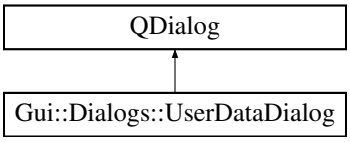
- /home/travis/build/FACT-Team/FactDev/tests/database/userdatabasetest.h
- /home/travis/build/FACT-Team/FactDev/tests/database/userdatabasetest.cpp

8.86 Gui::Dialogs::UserDataDialog Class Reference

The UserDataDialog class Window to fill user data.

`#include <userdatadialog.h>`

Inheritance diagram for Gui::Dialogs::UserDataDialog:



Public Slots

- void checkFields ()

  UserDataDialog::checkFields Check all fields of dialog components.
- void browseWorkspacePath ()

  UserDataDialog::browseWorkspacePath Open a new window to define the workspace path of the user.

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

8.86.1 Detailed Description

The UserDataDialog class Window to fill user data.

Author

Florent Berbie

See Also

Project

8.86.2 Constructor & Destructor Documentation

8.86.2.1 Gui::Dialogs::UserDataDialog::UserDataDialog (QWidget * *parent* = 0) [explicit]

UserDataDialog: Construct a window with user data.

Parameters

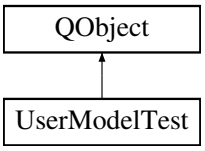
parent	

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

- /home/travis/build/FACT-Team/FactDev/src/gui/dialogs/userdatadialog.h
- /home/travis/build/FACT-Team/FactDev/src/gui/dialogs/userdatadialog.cpp

8.87 UserModelTest Class Reference

Inheritance diagram for UserModelTest:



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

- /home/travis/build/FACT-Team/FactDev/tests/models/usermodeltest.h
- /home/travis/build/FACT-Team/FactDev/tests/models/usermodeltest.cpp

Index

add
    Gui::Widgets::ContributoriesWidget, 58
addBill
    Gui::MainWindow, 100
addBilling
    Databases::BillingDatabase, 29
addBillingProject
    Databases::BillingDatabase, 29
addBillingToProject
    Utils::HierarchicalSystem, 92
addContributory
    Databases::ContributoryDatabase, 62
    Models::Billing, 25
addCustomer
    Databases::CustomerDatabase, 75
addDoc
    Gui::MainWindow, 100
addProject
    Databases::ProjectDatabase, 117
AddProjectDialog
    Gui::Dialogs::AddProjectDialog, 22
addProjectToCustomer
    Utils::HierarchicalSystem, 92
addQuote
    Gui::MainWindow, 101
AddQuoteDialog
    Gui::Dialogs::AddQuoteDialog, 23
append
    Gui::Widgets::WdgModels::BillingsTableModel, 33
    Gui::Widgets::WdgModels::ContributoriesTable-
        Model, 54
    Gui::Widgets::WdgModels::CustomersTableModel,
        80
    Gui::Widgets::WdgModels::ProjectsTableModel,
        121

Billing
    Models::Billing, 25
BillingDatabaseTest, 32
BillingModelTest, 32
BillingsTableModel
    Gui::Widgets::WdgModels::BillingsTableModel, 33

canEval
    CounterContext, 64, 65
    Mustache::Context, 51
    Mustache::QtVariantContext, 125
check
    Gui::Widgets::CheckFields::CheckEmail, 38
    Gui::Widgets::CheckFields::CheckFieldsLetters, 39
    Gui::Widgets::CheckFields::CheckPhone, 42
    Gui::Widgets::CheckFields::CheckPostalCode, 43
    Gui::Widgets::CheckFields::CheckSiretNumber, 46
    Gui::Widgets::CheckFields::CheckUntilField, 47
    Gui::Widgets::CheckFields::CheckValidField, 49
    Gui::Widgets::CheckFields::ICheckField, 93
CheckCity
    Gui::Widgets::CheckFields::CheckCity, 36
CheckCountry
    Gui::Widgets::CheckFields::CheckCountry, 37
CheckEmail
    Gui::Widgets::CheckFields::CheckEmail, 38
CheckFieldsLetters
    Gui::Widgets::CheckFields::CheckFieldsLetters, 39
CheckName
    Gui::Widgets::CheckFields::CheckName, 40
CheckPhone
    Gui::Widgets::CheckFields::CheckPhone, 41
CheckPostalCode
    Gui::Widgets::CheckFields::CheckPostalCode, 43
CheckQLineEdit
    Gui::Widgets::CheckFields::CheckQLineEdit, 45
CheckSiretNumber
    Gui::Widgets::CheckFields::CheckSiretNumber, 46
CheckUntilField
    Gui::Widgets::CheckFields::CheckUntilField, 47
CheckValidField
    Gui::Widgets::CheckFields::CheckValidField, 48
columnCount
    Gui::Widgets::WdgModels::BillingsTableModel, 33
    Gui::Widgets::WdgModels::ContributoriesTable-
        Model, 54
    Gui::Widgets::WdgModels::CustomersTableModel,
        81
    Gui::Widgets::WdgModels::ProjectContributories-
        TableModel, 114
    Gui::Widgets::WdgModels::ProjectsTableModel,
        121
Comment
    Mustache::Tag, 139
Context
    Mustache::Context, 51
ContributoriesDatabaseTest, 52
ContributoriesTableModel
    Gui::Widgets::WdgModels::ContributoriesTable-
        Model, 54
Contributory
    Models::Contributory, 59
ContributoryListTest, 63

ContributoryModelTest, 64
count
Gui::Widgets::WdgModels::BillingsTableModel, 34
Gui::Widgets::WdgModels::ContributoriesTable-Model, 54
Gui::Widgets::WdgModels::CustomersTableModel, 81
Gui::Widgets::WdgModels::ProjectsTableModel, 121
CounterContext, 64
canEval, 64, 65
eval, 65
stringValue, 65
Customer
Models::Customer, 67
CustomerContextualMenu
Gui::Widgets::CustomerContextualMenu, 74
CustomerDatabaseTest, 78
CustomerModelTest, 79
CustomersTableModel
Gui::Widgets::WdgModels::CustomersTableModel, 80

DB_FILENAME
Parameters, 104
data
Gui::Widgets::WdgModels::BillingsTableModel, 34
Gui::Widgets::WdgModels::ContributoriesTable-Model, 55
Gui::Widgets::WdgModels::CustomersTableModel, 81
Gui::Widgets::WdgModels::ProjectContributories-TableModel, 114
Gui::Widgets::WdgModels::ProjectsTableModel, 121
Databases, 17
Databases::BillingDatabase, 28
addBilling, 29
addBillingProject, 29
getAllBillingsOfProject, 30
getBilling, 30
getBillingsTable, 30
getMaxBillingNumber, 30
getMaxQuoteNumber, 31
instance, 31
removeBilling, 31
removeBillingProject, 31
Databases::ContributoryDatabase, 61
addContributory, 62
getContributory, 62, 63
instance, 63
removeContributory, 63
Databases::CustomerDatabase, 74
addCustomer, 75
getCustomer, 75
getCustomersTable, 76
getItemBillQuote, 76
getItemCustomer, 76
getItemProject, 76
getItemRoot, 77
getNbCustomers, 77
getTree, 77
instance, 77
removeCustomer, 77
updateCustomer, 78
Databases::Database, 83
executeFile, 85
instance, 85
lastError, 85
setDatabase, 85
value, 86
Databases::ProjectDatabase, 115
addProject, 117
getAllProjects, 117
getNbProjects, 117
getNbProjectsForACustomer, 117
getProject, 117
getProjectsOfCustomer, 118
getProjectsTable, 118
instance, 118
removeProject, 118
updateProject, 118
Databases::RateDatabase, 126
instance, 127
Databases::UserDatabase, 148
getUser, 149
instance, 149
DbException
Exceptions::DbException, 87
DialogAddCustomer
Gui::Dialogs::DialogAddCustomer, 89

editUser
Gui::MainWindow, 101
error
Mustache::Renderer, 130
errorPartial
Mustache::Renderer, 130
errorPos
Mustache::Renderer, 130
eval
CounterContext, 65
Mustache::Context, 51
Mustache::QtVariantContext, 125
Exceptions::DbException, 86
DbException, 87
popupMessage, 88
Exceptions::FileException, 89
FileException, 90
popupMessage, 90
executeFile
Databases::Database, 85

FileException
Exceptions::FileException, 90
filterOnBillsOrQuotes
Models::Search, 132
filterOnCompany

Models::Search, 132
filterOnContributories
Models::Search, 132
filterOnCustomersWithoutProject
Models::Search, 132
filterOnNumberElements
Models::Search, 132
filterOnProjects
Models::Search, 133
filterOnReferentLastname
Models::Search, 133
filterOnVarcharElements
Models::Search, 133
firstLetterToUpper
Utils::String, 138
flags
Gui::Widgets::WdgModels::BillingsTableModel, 34
Gui::Widgets::WdgModels::ContributoriesTableModel, 55
Gui::Widgets::WdgModels::CustomersTableModel, 81
Gui::Widgets::WdgModels::ProjectContributoriesTableModel, 114
Gui::Widgets::WdgModels::ProjectsTableModel, 121
fn_t
Mustache::QtVariantContext, 125

Generation, 90
Generator, 91
getAddress
Models::Customer, 68
Models::User, 143
getAllBillingsOfProject
Databases::BillingDatabase, 30
getAllProjects
Databases::ProjectDatabase, 117
getBeginDate
Models::Project, 110
getBilling
Databases::BillingDatabase, 30
getBillings
Gui::Widgets::WdgModels::BillingsTableModel, 34
getBillingsTable
Databases::BillingDatabase, 30
getBtnValid
Gui::Widgets::CheckFields::CheckQLineEdit, 45
getCity
Models::Customer, 68
Models::User, 143
getCompany
Models::Customer, 68
Models::User, 143
getContributories
Gui::Widgets::ContributoriesWidget, 58
Gui::Widgets::WdgModels::ContributoriesTableModel, 55
Models::Billing, 25
getContributory
Databases::ContributoryDatabase, 62, 63
getCountry
Gui::Widgets::CheckFields::CheckPhone, 42
Gui::Widgets::CheckFields::CheckPostalCode, 43
Models::Customer, 68
getCurrentCustomerId
Gui::MainWindow, 101
Gui::Widgets::searchWidget, 137
getCurrentCustomerName
Gui::MainWindow, 101
getCurrentProjectId
Gui::MainWindow, 101
getCurrentProjectName
Gui::MainWindow, 101
getCurrentQuoteId
Gui::MainWindow, 101
getCustomer
Databases::CustomerDatabase, 75
Models::Project, 110
getCustomers
Gui::Widgets::WdgModels::CustomersTableModel, 81
Utils::HierarchicalSystem, 92
getCustomersTable
Databases::CustomerDatabase, 76
getDailyRate
Gui::Widgets::RateWidget, 128
Models::Project, 110
getDataMap
Models::Billing, 25
Models::Contributory, 60
Models::Customer, 68
Models::IModel, 94
Models::Project, 110
Models::User, 143
getDate
Models::Billing, 25
getDescription
Models::Billing, 26
Models::Contributory, 60
Models::Project, 110
getEmail
Models::Customer, 68
Models::User, 144
getEndDate
Models::Project, 110
getFax
Models::Customer, 68
getFilter
Models::Search, 133
getFirstname
Models::User, 144
getFirstnameReferent
Models::Customer, 69
getGroupFilter
Models::Search, 133
getHourlyRate
Gui::Widgets::RateWidget, 128

getId
Models::IModel, 94
getItemBillQuote
Databases::CustomerDatabase, 76
getItemCustomer
Databases::CustomerDatabase, 76
getItemProject
Databases::CustomerDatabase, 76
getItemRoot
Databases::CustomerDatabase, 77
getLastname
Models::User, 144
getLastnameReferent
Models::Customer, 69
getMaxBillingNumber
Databases::BillingDatabase, 30
getMaxQuoteNumber
Databases::BillingDatabase, 31
getMobilePhone
Models::Customer, 69
Models::User, 144
getModel
Utils::ItemType, 96
getName
Models::Project, 110
Utils::ItemType, 96
getNameFolder
Models::Customer, 69
getNbCustomers
Databases::CustomerDatabase, 77
getNbHours
Models::Contributory, 60
getNbProjects
Databases::ProjectDatabase, 117
getNbProjectsForACustomer
Databases::ProjectDatabase, 117
getNoSiret
Models::User, 144
getNumber
Gui::Dialogs::AddQuoteDialog, 23
Models::Billing, 26
getPartial
Mustache::PartialFileLoader, 105
Mustache::PartialMap, 106
Mustache::PartialResolver, 106
getPath
Models::Customer, 69
getPhone
Models::Customer, 69
Models::User, 144
getPostalCode
Models::Customer, 69
Models::User, 145
getProject
Databases::ProjectDatabase, 117
Models::Contributory, 60
getProjects
Gui::Widgets::WdgModels::ProjectsTableModel, 121
Utils::HierarchicalSystem, 92
getProjectsOfCustomer
Databases::ProjectDatabase, 118
getProjectsTable
Databases::ProjectDatabase, 118
getSearchInBillsQuotes
Models::Search, 133
getSearchInCompanies
Models::Search, 134
getSearchInProjects
Models::Search, 134
getSearchInReferentLastname
Models::Search, 134
getText
Models::Search, 134
getTitle
Models::Billing, 26
Models::User, 145
getTree
Databases::CustomerDatabase, 77
getType
Utils::ItemType, 96
getUser
Databases::UserDatabase, 149
getWorkspaceName
Models::User, 145
getWorkspacePath
Models::User, 145
Gui, 17
Gui::Dialogs::AddProjectDialog, 21
AddProjectDialog, 22
Gui::Dialogs::AddQuoteDialog, 22
AddQuoteDialog, 23
getNumber, 23
Gui::Dialogs::DialogAddCustomer, 88
DialogAddCustomer, 89
Gui::Dialogs::MessageBox, 102
MessageBox, 103
setImage, 103
setText, 104
Gui::Dialogs::UserDataDialog, 150
UserDataDialog, 150
Gui::MainWindow, 99
addBill, 100
addDoc, 100
addQuote, 101
editUser, 101
getCurrentCustomerId, 101
getCurrentCustomerName, 101
getCurrentProjectId, 101
getCurrentProjectName, 101
getCurrentQuoteId, 101
MainWindow, 100
newProject, 102
rootTree, 102
search, 102

treeLevel, 102
Gui::Widgets, 18
Gui::Widgets::CheckFields::CheckCity, 35
CheckCity, 36
Gui::Widgets::CheckFields::CheckCountry, 36
CheckCountry, 37
Gui::Widgets::CheckFields::CheckEmail, 37
check, 38
CheckEmail, 38
Gui::Widgets::CheckFields::CheckFieldsLetters, 39
check, 39
CheckFieldsLetters, 39
Gui::Widgets::CheckFields::CheckName, 40
CheckName, 40
Gui::Widgets::CheckFields::CheckPhone, 41
check, 42
CheckPhone, 41
getCountry, 42
setCountry, 42
Gui::Widgets::CheckFields::CheckPostalCode, 42
check, 43
CheckPostalCode, 43
getCountry, 43
setCountry, 44
Gui::Widgets::CheckFields::CheckQLineEdit, 44
CheckQLineEdit, 45
getBtnValid, 45
isValid, 45
setBtnValid, 45
Gui::Widgets::CheckFields::CheckSiretNumber, 46
check, 46
CheckSiretNumber, 46
Gui::Widgets::CheckFields::CheckUntilField, 47
check, 47
CheckUntilField, 47
Gui::Widgets::CheckFields::CheckValidField, 48
check, 49
CheckValidField, 48
Gui::Widgets::CheckFields::ICheckField, 92
check, 93
Gui::Widgets::ComboBoxModelWidget, 50
Gui::Widgets::ContributoriesWidget, 57
add, 58
getContributories, 58
Gui::Widgets::CustomerContextualMenu, 73
CustomerContextualMenu, 74
Gui::Widgets::CustomerDataWidget, 78
printInformations, 79
Gui::Widgets::Delegates::ComboBoxDelegate, 49
Gui::Widgets::Delegates::DoubleSpinBoxDelegate, 89
Gui::Widgets::Delegates::ProjectComboDelegate, 112
Gui::Widgets::Delegates::UnitComboDelegate, 140
Gui::Widgets::Popup, 107
Gui::Widgets::ProjectsWidget, 123
ProjectsWidget, 123
updateBtn, 124
Gui::Widgets::RateWidget, 127
getDailyRate, 128
getHourlyRate, 128
RateWidget, 128
setDailyRate, 129
setHourlyRate, 129
setWidgetDailyRateValue, 129
setWidgetHourlyRateValue, 129
Gui::Widgets::WdgModels::BillingsTableModel, 32
append, 33
BillingsTableModel, 33
columnCount, 33
count, 34
data, 34
flags, 34
getBillings, 34
headerData, 34
remove, 35
rowCount, 35
setData, 35
Gui::Widgets::WdgModels::ContributoriesTableModel, 53
append, 54
columnCount, 54
ContributoriesTableModel, 54
count, 54
data, 55
flags, 55
getContributories, 55
headerData, 55
remove, 55
rowCount, 57
setData, 57
Gui::Widgets::WdgModels::CustomersTableModel, 79
append, 80
columnCount, 81
count, 81
CustomersTableModel, 80
data, 81
flags, 81
getCustomers, 81
headerData, 81
remove, 83
rowCount, 83
setData, 83
Gui::Widgets::WdgModels::ProjectContributoriesTableModel, 113
columnCount, 114
data, 114
flags, 114
headerData, 115
ProjectContributoriesTableModel, 114
rowCount, 115
setData, 115
Gui::Widgets::WdgModels::ProjectsTableModel, 119
append, 121
columnCount, 121
count, 121
data, 121
flags, 121

getProjects, 121
headerData, 122
ProjectsTableModel, 120
remove, 122
rowCount, 122
setData, 122
Gui::Widgets::searchWidget, 136
getCurrentCustomerId, 137
isCustomerSelected, 137
search, 137
searchWidget, 137

headerData
Gui::Widgets::WdgModels::BillingsTableModel, 34
Gui::Widgets::WdgModels::ContributoriesTable-Model, 55
Gui::Widgets::WdgModels::CustomersTableModel, 81
Gui::Widgets::WdgModels::ProjectContributories-TableModel, 115
Gui::Widgets::WdgModels::ProjectsTableModel, 122
hydrat
Models::Billing, 26
Models::Contributory, 60
Models::Customer, 70
Models::IModel, 94
Models::Project, 111
Models::User, 145

instance
Databases::BillingDatabase, 31
Databases::ContributoryDatabase, 63
Databases::CustomerDatabase, 77
Databases::Database, 85
Databases::ProjectDatabase, 118
Databases::RateDatabase, 127
Databases::UserDatabase, 149
Utils::Log, 98
InvertedSectionStart
Mustache::Tag, 139
isBilling
Models::Billing, 26
isCustomerSelected
Gui::Widgets::searchWidget, 137
isFalse
Mustache::Context, 51
Mustache::QtVariantContext, 125
isToRemoved
Models::IModel, 94
isValid
Gui::Widgets::CheckFields::CheckQLineEdit, 45
ItemType
Utils::ItemType, 96
ItemTypeTest, 97

lastError
Databases::Database, 85
listCount
Mustache::Context, 51
Mustache::QtVariantContext, 125

MainWindow
Gui::MainWindow, 100
MessageBox
Gui::Dialogs::MessageBox, 103
Models, 18
Models::Billing, 23
addContributory, 25
Billing, 25
getContributories, 25
getDataMap, 25
getDate, 25
getDescription, 26
getNumber, 26
getTitle, 26
hydrat, 26
isBilling, 26
operator<, 27
operator==, 27
setDate, 27
setDescription, 27
setIsBilling, 27
setNumber, 28
setTitle, 28
Models::ContributoriesList, 52
Models::Contributory, 58
Contributory, 59
getDataMap, 60
getDescription, 60
getNbHours, 60
getProject, 60
hydrat, 60
operator==, 61
setDescription, 61
setNbHours, 61
setProject, 61
Models::Customer, 65
Customer, 67
getAddress, 68
getCity, 68
getCompany, 68
getCountry, 68
getDataMap, 68
getEmail, 68
getFax, 68
getFirstnameReferent, 69
getLastnameReferent, 69
getMobilePhone, 69
getNameFolder, 69
getPath, 69
getPhone, 69
getPostalCode, 69
hydrat, 70
operator==, 70
setAddress, 70
setCity, 70
setCompany, 71

setCountry, 71
setEmail, 71
setFax, 71
setFirstnameReferent, 71
setLastnameReferent, 71
setMobilePhone, 71
setPhone, 73
setPostalCode, 73
Models::IModel, 93
getDataMap, 94
getId, 94
hydrat, 94
isToRemoved, 94
setId, 95
setToRemoved, 95
Models::Project, 107
getBeginDate, 110
getCustomer, 110
getDailyRate, 110
getDataMap, 110
getDescription, 110
getEndDate, 110
getName, 110
hydrat, 111
operator<, 111
operator==, 111
Project, 109
setBeginDate, 111
setCustomer, 112
setDailyRate, 112
setDescription, 112
setEndDate, 112
setName, 112
Models::Rate, 126
Models::Search, 130
filterOnBillsOrQuotes, 132
filterOnCompany, 132
filterOnContributories, 132
filterOnCustomersWithoutProject, 132
filterOnNumberElements, 132
filterOnProjects, 133
filterOnReferentLastname, 133
filterOnVarcharElements, 133
getFilter, 133
getGroupFilter, 133
getSearchInBillsQuotes, 133
getSearchInCompanies, 134
getSearchInProjects, 134
getSearchInReferentLastname, 134
getText, 134
searchInContributories, 134
setGroupFilter, 134
setSearchInBillsQuotes, 135
setSearchInCompanies, 135
setSearchInContributories, 135
setSearchInProjects, 135
setSearchInReferentLastname, 135
setText, 135
Models::User, 141
getAddress, 143
getCity, 143
getCompany, 143
getDataMap, 143
getEmail, 144
getFirstname, 144
getLastname, 144
getMobilePhone, 144
getNoSiret, 144
getPhone, 144
getPostalCode, 145
getTitle, 145
getWorkspaceName, 145
getWorkspacePath, 145
hydrat, 145
operator==, 146
setAddress, 146
setCity, 146
setCompany, 146
setEmail, 146
setFirstname, 147
setLastname, 147
setMobilePhone, 147
setNoSiret, 147
setPhone, 147
setPostalCode, 147
setTitle, 147
setWorkspaceName, 148
setWorkspacePath, 148
User, 143
Mustache::Tag
Comment, 139
InvertedSectionStart, 139
Partial, 139
SectionEnd, 139
SectionStart, 139
SetDelimiter, 139
Mustache::Context, 50
canEval, 51
Context, 51
eval, 51
isFalse, 51
listCount, 51
partialResolver, 51
partialValue, 51
pop, 51
push, 52
stringValue, 52
Mustache::PartialFileLoader, 105
getPartial, 105
Mustache::PartialMap, 105
getPartial, 106
Mustache::PartialResolver, 106
getPartial, 106
Mustache::QtVariantContext, 124
canEval, 125
eval, 125

fn_t, 125
isFalse, 125
listCount, 125
pop, 125
push, 125
stringValue, 126
Mustache::Renderer, 129
error, 130
errorPartial, 130
errorPos, 130
render, 130
setTagMarkers, 130
Mustache::Tag, 139
Type, 139

newProject
Gui::MainWindow, 102

operator<
Models::Billing, 27
Models::Project, 111
operator<<
Utils::Log, 98
operator==
Models::Billing, 27
Models::Contributory, 61
Models::Customer, 70
Models::Project, 111
Models::User, 146

Parameters, 104
DB_FILENAME, 104
VERSION, 104
Partial
Mustache::Tag, 139
partialResolver
Mustache::Context, 51
partialValue
Mustache::Context, 51
pop
Mustache::Context, 51
Mustache::QtVariantContext, 125
popupMessage
Exceptions::DbException, 88
Exceptions::FileException, 90
printInformations
Gui::Widgets::CustomerDataWidget, 79
Project
Models::Project, 109
ProjectContributoriesTableModel
Gui::Widgets::WdgModels::ProjectContributories-
TableModel, 114
ProjectDatabaseTest, 119
ProjectModelTest, 119
ProjectsTableModel
Gui::Widgets::WdgModels::ProjectsTableModel,
120
ProjectsWidget
Gui::Widgets::ProjectsWidget, 123

push
Mustache::Context, 52
Mustache::QtVariantContext, 125

RateModelTest, 127
RateWidget
Gui::Widgets::RateWidget, 128
remove
Gui::Widgets::WdgModels::BillingsTableModel, 35
Gui::Widgets::WdgModels::ContributoriesTable-
Model, 55
Gui::Widgets::WdgModels::CustomersTableModel,
83
Gui::Widgets::WdgModels::ProjectsTableModel,
122
removeBilling
Databases::BillingDatabase, 31
removeBillingProject
Databases::BillingDatabase, 31
removeContributory
Databases::ContributoryDatabase, 63
removeCustomer
Databases::CustomerDatabase, 77
removeProject
Databases::ProjectDatabase, 118
render
Mustache::Renderer, 130
rootTree
Gui::MainWindow, 102
rowCount
Gui::Widgets::WdgModels::BillingsTableModel, 35
Gui::Widgets::WdgModels::ContributoriesTable-
Model, 57
Gui::Widgets::WdgModels::CustomersTableModel,
83
Gui::Widgets::WdgModels::ProjectContributories-
TableModel, 115
Gui::Widgets::WdgModels::ProjectsTableModel,
122

search
Gui::MainWindow, 102
Gui::Widgets::searchWidget, 137
searchInContributories
Models::Search, 134
searchTest, 136
searchWidget
Gui::Widgets::searchWidget, 137
SectionEnd
Mustache::Tag, 139
SectionStart
Mustache::Tag, 139
SetDelimiter
Mustache::Tag, 139
setAddress
Models::Customer, 70
Models::User, 146
setBeginDate
Models::Project, 111

setBtnValid
Gui::Widgets::CheckFields::CheckQLineEdit, 45
setCity
Models::Customer, 70
Models::User, 146
setCompany
Models::Customer, 71
Models::User, 146
setCountry
Gui::Widgets::CheckFields::CheckPhone, 42
Gui::Widgets::CheckFields::CheckPostalCode, 44
Models::Customer, 71
setCustomer
Models::Project, 112
setDailyRate
Gui::Widgets::RateWidget, 129
Models::Project, 112
setData
Gui::Widgets::WdgModels::BillingsTableModel, 35
Gui::Widgets::WdgModels::ContributoriesTable-Model, 57
Gui::Widgets::WdgModels::CustomersTableModel, 83
Gui::Widgets::WdgModels::ProjectContributories-TableModel, 115
Gui::Widgets::WdgModels::ProjectsTableModel, 122
setDatabase
Databases::Database, 85
setDate
Models::Billing, 27
setDescription
Models::Billing, 27
Models::Contributory, 61
Models::Project, 112
setEmail
Models::Customer, 71
Models::User, 146
setEndDate
Models::Project, 112
setFax
Models::Customer, 71
setFirstname
Models::User, 147
setFirstnameReferent
Models::Customer, 71
setGroupFilter
Models::Search, 134
setHourlyRate
Gui::Widgets::RateWidget, 129
setId
Models::IModel, 95
setImage
Gui::Dialogs::MessageBox, 103
setIsBilling
Models::Billing, 27
setLastname
Models::User, 147
setLastnameReferent
Models::Customer, 71
setMobilePhone
Models::Customer, 71
Models::User, 147
setName
Models::Project, 112
Utils::ItemType, 96
setNbHours
Models::Contributory, 61
setNoSiret
Models::User, 147
setNumber
Models::Billing, 28
setPhone
Models::Customer, 73
Models::User, 147
setPostalCode
Models::Customer, 73
Models::User, 147
setProject
Models::Contributory, 61
setSearchInBillsQuotes
Models::Search, 135
setSearchInCompanies
Models::Search, 135
setSearchInContributories
Models::Search, 135
setSearchInProjects
Models::Search, 135
setSearchInReferentLastname
Models::Search, 135
setTagMarkers
Mustache::Renderer, 130
setText
Gui::Dialogs::MessageBox, 104
Models::Search, 135
setTitle
Models::Billing, 28
Models::User, 147
setToRemoved
Models::IModel, 95
setType
Utils::ItemType, 97
setWidgetDailyRateValue
Gui::Widgets::RateWidget, 129
setWidgetHourlyRateValue
Gui::Widgets::RateWidget, 129
setWorkspaceName
Models::User, 148
setWorkspacePath
Models::User, 148
StringTest, 138
stringValue
CounterContext, 65
Mustache::Context, 52
Mustache::QtVariantContext, 126

TestAdder< T >, 139

TestMustache, 140
TestRunner, 140
testadder, 140
treeLevel
    Gui::MainWindow, 102
Type
    Mustache::Tag, 139

updateBtn
    Gui::Widgets::ProjectsWidget, 124
updateCustomer
    Databases::CustomerDatabase, 78
updateProject
    Databases::ProjectDatabase, 118
User
    Models::User, 143
UserDataDialog
    Gui::Dialogs::UserDataDialog, 150
UserDatabaseTest, 149
UserModelTest, 151
Utils::HierarchicalSystem, 91
    addBillingToProject, 92
    addProjectToCustomer, 92
    getCustomers, 92
    getProjects, 92
Utils::ItemType, 95
    getModel, 96
    getName, 96
    getType, 96
    ItemType, 96
    setName, 96
    setType, 97
Utils::Log, 97
    instance, 98
    operator<<, 98
    write, 98
Utils::String, 138
    firstLetterToUpper, 138

VERSION
    Parameters, 104
value
    Databases::Database, 86

write
    Utils::Log, 98