

FactDev

0.1

Generated by Doxygen 1.8.6

Tue Mar 10 2015 09:42:09



# Contents

<b>1</b>	<b>FactDev documentation</b>	<b>1</b>
<b>2</b>	<b>README</b>	<b>3</b>
<b>3</b>	<b>QTestRunner</b>	<b>5</b>
<b>4</b>	<b>Namespace Index</b>	<b>7</b>
4.1	Namespace List . . . . .	7
<b>5</b>	<b>Hierarchical Index</b>	<b>9</b>
5.1	Class Hierarchy . . . . .	9
<b>6</b>	<b>Class Index</b>	<b>13</b>
6.1	Class List . . . . .	13
<b>7</b>	<b>Namespace Documentation</b>	<b>17</b>
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
<b>8</b>	<b>Class Documentation</b>	<b>21</b>
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
8.2	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	fillQuoteBilling	23
8.2.3.2	getCopy	23
8.2.3.3	getIdCustomer	24
8.2.3.4	getNumber	24
8.2.3.5	setCopy	24
8.2.3.6	setQuoteIdNumber	24
8.3	Models::Billing Class Reference	24
8.3.1	Detailed Description	26
8.3.2	Constructor & Destructor Documentation	26
8.3.2.1	Billing	26
8.3.3	Member Function Documentation	26
8.3.3.1	addContributory	26
8.3.3.2	getContributories	26
8.3.3.3	getDataMap	27
8.3.3.4	getDate	27
8.3.3.5	getDescription	27
8.3.3.6	getFilename	27
8.3.3.7	getFolder	27
8.3.3.8	getNumber	27
8.3.3.9	getPath	28
8.3.3.10	getSumQuantity	28
8.3.3.11	getSumRate	28
8.3.3.12	getTitle	28
8.3.3.13	hydrat	28
8.3.3.14	isBilling	28
8.3.3.15	isPaid	28
8.3.3.16	operator!=	29
8.3.3.17	operator<	29
8.3.3.18	operator==	29
8.3.3.19	setDate	29
8.3.3.20	setDescription	29
8.3.3.21	setIsBilling	29
8.3.3.22	setIsPaid	30
8.3.3.23	setNumber	30
8.3.3.24	setTitle	30
8.4	Databases::BillingDatabase Class Reference	30
8.4.1	Detailed Description	31
8.4.2	Member Function Documentation	32
8.4.2.1	addBilling	32

8.4.2.2	<a href="#">addBillingProject</a>	32
8.4.2.3	<a href="#">getAllBillingsOfProject</a>	32
8.4.2.4	<a href="#">getBilling</a>	32
8.4.2.5	<a href="#">getBilling</a>	32
8.4.2.6	<a href="#">getBillings</a>	32
8.4.2.7	<a href="#">getBillingsTable</a>	33
8.4.2.8	<a href="#">getMaxBillingNumber</a>	33
8.4.2.9	<a href="#">getMaxBillingNumberOfCustomer</a>	33
8.4.2.10	<a href="#">getMaxQuoteNumber</a>	33
8.4.2.11	<a href="#">getMaxQuoteNumberOfCustomer</a>	33
8.4.2.12	<a href="#">instance</a>	34
8.4.2.13	<a href="#">isBillingPaid</a>	34
8.4.2.14	<a href="#">removeBilling</a>	34
8.4.2.15	<a href="#">removeBillingProject</a>	34
8.5	<a href="#">BillingDatabaseTest Class Reference</a>	35
8.6	<a href="#">BillingModelTest Class Reference</a>	35
8.7	<a href="#">Gui::Widgets::WdgModels::BillingsTableModel Class Reference</a>	35
8.7.1	<a href="#">Detailed Description</a>	36
8.7.2	<a href="#">Constructor &amp; Destructor Documentation</a>	36
8.7.2.1	<a href="#">BillingsTableModel</a>	36
8.7.3	<a href="#">Member Function Documentation</a>	36
8.7.3.1	<a href="#">append</a>	36
8.7.3.2	<a href="#">columnCount</a>	37
8.7.3.3	<a href="#">count</a>	37
8.7.3.4	<a href="#">data</a>	37
8.7.3.5	<a href="#">flags</a>	37
8.7.3.6	<a href="#">getBillings</a>	37
8.7.3.7	<a href="#">headerData</a>	38
8.7.3.8	<a href="#">remove</a>	39
8.7.3.9	<a href="#">rowCount</a>	39
8.7.3.10	<a href="#">setData</a>	39
8.8	<a href="#">Gui::Widgets::CheckFields::CheckCity Class Reference</a>	39
8.8.1	<a href="#">Detailed Description</a>	40
8.8.2	<a href="#">Constructor &amp; Destructor Documentation</a>	40
8.8.2.1	<a href="#">CheckCity</a>	40
8.9	<a href="#">Gui::Widgets::CheckFields::CheckCountry Class Reference</a>	40
8.9.1	<a href="#">Detailed Description</a>	41
8.9.2	<a href="#">Constructor &amp; Destructor Documentation</a>	41
8.9.2.1	<a href="#">CheckCountry</a>	41
8.10	<a href="#">Gui::Widgets::CheckFields::CheckEmail Class Reference</a>	41

8.10.1 Detailed Description . . . . .	42
8.10.2 Constructor & Destructor Documentation . . . . .	42
8.10.2.1 CheckEmail . . . . .	42
8.10.3 Member Function Documentation . . . . .	42
8.10.3.1 check . . . . .	42
8.11 Gui::Widgets::CheckFields::CheckFieldsLetters Class Reference . . . . .	43
8.11.1 Detailed Description . . . . .	43
8.11.2 Constructor & Destructor Documentation . . . . .	43
8.11.2.1 CheckFieldsLetters . . . . .	43
8.11.3 Member Function Documentation . . . . .	43
8.11.3.1 check . . . . .	43
8.12 Gui::Widgets::CheckFields::CheckFieldsNumbers Class Reference . . . . .	44
8.12.1 Detailed Description . . . . .	44
8.12.2 Constructor & Destructor Documentation . . . . .	44
8.12.2.1 CheckFieldsNumbers . . . . .	44
8.12.3 Member Function Documentation . . . . .	45
8.12.3.1 check . . . . .	45
8.13 Gui::Widgets::CheckFields::CheckIpAddress Class Reference . . . . .	45
8.13.1 Detailed Description . . . . .	45
8.13.2 Constructor & Destructor Documentation . . . . .	46
8.13.2.1 CheckIpAddress . . . . .	46
8.13.3 Member Function Documentation . . . . .	46
8.13.3.1 check . . . . .	46
8.14 Gui::Widgets::CheckFields::CheckLogin Class Reference . . . . .	46
8.14.1 Detailed Description . . . . .	47
8.14.2 Constructor & Destructor Documentation . . . . .	47
8.14.2.1 CheckLogin . . . . .	47
8.14.3 Member Function Documentation . . . . .	47
8.14.3.1 check . . . . .	47
8.15 Gui::Widgets::CheckFields::CheckName Class Reference . . . . .	48
8.15.1 Detailed Description . . . . .	48
8.15.2 Constructor & Destructor Documentation . . . . .	48
8.15.2.1 CheckName . . . . .	48
8.16 Gui::Widgets::CheckFields::CheckPhone Class Reference . . . . .	48
8.16.1 Detailed Description . . . . .	49
8.16.2 Constructor & Destructor Documentation . . . . .	49
8.16.2.1 CheckPhone . . . . .	49
8.16.3 Member Function Documentation . . . . .	49
8.16.3.1 check . . . . .	49
8.16.3.2 getCountry . . . . .	50

8.16.3.3	setCountry	50
8.17	Gui::Widgets::CheckFields::CheckPortNumber Class Reference	50
8.17.1	Detailed Description	50
8.17.2	Constructor & Destructor Documentation	51
8.17.2.1	CheckPortNumber	51
8.17.3	Member Function Documentation	51
8.17.3.1	check	51
8.18	Gui::Widgets::CheckFields::CheckPostalCode Class Reference	51
8.18.1	Detailed Description	52
8.18.2	Constructor & Destructor Documentation	52
8.18.2.1	CheckPostalCode	52
8.18.3	Member Function Documentation	52
8.18.3.1	check	52
8.18.3.2	getCountry	52
8.18.3.3	setCountry	52
8.19	Gui::Widgets::CheckFields::CheckQLineEdit Class Reference	53
8.19.1	Detailed Description	53
8.19.2	Constructor & Destructor Documentation	53
8.19.2.1	CheckQLineEdit	53
8.19.3	Member Function Documentation	54
8.19.3.1	getBtnValid	54
8.19.3.2	isValid	54
8.19.3.3	setBtnValid	54
8.20	Gui::Widgets::CheckFields::CheckSiretNumber Class Reference	54
8.20.1	Detailed Description	55
8.20.2	Constructor & Destructor Documentation	55
8.20.2.1	CheckSiretNumber	55
8.20.3	Member Function Documentation	55
8.20.3.1	check	55
8.21	Gui::Widgets::CheckFields::CheckUntilField Class Reference	55
8.21.1	Detailed Description	56
8.21.2	Constructor & Destructor Documentation	56
8.21.2.1	CheckUntilField	56
8.21.3	Member Function Documentation	56
8.21.3.1	check	56
8.22	Gui::Widgets::CheckFields::CheckValidField Class Reference	56
8.22.1	Detailed Description	57
8.22.2	Constructor & Destructor Documentation	57
8.22.2.1	CheckValidField	57
8.22.3	Member Function Documentation	57

8.22.3.1	check	57
8.23	Gui::Widgets::Delegates::ComboBoxDelegate Class Reference	58
8.24	Gui::Widgets::ComboBoxModelWidget Class Reference	58
8.24.1	Detailed Description	58
8.25	Mustache::Context Class Reference	59
8.25.1	Detailed Description	59
8.25.2	Constructor & Destructor Documentation	59
8.25.2.1	Context	59
8.25.3	Member Function Documentation	59
8.25.3.1	canEval	59
8.25.3.2	eval	60
8.25.3.3	isFalse	60
8.25.3.4	listCount	60
8.25.3.5	partialResolver	60
8.25.3.6	partialValue	60
8.25.3.7	pop	60
8.25.3.8	push	60
8.25.3.9	stringValue	60
8.26	ContributoriesDatabaseTest Class Reference	61
8.27	Models::ContributoriesList Class Reference	61
8.27.1	Member Function Documentation	61
8.27.1.1	setAllIdContributories	61
8.28	Gui::Widgets::WdgModels::ContributoriesTableModel Class Reference	62
8.28.1	Detailed Description	63
8.28.2	Constructor & Destructor Documentation	63
8.28.2.1	ContributoriesTableModel	63
8.28.3	Member Function Documentation	63
8.28.3.1	append	63
8.28.3.2	columnCount	63
8.28.3.3	count	63
8.28.3.4	data	63
8.28.3.5	flags	64
8.28.3.6	getContributories	64
8.28.3.7	headerData	64
8.28.3.8	remove	64
8.28.3.9	rowCount	65
8.28.3.10	setData	65
8.29	Gui::Widgets::ContributoriesWidget Class Reference	65
8.29.1	Member Function Documentation	66
8.29.1.1	add	66



8.29.1.2	<a href="#">getContributories</a>	66
8.30	<a href="#">Models::Contributory Class Reference</a>	66
8.30.1	<a href="#">Detailed Description</a>	67
8.30.2	<a href="#">Constructor &amp; Destructor Documentation</a>	67
8.30.2.1	<a href="#">Contributory</a>	67
8.30.3	<a href="#">Member Function Documentation</a>	68
8.30.3.1	<a href="#">getDataMap</a>	68
8.30.3.2	<a href="#">getDescription</a>	68
8.30.3.3	<a href="#">getNbHours</a>	68
8.30.3.4	<a href="#">getProject</a>	68
8.30.3.5	<a href="#">hydrat</a>	68
8.30.3.6	<a href="#">operator!=</a>	68
8.30.3.7	<a href="#">operator==</a>	69
8.30.3.8	<a href="#">setDescription</a>	69
8.30.3.9	<a href="#">setNbHours</a>	69
8.30.3.10	<a href="#">setProject</a>	69
8.31	<a href="#">Databases::ContributoryDatabase Class Reference</a>	69
8.31.1	<a href="#">Detailed Description</a>	70
8.31.2	<a href="#">Member Function Documentation</a>	70
8.31.2.1	<a href="#">addContributory</a>	70
8.31.2.2	<a href="#">getContributoriesByBilling</a>	71
8.31.2.3	<a href="#">getContributoriesByBillingAndProject</a>	72
8.31.2.4	<a href="#">getContributory</a>	72
8.31.2.5	<a href="#">getContributory</a>	72
8.31.2.6	<a href="#">instance</a>	72
8.31.2.7	<a href="#">removeContributory</a>	73
8.32	<a href="#">ContributoryListTest Class Reference</a>	74
8.33	<a href="#">ContributoryModelTest Class Reference</a>	74
8.34	<a href="#">CounterContext Class Reference</a>	74
8.34.1	<a href="#">Member Function Documentation</a>	75
8.34.1.1	<a href="#">canEval</a>	75
8.34.1.2	<a href="#">canEval</a>	75
8.34.1.3	<a href="#">eval</a>	75
8.34.1.4	<a href="#">eval</a>	75
8.34.1.5	<a href="#">stringValue</a>	76
8.34.1.6	<a href="#">stringValue</a>	76
8.35	<a href="#">Models::Customer Class Reference</a>	76
8.35.1	<a href="#">Detailed Description</a>	78
8.35.2	<a href="#">Constructor &amp; Destructor Documentation</a>	78
8.35.2.1	<a href="#">Customer</a>	78

8.35.3	Member Function Documentation	78
8.35.3.1	getAddress	78
8.35.3.2	getCity	78
8.35.3.3	getCompany	78
8.35.3.4	getCountry	78
8.35.3.5	getDataMap	79
8.35.3.6	getEmail	79
8.35.3.7	getFax	79
8.35.3.8	getFirstnameReferent	79
8.35.3.9	getLastnameReferent	79
8.35.3.10	getMobilePhone	79
8.35.3.11	getNameFolder	80
8.35.3.12	getPath	80
8.35.3.13	getPhone	80
8.35.3.14	getPostalCode	80
8.35.3.15	hydrat	80
8.35.3.16	operator!=	80
8.35.3.17	operator==	81
8.35.3.18	setAddress	82
8.35.3.19	setCity	82
8.35.3.20	setCompany	82
8.35.3.21	setCountry	82
8.35.3.22	setEmail	82
8.35.3.23	setFax	82
8.35.3.24	setFirstnameReferent	83
8.35.3.25	setLastnameReferent	83
8.35.3.26	setMobilePhone	83
8.35.3.27	setPhone	83
8.35.3.28	setPostalCode	83
8.36	Gui::Widgets::CustomerContextualMenu Class Reference	83
8.36.1	Detailed Description	84
8.36.2	Constructor & Destructor Documentation	84
8.36.2.1	CustomerContextualMenu	84
8.37	Databases::CustomerDatabase Class Reference	84
8.37.1	Detailed Description	85
8.37.2	Member Function Documentation	86
8.37.2.1	addCustomer	86
8.37.2.2	getCustomer	86
8.37.2.3	getCustomer	86
8.37.2.4	getCustomersTable	86

8.37.2.5	<a href="#">getItemBillQuote</a>	86
8.37.2.6	<a href="#">getItemCustomer</a>	87
8.37.2.7	<a href="#">getItemProject</a>	87
8.37.2.8	<a href="#">getItemRoot</a>	87
8.37.2.9	<a href="#">getNbCustomers</a>	87
8.37.2.10	<a href="#">getTree</a>	87
8.37.2.11	<a href="#">instance</a>	88
8.37.2.12	<a href="#">removeCustomer</a>	88
8.37.2.13	<a href="#">updateCustomer</a>	88
8.38	<a href="#">CustomerDatabaseTest Class Reference</a>	88
8.39	<a href="#">Gui::Widgets::CustomerDataWidget Class Reference</a>	89
8.39.1	<a href="#">Detailed Description</a>	89
8.39.2	<a href="#">Member Function Documentation</a>	89
8.39.2.1	<a href="#">printInformations</a>	89
8.40	<a href="#">CustomerModelTest Class Reference</a>	89
8.41	<a href="#">Gui::Widgets::WdgModels::CustomersTableModel Class Reference</a>	90
8.41.1	<a href="#">Detailed Description</a>	91
8.41.2	<a href="#">Constructor &amp; Destructor Documentation</a>	91
8.41.2.1	<a href="#">CustomersTableModel</a>	91
8.41.3	<a href="#">Member Function Documentation</a>	91
8.41.3.1	<a href="#">append</a>	91
8.41.3.2	<a href="#">columnCount</a>	91
8.41.3.3	<a href="#">count</a>	91
8.41.3.4	<a href="#">data</a>	91
8.41.3.5	<a href="#">flags</a>	92
8.41.3.6	<a href="#">getCustomers</a>	92
8.41.3.7	<a href="#">headerData</a>	92
8.41.3.8	<a href="#">remove</a>	92
8.41.3.9	<a href="#">rowCount</a>	93
8.41.3.10	<a href="#">setData</a>	93
8.42	<a href="#">Databases::Database Class Reference</a>	93
8.42.1	<a href="#">Detailed Description</a>	94
8.42.2	<a href="#">Member Function Documentation</a>	95
8.42.2.1	<a href="#">executeFile</a>	95
8.42.2.2	<a href="#">instance</a>	96
8.42.2.3	<a href="#">lastError</a>	96
8.42.2.4	<a href="#">setDatabase</a>	96
8.42.2.5	<a href="#">value</a>	96
8.43	<a href="#">Gui::Widgets::DatabaseSettingsWidget Class Reference</a>	96
8.43.1	<a href="#">Detailed Description</a>	97

8.43.2	Constructor & Destructor Documentation	98
8.43.2.1	DatabaseSettingsWidget	98
8.43.3	Member Function Documentation	99
8.43.3.1	getDatabaseName	99
8.43.3.2	getDomainNameOrIP	99
8.43.3.3	getLogin	99
8.43.3.4	getPassword	99
8.43.3.5	getPort	99
8.43.3.6	isValid	99
8.44	Exceptions::DbException Class Reference	100
8.44.1	Detailed Description	100
8.44.2	Constructor & Destructor Documentation	100
8.44.2.1	DbException	100
8.44.3	Member Function Documentation	101
8.44.3.1	popupMessage	101
8.45	Gui::Dialogs::DialogAddCustomer Class Reference	102
8.45.1	Detailed Description	102
8.45.2	Constructor & Destructor Documentation	102
8.45.2.1	DialogAddCustomer	102
8.46	Utils::Directories Class Reference	103
8.46.1	Member Function Documentation	103
8.46.1.1	makeDirectory	103
8.47	Gui::Widgets::Delegates::DoubleSpinBoxDelegate Class Reference	103
8.48	Exceptions::FileException Class Reference	104
8.48.1	Detailed Description	104
8.48.2	Constructor & Destructor Documentation	104
8.48.2.1	FileException	104
8.48.3	Member Function Documentation	104
8.48.3.1	popupMessage	104
8.49	Generation Class Reference	105
8.50	Utils::HierarchicalSystem Class Reference	105
8.50.1	Detailed Description	106
8.50.2	Member Function Documentation	106
8.50.2.1	addBillingToProject	106
8.50.2.2	addProjectToCustomer	106
8.50.2.3	getCustomers	106
8.50.2.4	getProjects	106
8.51	Gui::Widgets::CheckFields::ICheckField Class Reference	107
8.51.1	Detailed Description	107
8.51.2	Member Function Documentation	107

8.51.2.1	check	107
8.52	Models::IModel Class Reference	107
8.52.1	Detailed Description	108
8.52.2	Member Function Documentation	108
8.52.2.1	getDataMap	108
8.52.2.2	getId	108
8.52.2.3	hydrat	109
8.52.2.4	isToRemoved	109
8.52.2.5	setId	109
8.52.2.6	setToRemoved	109
8.53	Utils::ItemType Class Reference	109
8.53.1	Detailed Description	110
8.53.2	Constructor & Destructor Documentation	110
8.53.2.1	ItemType	110
8.53.3	Member Function Documentation	110
8.53.3.1	getModel	110
8.53.3.2	getName	110
8.53.3.3	getType	111
8.53.3.4	setName	111
8.53.3.5	setType	111
8.54	ItemTypeTest Class Reference	111
8.55	Utils::Log Class Reference	111
8.55.1	Detailed Description	112
8.55.2	Member Function Documentation	112
8.55.2.1	instance	112
8.55.2.2	write	112
8.55.3	Friends And Related Function Documentation	112
8.55.3.1	operator<<	112
8.56	Gui::MainWindow Class Reference	113
8.56.1	Detailed Description	114
8.56.2	Constructor & Destructor Documentation	114
8.56.2.1	MainWindow	114
8.56.3	Member Function Documentation	115
8.56.3.1	addBill	115
8.56.3.2	addDoc	115
8.56.3.3	addProject	115
8.56.3.4	addQuote	115
8.56.3.5	editUser	115
8.56.3.6	getCurrentCustomerId	116
8.56.3.7	getCurrentCustomerName	116

8.56.3.8	<a href="#">getCurrentProjectId</a>	116
8.56.3.9	<a href="#">getCurrentProjectName</a>	116
8.56.3.10	<a href="#">getCurrentQuotId</a>	116
8.56.3.11	<a href="#">rootTree</a>	116
8.56.3.12	<a href="#">search</a>	116
8.56.3.13	<a href="#">treeLevel</a>	117
8.57	<a href="#">Gui::Dialogs::MessageBox Class Reference</a>	117
8.57.1	<a href="#">Detailed Description</a>	118
8.57.2	<a href="#">Constructor &amp; Destructor Documentation</a>	118
8.57.2.1	<a href="#">MessageBox</a>	118
8.57.3	<a href="#">Member Function Documentation</a>	118
8.57.3.1	<a href="#">setImage</a>	118
8.57.3.2	<a href="#">setText</a>	118
8.58	<a href="#">Parameters Class Reference</a>	118
8.58.1	<a href="#">Detailed Description</a>	119
8.58.2	<a href="#">Member Data Documentation</a>	119
8.58.2.1	<a href="#">DB_FILENAME</a>	119
8.58.2.2	<a href="#">VERSION</a>	119
8.59	<a href="#">Mustache::PartialFileLoader Class Reference</a>	119
8.59.1	<a href="#">Detailed Description</a>	119
8.59.2	<a href="#">Member Function Documentation</a>	119
8.59.2.1	<a href="#">getPartial</a>	119
8.60	<a href="#">Mustache::PartialMap Class Reference</a>	120
8.60.1	<a href="#">Detailed Description</a>	120
8.60.2	<a href="#">Member Function Documentation</a>	120
8.60.2.1	<a href="#">getPartial</a>	120
8.61	<a href="#">Mustache::PartialResolver Class Reference</a>	120
8.61.1	<a href="#">Detailed Description</a>	121
8.61.2	<a href="#">Member Function Documentation</a>	121
8.61.2.1	<a href="#">getPartial</a>	121
8.62	<a href="#">Generator::PdfGenerator Class Reference</a>	121
8.63	<a href="#">Utils::pointers Class Reference</a>	121
8.64	<a href="#">Gui::Widgets::Popup Class Reference</a>	121
8.64.1	<a href="#">Detailed Description</a>	122
8.65	<a href="#">Models::Project Class Reference</a>	122
8.65.1	<a href="#">Detailed Description</a>	123
8.65.2	<a href="#">Constructor &amp; Destructor Documentation</a>	124
8.65.2.1	<a href="#">Project</a>	124
8.65.3	<a href="#">Member Function Documentation</a>	125
8.65.3.1	<a href="#">costCompute</a>	125

8.65.3.2	<a href="#">getBeginDate</a>	125
8.65.3.3	<a href="#">getCost</a>	125
8.65.3.4	<a href="#">getCustomer</a>	125
8.65.3.5	<a href="#">getDailyRate</a>	125
8.65.3.6	<a href="#">getDataMap</a>	125
8.65.3.7	<a href="#">getDescription</a>	126
8.65.3.8	<a href="#">getEndDate</a>	126
8.65.3.9	<a href="#">getName</a>	126
8.65.3.10	<a href="#">hydrat</a>	126
8.65.3.11	<a href="#">operator!=</a>	126
8.65.3.12	<a href="#">operator&lt;</a>	126
8.65.3.13	<a href="#">operator==</a>	127
8.65.3.14	<a href="#">setBeginDate</a>	127
8.65.3.15	<a href="#">setCost</a>	127
8.65.3.16	<a href="#">setCustomer</a>	127
8.65.3.17	<a href="#">setDailyRate</a>	127
8.65.3.18	<a href="#">setDescription</a>	127
8.65.3.19	<a href="#">setEndDate</a>	128
8.65.3.20	<a href="#">setName</a>	128
8.66	<a href="#">Gui::Widgets::Delegates::ProjectComboDelegate Class Reference</a>	128
8.67	<a href="#">Gui::Widgets::WdgModels::ProjectContributoriesTableModel Class Reference</a>	129
8.67.1	<a href="#">Constructor &amp; Destructor Documentation</a>	129
8.67.1.1	<a href="#">ProjectContributoriesTableModel</a>	129
8.67.2	<a href="#">Member Function Documentation</a>	129
8.67.2.1	<a href="#">columnCount</a>	129
8.67.2.2	<a href="#">data</a>	130
8.67.2.3	<a href="#">flags</a>	130
8.67.2.4	<a href="#">headerData</a>	130
8.67.2.5	<a href="#">rowCount</a>	130
8.67.2.6	<a href="#">setData</a>	130
8.68	<a href="#">Databases::ProjectDatabase Class Reference</a>	131
8.68.1	<a href="#">Detailed Description</a>	132
8.68.2	<a href="#">Member Function Documentation</a>	132
8.68.2.1	<a href="#">addProject</a>	132
8.68.2.2	<a href="#">getAllProjects</a>	132
8.68.2.3	<a href="#">getNbProjects</a>	132
8.68.2.4	<a href="#">getNbProjectsForACustomer</a>	132
8.68.2.5	<a href="#">getProject</a>	133
8.68.2.6	<a href="#">getProject</a>	133
8.68.2.7	<a href="#">getProjectsOfCustomer</a>	133

8.68.2.8	<a href="#">getProjectsTable</a>	133
8.68.2.9	<a href="#">instance</a>	134
8.68.2.10	<a href="#">removeProject</a>	134
8.68.2.11	<a href="#">updateProject</a>	134
8.69	<a href="#">ProjectDatabaseTest Class Reference</a>	134
8.70	<a href="#">ProjectModelTest Class Reference</a>	135
8.71	<a href="#">Gui::Widgets::WdgModels::ProjectsTableModel Class Reference</a>	135
8.71.1	<a href="#">Detailed Description</a>	136
8.71.2	<a href="#">Constructor &amp; Destructor Documentation</a>	136
8.71.2.1	<a href="#">ProjectsTableModel</a>	136
8.71.3	<a href="#">Member Function Documentation</a>	136
8.71.3.1	<a href="#">append</a>	136
8.71.3.2	<a href="#">columnCount</a>	136
8.71.3.3	<a href="#">count</a>	136
8.71.3.4	<a href="#">data</a>	136
8.71.3.5	<a href="#">flags</a>	137
8.71.3.6	<a href="#">getProjects</a>	137
8.71.3.7	<a href="#">headerData</a>	137
8.71.3.8	<a href="#">remove</a>	137
8.71.3.9	<a href="#">rowCount</a>	138
8.71.3.10	<a href="#">setData</a>	138
8.72	<a href="#">Gui::Widgets::ProjectsWidget Class Reference</a>	138
8.72.1	<a href="#">Detailed Description</a>	139
8.72.2	<a href="#">Constructor &amp; Destructor Documentation</a>	139
8.72.2.1	<a href="#">ProjectsWidget</a>	139
8.72.3	<a href="#">Member Function Documentation</a>	139
8.72.3.1	<a href="#">updateBtn</a>	139
8.73	<a href="#">Mustache::QtVariantContext Class Reference</a>	139
8.73.1	<a href="#">Detailed Description</a>	140
8.73.2	<a href="#">Member Typedef Documentation</a>	140
8.73.2.1	<a href="#">fn_t</a>	140
8.73.3	<a href="#">Member Function Documentation</a>	140
8.73.3.1	<a href="#">canEval</a>	140
8.73.3.2	<a href="#">eval</a>	140
8.73.3.3	<a href="#">isFalse</a>	141
8.73.3.4	<a href="#">listCount</a>	141
8.73.3.5	<a href="#">pop</a>	141
8.73.3.6	<a href="#">push</a>	141
8.73.3.7	<a href="#">stringValue</a>	141
8.74	<a href="#">Models::Rate Class Reference</a>	141



8.75 Databases::RateDatabase Class Reference . . . . .	142
8.75.1 Member Function Documentation . . . . .	142
8.75.1.1 instance . . . . .	142
8.76 RateModelTest Class Reference . . . . .	143
8.77 Gui::Widgets::RateWidget Class Reference . . . . .	143
8.77.1 Detailed Description . . . . .	144
8.77.2 Constructor & Destructor Documentation . . . . .	144
8.77.2.1 RateWidget . . . . .	144
8.77.3 Member Function Documentation . . . . .	144
8.77.3.1 getDailyRate . . . . .	144
8.77.3.2 getHourlyRate . . . . .	144
8.77.3.3 setDailyRate . . . . .	144
8.77.3.4 setHourlyRate . . . . .	144
8.77.3.5 setWidgetDailyRateValue . . . . .	144
8.77.3.6 setWidgetHourlyRateValue . . . . .	145
8.78 Mustache::Renderer Class Reference . . . . .	145
8.78.1 Detailed Description . . . . .	145
8.78.2 Member Function Documentation . . . . .	145
8.78.2.1 error . . . . .	145
8.78.2.2 errorPartial . . . . .	145
8.78.2.3 errorPos . . . . .	145
8.78.2.4 render . . . . .	146
8.78.2.5 setTagMarkers . . . . .	146
8.79 Models::Search Class Reference . . . . .	146
8.79.1 Detailed Description . . . . .	147
8.79.2 Member Function Documentation . . . . .	147
8.79.2.1 filterOnBillsOrQuotes . . . . .	147
8.79.2.2 filterOnCompany . . . . .	147
8.79.2.3 filterOnContributories . . . . .	147
8.79.2.4 filterOnCustomersWithoutProject . . . . .	148
8.79.2.5 filterOnNumberElements . . . . .	148
8.79.2.6 filterOnProjects . . . . .	148
8.79.2.7 filterOnReferentLastname . . . . .	148
8.79.2.8 filterOnVarcharElements . . . . .	148
8.79.2.9 getFilter . . . . .	149
8.79.2.10 getGroupFilter . . . . .	149
8.79.2.11 getSearchInBillsQuotes . . . . .	149
8.79.2.12 getSearchInCompanies . . . . .	149
8.79.2.13 getSearchInProjects . . . . .	149
8.79.2.14 getSearchInReferentLastname . . . . .	149

8.79.2.15	getText	149
8.79.2.16	searchInContributories	150
8.79.2.17	setGroupFilter	150
8.79.2.18	setSearchInBillsQuotes	150
8.79.2.19	setSearchInCompanies	150
8.79.2.20	setSearchInContributories	150
8.79.2.21	setSearchInProjects	150
8.79.2.22	setSearchInReferentLastname	150
8.79.2.23	setText	151
8.80	Gui::Docks::SearchDock Class Reference	151
8.81	searchTest Class Reference	151
8.82	Gui::Widgets::searchWidget Class Reference	152
8.82.1	Detailed Description	153
8.82.2	Constructor & Destructor Documentation	153
8.82.2.1	searchWidget	153
8.82.3	Member Function Documentation	153
8.82.3.1	getCurrentCustomerId	153
8.82.3.2	isCustomerSelected	153
8.82.3.3	search	153
8.83	Gui::Dialogs::StartedWindowsDialog Class Reference	153
8.83.1	Member Function Documentation	154
8.83.1.1	databaseTypeChanged	154
8.83.1.2	getImage	154
8.83.1.3	isDatabaseCentralized	155
8.83.1.4	isDatabaseTypeValid	155
8.84	Utils::String Class Reference	155
8.84.1	Detailed Description	155
8.84.2	Member Function Documentation	155
8.84.2.1	firstLetterToUpper	155
8.85	StringTest Class Reference	156
8.86	Mustache::Tag Struct Reference	156
8.86.1	Detailed Description	156
8.86.2	Member Enumeration Documentation	157
8.86.2.1	Type	157
8.87	TestAdder< T > Class Template Reference	157
8.88	testadder Class Reference	157
8.89	TestMustache Class Reference	157
8.90	TestRunner Class Reference	158
8.91	Generator::TexGenerator Class Reference	158
8.92	Gui::Widgets::Delegates::UnitComboDelegate Class Reference	158

8.93 Models::User Class Reference . . . . .	159
8.93.1 Detailed Description . . . . .	160
8.93.2 Constructor & Destructor Documentation . . . . .	160
8.93.2.1 User . . . . .	160
8.93.3 Member Function Documentation . . . . .	161
8.93.3.1 getAddress . . . . .	161
8.93.3.2 getCity . . . . .	161
8.93.3.3 getCompany . . . . .	161
8.93.3.4 getDataMap . . . . .	161
8.93.3.5 getEmail . . . . .	161
8.93.3.6 getFirstname . . . . .	161
8.93.3.7 getLastName . . . . .	162
8.93.3.8 getMobilePhone . . . . .	162
8.93.3.9 getNoSiret . . . . .	162
8.93.3.10 getPhone . . . . .	162
8.93.3.11 getPostalCode . . . . .	162
8.93.3.12 getTitle . . . . .	162
8.93.3.13 getWorkspaceName . . . . .	162
8.93.3.14 getWorkspacePath . . . . .	163
8.93.3.15 hydrat . . . . .	163
8.93.3.16 operator!= . . . . .	163
8.93.3.17 operator== . . . . .	163
8.93.3.18 setAddress . . . . .	163
8.93.3.19 setCity . . . . .	163
8.93.3.20 setCompany . . . . .	164
8.93.3.21 setEmail . . . . .	164
8.93.3.22 setFirstname . . . . .	164
8.93.3.23 setLastName . . . . .	164
8.93.3.24 setMobilePhone . . . . .	164
8.93.3.25 setNoSiret . . . . .	164
8.93.3.26 setPhone . . . . .	165
8.93.3.27 setPostalCode . . . . .	166
8.93.3.28 setTitle . . . . .	166
8.93.3.29 setWorkspaceName . . . . .	166
8.93.3.30 setWorkspacePath . . . . .	166
8.93.3.31 updateFolders . . . . .	166
8.94 Databases::UserDatabase Class Reference . . . . .	167
8.94.1 Detailed Description . . . . .	167
8.94.2 Member Function Documentation . . . . .	167
8.94.2.1 getUser . . . . .	167

---

8.94.2.2 instance . . . . .	168
8.95 UserDatabaseTest Class Reference . . . . .	168
8.96 Gui::Dialogs::UserDataDialog Class Reference . . . . .	168
8.96.1 Detailed Description . . . . .	169
8.96.2 Constructor & Destructor Documentation . . . . .	169
8.96.2.1 UserDataDialog . . . . .	169
8.97 UserModelTest Class Reference . . . . .	169
<b>Index</b>	<b>170</b>

# Chapter 1

## FactDev documentation

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

### Team

FACT team is a team for University 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 developed 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 = "<b>{{name}}</b> <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:

<a href="#">Databases</a>		
	Class for <a href="#">Database</a> , contains queries . . . . .	??
<a href="#">Gui</a>	. . . . .	??
<a href="#">Gui::Widgets</a>		
	Contains <a href="#">Widgets</a> classes . . . . .	??
<a href="#">Models</a>		
	<a href="#">Models</a> 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 . . . . .	??
Databases::Database . . . . .	??
Databases::BillingDatabase . . . . .	??
Databases::ContributoryDatabase . . . . .	??
Databases::CustomerDatabase . . . . .	??
Databases::ProjectDatabase . . . . .	??
Databases::RateDatabase . . . . .	??
Databases::UserDatabase . . . . .	??
Utils::Directories . . . . .	??
exception	
Exceptions::DbException . . . . .	??
Exceptions::FileException . . . . .	??
Utils::HierarchicalSystem . . . . .	??
Gui::Widgets::CheckFields::ICheckField . . . . .	??
Gui::Widgets::CheckFields::CheckQLineEdit . . . . .	??
Gui::Widgets::CheckFields::CheckEmail . . . . .	??
Gui::Widgets::CheckFields::CheckPortNumber . . . . .	??
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::CheckFieldsNumbers . . . . .	??
Gui::Widgets::CheckFields::CheckIpAddress . . . . .	??
Gui::Widgets::CheckFields::CheckLogin . . . . .	??
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 . . . . .	??
Generator::PdfGenerator . . . . .	??
Utils::pointers . . . . .	??
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::StartedWindowsDialog . . . . .	??
Gui::Dialogs::UserDataDialog . . . . .	??
QDockWidget	
Gui::Docks::SearchDock . . . . .	??
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 . . . . .	??
ContributoryListTest . . . . .	??
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::DatabaseSettingsWidget . . . . .	??
Gui::Widgets::ProjectsWidget . . . . .	??
Gui::Widgets::RateWidget . . . . .	??
Gui::Widgets::searchWidget . . . . .	??
Models::Rate . . . . .	??
Mustache::Renderer . . . . .	??
Models::Search . . . . .	??
Utils::String . . . . .	??
Mustache::Tag . . . . .	??
TestAdder< T > . . . . .	??
testadder . . . . .	??
TestRunner . . . . .	??
Generator::TexGenerator . . . . .	??





## Chapter 6

# Class Index

### 6.1 Class List

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

<a href="#">Gui::Dialogs::AddProjectDialog</a>	
Windows to add a new Project . . . . .	??
<a href="#">Gui::Dialogs::AddQuoteDialog</a>	
Window to add or modify a Quote . . . . .	??
<a href="#">Models::Billing</a>	
: <a href="#">Billing</a> or Quote of a <a href="#">Customer</a> . . . . .	??
<a href="#">Databases::BillingDatabase</a>	
The <a href="#">BillingDatabase</a> class Billing (or Quote) table database . . . . .	??
<a href="#">BillingDatabaseTest</a> . . . . .	??
<a href="#">BillingModelTest</a> . . . . .	??
<a href="#">Gui::Widgets::WdgModels::BillingsTableModel</a>	
For a Billing table . . . . .	??
<a href="#">Gui::Widgets::CheckFields::CheckCity</a>	
Line Edit of City with a check icon . . . . .	??
<a href="#">Gui::Widgets::CheckFields::CheckCountry</a>	
<a href="#">CheckCountry::CheckCountry</a> Line Edit of country with a check icon . . . . .	??
<a href="#">Gui::Widgets::CheckFields::CheckEmail</a>	
Line Edit of email with a check icon . . . . .	??
<a href="#">Gui::Widgets::CheckFields::CheckFieldsLetters</a>	
Field with only letters (no numbers) . . . . .	??
<a href="#">Gui::Widgets::CheckFields::CheckFieldsNumbers</a>	
Line Edit of number with a check icon . . . . .	??
<a href="#">Gui::Widgets::CheckFields::CheckIpAddress</a>	
Line Edit of IP address with a check icon . . . . .	??
<a href="#">Gui::Widgets::CheckFields::CheckLogin</a>	
Line Edit of login with a check icon . . . . .	??
<a href="#">Gui::Widgets::CheckFields::CheckName</a>	
Line edit of name with a check icon . . . . .	??
<a href="#">Gui::Widgets::CheckFields::CheckPhone</a>	
Line Edit of Phone number with a check icon . . . . .	??
<a href="#">Gui::Widgets::CheckFields::CheckPortNumber</a>	
The <a href="#">CheckFieldsNumbers</a> class Line Edit of number with a check icon . . . . .	??
<a href="#">Gui::Widgets::CheckFields::CheckPostalCode</a>	
Line Edit of postal code with a check icon . . . . .	??
<a href="#">Gui::Widgets::CheckFields::CheckQLineEdit</a>	
LineEdit custom with a check of text inputed . . . . .	??
<a href="#">Gui::Widgets::CheckFields::CheckSiretNumber</a>	
Line Edit with a check icon . . . . .	??

<a href="#">Gui::Widgets::CheckFields::CheckUntilField</a>	
The <a href="#">CheckUntilField</a> class	??
<a href="#">Gui::Widgets::CheckFields::CheckValidField</a>	
Check field not required	??
<a href="#">Gui::Widgets::Delegates::ComboBoxDelegate</a>	??
<a href="#">Gui::Widgets::ComboBoxModelWidget</a>	
The <a href="#">ComboBoxModelWidget</a> class	??
<a href="#">Mustache::Context</a>	??
<a href="#">ContributoriesDatabaseTest</a>	??
<a href="#">Models::ContributoriesList</a>	??
<a href="#">Gui::Widgets::WdgModels::ContributoriesTableModel</a>	
For a custom table for contributories widget	??
<a href="#">Gui::Widgets::ContributoriesWidget</a>	??
<a href="#">Models::Contributory</a>	
The <a href="#">Contributory</a> class	??
<a href="#">Databases::ContributoryDatabase</a>	
The <a href="#">ContributoryDatabase</a> class Contributory (or Quote) table database	??
<a href="#">ContributoryListTest</a>	??
<a href="#">ContributoryModelTest</a>	??
<a href="#">CounterContext</a>	??
<a href="#">Models::Customer</a>	
<a href="#">Customer</a>	??
<a href="#">Gui::Widgets::CustomerContextualMenu</a>	
Display contextual menu on a customer	??
<a href="#">Databases::CustomerDatabase</a>	
The <a href="#">CustomerDatabase</a> class Customer table database	??
<a href="#">CustomerDatabaseTest</a>	??
<a href="#">Gui::Widgets::CustomerDataWidget</a>	
Class for display info of a customer	??
<a href="#">CustomerModelTest</a>	??
<a href="#">Gui::Widgets::WdgModels::CustomersTableModel</a>	
For a customer table	??
<a href="#">Databases::Database</a>	
The <a href="#">Database</a> class Master class for all database access	??
<a href="#">Gui::Widgets::DatabaseSettingsWidget</a>	
Windows of database settings	??
<a href="#">Exceptions::DbException</a>	
For database exception : queries, db file, ...	??
<a href="#">Gui::Dialogs::DialogAddCustomer</a>	
Window to add or modify a Customer	??
<a href="#">Utils::Directories</a>	??
<a href="#">Gui::Widgets::Delegates::DoubleSpinBoxDelegate</a>	??
<a href="#">Exceptions::FileException</a>	
For file/access file exception	??
<a href="#">Generation</a>	??
<a href="#">Utils::HierarchicalSystem</a>	
Create class which contains hierarchical system of FactDev	??
<a href="#">Gui::Widgets::CheckFields::ICheckField</a>	
Interface to check fields validity	??
<a href="#">Models::IModel</a>	
The <a href="#">IModel</a> class	??
<a href="#">Utils::ItemType</a>	
Item type model	??
<a href="#">ItemTypeTest</a>	??
<a href="#">Utils::Log</a>	
For Simple management of log	??
<a href="#">Gui::MainWindow</a>	
Main Window of the software	??

<a href="#">Gui::Dialogs::MessageBox</a>	
Information window with message	??
<a href="#">Parameters</a>	
Class for simple user parameters	??
<a href="#">Mustache::PartialFileLoader</a>	??
<a href="#">Mustache::PartialMap</a>	??
<a href="#">Mustache::PartialResolver</a>	??
<a href="#">Generator::PdfGenerator</a>	??
<a href="#">Utils::pointers</a>	??
<a href="#">Gui::Widgets::Popup</a>	
Class for display popup quickly	??
<a href="#">Models::Project</a>	
: <a href="#">Project</a> linked to a <a href="#">Customer</a>	??
<a href="#">Gui::Widgets::Delegates::ProjectComboDelegate</a>	??
<a href="#">Gui::Widgets::WdgModels::ProjectContributoriesTableModel</a>	??
<a href="#">Databases::ProjectDatabase</a>	
Project table database	??
<a href="#">ProjectDatabaseTest</a>	??
<a href="#">ProjectModelTest</a>	??
<a href="#">Gui::Widgets::WdgModels::ProjectsTableModel</a>	
For a Project table	??
<a href="#">Gui::Widgets::ProjectsWidget</a>	
Actions on Project	??
<a href="#">Mustache::QtVariantContext</a>	??
<a href="#">Models::Rate</a>	??
<a href="#">Databases::RateDatabase</a>	??
<a href="#">RateModelTest</a>	??
<a href="#">Gui::Widgets::RateWidget</a>	
Class for display Rate	??
<a href="#">Mustache::Renderer</a>	??
<a href="#">Models::Search</a>	
The <a href="#">Search</a> class	??
<a href="#">Gui::Docks::SearchDock</a>	??
<a href="#">searchTest</a>	??
<a href="#">Gui::Widgets::searchWidget</a>	
Class for search in database	??
<a href="#">Gui::Dialogs::StartedWindowsDialog</a>	??
<a href="#">Utils::String</a>	
The <a href="#">Utils</a> class	??
<a href="#">StringTest</a>	??
<a href="#">Mustache::Tag</a>	??
<a href="#">TestAdder&lt; T &gt;</a>	??
<a href="#">testadder</a>	??
<a href="#">TestMustache</a>	??
<a href="#">TestRunner</a>	??
<a href="#">Generator::TexGenerator</a>	??
<a href="#">Gui::Widgets::Delegates::UnitComboDelegate</a>	??
<a href="#">Models::User</a>	
<a href="#">User</a> of it application	??
<a href="#">Databases::UserDatabase</a>	
Access to User data in the the table User of the <a href="#">Database</a>	??
<a href="#">UserDatabaseTest</a>	??
<a href="#">Gui::Dialogs::UserDataDialog</a>	
Window to fill user data	??
<a href="#">UserModelTest</a>	??



## 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 [DatabaseSettingsWidget](#)  
*The [DatabaseSettingsWidget](#) class Windows of database settings.*
- 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 idProject=0, int noRowCustomer=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 idProject = 0, int noRowCustomer = 0, QWidget * parent = 0 )`  
`[explicit]`

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

Parameters

<i>noRowCustomer</i>	Row number of the Customer
<i>idProject</i>	Project identify
<i>parent</i>	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)
- void **changeDocType** ()

### Public Member Functions

- [AddQuoteDialog](#) (bool isBilling, int idCustomer=0, int id=0, bool copy=false, 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.
- bool [getCopy](#) () const

- [AddQuoteDialog::getCopy](#) return if [AddQuoteDialog](#) is an add by copy of Billing/quote or if it's edition or add of Billing/quote.
- void [setCopy](#) (bool copy)  
[AddQuoteDialog::setCopy](#) Change the `_copy` value to define if it's a copy of a Billing/quote or if it's a new Billing or a Billing edition.
  - int [getIdCustomer](#) () const  
[AddQuoteDialog::getIdCustomer](#) get the id of the customer changing his quote/billing.
  - void [fillQuoteBilling](#) (bool isBilling)  
[AddQuoteDialog::fillQuoteBilling](#) fill fields for a Billing or a quote.
  - void [fillQuoteBillingCopy](#) (bool isBilling)  
[AddQuoteDialog::fillQuoteBilling](#) fill fields if it's a Billing or a quote for a copy.
  - void [setQuoteIdNumber](#) (int id, int idCustomer, bool isBilling)  
[AddQuoteDialog::setQuoteIdNumber](#) set id and number of the `_quote`

### 8.2.1 Detailed Description

The [AddQuoteDialog](#) class Window to add or modify a Quote.

Author

### 8.2.2 Constructor & Destructor Documentation

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

[AddQuoteDialog::AddQuoteDialog](#) Construct a windows [AddQuoteDialog](#).

Parameters

<i>isBilling</i>	displaying of a quote or a Billing
<i>idCustomer</i>	the id of the Customer
<i>id</i>	the id of the quote or the billing of the Customer's Project
<i>edit</i>	if it's an edition or a copy of a quote/Billing
<i>parent</i>	QWidget of the current windows

### 8.2.3 Member Function Documentation

- 8.2.3.1 `void Gui::Dialogs::AddQuoteDialog::fillQuoteBilling ( bool isBilling )`

[AddQuoteDialog::fillQuoteBilling](#) fill fields for a Billing or a quote.

Parameters

<i>isBilling</i>	if it's a quote or a Billing
------------------	------------------------------

- 8.2.3.2 `bool Gui::Dialogs::AddQuoteDialog::getCopy ( ) const`

[AddQuoteDialog::getCopy](#) return if [AddQuoteDialog](#) is an add by copy of Billing/quote or if it's edition or add of Billing/quote.

Returns

if it's a copy or not

8.2.3.3 `int Gui::Dialogs::AddQuoteDialog::getIdCustomer ( ) const`

[AddQuoteDialog::getIdCustomer](#) get the id of the customer changing his quote/billing.

Returns

the id of the customer

8.2.3.4 `int Gui::Dialogs::AddQuoteDialog::getNumber ( )`

[AddQuoteDialog::getNumber](#) return the number of bill or quote.

Returns

int

8.2.3.5 `void Gui::Dialogs::AddQuoteDialog::setCopy ( bool copy )`

[AddQuoteDialog::setCopy](#) Change the `_copy` value to define if it's a copy of a Billing/quote or if it's a new Billing or a Billing edition.

Parameters

<i>copy</i>	
-------------	--

8.2.3.6 `void Gui::Dialogs::AddQuoteDialog::setQuoteIdNumber ( int id, int idCustomer, bool isBilling )`

[AddQuoteDialog::setQuoteIdNumber](#) set id and number of the `_quote`

Parameters

<i>id</i>	the id of the Billing
<i>idCustomer</i>	the id of the customer to set the <i>number</i>
<i>isBilling</i>	if it's a quote or a Billing

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

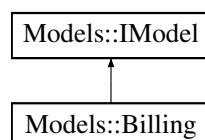
- `/home/travis/build/FACT-Team/FactDev/src/gui/dialogs/addquotedialog.h`
- `/home/travis/build/FACT-Team/FactDev/src/gui/dialogs/addquotedialog.cpp`

## 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
- void [generatePdf](#) ()  
*generatePdf* Generate a .pdf file for the billing
- QString [getPath](#) ()  
*getPath* Return the path of billing filename (without extension)
- QString [getFolder](#) ()  
*getFolder* Return the directory of billing
- QString [getFilename](#) ()  
*getFilename* Return the filename of billing (without extension)
- [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](#) ()  
*Billing::getSumRate.* return [Billing](#) total by money.
- double [getSumQuantity](#) ()  
*Billing::getSumQuantity.* return [Billing](#) total by time.
- 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 **isPaid** () const
- *Billing::isPaid* Return TRUE if thee current billing is paid else return FALSE.
- void **setIsPaid** (bool **isPaid**)
- *Billing::setIsPaid* Define the current billing according the argument **isPaid**
- 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

### 8.3.2 Constructor & Destructor Documentation

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

**Billing::Billing.** Construct a **Billing** or quote.

Parameters

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

### 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

<i>p</i>	The <b>Project</b> who contain <b>Contributory</b>
<i>c</i>	The new <b>Contributory</b>

#### 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 QString Models::Billing::getFilename ( )

getFilename Return the filename of billing (without extension)

Returns

#### 8.3.3.7 QString Models::Billing::getFolder ( )

getFolder Return the directory of billing

Returns

#### 8.3.3.8 int Models::Billing::getNumber ( ) const

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

Returns

\_number of [Billing](#)

#### 8.3.3.9 QString Models::Billing::getPath ( )

getPath Return the path of billing filename (without extension)

Returns

#### 8.3.3.10 double Models::Billing::getSumQuantity ( )

[Billing::getSumQuantity](#). return **Billing** total by time.

Returns

time in hours

#### 8.3.3.11 double Models::Billing::getSumRate ( )

[Billing::getSumRate](#). return **Billing** total by money.

Returns

money

#### 8.3.3.12 QString Models::Billing::getTitle ( ) const

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

Returns

title of **Billing**

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

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

Parameters

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

Implements [Models::IModel](#).

#### 8.3.3.14 bool Models::Billing::isBilling ( ) const

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

Returns

if it's billing or a quote

#### 8.3.3.15 bool Models::Billing::isPaid ( ) const

[Billing::isPaid](#) Return TRUE if thee current billing is paid else return FALSE.

Returns

Boolean



8.3.3.16 `bool Models::Billing::operator!= ( const Billing & b )`

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

Parameters

<i>b</i>	the <code>Billing</code> to compare with the current <code>Billing</code>
----------	---

Returns

true if the `Billing` are different else false

8.3.3.17 `bool Models::Billing::operator< ( const Billing & b ) const`

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

Parameters

<i>b</i>	the <code>Billing</code> to compare with the current <code>Billing</code>
----------	---

Returns

true if the `Billing` are different else false

8.3.3.18 `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

<i>b</i>	the <code>Billing</code> to compare with the current <code>Billing</code>
----------	---

Returns

true if they are the same billings else false

8.3.3.19 `void Models::Billing::setDate ( const QDate & getDate )`

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

Parameters

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

8.3.3.20 `void Models::Billing::setDescription ( const QString & getDescription )`

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

Parameters

<i>getDescription</i>	Modify the description with <i>getDescription</i>
-----------------------	---

8.3.3.21 `void Models::Billing::setIsBilling ( bool isBilling )`

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

## Parameters

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

8.3.3.22 void Models::Billing::setIsPaid ( bool *isPaid* )

[Billing::setIsPaid](#) Define the current billing according the argument *isPaid*

## Parameters

<i>isPaid</i>	Boolean
---------------	---------

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

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

## Parameters

<i>getNumber</i>	the new number of the <a href="#">Billing</a>
------------------	---

8.3.3.24 void Models::Billing::setTitle ( const QString & *getTitle* )

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

## Parameters

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

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

- /home/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 pld)

*BillingDatabase::removeCustomer* Remove the billing with the id pld
- 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.
- bool **isBillingPaid** (const int pld)

*BillingDatabase::isBillingPaid* Return TRUE if the id pld correspond to a Billing and not quote (isBilling = 1) and if this billing is paid (isPaid = 1) else return FALSE.
- 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
- int **getMaxBillingNumberOfCustomer** (const int idCustomer)

*BillingDatabase::getMaxBillingNumberOfCustomer* Get the last number of a **Customer's** Billing.
- int **getMaxQuoteNumberOfCustomer** (const int idCustomer)

*BillingDatabase::getMaxQuoteNumberOfCustomer* Get the last number of a **Customer's** quote.
- QSharedPointer< **Models::Billing** > **getBilling** (QStringQuery &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.
- QList< **Billing** > **getBillings** (const int projectId)

*BillingDatabase::getBillings* get bills by project.

## 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

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

<i>idProject</i>	Project id
<i>idBilling</i>	Billing id
<i>idContributory</i>	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 pld )`

`BillingDatabase::getCustomer` get informations about the billing identified by *pld*

Parameters

<i>pld</i>	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

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

Returns

a billing formed according to QSharedPointer

### 8.4.2.6 `QList< Billing > Databases::BillingDatabase::getBillings ( const int projectId )`

[BillingDatabase::getBillings](#) get bills by project.

## Parameters

<i>projectId</i>	
------------------	--

## Returns

List with bills

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

[BillingDatabase::getBillingsTable](#) Return an item model of billings for QTableView.

## Parameters

<i>pId</i>	the project id of the billings returned
------------	---

## Exceptions

<i>DbException</i>	
--------------------	--

## Returns

QStandardItemModel an item model for QTableView

**8.4.2.8** `int Databases::BillingDatabase::getMaxBillingNumber ( )`

`getMaxBillingNumber` Get the last number of a billing

## Returns

The max number

**8.4.2.9** `int Databases::BillingDatabase::getMaxBillingNumberOfCustomer ( const int idCustomer )`

[BillingDatabase::getMaxBillingNumberOfCustomer](#) Get the last number of a **Customer's** Billing.

## Parameters

<i>idCustomer</i>	Customer id
-------------------	-------------

## Returns

the last number of the **Customer's** Billing

**8.4.2.10** `int Databases::BillingDatabase::getMaxQuoteNumber ( )`

`getMaxQuoteNumber` Get the last number of a quote

## Returns

The last number

**8.4.2.11** `int Databases::BillingDatabase::getMaxQuoteNumberOfCustomer ( const int idCustomer )`

[BillingDatabase::getMaxQuoteNumberOfCustomer](#) Get the last number of a **Customer's** quote.

## Parameters

<i>idCustomer</i>	Customer id
-------------------	-------------

## Returns

the last number of the **Customer's** quote

#### 8.4.2.12 **BillingDatabase** \* **Databases::BillingDatabase::instance** ( ) throw **DbException** \*) [static]

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

## See Also

**DbException**

## Returns

Instance of **BillingDatabase**

#### 8.4.2.13 **bool** **Databases::BillingDatabase::isBillingPaid** ( const int *pld* )

**BillingDatabase::isBillingPaid** Return TRUE if the id *pld* correspond to a Billing and not quote (isBilling = 1) and if this billing is paid (isPaid = 1) else return FALSE.

## Parameters

<i>pld</i>	Billing id
------------	------------

## Returns

TRUE if billing is paid

#### 8.4.2.14 **void** **Databases::BillingDatabase::removeBilling** ( const int *pld* )

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

## Parameters

<i>pld</i>	billing id
------------	------------

#### 8.4.2.15 **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

<i>idProject</i>	Project id
<i>idBilling</i>	Billing id

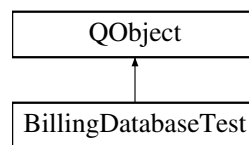
<i>idContributory</i>	Contributory id
-----------------------	-----------------

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:

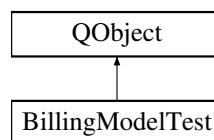


The documentation for this class was generated from the following files:

- /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:

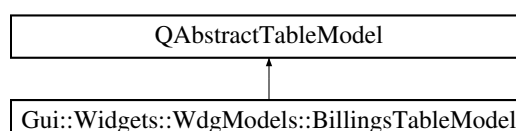
- /home/travis/build/FACT-Team/FactDev/tests/models/billingmodeltest.h
- /home/travis/build/FACT-Team/FactDev/tests/models/billingmodeltest.cpp

## 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

<i>parent</i>	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

<i>Billing</i>	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

<i>index</i>	The cell who we want data
<i>role</i>	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

<i>index</i>	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

<i>section</i>	The number of column
<i>orientation</i>	The table orientation
<i>role</i>	

## 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>i</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

<i>index</i>	The cell to change data
<i>value</i>	The new value
<i>role</i>	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/widgetmodels/billingstablemodel.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/widgetmodels/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

<i>w</i>	QWidget linked to <a href="#">CheckCity</a>
----------	---

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

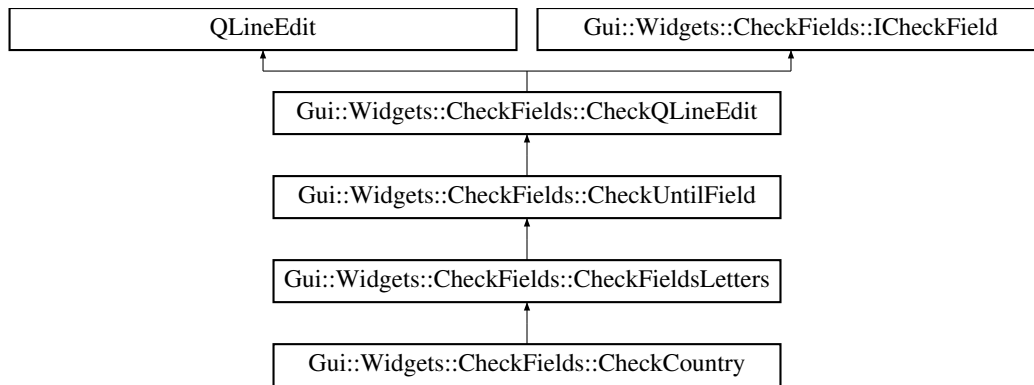
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkcity.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkcity.cpp

## 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

<i>w</i>	QWidget linked to <a href="#">CheckCountry</a>
----------	--

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

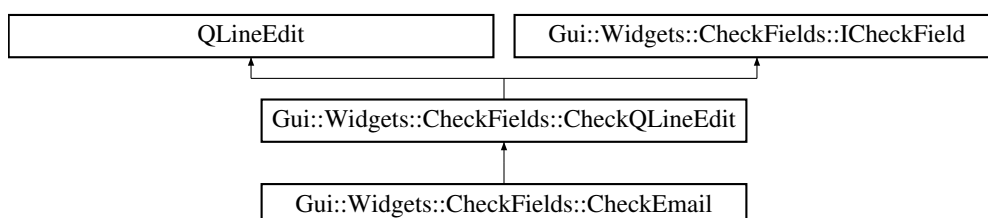
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkcountry.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkcountry.cpp

## 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

<i>w</i>	<a href="#">QWidget</a> linked to <a href="#">CheckEmail</a>
----------	--

### 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

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

Returns

*boolean*

Implements [Gui::Widgets::CheckFields::ICheckField](#).

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

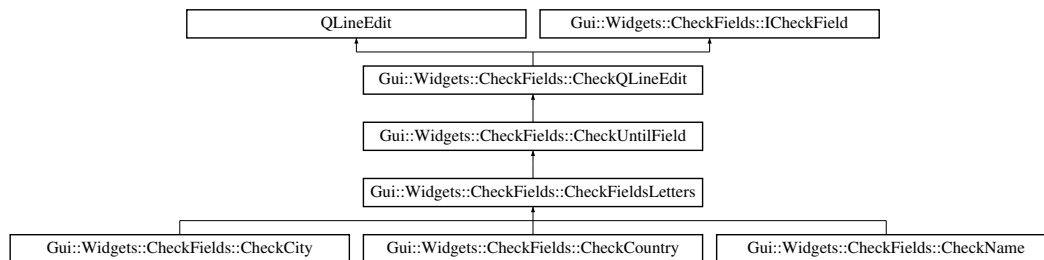
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkemail.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkemail.cpp

## 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)  
[CheckFieldsLetters::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

<i>w</i>	QWidget linked to <a href="#">CheckFieldsLetters</a>
----------	--

#### 8.11.3 Member Function Documentation

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

[CheckFieldsLetters::check](#) Check if the field contains only letters.

Parameters

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

**Returns**

boolean

Implements [Gui::Widgets::CheckFields::ICheckField](#).

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

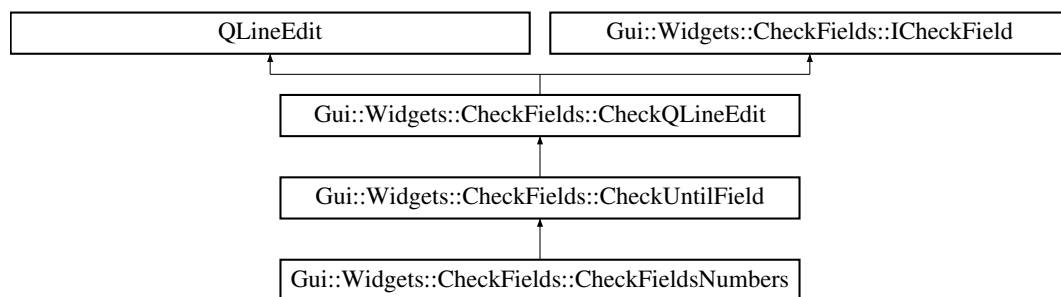
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkfieldsletters.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkfieldsletters.cpp

## 8.12 Gui::Widgets::CheckFields::CheckFieldsNumbers Class Reference

The [CheckFieldsNumbers](#) class Line Edit of number with a check icon.

```
#include <checkfieldsnumbers.h>
```

Inheritance diagram for Gui::Widgets::CheckFields::CheckFieldsNumbers:

**Public Member Functions**

- [CheckFieldsNumbers](#) (QWidget \*w=0, QPushButton \*btn=0)  
*CheckFieldsNumbers::CheckFieldsNumbers* Construct a [CheckFieldsNumbers](#).
- bool [check](#) (QString text)  
*CheckFieldsNumbers::check* Check if the field contains only numbers.

**Additional Inherited Members**

### 8.12.1 Detailed Description

The [CheckFieldsNumbers](#) class Line Edit of number with a check icon.

**Author**

Florent BERBIE

**See Also**

[CheckQLineEdit](#)  
[CheckUntilField](#)

### 8.12.2 Constructor & Destructor Documentation

8.12.2.1 [Gui::Widgets::CheckFields::CheckFieldsNumbers::CheckFieldsNumbers](#) ( [QWidget](#) \* *w* = 0, [QPushButton](#) \* *btn* = 0 )

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



## Parameters

<i>w</i>	QWidget linked to <a href="#">CheckFieldsNumbers</a>
----------	--

## 8.12.3 Member Function Documentation

8.12.3.1 bool Gui::Widgets::CheckFields::CheckFieldsNumbers::check ( QString *text* ) [virtual]

[CheckFieldsNumbers::check](#) Check if the field contains only numbers.

## Parameters

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

## Returns

boolean

Implements [Gui::Widgets::CheckFields::ICheckField](#).

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

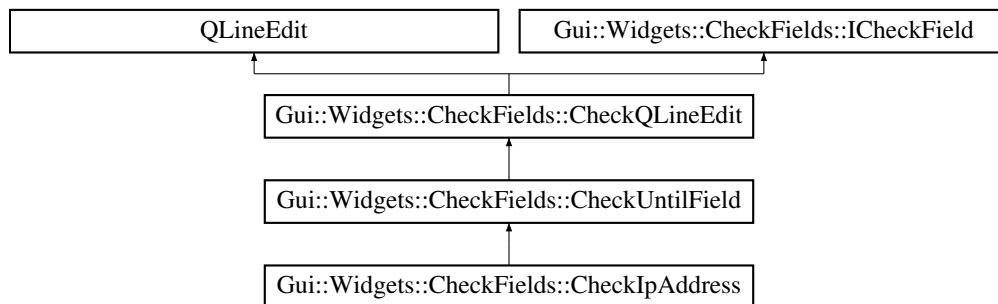
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkfieldsnumbers.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkfieldsnumbers.cpp

## 8.13 Gui::Widgets::CheckFields::CheckIpAddress Class Reference

The [CheckIpAddress](#) class Line Edit of IP address with a check icon.

```
#include <checkipaddress.h>
```

Inheritance diagram for Gui::Widgets::CheckFields::CheckIpAddress:



## Public Member Functions

- [CheckIpAddress](#) (QWidget \*w=0, QPushButton \*btn=0)  
[CheckIpAddress::CheckIpAddress](#) Construct a [CheckIpAddress](#).
- bool [check](#) (QString text)  
[CheckFieldsNumbers::check](#) Check if the field contains only numbers.

## Additional Inherited Members

## 8.13.1 Detailed Description

The [CheckIpAddress](#) class Line Edit of IP address with a check icon.

## Author

Florent BERBIE

## See Also

[CheckQLineEdit](#)  
[CheckUntilField](#)

## 8.13.2 Constructor &amp; Destructor Documentation

8.13.2.1 `Gui::Widgets::CheckFields::CheckIpAddress::CheckIpAddress ( QWidget * w = 0, QPushButton * btn = 0 )`[CheckIpAddress::CheckIpAddress](#) Construct a [CheckIpAddress](#).

## Parameters

<i>w</i>	QWidget linked to <a href="#">CheckIpAddress</a>
----------	--

## 8.13.3 Member Function Documentation

8.13.3.1 `bool Gui::Widgets::CheckFields::CheckIpAddress::check ( QString text ) [virtual]`[CheckFieldsNumbers::check](#) Check if the field contains only numbers.

## Parameters

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

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

## 8.14 Gui::Widgets::CheckFields::CheckLogin Class Reference

The [CheckLogin](#) class Line Edit of login with a check icon.

#include &lt;checklogin.h&gt;

Inheritance diagram for Gui::Widgets::CheckFields::CheckLogin:



## Public Slots

- void [passwordPreviousInputed](#) (const QString &text)  
[CheckLogin::fieldTextChanged](#) For each new characater inputed or removed, displays an icon to show if the field is valid or not.

## Public Member Functions

- [CheckLogin](#) (QWidget \*w=0, QPushButton \*btn=0)  
[CheckLogin::CheckLogin](#) Construct a [CheckLogin](#).
- bool [check](#) (QString text)  
[CheckLogin::check](#) Check if the field contains only numbers.

### 8.14.1 Detailed Description

The [CheckLogin](#) class Line Edit of login with a check icon.

#### Author

Florent BERBIE

#### See Also

[CheckQLineEdit](#)  
[CheckUntilField](#)

### 8.14.2 Constructor & Destructor Documentation

8.14.2.1 [Gui::Widgets::CheckFields::CheckLogin::CheckLogin](#) ( [QWidget](#) \* *w* = 0, [QPushButton](#) \* *btn* = 0 )

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

#### Parameters

<i>w</i>	QWidget linked to <a href="#">CheckIpAddress</a>
----------	--

### 8.14.3 Member Function Documentation

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

[CheckLogin::check](#) Check if the field contains only numbers.

#### Parameters

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

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

## 8.15 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

#### 8.15.1 Detailed Description

The [CheckName](#) class Line edit of name with a check icon.

#### 8.15.2 Constructor & Destructor Documentation

##### 8.15.2.1 Gui::Widgets::CheckFields::CheckName::CheckName ( QWidget \* w = 0, QPushButton \* btn = 0 )

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

Parameters

<i>w</i>	QWidget linked to <a href="#">CheckName</a>
----------	---

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

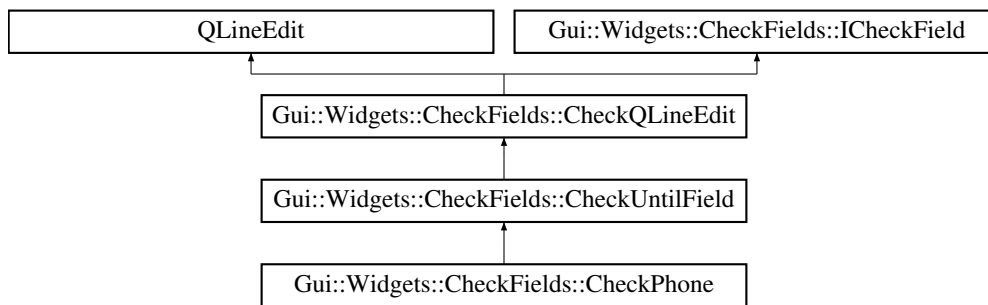
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkname.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkname.cpp

## 8.16 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

### 8.16.1 Detailed Description

The [CheckPhone](#) class Line Edit of Phone number with a check icon.

### 8.16.2 Constructor & Destructor Documentation

#### 8.16.2.1 Gui::Widgets::CheckFields::CheckPhone::CheckPhone ( QWidget \* w = 0, QPushButton \* btn = 0 )

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

Parameters

<i>w</i>	QWidget linked to <a href="#">CheckPhone</a>
----------	--

### 8.16.3 Member Function Documentation

#### 8.16.3.1 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

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

Returns

boolean

Implements [Gui::Widgets::CheckFields::ICheckField](#).

### 8.16.3.2 QString Gui::Widgets::CheckFields::CheckPhone::getCountry ( ) const

[CheckPhone::getCountry](#) Return the country linked to current field.

Returns

### 8.16.3.3 void Gui::Widgets::CheckFields::CheckPhone::setCountry ( const QString & country )

[CheckPhone::setCountry](#) Modify the *country* linked to field.

Parameters

<i>country</i>	New country
----------------	-------------

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

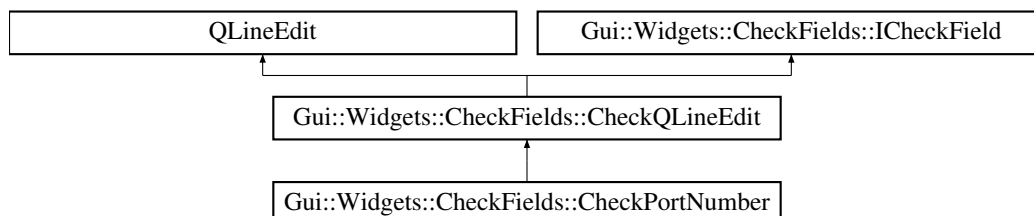
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkphone.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkphone.cpp

## 8.17 Gui::Widgets::CheckFields::CheckPortNumber Class Reference

The [CheckFieldsNumbers](#) class Line Edit of number with a check icon.

```
#include <checkportnumber.h>
```

Inheritance diagram for Gui::Widgets::CheckFields::CheckPortNumber:



### Public Member Functions

- [CheckPortNumber](#) (QWidget \*w=0, QPushButton \*btn=0)  
[CheckPortNumber](#).
- bool [check](#) (QString text)  
[CheckPortNumber::check](#) Check if the field contains only numbers or an empty text.

### Additional Inherited Members

#### 8.17.1 Detailed Description

The [CheckFieldsNumbers](#) class Line Edit of number with a check icon.

Author

Florent BERBIE

See Also

[CheckQLineEdit](#)  
[CheckUntilField](#)

## 8.17.2 Constructor & Destructor Documentation

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

[CheckPortNumber.](#)

Parameters

<i>w</i>	
<i>btn</i>	

## 8.17.3 Member Function Documentation

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

[CheckPortNumber::check](#) Check if the field contains only numbers or an empty text.

Parameters

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

Returns

boolean

Implements [Gui::Widgets::CheckFields::ICheckField](#).

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

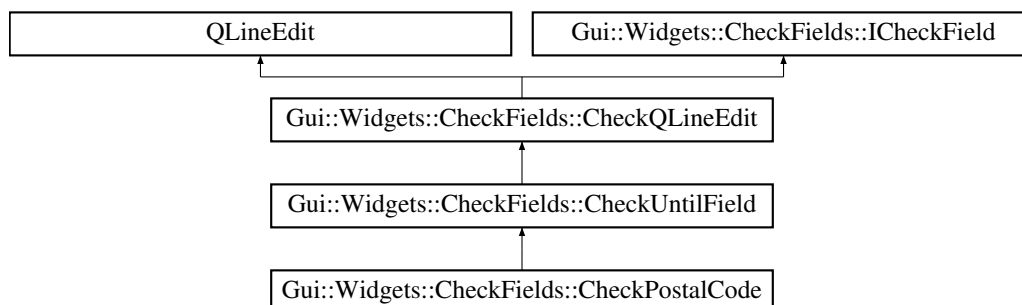
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkportnumber.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkportnumber.cpp

## 8.18 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

### 8.18.1 Detailed Description

The [CheckPostalCode](#) class Line Edit of postal code with a check icon.

### 8.18.2 Constructor & Destructor Documentation

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

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

Parameters

<i>w</i>	QWidget linked to <a href="#">CheckPostalCode</a>
----------	---

### 8.18.3 Member Function Documentation

8.18.3.1 `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

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

Returns

boolean

Implements [Gui::Widgets::CheckFields::ICheckField](#).

8.18.3.2 `QString Gui::Widgets::CheckFields::CheckPostalCode::getCountry ( ) const`

[CheckPostalCode::getCountry](#) Return the country linked to current field.

Returns

country

8.18.3.3 `void Gui::Widgets::CheckFields::CheckPostalCode::setCountry ( const QString & country )`

[CheckPostalCode::setCountry](#) Modify the *country* linked to field.



## Parameters

<i>country</i>	New country
----------------	-------------

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

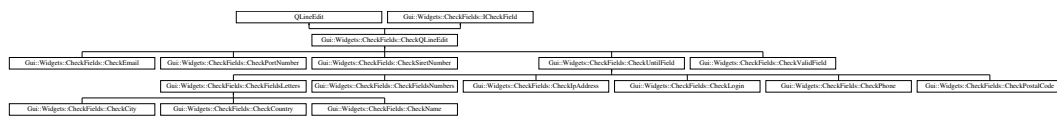
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkpostalcode.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkpostalcode.cpp

## 8.19 Gui::Widgets::CheckFields::CheckQLineEdit Class Reference

The [CheckQLineEdit](#) class QLineEdit 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

#### 8.19.1 Detailed Description

The [CheckQLineEdit](#) class QLineEdit custom with a check of text inputed.

#### 8.19.2 Constructor & Destructor Documentation

- 8.19.2.1 [Gui::Widgets::CheckFields::CheckQLineEdit::CheckQLineEdit](#) ( QWidget \* *parent* = 0, QPushButton \* *btn* = 0 )  
[explicit]

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

## Parameters

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

### 8.19.3 Member Function Documentation

8.19.3.1 `QPushButton * Gui::Widgets::CheckFields::CheckQLineEdit::getBtnValid ( ) const`

[CheckQLineEdit::getBtnValid.](#)

## Returns

a

8.19.3.2 `bool Gui::Widgets::CheckFields::CheckQLineEdit::isValid ( )`

isValid Return true if the current field if valid

## Returns

boolean

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

[CheckQLineEdit::setBtnValid.](#)

## Parameters

<i>getBtnValid</i>	
--------------------	--

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

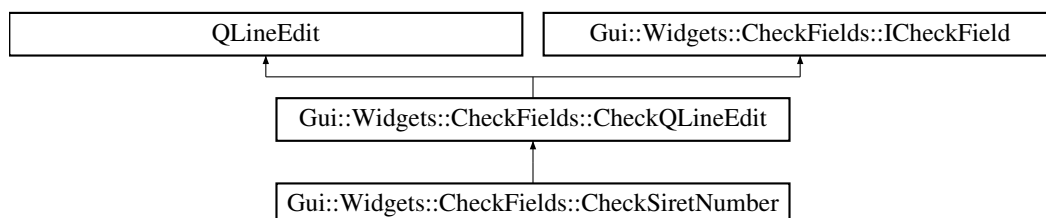
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkqlineedit.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/checkfields/checkqlineedit.cpp

## 8.20 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

### 8.20.1 Detailed Description

The [CheckSiretNumber](#) class Line Edit with a check icon.

### 8.20.2 Constructor & Destructor Documentation

8.20.2.1 [Gui::Widgets::CheckFields::CheckSiretNumber::CheckSiretNumber](#) ( [QWidget](#) \* *w* = 0, [QPushButton](#) \* *btn* = 0 )

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

Parameters

<i>w</i>	<a href="#">QWidget</a> linked to <a href="#">CheckSiretNumber</a>
----------	--

### 8.20.3 Member Function Documentation

8.20.3.1 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

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

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

## 8.21 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

### 8.21.1 Detailed Description

The [CheckUntilField](#) class.

### 8.21.2 Constructor & Destructor Documentation

8.21.2.1 [Gui::Widgets::CheckFields::CheckUntilField::CheckUntilField \( QWidget \\* w = 0, QPushButton \\* btn = 0 \)](#)

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

Parameters

<i>w</i>	QWidget linked to <a href="#">CheckUntilField</a>
----------	---

### 8.21.3 Member Function Documentation

8.21.3.1 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

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

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

## 8.22 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

### 8.22.1 Detailed Description

The [CheckValidField](#) class Check field not required.

### 8.22.2 Constructor & Destructor Documentation

8.22.2.1 [Gui::Widgets::CheckFields::CheckValidField::CheckValidField](#) ( QWidget \* *w* = 0, QPushButton \* *btn* = 0 )

[CheckValidField::CheckValidField](#).

Parameters

<i>w</i>	QWidget linked to <a href="#">CheckValidField</a>
----------	---

### 8.22.3 Member Function Documentation

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

[CheckValidField::check](#) Return TRUE : the field is not required.

Parameters

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

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

## 8.23 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:

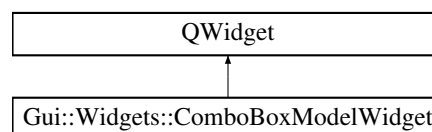
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/delegates/comboboxdelegate.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/delegates/comboboxdelegate.cpp

## 8.24 Gui::Widgets::ComboBoxModelWidget Class Reference

The [ComboBoxModelWidget](#) class.

```
#include <comboboxmodelwidget.h>
```

Inheritance diagram for Gui::Widgets::ComboBoxModelWidget:



### Public Member Functions

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

### 8.24.1 Detailed Description

The [ComboBoxModelWidget](#) class.

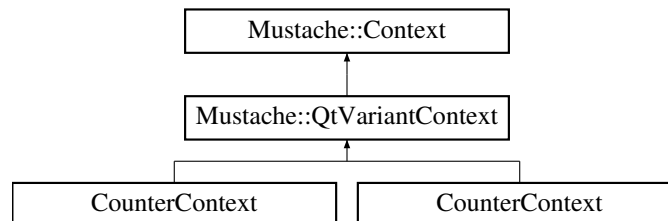
The documentation for this class was generated from the following files:

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

## 8.25 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)

### 8.25.1 Detailed Description

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

### 8.25.2 Constructor & Destructor Documentation

8.25.2.1 [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.

### 8.25.3 Member Function Documentation

8.25.3.1 [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](#).

### 8.25.3.2 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](#).

### 8.25.3.3 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](#).

### 8.25.3.4 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](#).

### 8.25.3.5 PartialResolver \* Context::partialResolver ( ) const

Returns the partial resolver passed to the constructor.

### 8.25.3.6 QString Context::partialValue ( const QString & key ) const

Returns the partial template for a given `key`.

### 8.25.3.7 virtual void Mustache::Context::pop ( ) [pure virtual]

Exit the current context.

Implemented in [Mustache::QtVariantContext](#).

### 8.25.3.8 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  $\geq 0$ , set the current context to the `index`'th value in the list value for `key`.

Implemented in [Mustache::QtVariantContext](#).

### 8.25.3.9 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](#).

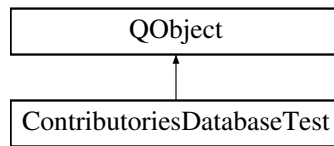
The documentation for this class was generated from the following files:

- /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



## 8.26 ContributoriesDatabaseTest Class Reference

Inheritance diagram for ContributoriesDatabaseTest:

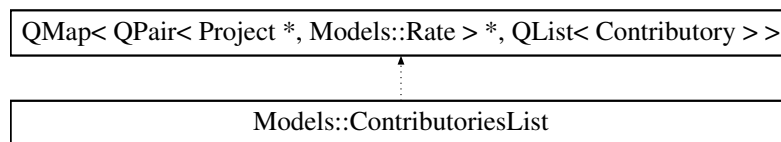


The documentation for this class was generated from the following files:

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

## 8.27 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)
- void **setAllIdContributories** (int idContributory)  
*[ContributoriesList::setAllIdContributories](#) Change all [Contributory](#) id with the same id.*
- 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** ()

### 8.27.1 Member Function Documentation

#### 8.27.1.1 void Models::ContributoriesList::setAllIdContributories ( int idContributory )

[ContributoriesList::setAllIdContributories](#) Change all [Contributory](#) id with the same id.

## Parameters

<i>idContributory</i>	the new <a href="#">Contributory</a> id
-----------------------	---

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

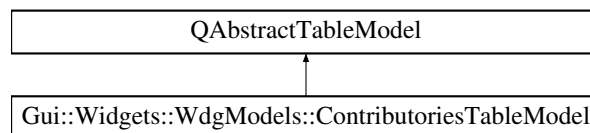
- /home/travis/build/FACT-Team/FactDev/src/models/contributorieslist.h
- /home/travis/build/FACT-Team/FactDev/src/models/contributorieslist.cpp

## 8.28 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](#) ()

### 8.28.1 Detailed Description

The [ContributoriesTableModel](#) class for a custom table for contributories widget.

#### Author

Antoine de Roquemaurel

#### See Also

[Contributory](#)

### 8.28.2 Constructor & Destructor Documentation

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

[ContributoriesTableModel](#) Construct a [ContributoriesTableModel](#).

#### Parameters

<i>parent</i>	Parent widget
---------------	---------------

### 8.28.3 Member Function Documentation

8.28.3.1 `void Gui::Widgets::WdgModels::ContributoriesTableModel::append ( const Contributory & contributory )`

append Add a new line in table

#### Parameters

<i>contributory</i>	The new contributory
---------------------	----------------------

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

columnCount Number of column of a contributory

#### Returns

The number of column

8.28.3.3 `int Gui::Widgets::WdgModels::ContributoriesTableModel::count ( )`

count Number of contributories in table

#### Returns

The number of contributories

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

data Obtains data of a specify cell

## Parameters

<i>index</i>	The cell who we want data
<i>role</i>	The role of set

## Returns

The data of cell

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

flags Differeents table flags

## Parameters

<i>index</i>	The cell who we want to know flags
--------------	------------------------------------

## Returns

Flags

**8.28.3.6** `QList< Contributory > Gui::Widgets::WdgModels::ContributoriesTableModel::getContributories ( )`

getContributories Get all contributories of table

## Returns

The contributory list

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

headerData Obtains header title of table

## Parameters

<i>section</i>	The number of column
<i>orientation</i>	The table orientation
<i>role</i>	

## Returns

The Title header of column

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

remove Remove a line

## Parameters

<i>i</i>	The number of line to remove
----------	------------------------------

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

rowCount Number of contributories row

#### Returns

The number of contributories

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

setData Change data of a cell

#### Parameters

<i>index</i>	The cell to change data
<i>value</i>	The new value
<i>role</i>	The role of cell

#### Returns

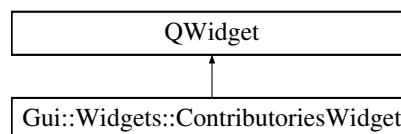
True if we could edit

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

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

## 8.29 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*

### 8.29.1 Member Function Documentation

#### 8.29.1.1 void Gui::Widgets::ContributoriesWidget::add ( [ContributoriesList](#) & *list* )

add Add the Contributory c in the model

Parameters

<i>c</i>	the <b>Contributory</b>
----------	-------------------------

#### 8.29.1.2 [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

## 8.30 Models::Contributory Class Reference

The [Contributory](#) class.

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

Inheritance diagram for Models::Contributory:



## Public Member Functions

- [Contributory](#) ()  
*[Contributory::Contributory](#) Construct a [Contributory](#).*
- [Contributory](#) (int id)  
*[Contributory::Contributory](#) Construct 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*
- QString **getLongDescription** () const
- void **setLongDescription** (const QString &getLongDescription)

## Additional Inherited Members

### 8.30.1 Detailed Description

The *Contributory* class.

Author

### 8.30.2 Constructor & Destructor Documentation

#### 8.30.2.1 Models::Contributory::Contributory ( int id )

*Contributory::Contributory* Construct a *Contributory* and get data in database.

Parameters

<i>id</i>	<i>Contributory</i> 's id
-----------	---------------------------

### 8.30.3 Member Function Documentation

#### 8.30.3.1 QVariantHash Models::Contributory::getDataMap ( ) [virtual]

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

Returns

Model's data

Implements [Models::IModel](#).

#### 8.30.3.2 QString Models::Contributory::getDescription ( ) const

getDescription Description of a contributory

Returns

The description

#### 8.30.3.3 double Models::Contributory::getNbHours ( ) const

getNbHours Number of work hour of a contributory

Returns

Then number of hours

#### 8.30.3.4 Project \* Models::Contributory::getProject ( ) const

[Contributory::getProject](#) Return the project linked to this [Contributory](#).

Returns

[Project](#) linked to this [Contributory](#)

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

[Contributory::hydrat](#) Get data about the [Contributory](#) which is specified by the identify *id*

Parameters

<i>id</i>	<a href="#">Contributory</a> identify
-----------	---------------------------------------

Implements [Models::IModel](#).

#### 8.30.3.6 bool Models::Contributory::operator!= ( const Contributory & c )

operator != define the operator "!=" to compare two [Contributory](#)



## Parameters

<i>c</i>	the <a href="#">Contributory</a> to compare with the current <a href="#">Contributory</a>
----------	---

## Returns

true if the [Contributory](#) are different else false

## 8.30.3.7 bool Models::Contributory::operator==( const Contributory &amp; c )

operator == define the operator "==" to compare two [Contributory](#)

## Parameters

<i>c</i>	the <a href="#">Contributory</a> to compare with the current <a href="#">Contributory</a>
----------	---

## Returns

true if the [Contributory](#) are equals else false

## 8.30.3.8 void Models::Contributory::setDescription ( const QString &amp; getDescription )

setDescription Change the contributory description

## Parameters

<i>getDescription</i>	The new description
-----------------------	---------------------

## 8.30.3.9 void Models::Contributory::setNbHours ( double value )

setNbHours Change nbHours

## Parameters

<i>value</i>	The new value of nbHours
--------------	--------------------------

## 8.30.3.10 void Models::Contributory::setProject ( Project \* id )

[Contributory::setProject](#) Modify the identify *id* of the [Project](#) linked to this [Contributory](#).

## Parameters

<i>id</i>	<a href="#">Project</a> 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

## 8.31 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 pld
- [Models::ContributoriesList](#) [getContributoriesByBilling](#) (const int billingId)  
*ContributoryDatabase::getContributoriesByBilling* get informations about the Contributory identified by Billing
- 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 pld)  
*ContributoryDatabase::removeCustomer* Remove the Contributory with the id pld
- [Models::Contributory](#) \* [getContributory](#) (QString &q)  
*getContributory* Obtain a contributory without new query
- [Models::ContributoriesList](#) [getContributoriesByBillingAndProject](#) (const int billingId, const int projectId)  
*getContributory* Get contributories list by project and billing

## Static Public Member Functions

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

## Additional Inherited Members

### 8.31.1 Detailed Description

The [ContributoryDatabase](#) class Contributory (or Quote) table database.

Author

See Also

[Database](#)  
Contributory/Quote

### 8.31.2 Member Function Documentation

#### 8.31.2.1 int Databases::ContributoryDatabase::addContributory ( const [Models::Contributory](#) & pContributory )

[ContributoryDatabase::addContributory](#) Add the Contributory pContributory to the database.

Returns

Contributory id

8.31.2.2 Models::ContributorList Databases::ContributoryDatabase::getContributorByBilling ( const int *billingId* )

[ContributoryDatabase::getContributorByBilling](#) get informations about the Contributor identified by *Billing*

## Parameters

<i>idBilling</i>	Contributory id
------------------	-----------------

## Returns

the Contributory

### 8.31.2.3 **Models::ContributoriesList** Databases::ContributoryDatabase::getContributoriesByBillingAndProject ( const int *billingId*, const int *projectId* )

getContributory Get contributories list by project and billing

## Parameters

<i>billingId</i>	
<i>projectId</i>	

## Returns

The contributories list by project and billing

### 8.31.2.4 **Models::Contributory** \* Databases::ContributoryDatabase::getContributory ( const int *idContributory* )

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

## Parameters

<i>idContributory</i>	Contributory id
-----------------------	-----------------

## Returns

the Contributory

### 8.31.2.5 **Models::Contributory** \* Databases::ContributoryDatabase::getContributory ( QSqlQuery & *q* )

getContributory Obtain a contributory without new query

## Parameters

<i>q</i>	The query to use
----------	------------------

## Returns

The contributory linked to q

### 8.31.2.6 **ContributoryDatabase** \* Databases::ContributoryDatabase::instance ( ) throw DbException \* ) [static]

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

## See Also

DbException

## Returns

Instance of [ContributoryDatabase](#)

8.31.2.7 void Databases::ContributoryDatabase::removeContributory ( const int *pld* )

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

## Parameters

<i>pld</i>	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

## 8.32 ContributoryListTest Class Reference

Inheritance diagram for ContributoryListTest:



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

## 8.33 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

## 8.34 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

### 8.34.1 Member Function Documentation

8.34.1.1 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](#).

8.34.1.2 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](#).

8.34.1.3 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](#).

8.34.1.4 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](#).

8.34.1.5 `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](#).

8.34.1.6 `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:

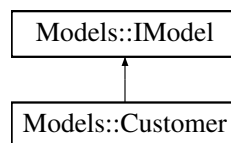
- `/home/travis/build/FACT-Team/FactDev/src/libs/qt-mustache/tests/test_mustache.cpp`

## 8.35 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* Construct a [Customer](#) who is specified by id
- void [commit](#) ()  
*Customer::commit* Update customer data on the database.
- void [hydrat](#) (int id)  
*Customer::hydrat* Insert into database informations related to the [Customer](#) who is specified by id
- void [remove](#) ()  
*Customer::remove* Remove the current customer.
- 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::setLastnameReferent* 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 different to the other [Customer](#) c Return TRUE if both customers are different, else FALSE.

## Additional Inherited Members

### 8.35.1 Detailed Description

The [Customer](#) class [Customer](#).

#### Author

Antoine de Roquemaurel  
Florent Berbie

### 8.35.2 Constructor & Destructor Documentation

#### 8.35.2.1 `Models::Customer::Customer ( int id )`

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

#### Parameters

<i>id</i>	<a href="#">Customer</a> identify
-----------	-----------------------------------

### 8.35.3 Member Function Documentation

#### 8.35.3.1 `QString Models::Customer::getAddress ( ) const`

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

#### Returns

company address

#### 8.35.3.2 `QString Models::Customer::getCity ( ) const`

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

#### Returns

city of the customer company

#### 8.35.3.3 `QString Models::Customer::getCompany ( ) const`

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

#### Returns

name of the company customer

#### 8.35.3.4 `QString Models::Customer::getCountry ( ) const`

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

#### Returns

country of the customer company

#### 8.35.3.5 QVariantHash Models::Customer::getDataMap ( ) [virtual]

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

##### Returns

Model's data

Implements [Models::IModel](#).

#### 8.35.3.6 QString Models::Customer::getEmail ( ) const

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

##### Returns

email of the customer company

#### 8.35.3.7 QString Models::Customer::getFax ( ) const

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

##### Returns

fax number

#### 8.35.3.8 QString Models::Customer::getFirstnameReferent ( ) const

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

##### Returns

Referent firstname of the customer

#### 8.35.3.9 QString Models::Customer::getLastnameReferent ( ) const

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

##### Returns

Referent lastname of the customer

#### 8.35.3.10 QString Models::Customer::getMobilePhone ( ) const

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

##### Returns

number of mobile phone

#### 8.35.3.11 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

#### 8.35.3.12 QString Models::Customer::getPath ( ) const

[Customer::getPath](#) Return the path of the workspace for the current [Customer](#).

##### Returns

workspace path

#### 8.35.3.13 QString Models::Customer::getPhone ( ) const

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

##### Returns

number of desktop phone

#### 8.35.3.14 QString Models::Customer::getPostalCode ( ) const

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

##### Returns

postal code of the customer company

#### 8.35.3.15 void Models::Customer::hydrat ( int *id* ) [virtual]

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

##### Parameters

<i>id</i>	<a href="#">Customer</a> identify
-----------	-----------------------------------

Implements [Models::IModel](#).

#### 8.35.3.16 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

<i>c</i>	<a href="#">Customer</a> to compare
----------	-------------------------------------

##### Returns

boolean

8.35.3.17 `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

<i>c</i>	<a href="#">Customer</a> to compare
----------	-------------------------------------

## Returns

boolean

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

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

## Parameters

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

8.35.3.19 void Models::Customer::setCity ( const QString & *city* )

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

## Parameters

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

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

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

## Parameters

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

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

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

## Parameters

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

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

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

## Parameters

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

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

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

## Parameters

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

8.35.3.24 void Models::Customer::setFirstnameReferent ( const QString & *firstnameReferent* )

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

## Parameters

<i>firstname-Referent</i>	New referent firstname of the <a href="#">Customer</a>
---------------------------	--

8.35.3.25 void Models::Customer::setLastnameReferent ( const QString & *lastnameReferent* )

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

## Parameters

<i>lastname-Referent</i>	New referent lastname of the <a href="#">Customer</a>
--------------------------	---

8.35.3.26 void Models::Customer::setMobilePhone ( const QString & *mobilePhone* )

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

## Parameters

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

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

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

## Parameters

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

8.35.3.28 void Models::Customer::setPostalCode ( const QString & *postalCode* )

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

## Parameters

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

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

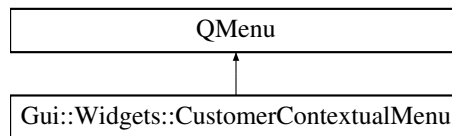
- /home/travis/build/FACT-Team/FactDev/src/models/customer.h
- /home/travis/build/FACT-Team/FactDev/src/models/customer.cpp

## 8.36 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.

### 8.36.1 Detailed Description

Display contextual menu on a customer.

Author

Antoine de Roquemaurel

### 8.36.2 Constructor & Destructor Documentation

#### 8.36.2.1 Gui::Widgets::CustomerContextualMenu::CustomerContextualMenu ( QWidget \* w = 0 )

[CustomerContextualMenu](#) Construct a new contextual menu.

Parameters

<i>w</i>	Parent
----------	--------

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

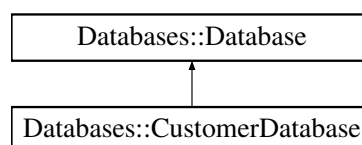
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/customercontextualmenu.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/customercontextualmenu.cpp

## 8.37 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 pld)  
*CustomerDatabase::getCustomer* get informations about the customer identified by pld
- 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 pld)  
*CustomerDatabase::removeCustomer* Remove the customer with the id pld
- int [getNbCustomers](#) ()  
*CustomerDatabase::getNbCustomers* Return the number of customers existing.
- QStandardItem \* [getItemRoot](#) ()  
*CustomerDatabase::getItemRoot* Return the first item for the QStandardItemModel.
- QStandardItem \* [getItemCustomer](#) (QString q1)  
*CustomerDatabase::getItemCustomer* Return the customer item for the QStandardItemModel.
- QStandardItem \* [getItemProject](#) (QString q2)  
*CustomerDatabase::getItemProject* Return the project item for the QStandardItemModel.
- QStandardItem \* [getItemBillQuote](#) (QString q3)  
*CustomerDatabase::getItemCustomer* Return the bill/quote item for the QStandardItemModel.
- QSharedPointer< [Models::Customer](#) > [getCustomer](#) (QString q)  
*CustomerDatabase::getCustomer* Add the element of the q request and return their.
- void [updateCustomer](#) (QString 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

### 8.37.1 Detailed Description

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

#### Author

Antoine de Roquemaurel  
 Manantsoa Razanajatovo  
 Florent Berbie

#### See Also

[Database](#)  
[Customer](#)

### 8.37.2 Member Function Documentation

8.37.2.1 `int Databases::CustomerDatabase::addCustomer ( const Models::Customer & pCustomer )`

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

Returns

customer id

8.37.2.2 `QSharedPointer< Models::Customer > Databases::CustomerDatabase::getCustomer ( const int pld )`

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

Parameters

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

Returns

the Customer

8.37.2.3 `QSharedPointer< Models::Customer > Databases::CustomerDatabase::getCustomer ( QSqlQuery & q )`

[CustomerDatabase::getCustomer](#) Add the element of the *q* request and return their.

Parameters

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

Returns

a customer formed according to QSharedPointer

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

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

Parameters

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

Exceptions

<i>DbException</i>	
--------------------	--

Returns

QStandardItemModel an item model for QTableView

8.37.2.5 `QStandardItem * Databases::CustomerDatabase::getItemBillQuote ( QSqlQuery q3 )`

[CustomerDatabase::getItemCustomer](#) Return the bill/quote item for the QStandardItemModel.

## Parameters

<i>q3</i>	the first row of the sql query for bills/quotes
-----------	---

## Returns

QStandardItem an item for QTree (level/depth 3)

**8.37.2.6** QStandardItem \* Databases::CustomerDatabase::getItemCustomer ( QSqlQuery *q1* )

[CustomerDatabase::getItemCustomer](#) Return the customer item for the QStandardItemModel.

## Parameters

<i>q1</i>	the row of the sql query for customers
-----------	--

## Returns

QStandardItem an item for QTree (level/depth 1)

**8.37.2.7** QStandardItem \* Databases::CustomerDatabase::getItemProject ( QSqlQuery *q2* )

[CustomerDatabase::getItemProject](#) Return the project item for the QStandardItemModel.

## Parameters

<i>q2</i>	the row of the sql query for projects
-----------	---------------------------------------

## Returns

QStandardItem an item for QTree (level/depth 2)

**8.37.2.8** QStandardItem \* Databases::CustomerDatabase::getItemRoot ( )

[CustomerDatabase::getItemRoot](#) Return the first item for the QStandardItemModel.

## Returns

QStandardItem an item for QTree (level/depth 0)

**8.37.2.9** int Databases::CustomerDatabase::getNbCustomers ( )

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

## Returns

number of customers

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

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

## Parameters

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

## Exceptions

<i>DbException</i>	
--------------------	--

## Returns

QStandardItemModel an item model for QTreeView

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

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

## See Also

[DbException](#)

## Returns

Instance of [CustomerDatabase](#)

**8.37.2.12** **void Databases::CustomerDatabase::removeCustomer ( const int *pld* )**

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

## Parameters

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

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

[CustomerDatabase::updateCustomer](#) Update customer data according to the request *q*

## Parameters

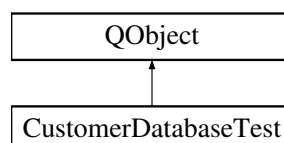
<i>q</i>	SQL request
----------	-------------

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

## 8.38 CustomerDatabaseTest Class Reference

Inheritance diagram for CustomerDatabaseTest:



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

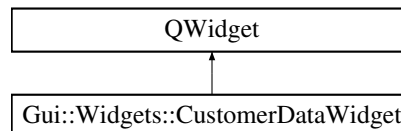
- /home/travis/build/FACT-Team/FactDev/tests/database/customerdatabasetest.h
- /home/travis/build/FACT-Team/FactDev/tests/database/customerdatabasetest.cpp

## 8.39 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

#### 8.39.1 Detailed Description

Class for display info of a customer.

Author

#### 8.39.2 Member Function Documentation

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

printInformations Print Data of customer id

Parameters

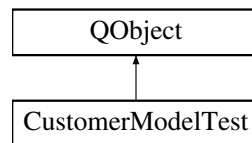
<i>id</i>	of customer to print
-----------	----------------------

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

## 8.40 CustomerModelTest Class Reference

Inheritance diagram for CustomerModelTest:



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

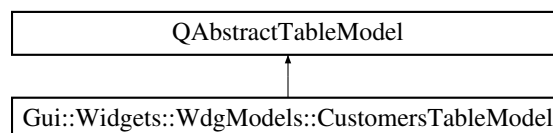
- /home/travis/build/FACT-Team/FactDev/tests/models/customermodeltest.h
- /home/travis/build/FACT-Team/FactDev/tests/models/customermodeltest.cpp

## 8.41 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.

### 8.41.1 Detailed Description

The [CustomersTableModel](#) class for a customer table.

#### Author

Florent Berbie

#### See Also

[Customer](#)

### 8.41.2 Constructor & Destructor Documentation

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

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

#### Parameters

<i>parent</i>	Parent widget
---------------	---------------

### 8.41.3 Member Function Documentation

8.41.3.1 `void Gui::Widgets::WdgModels::CustomersTableModel::append ( const Customer & customer )`

[CustomersTableModel::append](#) Add a new line in table.

#### Parameters

<i>Customer</i>	The new customer
-----------------	------------------

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

[CustomersTableModel::columnCount](#) Number of column of a customer.

#### Returns

The number of column

8.41.3.3 `int Gui::Widgets::WdgModels::CustomersTableModel::count ( )`

[CustomersTableModel::count](#) Number of customers in table.

#### Returns

The number of customers

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

[CustomersTableModel::data](#) Obtains data of a specify cell.

## Parameters

<i>index</i>	The cell who we want data
<i>role</i>	The role of set

## Returns

The data of cell

8.41.3.5 `Qt::ItemFlags Gui::Widgets::WdgModels::CustomersTableModel::flags ( const QModelIndex & index ) const`

[CustomersTableModel::flags](#) Differents table flags.

## Parameters

<i>index</i>	The cell who we want to know flags
--------------	------------------------------------

## Returns

Flags

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

[CustomersTableModel::getCustomers](#) Return the list of customers.

## Returns

list of Customers

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

[CustomersTableModel::headerData](#) Obtains header title of table.

## Parameters

<i>section</i>	The number of column
<i>orientation</i>	The table orientation
<i>role</i>	

## Returns

The Title header of column

8.41.3.8 `void Gui::Widgets::WdgModels::CustomersTableModel::remove ( const int i )`

[CustomersTableModel::remove](#) Remove a line.

## Parameters

<i>i</i>	The number of line to remove
----------	------------------------------



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

[CustomersTableModel::rowCount](#) Number of customers row.

#### Returns

The number of customers

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

[CustomersTableModel::setData](#) Change data of a cell.

#### Parameters

<i>index</i>	The cell to change data
<i>value</i>	The new value
<i>role</i>	The role of cell

#### Returns

True if we could edit

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

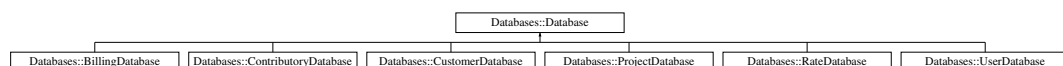
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/widgetsmodels/customerstablemodel.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/widgetsmodels/customerstablemodel.cpp

## 8.42 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 ocured 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.*

## 8.42.1 Detailed Description

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

Author

Antoine de Roquemaurel

## 8.42.2 Member Function Documentation

### 8.42.2.1 void Databases::Database::executeFile ( QString *pName* )

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

## Parameters

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

## 8.42.2.2 Database \* Databases::Database::instance ( ) throw DbException \*) [static]

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

## Returns

Instance of [Database](#)

## 8.42.2.3 QString Databases::Database::lastError ( const QSqlQuery &amp; q ) const [inline]

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

## Parameters

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

## Returns

an error message

## 8.42.2.4 void Databases::Database::setDatabase ( QSqlDatabase sql )

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

## Parameters

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

## 8.42.2.5 QVariant Databases::Database::value ( const QSqlQuery &amp; q, const QString &amp; champ ) const [protected]

Database::valeur Value of database field.

## Parameters

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

## Returns

The value

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

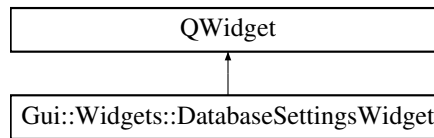
- /home/travis/build/FACT-Team/FactDev/src/database/database.h
- /home/travis/build/FACT-Team/FactDev/src/database/database.cpp

## 8.43 Gui::Widgets::DatabaseSettingsWidget Class Reference

The [DatabaseSettingsWidget](#) class Windows of database settings.

```
#include <databasesettingswidget.h>
```

Inheritance diagram for Gui::Widgets::DatabaseSettingsWidget:



## Public Slots

- bool [isValid](#) ()  
*DatabaseSettingsWidget::isValid* Return TRUE if all fields are correctly inputed else FALSE.
- void [checkRepeatPassword](#) (QString text)  
*DatabaseSettingsWidget::checkRepeatLogin* Check if the second login field is the same than the first.
- void [userInterfaceChanged](#) ()  
*DatabaseSettingsWidget::userInterfaceChanged* User interface has changed.

## Signals

- void [textfieldChanged](#) ()  
*DatabaseSettingsWidget::textfieldChanged* Signal which indicates if a fieldtext has changed.

## Public Member Functions

- [DatabaseSettingsWidget](#) (QWidget \*parent=0)  
*DatabaseSettingsWidget::DatabaseSettingsWidget* Construct a [DatabaseSettingsWidget](#).
- void [fillFields](#) ()  
*DatabaseSettingsWidget::fillFields* Complete fields with a default value for field Database name, Username, IP address and port.
- QString [getDatabaseName](#) ()  
*DatabaseSettingsWidget::getDatabaseName* Return the database name.
- QString [getLogin](#) ()  
*DatabaseSettingsWidget::getLogin* Return the login of the user.
- QString [getPassword](#) ()  
*DatabaseSettingsWidget::getPassword* Return the password of the user.
- QString [getDomainNameOrIP](#) ()  
*DatabaseSettingsWidget::getDomainNameOrIP* Return the domain name or IP address on the machine where is the database.
- QString [getPort](#) ()  
*DatabaseSettingsWidget::getPort* Return the port of the database.

### 8.43.1 Detailed Description

The [DatabaseSettingsWidget](#) class Windows of database settings.

Author

## 8.43.2 Constructor & Destructor Documentation

8.43.2.1 `Gui::Widgets::DatabaseSettingsWidget::DatabaseSettingsWidget ( QWidget * parent = 0 )` `[explicit]`

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

## Parameters

<i>parent</i>	Parent widget of this windows
---------------	-------------------------------

### 8.43.3 Member Function Documentation

#### 8.43.3.1 QString Gui::Widgets::DatabaseSettingsWidget::getDatabaseName ( )

[DatabaseSettingsWidget::getDatabaseName](#) Return the database name.

## Returns

Database name

#### 8.43.3.2 QString Gui::Widgets::DatabaseSettingsWidget::getDomainNameOrIP ( )

[DatabaseSettingsWidget::getDomainNameOrIP](#) Return the domain name or IP address on the machine where is the database.

## Returns

Domain name or IP address

#### 8.43.3.3 QString Gui::Widgets::DatabaseSettingsWidget::getLogin ( )

[DatabaseSettingsWidget::getLogin](#) Return the login of the user.

## Returns

Login of the user

#### 8.43.3.4 QString Gui::Widgets::DatabaseSettingsWidget::getPassword ( )

[DatabaseSettingsWidget::getPassword](#) Return the password of the user.

## Returns

User password

#### 8.43.3.5 QString Gui::Widgets::DatabaseSettingsWidget::getPort ( )

[DatabaseSettingsWidget::getPort](#) Return the port of the database.

## Returns

Database port

#### 8.43.3.6 bool Gui::Widgets::DatabaseSettingsWidget::isValid ( ) [slot]

[DatabaseSettingsWidget::isValid](#) Return TRUE if all fields are correctly inputed else FALSE.

## Returns

boolean

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

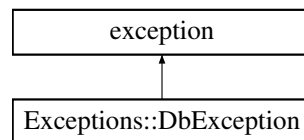
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/databasesettingswidget.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/databasesettingswidget.cpp

## 8.44 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.

### 8.44.1 Detailed Description

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

#### Author

Antoine de Roquemaurel

### 8.44.2 Constructor & Destructor Documentation

8.44.2.1 Exceptions::DbException::DbException ( const QString *fct*, const QString *fctName*, const QString *logError*, float *errorCode* )

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

#### Parameters

<i>userError</i>	ClassName of error
------------------	--------------------



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

### 8.44.3 Member Function Documentation

#### 8.44.3.1 void Exceptions::DbException::popupMessage ( QWidget \* *parent* )

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

Parameters

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

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

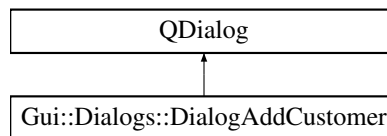
- /home/travis/build/FACT-Team/FactDev/src/exceptions/dbexception.h
- /home/travis/build/FACT-Team/FactDev/src/exceptions/dbexception.cpp

## 8.45 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*

### 8.45.1 Detailed Description

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

Author

## 8.45.2 Constructor & Destructor Documentation

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

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

Parameters

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

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

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

## 8.46 Utils::Directories Class Reference

### Static Public Member Functions

- static QString [makeDirectory](#) (QDir &directory, const QString path, const QString folder) throw (Exceptions::FileNotFoundException\*)

*MainWindow::makeDirectory If not exists make a new directory folder*

### 8.46.1 Member Function Documentation

8.46.1.1 `QString Utils::Directories::makeDirectory ( QDir & directory, const QString path, const QString folder ) throw Exceptions::FileNotFoundException *) [static]`

*MainWindow::makeDirectory If not exists make a new directory folder*

Parameters

<i>path</i>	Return the path of the folder just created
<i>folder</i>	Folder name to create

Returns

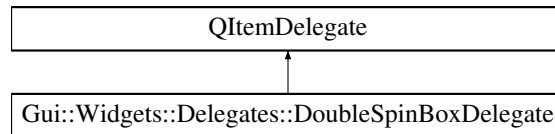
Path of the folder just created

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

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

## 8.47 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

## 8.48 Exceptions::FileException Class Reference

The [FileException](#) class for file/access 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.*

### 8.48.1 Detailed Description

The [FileException](#) class for file/access file exception.

Author

Florent Berbie

### 8.48.2 Constructor & Destructor Documentation

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

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

## Parameters

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

### 8.48.3 Member Function Documentation

#### 8.48.3.1 void Exceptions::FileException::popupMessage ( QWidget \* *parent* )

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

## Parameters

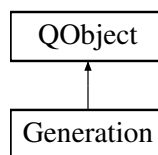
<i>parent</i>	
---------------	--

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

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

## 8.49 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

## 8.50 Utils::HierarchicalSystem Class Reference

The [HierarchicalSystem](#) class Create class which contains hierarchical system of FactDev.

```
#include <hierarchicalsistem.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.

### 8.50.1 Detailed Description

The [HierarchicalSystem](#) class Create class which contains hierarchical system of FactDev.

#### Author

Florent Berbie

#### See Also

[Customer](#)  
[Project](#)  
[Billing](#)

### 8.50.2 Member Function Documentation

#### 8.50.2.1 void Utils::HierarchicalSystem::addBillingToProject ( [Billing](#) \* b, [Project](#) \* p )

[\*HierarchicalSystem::addBillingToProject\*](#) Add the Billing b to the Project p

##### Parameters

<i>b</i>	<a href="#">Billing</a>
<i>p</i>	<a href="#">Project</a>

#### 8.50.2.2 void Utils::HierarchicalSystem::addProjectToCustomer ( [Project](#) \* p, [Customer](#) c )

[\*HierarchicalSystem::addProjectToCustomer\*](#) Add the Project p to the Customer c

##### Parameters

<i>p</i>	<a href="#">Project</a>
<i>c</i>	<a href="#">Customer</a>

#### 8.50.2.3 [QMap](#)< [Project](#) \*, [Customer](#) > [Utils::HierarchicalSystem::getCustomers](#) ( ) const

[\*HierarchicalSystem::getCustomers\*](#) Return all customers and these projects linked.

#### Returns

Projects linked to Customers

8.50.2.4 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:

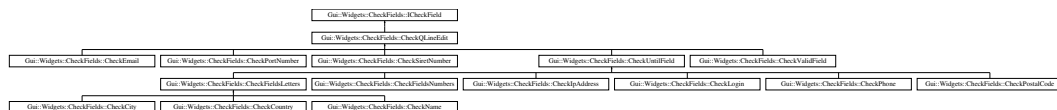
- /home/travis/build/FACT-Team/FactDev/src/utis/hierarchicalsistem.h
- /home/travis/build/FACT-Team/FactDev/src/utis/hierarchicalsistem.cpp

## 8.51 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

#### 8.51.1 Detailed Description

The [ICheckField](#) class Interface to check fields validity.

#### 8.51.2 Member Function Documentation

8.51.2.1 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::CheckPortNumber](#), [Gui::Widgets::CheckFields::CheckFieldsNumbers](#), [Gui::Widgets::CheckFields::CheckIpAddress](#), [Gui::Widgets::CheckFields::CheckLogin](#), [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

## 8.52 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.*

#### 8.52.1 Detailed Description

The `IModel` class.

Author

Antoine de Roquemaurel

## 8.52.2 Member Function Documentation

### 8.52.2.1 `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::User](#), [Models::Contributory](#), and [Models::Customer](#).

### 8.52.2.2 `int Models::IModel::getId ( ) const [inline]`

[IModel::getId](#) Return the identify of the element of the database.

Returns

identity

### 8.52.2.3 `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

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

Implemented in [Models::Billing](#), [Models::Project](#), [Models::User](#), [Models::Contributory](#), and [Models::Customer](#).

### 8.52.2.4 `bool Models::IModel::isToRemoved ( ) const [inline]`

toRemoved return if object must be removed.

Returns

### 8.52.2.5 `void Models::IModel::setId ( int id ) [inline]`

[IModel::setId](#) Replace the current identify by *id*

Parameters

<i>id</i>	New identify
-----------	--------------

### 8.52.2.6 `void Models::IModel::setToRemoved ( bool toRemoved ) [inline]`

setToRemoved Change the flag for removed object

Parameters

<i>toRemoved</i>	The new flag
------------------	--------------

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

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



## 8.53 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*

#### 8.53.1 Detailed Description

The `ItemType` class Item type model.

#### 8.53.2 Constructor & Destructor Documentation

##### 8.53.2.1 Utils::ItemType::ItemType ( int type, QString name )

`ItemType::ItemType` Construct an Item type.

Parameters

<i>type</i>	Type of the item
<i>name</i>	Name of the item

#### 8.53.3 Member Function Documentation

##### 8.53.3.1 Models::IModel \* Utils::ItemType::getModel ( int id )

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

## Parameters

<i>id</i>	Item type identity
-----------	--------------------

## Returns

database model

## 8.53.3.2 QString Utils::ItemType::getName ( ) const

[ItemType::getName](#) Get item name.

## Returns

item name

## 8.53.3.3 int Utils::ItemType::getType ( ) const

[ItemType::getType](#) Get the type of the current item.

## Returns

type of the current item

## 8.53.3.4 void Utils::ItemType::setName ( const QString &amp; name )

[ItemType::setName](#) Modify the item name.

## Parameters

<i>name</i>	New Item name
-------------	---------------

## 8.53.3.5 void Utils::ItemType::setType ( int type )

[ItemType::setType](#) Modify the type of the current item.

## Parameters

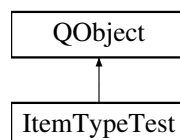
<i>type</i>	New item type
-------------	---------------

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

## 8.54 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

## 8.55 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*

#### 8.55.1 Detailed Description

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

#### 8.55.2 Member Function Documentation

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

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

Parameters

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

Returns

Instance of logger.

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

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

## Parameters

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

### 8.55.3 Friends And Related Function Documentation

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

operator << for log writing

## Parameters

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

## Returns

New logger.

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

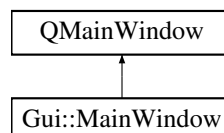
- /home/travis/build/FACT-Team/FactDev/src/utils/log.h
- /home/travis/build/FACT-Team/FactDev/src/utils/log.cpp

## 8.56 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 [billingsIsPaid](#) ()  
*MainWindow::billingsIsPaid* Define the current billing as "paid".
- void [editUser](#) ()  
*MainWindow::editUser* modify the user.

- void [search](#) (QString s)  
*MainWindow::search* launch a new search.
- void [addProject](#) ()  
*MainWindow::addProject* 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.
- void [copyDoc](#) ()  
*MainWindow::copyDoc* Copy all elements of a quote or a bill and Display these elements in a new quote or bill.
- void [openPdf](#) ()  
*MainWindow::editQuote* Edit the qupte or.

## 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.

### 8.56.1 Detailed Description

The [MainWindow](#) class Main Window of the software.

Author

Everybody

### 8.56.2 Constructor & Destructor Documentation

8.56.2.1 `Gui::MainWindow::MainWindow ( QWidget * parent = 0 ) [explicit]`

[MainWindow](#): Construct a window.

Parameters

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

### 8.56.3 Member Function Documentation

8.56.3.1 `void Gui::MainWindow::addBill ( ) [slot]`

[MainWindow::addBill](#) open window to add a new bill.

See Also

AddQuoteDialog

8.56.3.2 `void Gui::MainWindow::addDoc ( bool isBilling )`

[MainWindow::addDoc](#) open window to add a new document.

Parameters

<i>bool</i>	quote or bill
-------------	---------------

See Also

[addBill](#) [addQuote](#)

8.56.3.3 `void Gui::MainWindow::addProject ( ) [slot]`

[MainWindow::addProject](#) Create a new project for a customer.

See Also

AddProjectDialog

8.56.3.4 `void Gui::MainWindow::addQuote ( ) [slot]`

[MainWindow::addQuote](#) open window to add a new quote.

See Also

AddQuoteDialog

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

[MainWindow::editUser](#) modify the user.

See Also

UserDataDialog

8.56.3.6 int Gui::MainWindow::getCurrentCustomerId ( )

[MainWindow::getCurrentCustomerId](#) get the selected customer.

Returns

id of the selected customer

8.56.3.7 QString Gui::MainWindow::getCurrentCustomerName ( )

[MainWindow::getCurrentCustomerName](#) get the selected customer name in the customers' table.

Returns

name of the selected customer

8.56.3.8 int Gui::MainWindow::getCurrentProjectId ( )

[MainWindow::getCurrentProjectId](#) get the selected project id.

Returns

id of the selected project

8.56.3.9 QString Gui::MainWindow::getCurrentProjectName ( )

[MainWindow::getCurrentProjectName](#) get the selected project name in the table of projects.

Returns

name of the selected project

8.56.3.10 int Gui::MainWindow::getCurrentQuoteId ( )

[MainWindow::getCurrentQuoteId](#) get the selected quote id.

Returns

id of the selected quote

8.56.3.11 QModelIndex Gui::MainWindow::rootTree ( )

[MainWindow::rootTree](#) return the root of the tree "Tous les clients".

Returns

QModelIndex

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

[MainWindow::search](#) launch a new search.



## Parameters

<code>s</code>	text in field
----------------	---------------

## 8.56.3.13 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:

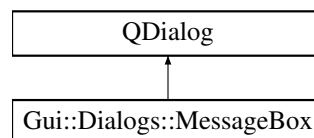
- /home/travis/build/FACT-Team/FactDev/src/gui/mainwindow/mainwindow.h
- /home/travis/build/FACT-Team/FactDev/src/gui/mainwindow/mainwindow.cpp

## 8.57 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.*

### 8.57.1 Detailed Description

The [MessageBox](#) class Information window with message.

Author

Florent Berbie

### 8.57.2 Constructor & Destructor Documentation

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

[MessageBox::MessageBox](#) Construt a [MessageBox](#)

Parameters

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

### 8.57.3 Member Function Documentation

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

[MessageBox::setImage](#) Add the icon *img* to the current window.

Parameters

<i>img</i>	Icon
<i>width</i>	Icon width (default: 128)
<i>height</i>	Icon height (default: 128)

8.57.3.2 `void Gui::Dialogs::MessageBox::setText ( QString txt )`

[MessageBox::setText](#) Add the text *txt* to the current window.

Parameters

<i>txt</i>	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

## 8.58 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](#) = 1.0  
*VERSION* Version number of software.

### 8.58.1 Detailed Description

The [Parameters](#) class Class for simple user parameters.

Author

Antoine de Roquemaurel

### 8.58.2 Member Data Documentation

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

DB\_FILENAME The database sqlite file name.

Database file name

8.58.2.2 `const double Parameters::VERSION = 1.0 [static]`

VERSION Version number of software.

Application version

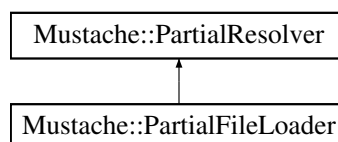
The documentation for this class was generated from the following files:

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

## 8.59 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)

### 8.59.1 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.

### 8.59.2 Member Function Documentation

8.59.2.1 `QString PartialFileLoader::getPartial ( const QString & name ) [virtual]`

Returns the partial template with a given `name`.

Implements [Mustache::PartialResolver](#).

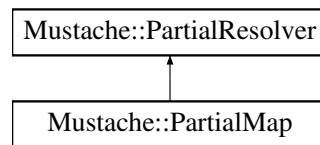
The documentation for this class was generated from the following files:

- /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

## 8.60 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)

### 8.60.1 Detailed Description

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

### 8.60.2 Member Function Documentation

#### 8.60.2.1 QString PartialMap::getPartial ( const QString & name ) [virtual]

Returns the partial template with a given `name`.

Implements [Mustache::PartialResolver](#).

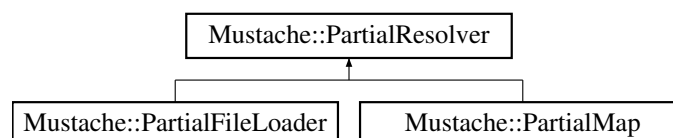
The documentation for this class was generated from the following files:

- /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

## 8.61 Mustache::PartialResolver Class Reference

```
#include <mustache.h>
```

Inheritance diagram for Mustache::PartialResolver:



## Public Member Functions

- virtual QString [getPartial](#) (const QString &name)=0

### 8.61.1 Detailed Description

Interface for fetching template partials.

### 8.61.2 Member Function Documentation

8.61.2.1 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

## 8.62 Generator::PdfGenerator Class Reference

### Public Member Functions

- **PdfGenerator** (QString pdflatexPath="pdflatex")
- void **generate** (QString inputDir, QString filename)

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

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

## 8.63 Utils::pointers Class Reference

### Static Public Member Functions

- static void **deleteIfNotNull** (QObject \*p)

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

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

## 8.64 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.*

### 8.64.1 Detailed Description

Class for display popup quickly.

#### Author

Antoine de Roquemaurel

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

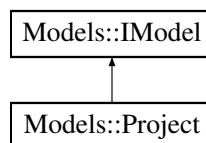
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/popup.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/popup.cpp

## 8.65 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](#) Destruct 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 [getCost](#) () const  
*[Project::getCost](#) Return the **Project** cost*
- void [setCost](#) (double cost)  
*[Project::setCost](#) Modify the **Project** cost*
- 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 first is anterior to the second.*
- bool [operator!=](#) (const [Project](#) &p)  
*[Project::operator !=](#) Re-define the operator "!=" to compare if the current project is different to the other **Project** p  
Return TRUE if both projects are different, else FALSE.*
- double [costCompute](#) ()  
*[Project::costCompute](#) compute the **Project** cost*

## Additional Inherited Members

### 8.65.1 Detailed Description

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

#### Author

Florent Berbie

#### See Also

[IModel](#)

## 8.65.2 Constructor & Destructor Documentation

### 8.65.2.1 Models::Project::Project ( int *id* )

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



## Parameters

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

### 8.65.3 Member Function Documentation

#### 8.65.3.1 double Models::Project::costCompute ( )

[Project::costCompute](#) compute the **Project** cost

**Returns**

the project cost

#### 8.65.3.2 QDate Models::Project::getBeginDate ( ) const

[Project::getBeginDate](#) return the date of creation of the **Project**

**Returns**

the begin date of the **Project**

#### 8.65.3.3 double Models::Project::getCost ( ) const

[Project::getCost](#) Return the **Project** cost

**Returns**

the project cost

#### 8.65.3.4 QSharedPointer< Customer > Models::Project::getCustomer ( ) const

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

**Returns**

customer linked to this project

#### 8.65.3.5 double Models::Project::getDailyRate ( ) const

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

**Returns**

the daily rate linket to the current project

#### 8.65.3.6 QVariantHash Models::Project::getDataMap ( ) [virtual]

[getDataMap](#) Get all data of model with a HashMap key/value

**Returns**

Model's data

Implements [Models::IModel](#).

#### 8.65.3.7 QString Models::Project::getDescription ( ) const

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

##### Returns

project description

#### 8.65.3.8 QDate Models::Project::getEndDate ( ) const

[Project::getEndDate](#) Return the *endDate* of the **Project**

##### Returns

the end date of the project

#### 8.65.3.9 QString Models::Project::getName ( ) const

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

##### Returns

project name

#### 8.65.3.10 void Models::Project::hydrat ( int *id* ) [virtual]

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

##### Parameters

<i>id</i>	<a href="#">Project</a> identify
-----------	----------------------------------

Implements [Models::IModel](#).

#### 8.65.3.11 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

<i>c</i>	<a href="#">Project</a> to compare
----------	------------------------------------

##### Returns

boolean

#### 8.65.3.12 bool Models::Project::operator< ( const Project & *p* ) const

[Project::operator <](#) defines the operator "<" to compare two **Project** and to see if the first is anterior to the second.

## Parameters

<i>b</i>	the <a href="#">Project</a> to compare with the current <a href="#">Project</a>
----------	---

## Returns

true if the [Project](#) are different else false

8.65.3.13 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

<i>c</i>	<a href="#">Project</a> to compare
----------	------------------------------------

## Returns

boolean

8.65.3.14 void Models::Project::setBeginDate ( [QDate](#) *beginDate* )

[Project::setBeginDate](#) Modify *beginDate* of a [Project](#)

## Parameters

<i>beginDate</i>	the new date of creation of the project
------------------	---

8.65.3.15 void Models::Project::setCost ( double *cost* )

[Project::setCost](#) Modify the [Project](#) *cost*

## Parameters

<i>cost</i>	the projectCost
-------------	-----------------

8.65.3.16 void Models::Project::setCustomer ( [QSharedPointer](#)< [Customer](#) > *customer* )

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

## Parameters

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

8.65.3.17 void Models::Project::setDailyRate ( double *dailyRate* )

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

## Parameters

---

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

8.65.3.18 void Models::Project::setDescription ( const QString & *description* )

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

Parameters

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

8.65.3.19 void Models::Project::setEndDate ( QDate *endDate* )

[Project::setEndDate](#) Modify *endDate* of [Project](#)

Parameters

<i>endDate</i>	the new end date of the project
----------------	---------------------------------

8.65.3.20 void Models::Project::setName ( const QString & *name* )

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

Parameters

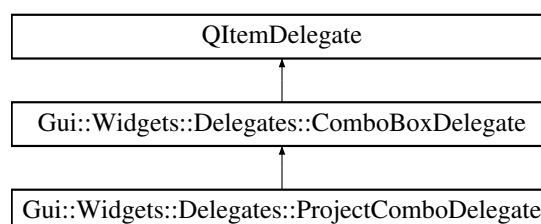
<i>name</i>	<a href="#">Project</a> name
-------------	------------------------------

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

## 8.66 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

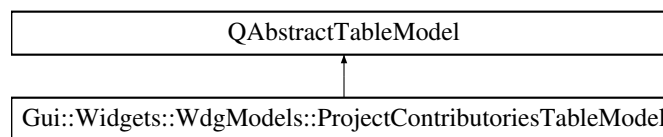
- void **setLocked** (bool getLocked)

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

## 8.67 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)

### 8.67.1 Constructor & Destructor Documentation

8.67.1.1 Gui::Widgets::WdgModels::ProjectContributoriesTableModel::ProjectContributoriesTableModel ( QObject \* parent = 0 )

[ContributoriesTableModel](#) Construct a [ContributoriesTableModel](#).

## Parameters

<i>parent</i>	Parent widget
---------------	---------------

## 8.67.2 Member Function Documentation

8.67.2.1 `int Gui::Widgets::WdgModels::ProjectContributoriesTableModel::columnCount ( const QModelIndex & ) const`

columnCount Number of column of a contributory

## Returns

The number of column

8.67.2.2 `QVariant Gui::Widgets::WdgModels::ProjectContributoriesTableModel::data ( const QModelIndex & index, int role ) const`

data Obtains data of a specify cell

## Parameters

<i>index</i>	The cell who we want data
<i>role</i>	The role of set

## Returns

The data of cell

8.67.2.3 `Qt::ItemFlags Gui::Widgets::WdgModels::ProjectContributoriesTableModel::flags ( const QModelIndex & index ) const`

flags Differents table flags

## Parameters

<i>index</i>	The cell who we want to know flags
--------------	------------------------------------

## Returns

Flags

8.67.2.4 `QVariant Gui::Widgets::WdgModels::ProjectContributoriesTableModel::headerData ( int section, Qt::Orientation orientation, int role ) const`

headerData Obtains header title of table

## Parameters

<i>section</i>	The number of column
<i>orientation</i>	The table orientation
<i>role</i>	

## Returns

The Title header of column

8.67.2.5 `int Gui::Widgets::WdgModels::ProjectContributoriesTableModel::rowCount ( const QModelIndex & ) const`

rowCount Number of contributories row

#### Returns

The number of contributories

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

setData Change data of a cell

#### Parameters

<i>index</i>	The cell to change data
<i>value</i>	The new value
<i>role</i>	The role of cell

#### Returns

True if we could edit

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

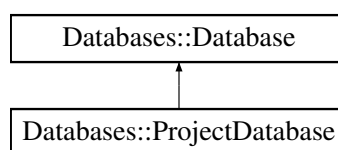
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/widgetmodels/projectcontributoriestablemodel.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/widgetmodels/projectcontributoriestablemodel.cpp

## 8.68 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 pld)`  
*ProjectDatabase::getProject* Get informations about the project identified by 'pld'.
- `Models::Project * getProject (QStringQuery &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 pld)`  
*removeProject* Remove the project with the id 'pld'

- int [getNbProjects](#) ()  
*ProjectDatabase::getNbProjects* Return the number of projects existing.
- int [getNbProjectsForACustomer](#) (const int pld)  
*ProjectDatabase::getNbProjectsForACustomer* Return the number of projects existing for an identify customer pld
- QMap< int, [Models::Project](#) > [getProjectsOfCustomer](#) (QSharedPointer< [Models::Customer](#) > c)  
*getProjectsOfCustomer* Return all projects of a customer
- [WdgModels::ProjectsTableModel](#) \* [getProjectsTable](#) (const int pld) 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

### 8.68.1 Detailed Description

The [ProjectDatabase](#) class Project table database.

#### Author

Florent Berbie

#### See Also

[Database](#)  
[Project](#)

### 8.68.2 Member Function Documentation

#### 8.68.2.1 int Databases::ProjectDatabase::addProject ( const Models::Project & pProject )

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

#### Returns

project id

#### 8.68.2.2 QList< Project \* > Databases::ProjectDatabase::getAllProjects ( )

[Project::getAllProjects](#) Return the list of projects.

#### Returns

Projects list



**8.68.2.3** `int Databases::ProjectDatabase::getNbProjects ( )`

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

**Returns**

number of projects

**8.68.2.4** `int Databases::ProjectDatabase::getNbProjectsForACustomer ( const int pld )`

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

**Parameters**

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

**Returns**

number of projects

**8.68.2.5** `Models::Project * Databases::ProjectDatabase::getProject ( const int pld )`

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

**Parameters**

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

**Returns**

the project

**8.68.2.6** `Models::Project * Databases::ProjectDatabase::getProject ( QSqlQuery & q )`

[ProjectDatabase::getProject](#).

**Parameters**

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

**Returns****8.68.2.7** `QMap< int, Models::Project > Databases::ProjectDatabase::getProjectsOfCustomer ( QSharedPointer< Models::Customer > c )`

[getProjectsOfCustomer](#) Return all projects of a customer

**Parameters**

<i>c</i>	The customer
----------	--------------

**Returns**

All projects of *c* with id in key

**8.68.2.8 WdgModels::ProjectsTableModel \* Databases::ProjectDatabase::getProjectsTable ( const int *pld* ) throw DbException \*)**

getProjectsTable Return all projects of a customer in QStandardItemModel

## Parameters

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

## Returns

QStandardItemModel an item model for QTableView

### 8.68.2.9 ProjectDatabase \* Databases::ProjectDatabase::instance ( ) throw DbException \*) [static]

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

## Returns

Instance of [ProjectDatabase](#)

### 8.68.2.10 void Databases::ProjectDatabase::removeProject ( const int *pld* )

removeProject Remove the project with the id 'pld'

## Parameters

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

### 8.68.2.11 QSharedPointer< Models::Project > Databases::ProjectDatabase::updateProject ( QSqlQuery & *q* )

getProject Obtain a project without new query

## Parameters

<i>q</i>	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

## 8.69 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

## 8.70 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

## 8.71 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.

### 8.71.1 Detailed Description

The [ProjectsTableModel](#) class for a Project table.

#### Author

Florent Berbie

#### See Also

[Project](#)

### 8.71.2 Constructor & Destructor Documentation

#### 8.71.2.1 Gui::Widgets::WdgModels::ProjectsTableModel::ProjectsTableModel ( )

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

##### Parameters

<i>parent</i>	Parent widget
---------------	---------------

### 8.71.3 Member Function Documentation

#### 8.71.3.1 void Gui::Widgets::WdgModels::ProjectsTableModel::append ( const Project & *project* )

[ProjectsTableModel::append](#) Add a new line in table.

##### Parameters

<i>Project</i>	The new Project
----------------	-----------------

#### 8.71.3.2 int Gui::Widgets::WdgModels::ProjectsTableModel::columnCount ( const QModelIndex & ) const

[ProjectsTableModel::columnCount](#) Number of column of a Project.

##### Returns

The number of column

#### 8.71.3.3 int Gui::Widgets::WdgModels::ProjectsTableModel::count ( )

[ProjectsTableModel::count](#) Number of projects in table.

##### Returns

The number of projects

#### 8.71.3.4 QVariant Gui::Widgets::WdgModels::ProjectsTableModel::data ( const QModelIndex & *index*, int *role* = Qt::DisplayRole ) const

[ProjectsTableModel::data](#) Obtains data of a specify cell.

## Parameters

<i>index</i>	The cell who we want data
<i>role</i>	The role of set

## Returns

The data of cell

8.71.3.5 `Qt::ItemFlags Gui::Widgets::WdgModels::ProjectsTableModel::flags ( const QModelIndex & index ) const`

[ProjectsTableModel::flags](#) Differents table flags.

## Parameters

<i>index</i>	The cell who we want to know flags
--------------	------------------------------------

## Returns

Flags

8.71.3.6 `QList< Project > Gui::Widgets::WdgModels::ProjectsTableModel::getProjects ( void ) const`

[ProjectsTableModel::getProjects](#) Return the list of projects.

## Returns

list of projects

8.71.3.7 `QVariant Gui::Widgets::WdgModels::ProjectsTableModel::headerData ( int section, Qt::Orientation orientation, int role = Qt::DisplayRole ) const`

[ProjectsTableModel::headerData](#) Obtains header title of table.

## Parameters

<i>section</i>	The number of column
<i>orientation</i>	The table orientation
<i>role</i>	

## Returns

The Title header of column

8.71.3.8 `void Gui::Widgets::WdgModels::ProjectsTableModel::remove ( const int i )`

[ProjectsTableModel::remove](#) Remove a line.

## Parameters

<i>i</i>	The number of line to remove
----------	------------------------------

8.71.3.9 `int Gui::Widgets::WdgModels::ProjectsTableModel::rowCount ( const QModelIndex & ) const`

[ProjectsTableModel::rowCount](#) Number of projects row.

#### Returns

The number of projects

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

[ProjectsTableModel::setData](#) Change data of a cell.

#### Parameters

<i>index</i>	The cell to change data
<i>value</i>	The new value
<i>role</i>	The role of cell

#### Returns

True if we could edit

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

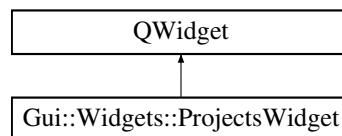
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/widgetmodels/projectstablemodel.h
- /home/travis/build/FACT-Team/FactDev/src/gui/widgets/widgetmodels/projectstablemodel.cpp

## 8.72 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::addProject* 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](#).*

### 8.72.1 Detailed Description

The [ProjectsWidget](#) class Actions on Project.

#### Author

Florent Berbie

### 8.72.2 Constructor & Destructor Documentation

#### 8.72.2.1 [Gui::Widgets::ProjectsWidget::ProjectsWidget](#) ( [QWidget](#) \* *parent* = 0 ) [explicit]

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

#### Parameters

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

### 8.72.3 Member Function Documentation

#### 8.72.3.1 void [Gui::Widgets::ProjectsWidget::updateBtn](#) ( bool *b* ) [slot]

[ProjectsWidget::updateBtn](#) Update the toolbar in tblProjects.

#### Parameters

<i>boolean</i>	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

## 8.73 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)

### 8.73.1 Detailed Description

A context implementation which wraps a QVariantHash or QVariantMap.

### 8.73.2 Member Typedef Documentation

8.73.2.1 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.

### 8.73.3 Member Function Documentation

8.73.3.1 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](#).

8.73.3.2 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](#).

#### 8.73.3.3 `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](#).

#### 8.73.3.4 `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](#).

#### 8.73.3.5 `void QtVariantContext::pop ( )` [virtual]

Exit the current context.

Implements [Mustache::Context](#).

#### 8.73.3.6 `void QtVariantContext::push ( const QString & key, int index = -1 )` [virtual]

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

Implements [Mustache::Context](#).

#### 8.73.3.7 `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`

## 8.74 Models::Rate Class Reference

### Public Member Functions

- **Rate** (double hourly)
- **Rate** (int idBilling, int idProject)
- void **setDailyRate** (const double &)
- void **setHourlyRate** (const double &)

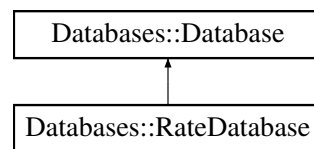
- double **getDailyRate** () const
- double **getHourlyRate** () const
- int **getNbDailyHours** () const
- double **getLegalRate** () const

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

## 8.75 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

#### 8.75.1 Member Function Documentation

8.75.1.1 **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

## 8.76 RateModelTest Class Reference

Inheritance diagram for RateModelTest:



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

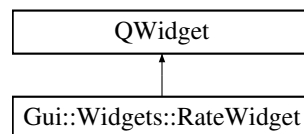
- /home/travis/build/FACT-Team/FactDev/tests/models/ratemodeltest.h
- /home/travis/build/FACT-Team/FactDev/tests/models/ratemodeltest.cpp

## 8.77 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*

### 8.77.1 Detailed Description

Class for display Rate.

Author

Florent Berbie

### 8.77.2 Constructor & Destructor Documentation

**8.77.2.1** `Gui::Widgets::RateWidget::RateWidget ( QWidget * parent = 0 )` `[explicit]`

[RateWidget::RateWidget](#) Construct a rate widget.

Parameters

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

### 8.77.3 Member Function Documentation

**8.77.3.1** `double Gui::Widgets::RateWidget::getDailyRate ( )`

[RateWidget::getDailyRate](#) Get the daily rate.

Returns

The daily rate

**8.77.3.2** `double Gui::Widgets::RateWidget::getHourlyRate ( )`

[RateWidget::getHourlyRate](#) Get the hourly rate.

Returns

The hourly rate

**8.77.3.3** `void Gui::Widgets::RateWidget::setDailyRate ( )` `[slot]`

[setDailyRate](#) Set a new value for the daily rate

Parameters

<i>dailyRate</i>	The new daily rate
------------------	--------------------

**8.77.3.4** `void Gui::Widgets::RateWidget::setHourlyRate ( )` `[slot]`

[setHourlyRate](#) Set a new value for the hourly rate

Parameters

<i>hourlyRate</i>	The new hourly rate
-------------------	---------------------

**8.77.3.5** `void Gui::Widgets::RateWidget::setWidgetDailyRateValue ( double value )`

[RateWidget::setWidgetDailyRateValue](#) Modify the *value* of the daily rate spin box component.

## Parameters

<i>value</i>	New Value
--------------	-----------

8.77.3.6 void Gui::Widgets::RateWidget::setWidgetHourlyRateValue ( double *value* )

[RateWidget::setWidgetHourlyRateValue](#) Modify the *value* of the hourly rate spin box component.

## Parameters

<i>value</i>	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

## 8.78 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)

### 8.78.1 Detailed Description

Renders Mustache templates, replacing mustache tags with values from a provided context.

### 8.78.2 Member Function Documentation

#### 8.78.2.1 QString Renderer::error ( ) const

Returns a message describing the last error encountered by the previous [render\(\)](#) call.

#### 8.78.2.2 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.

#### 8.78.2.3 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.

#### 8.78.2.4 QString Renderer::render ( const QString & *\_template*, Context \* *context* )

Render a Mustache template, using `context` to fetch the values used to replace Mustache tags.

#### 8.78.2.5 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

## 8.79 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 activated.*
- 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.*

### 8.79.1 Detailed Description

The [Search](#) class.

Author

Antoine de Roquemaurel  
 Florent Berbie

### 8.79.2 Member Function Documentation

#### 8.79.2.1 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

<i>filter</i>	Text inputed in searchbar
<i>list</i>	List of bills or quotes

#### 8.79.2.2 void Models::Search::filterOnCompany ( QString & filter, const QStringList list )

[Search::filterOnCompany](#) [Search](#) all companies contained in the *list* of the *filter*

Parameters

<i>filter</i>	Text inputed in searchbar
<i>list</i>	List of companies

#### 8.79.2.3 void Models::Search::filterOnContributories ( QString & filter, const QStringList list )

[Search::filterOnContributories](#) [Search](#) all contributories contained in the *list* of the *filter*



## Parameters

<i>filter</i>	Text inputed in searchbar
<i>list</i>	List of contributories

8.79.2.4 void Models::Search::filterOnCustomersWithoutProject ( QString & *filter*, const QStringList *list* )

[Search::filterOnCustomersWithoutProject Search](#) all customers which are contained in the *list* of the *filter*

## Parameters

<i>filter</i>	Text inputed in searchbar
<i>list</i>	List of bills or quotes

8.79.2.5 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

<i>filter</i>	Text inputed in the searchbar
<i>list</i>	List of numbers from the <i>filter</i>
<i>element</i>	Attribute name into the database

8.79.2.6 void Models::Search::filterOnProjects ( QString & *filter*, const QStringList *list* )

[Search::filterOnProjects Search](#) all projects contained in the *list* of the *filter*

## Parameters

<i>filter</i>	Text inputed in searchbar
<i>list</i>	List of projects

8.79.2.7 void Models::Search::filterOnReferentLastname ( QString & *filter*, const QStringList *list* )

[Search::filterOnReferentLastname Search](#) all referents last name contained in the *list* of the *filter*

## Parameters

<i>filter</i>	Text inputed in searchbar
<i>list</i>	List of referent last name

8.79.2.8 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

<i>filter</i>	Text inputed in the searchbar
<i>list</i>	List of words from the <i>filter</i>

<i>element</i>	Attribute name into the database
----------------	----------------------------------

#### 8.79.2.9 QString Models::Search::getFilter ( )

[Search::getFilter](#) Return the search filter.

##### Returns

filter selected (sql portion)

#### 8.79.2.10 bool Models::Search::getGroupFilter ( ) const

[Search::getGroupFilter](#) Return if the filter is activated.

##### Returns

boolean if search filter is activated

#### 8.79.2.11 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

#### 8.79.2.12 bool Models::Search::getSearchInCompanies ( ) const

[Search::getSearchInCompanies](#) Return if we search a company.

##### Returns

boolean if we search a company

#### 8.79.2.13 bool Models::Search::getSearchInProjects ( ) const

[Search::getSearchInProjects](#) Return TRUE if it exists one or more projects in database.

##### Returns

boolean if project are existing

#### 8.79.2.14 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

## 8.79.2.15 QString Models::Search::getText ( ) const

[Search::getText](#) Return sql portion of filter.

## Returns

QString the sql portion

## 8.79.2.16 bool Models::Search::searchInContributories ( ) const

[Search::getSearchInContributories](#) Return TRUE if it exists one or more projets in database.

## Returns

boolean if contributories are existing

8.79.2.17 void Models::Search::setGroupFilter ( bool *getGroupFilter* )

[Search::setGroupFilter](#) Modify if we active search filter.

## Parameters

<i>getGroupFilter</i>	Get if filter is actived
-----------------------	--------------------------

8.79.2.18 void Models::Search::setSearchInBillsQuotes ( bool *searchInBillsQuotes* )

[Search::setSearchInBillsQuotes](#) Modify the filter of bills and quotes search.

## Parameters

<i>searchInBills-Quotes</i>	<a href="#">Search</a> in bills or quotes which are concerned
-----------------------------	---

8.79.2.19 void Models::Search::setSearchInCompanies ( bool *searchInCompanies* )

[Search::setSearchInCompanies](#) Modify the filter of companies search.

## Parameters

<i>getSearchIn-Companies</i>	<a href="#">Search</a> in companies is concerned
------------------------------	--

8.79.2.20 void Models::Search::setSearchInContributories ( bool *searchInContributories* )

[Search::setSearchInContributories](#) Modify the filter of contributories search.

## Parameters

<i>searchIn-Contributories</i>	<a href="#">Search</a> in contributories which are concerned
--------------------------------	--

8.79.2.21 void Models::Search::setSearchInProjects ( bool *searchInProjects* )

[Search::setSearchInProjects](#) Modify the filter of projects search.

## Parameters

<i>searchInProjects</i>	<a href="#">Search</a> in projects which are concerned
-------------------------	--

8.79.2.22 void Models::Search::setSearchInReferentLastname ( bool *searchInReferentLastname* )

[Search::setSearchInReferentLastname](#) Modify the filter of referents last name search.

## Parameters

<i>searchIn-Referent-Lastname</i>	<a href="#">Search</a> in referents last name which are concerned
-----------------------------------	---

8.79.2.23 void Models::Search::setText ( const QString & *getText* )

[Search::setText](#) Modify sql portion.

## Parameters

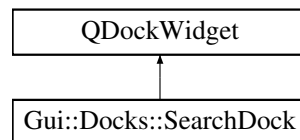
<i>getText</i>	Get sql portion
----------------	-----------------

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

## 8.80 Gui::Docks::SearchDock Class Reference

Inheritance diagram for Gui::Docks::SearchDock:



### Public Slots

- void **search** (QString text)

### Signals

- void **textChanged** (QString text)

### Public Member Functions

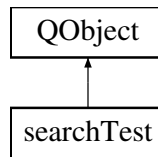
- **SearchDock** (QWidget \*parent=0, Qt::WindowFlags flags=0)

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

- /home/travis/build/FACT-Team/FactDev/src/gui/docks/searchdock.h
- /home/travis/build/FACT-Team/FactDev/src/gui/docks/searchdock.cpp

## 8.81 searchTest Class Reference

Inheritance diagram for searchTest:



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

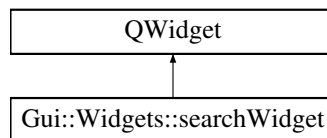
- /home/travis/build/FACT-Team/FactDev/tests/models/searchtest.h
- /home/travis/build/FACT-Team/FactDev/tests/models/searchtest.cpp

## 8.82 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 (First-name, 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)
- int **getIdCustomer** () const
- void **setIdCustomer** (int getIdCustomer)

### 8.82.1 Detailed Description

Class for search in database.

Author

Antoine de Roquemaurel

### 8.82.2 Constructor & Destructor Documentation

8.82.2.1 `Gui::Widgets::searchWidget::searchWidget ( QWidget * parent = 0 ) [explicit]`

[searchWidget::searchWidget](#) Construct a search widget

Parameters

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

### 8.82.3 Member Function Documentation

8.82.3.1 `int Gui::Widgets::searchWidget::getCurrentCustomerId ( )`

[searchWidget::getCurrentCustomerId](#) Return the id of the customer selected in the table

Returns

id of the current customer

8.82.3.2 `bool Gui::Widgets::searchWidget::isCustomerSelected ( ) const`

`isCustomerSelected` Return TRUE if a customer is selected, else FALSE

Returns

boolean

8.82.3.3 `void Gui::Widgets::searchWidget::search ( QString toSearch ) [slot]`

`searchWidget::launch` a search

Parameters

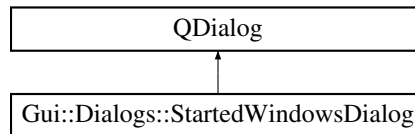
<i>toSearch</i>	The value to search
-----------------	---------------------

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`

## 8.83 Gui::Dialogs::StartedWindowsDialog Class Reference

Inheritance diagram for `Gui::Dialogs::StartedWindowsDialog`:



## Public Slots

- void [checkFields](#) ()  
*StartedWindowsDialog::checkFields* Check all fields of dialog components.
- void [backToPage2](#) ()  
*StartedWindowsDialog::backToPage2* Return to the second page.
- void [nextToPage2](#) ()  
*StartedWindowsDialog::nextToPage2* Go to the second page.
- void [nextToPage3](#) ()  
*StartedWindowsDialog::nextToPage3* Go to the third page.
- void [databaseTypeChanged](#) (const int index)  
*StartedWindowsDialog::databaseTypeChanged* Listener on the Database type combobox.
- void [updateNextButton](#) ()  
*StartedWindowsDialog::updateNextButton* Check if the next button is enabled.
- void [accept](#) ()  
*StartedWindowsDialog::accept* Valid data inputed by user and add these data in Database.

## Public Member Functions

- **StartedWindowsDialog** (QWidget \*parent=0)
- void [fillFields](#) ()  
*StartedWindowsDialog::fillFields* Fill line edits with the data of the user.
- QPixmap [getImage](#) (QString path, int width=256, int height=256)  
*StartedWindowsDialog::getImage*.
- bool [isDatabaseTypeValid](#) ()  
*StartedWindowsDialog::isDatabaseTypeValid* Return TRUE if settings of database type is valid else return FALSE.
- bool [isDatabaseCentralized](#) ()  
*StartedWindowsDialog::isDatabaseCentralized* Return TRUE if the current type of database is centralized else FALSE.

### 8.83.1 Member Function Documentation

8.83.1.1 void [Gui::Dialogs::StartedWindowsDialog::databaseTypeChanged](#) ( const int *index* ) [slot]

[StartedWindowsDialog::databaseTypeChanged](#) Listener on the Database type combobox.

#### Parameters

<i>index</i>	Current index selected
--------------	------------------------

8.83.1.2 QPixmap [Gui::Dialogs::StartedWindowsDialog::getImage](#) ( QString *path*, int *width* = 256, int *height* = 256 )

[StartedWindowsDialog::getImage](#).

## Parameters

<i>path</i>	
<i>width</i>	
<i>height</i>	

## Returns

8.83.1.3 `bool Gui::Dialogs::StartedWindowsDialog::isDatabaseCentralized ( )`

[StartedWindowsDialog::isDatabaseCentralized](#) Return TRUE if the current type of database is centralized else FALSE.

## Returns

boolean

8.83.1.4 `bool Gui::Dialogs::StartedWindowsDialog::isDatabaseTypeValid ( )`

[StartedWindowsDialog::isDatabaseTypeValid](#) Return TRUE if settings of database type is valid else return FALSE.

## Returns

boolean on the database type validity

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

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

## 8.84 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*

### 8.84.1 Detailed Description

The Utils class.

#### Author

Antoine de Roquemaurel

### 8.84.2 Member Function Documentation

8.84.2.1 `QString Utils::String::firstLetterToUpper ( QString s )` [static]

[firstLetterToUpper](#) Put the first letter of a string in capslock



## Parameters

<i>s</i>	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/utls/string.h
- /home/travis/build/FACT-Team/FactDev/src/utls/string.cpp

## 8.85 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/utls/stringtest.h
- /home/travis/build/FACT-Team/FactDev/tests/utls/stringtest.cpp

## 8.86 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**

### 8.86.1 Detailed Description

Holds properties of a tag in a mustache template.

## 8.86.2 Member Enumeration Documentation

### 8.86.2.1 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`

## 8.87 `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/QStringRunner/testadder.cpp`

## 8.88 `testadder` Class Reference

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

- `/home/travis/build/FACT-Team/FactDev/tests/QStringRunner/testadder.h`

## 8.89 `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`

## 8.90 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`

## 8.91 Generator::TexGenerator Class Reference

### Public Member Functions

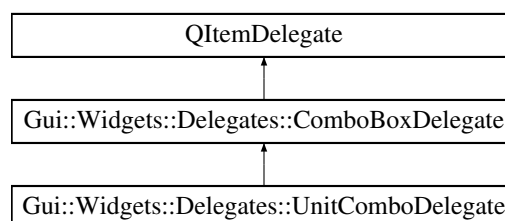
- `TexGenerator (QString tpl)`
- `void generate (QVariantHash data, QString path)`

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

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

## 8.92 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:

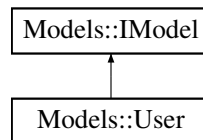
- `/home/travis/build/FACT-Team/FactDev/src/gui/widgets/delegates/unitcombodelegate.h`
- `/home/travis/build/FACT-Team/FactDev/src/gui/widgets/delegates/unitcombodelegate.cpp`

## 8.93 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. Construct an User.*
- `User (int id)`  
*User::User. Construct a User with the identify id*
- `void commit ()`  
*User::commit Update user data in User table on the database.*
- `void hydrat (int id=1)`  
*User::hydrat Get data of the user who is specified by id from the database.*
- `void remove ()`  
*remove Remove the current User*
- `QVariantHash getDataMap ()`  
*getDataMap Get all data of model with a HashMap key/value*
- `void updateFolders (void)`  
*MainWindow::updateFolders Make directories which contain quotes and billings. Directories are the same than theirs of the Tree organisation (without Projects).*
- `QString getFirstname () const`  
*User::getFirstname Return the user firstname.*
- `void setFirstname (const QString &firstname)`  
*User::setFirstname Modify the user firstname*
- `QString getLastName () const`  
*User::getLastName Return the user lastname.*
- `void setLastName (const QString &lastname)`  
*User::setLastName Modify the user lastname*
- `QString getCompany () const`  
*User::getCompany Return the user company.*
- `void setCompany (const QString &company)`  
*User::setCompany Modify the user company name.*
- `QString getTitle () const`  
*User::getTitle Return a short description of User (company) activity.*
- `void setTitle (const QString &title)`  
*User::setTitle Modify the user/company activities description*
- `QString getAddress () const`  
*User::getAddress Return the company address (Number and name of street)*
- `void setAddress (const QString &address)`  
*User::setAddress Modify the user company address*
- `QString getPostalCode () const`

- User::getPostalCode* Return the postal code.
- void **setPostalCode** (const QString &postalCode)
  - User::setPostalCode* Modify the postal code postalCode
- QString **getCity** () const
  - User::getCity* Return the city.
- void **setCity** (const QString &city)
  - User::setCity* Modify the city
- QString **getEmail** () const
  - User::getEmail* Return the user professional email
- void **setEmail** (const QString &email)
  - User::setEmail* Modify the user professional email
- QString **getMobilePhone** () const
  - User::getMobilePhone* Return the number of the professional mobile phone.
- void **setMobilePhone** (const QString &mobilePhone)
  - User::setMobilePhone* Modify the number of the professional user mobile phone mobilePhone
- QString **getPhone** () const
  - User::getPhone* Return the number of the desktop phone.
- void **setPhone** (const QString &phone)
  - User::setPhone* Modify the number of the desktop phone
- 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 different to the other **User** c Return TRUE if both Users are different, else FALSE.

## Additional Inherited Members

### 8.93.1 Detailed Description

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

Author

Florent Berbie

### 8.93.2 Constructor & Destructor Documentation

#### 8.93.2.1 Models::User::User ( int id )

**User::User**. Construct a **User** with the identify *id*

## Parameters

<i>id</i>	<a href="#">User</a> id
-----------	-------------------------

### 8.93.3 Member Function Documentation

#### 8.93.3.1 QString Models::User::getAddress ( ) const

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

## Returns

Address company

#### 8.93.3.2 QString Models::User::getCity ( ) const

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

## Returns

city

#### 8.93.3.3 QString Models::User::getCompany ( ) const

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

## Returns

New company name

#### 8.93.3.4 QVariantHash Models::User::getDataMap ( ) [virtual]

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

## Returns

Model's data

Implements [Models::IModel](#).

#### 8.93.3.5 QString Models::User::getEmail ( ) const

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

## Returns

professional email

#### 8.93.3.6 QString Models::User::getFirstname ( ) const

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

## Returns

[User](#) firstname

#### 8.93.3.7 QString Models::User::getLastname ( ) const

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

Returns

[User](#) lastname

#### 8.93.3.8 QString Models::User::getMobilePhone ( ) const

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

Returns

number of mobile phone

#### 8.93.3.9 QString Models::User::getNoSiret ( ) const

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

Returns

SIRET number

#### 8.93.3.10 QString Models::User::getPhone ( ) const

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

Returns

number of the desktop phone

#### 8.93.3.11 QString Models::User::getPostalCode ( ) const

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

Returns

postal code

#### 8.93.3.12 QString Models::User::getTitle ( ) const

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

Returns

a short description of user (company) activity

#### 8.93.3.13 QString Models::User::getWorkspaceName ( ) const

[User::getWorkspaceName](#) Return the name of the workspace user.

Returns

workspace name

#### 8.93.3.14 QString Models::User::getWorkspacePath ( ) const

[User::getWorkspacePath](#) Return the path of the workspace user.

##### Returns

workspace path

#### 8.93.3.15 void Models::User::hydrat ( int *id* = 1 ) [virtual]

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

##### Parameters

<i>id</i>	<a href="#">User</a> identify
-----------	-------------------------------

Implements [Models::IModel](#).

#### 8.93.3.16 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

<i>u</i>	<a href="#">User</a> to compare
----------	---------------------------------

##### Returns

boolean

#### 8.93.3.17 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

<i>u</i>	<a href="#">User</a> to compare
----------	---------------------------------

##### Returns

boolean

#### 8.93.3.18 void Models::User::setAddress ( const QString & *address* )

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

##### Parameters

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

#### 8.93.3.19 void Models::User::setCity ( const QString & *city* )

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



## Parameters

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

8.93.3.20 void Models::User::setCompany ( const QString & *company* )

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

## Parameters

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

8.93.3.21 void Models::User::setEmail ( const QString & *email* )

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

## Parameters

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

8.93.3.22 void Models::User::setFirstname ( const QString & *firstname* )

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

## Parameters

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

8.93.3.23 void Models::User::setLastname ( const QString & *lastname* )

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

## Parameters

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

8.93.3.24 void Models::User::setMobilePhone ( const QString & *mobilePhone* )

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

## Parameters

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

8.93.3.25 void Models::User::setNoSiret ( const QString & *noSiret* )

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

## Parameters

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

8.93.3.26 void Models::User::setPhone ( const QString & *phone* )

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

## Parameters

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

8.93.3.27 void Models::User::setPostalCode ( const QString & *postalCode* )

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

## Parameters

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

8.93.3.28 void Models::User::setTitle ( const QString & *title* )

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

## Parameters

<i>title</i>	Short description on activity(ies) of <a href="#">User</a> company
--------------	--

8.93.3.29 void Models::User::setWorkspaceName ( const QString & *workspaceName* )

[User::setWorkspaceName](#) Change the current workspace name by the new *workspaceName*

## Parameters

<i>workspaceName</i>	
----------------------	--

8.93.3.30 void Models::User::setWorkspacePath ( const QString & *workspacePath* )

[User::setWorkspacePath](#) Change the current workspace path by the new *workspacePath*

## Parameters

<i>workspacePath</i>	
----------------------	--

8.93.3.31 void Models::User::updateFolders ( void )

MainWindow::updateFolders Make directories which contain quotes and billings. Directories are the same than theirs of the Tree organisation (without Projects).

Organisation of folders are formed like this:

- COMPANY CustomerLastname CustomerFirstname/
  - Quotes/
    - \* quote1 ...
  - Billings/
    - \* billing1 ...

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

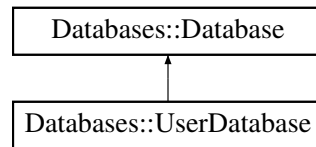
- /home/travis/build/FACT-Team/FactDev/src/models/user.h
- /home/travis/build/FACT-Team/FactDev/src/models/user.cpp

## 8.94 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

#### 8.94.1 Detailed Description

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

#### Author

Florent Berbie

#### See Also

[Database](#)  
[User](#)

#### 8.94.2 Member Function Documentation

##### 8.94.2.1 [Models::User](#) \* [Databases::UserDatabase::getUser](#) ( const int *pId* = 1 )

[getUser](#) Get informations about the user (identified by 'pId')

#### Parameters

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

#### Returns

the user

8.94.2.2 **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

## 8.95 **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

## 8.96 **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*

### 8.96.1 Detailed Description

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

#### Author

Florent Berbie

#### See Also

Project

### 8.96.2 Constructor & Destructor Documentation

#### 8.96.2.1 Gui::Dialogs::UserDataDialog::UserDataDialog ( QWidget \* *parent* = 0 ) [explicit]

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

#### Parameters

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

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

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

## 8.97 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, [66](#)
- addBill
  - Gui::MainWindow, [115](#)
- addBilling
  - Databases::BillingDatabase, [32](#)
- addBillingProject
  - Databases::BillingDatabase, [32](#)
- addBillingToProject
  - Utils::HierarchicalSystem, [106](#)
- addContributory
  - Databases::ContributoryDatabase, [70](#)
  - Models::Billing, [26](#)
- addCustomer
  - Databases::CustomerDatabase, [86](#)
- addDoc
  - Gui::MainWindow, [115](#)
- addProject
  - Databases::ProjectDatabase, [132](#)
  - Gui::MainWindow, [115](#)
- AddProjectDialog
  - Gui::Dialogs::AddProjectDialog, [22](#)
- addProjectToCustomer
  - Utils::HierarchicalSystem, [106](#)
- addQuote
  - Gui::MainWindow, [115](#)
- AddQuoteDialog
  - Gui::Dialogs::AddQuoteDialog, [23](#)
- append
  - Gui::Widgets::WdgModels::BillingsTableModel, [36](#)
  - Gui::Widgets::WdgModels::ContributoriesTableModel, [63](#)
  - Gui::Widgets::WdgModels::CustomersTableModel, [91](#)
  - Gui::Widgets::WdgModels::ProjectsTableModel, [136](#)
- Billing
  - Models::Billing, [26](#)
- BillingDatabaseTest, [35](#)
- BillingModelTest, [35](#)
- BillingsTableModel
  - Gui::Widgets::WdgModels::BillingsTableModel, [36](#)
- canEval
  - CounterContext, [75](#)
  - Mustache::Context, [59](#)
  - Mustache::QtVariantContext, [140](#)
- check
  - Gui::Widgets::CheckFields::CheckEmail, [42](#)
  - Gui::Widgets::CheckFields::CheckFieldsLetters, [43](#)
  - Gui::Widgets::CheckFields::CheckFieldsNumbers, [45](#)
  - Gui::Widgets::CheckFields::CheckIpAddress, [46](#)
  - Gui::Widgets::CheckFields::CheckLogin, [47](#)
  - Gui::Widgets::CheckFields::CheckPhone, [49](#)
  - Gui::Widgets::CheckFields::CheckPortNumber, [51](#)
  - Gui::Widgets::CheckFields::CheckPostalCode, [52](#)
  - Gui::Widgets::CheckFields::CheckSiretNumber, [55](#)
  - Gui::Widgets::CheckFields::CheckUntilField, [56](#)
  - Gui::Widgets::CheckFields::CheckValidField, [57](#)
  - Gui::Widgets::CheckFields::ICheckField, [107](#)
- CheckCity
  - Gui::Widgets::CheckFields::CheckCity, [40](#)
- CheckCountry
  - Gui::Widgets::CheckFields::CheckCountry, [41](#)
- CheckEmail
  - Gui::Widgets::CheckFields::CheckEmail, [42](#)
- CheckFieldsLetters
  - Gui::Widgets::CheckFields::CheckFieldsLetters, [43](#)
- CheckFieldsNumbers
  - Gui::Widgets::CheckFields::CheckFieldsNumbers, [44](#)
- CheckIpAddress
  - Gui::Widgets::CheckFields::CheckIpAddress, [46](#)
- CheckLogin
  - Gui::Widgets::CheckFields::CheckLogin, [47](#)
- CheckName
  - Gui::Widgets::CheckFields::CheckName, [48](#)
- CheckPhone
  - Gui::Widgets::CheckFields::CheckPhone, [49](#)
- CheckPortNumber
  - Gui::Widgets::CheckFields::CheckPortNumber, [51](#)
- CheckPostalCode
  - Gui::Widgets::CheckFields::CheckPostalCode, [52](#)
- CheckQLineEdit
  - Gui::Widgets::CheckFields::CheckQLineEdit, [53](#)
- CheckSiretNumber
  - Gui::Widgets::CheckFields::CheckSiretNumber, [55](#)
- CheckUntilField
  - Gui::Widgets::CheckFields::CheckUntilField, [56](#)
- CheckValidField
  - Gui::Widgets::CheckFields::CheckValidField, [57](#)
- columnCount
  - Gui::Widgets::WdgModels::BillingsTableModel, [37](#)
  - Gui::Widgets::WdgModels::ContributoriesTableModel, [63](#)
  - Gui::Widgets::WdgModels::CustomersTableModel, [91](#)

- Gui::Widgets::WdgModels::ProjectContributoriesTableModel, 129
  - Gui::Widgets::WdgModels::ProjectsTableModel, 136
- Comment
  - Mustache::Tag, 157
- Context
  - Mustache::Context, 59
- ContributoriesDatabaseTest, 61
- ContributoriesTableModel
  - Gui::Widgets::WdgModels::ContributoriesTableModel, 63
- Contributory
  - Models::Contributory, 67
- ContributoryListTest, 74
- ContributoryModelTest, 74
- costCompute
  - Models::Project, 125
- count
  - Gui::Widgets::WdgModels::BillingsTableModel, 37
  - Gui::Widgets::WdgModels::ContributoriesTableModel, 63
  - Gui::Widgets::WdgModels::CustomersTableModel, 91
  - Gui::Widgets::WdgModels::ProjectsTableModel, 136
- CounterContext, 74
  - canEval, 75
  - eval, 75
  - stringValue, 75, 76
- Customer
  - Models::Customer, 78
- CustomerContextualMenu
  - Gui::Widgets::CustomerContextualMenu, 84
- CustomerDatabaseTest, 88
- CustomerModelTest, 89
- CustomersTableModel
  - Gui::Widgets::WdgModels::CustomersTableModel, 91
- DB\_FILENAME
  - Parameters, 119
- data
  - Gui::Widgets::WdgModels::BillingsTableModel, 37
  - Gui::Widgets::WdgModels::ContributoriesTableModel, 63
  - Gui::Widgets::WdgModels::CustomersTableModel, 91
  - Gui::Widgets::WdgModels::ProjectContributoriesTableModel, 130
  - Gui::Widgets::WdgModels::ProjectsTableModel, 136
- DatabaseSettingsWidget
  - Gui::Widgets::DatabaseSettingsWidget, 98
- databaseTypeChanged
  - Gui::Dialogs::StartedWindowsDialog, 154
- Databases, 17
- Databases::BillingDatabase, 30
  - addBilling, 32
  - addBillingProject, 32
  - getAllBillingsOfProject, 32
  - getBilling, 32
  - getBillings, 32
  - getBillingsTable, 33
  - getMaxBillingNumber, 33
  - getMaxBillingNumberOfCustomer, 33
  - getMaxQuoteNumber, 33
  - getMaxQuoteNumberOfCustomer, 33
  - instance, 34
  - isBillingPaid, 34
  - removeBilling, 34
  - removeBillingProject, 34
- Databases::ContributoryDatabase, 69
  - addContributory, 70
  - getContributoriesByBilling, 70
  - getContributoriesByBillingAndProject, 72
  - getContributory, 72
  - instance, 72
  - removeContributory, 72
- Databases::CustomerDatabase, 84
  - addCustomer, 86
  - getCustomer, 86
  - getCustomersTable, 86
  - getItemBillQuote, 86
  - getItemCustomer, 87
  - getItemProject, 87
  - getItemRoot, 87
  - getNbCustomers, 87
  - getTree, 87
  - instance, 88
  - removeCustomer, 88
  - updateCustomer, 88
- Databases::Database, 93
  - executeFile, 95
  - instance, 96
  - lastError, 96
  - setDatabase, 96
  - value, 96
- Databases::ProjectDatabase, 131
  - addProject, 132
  - getAllProjects, 132
  - getNbProjects, 132
  - getNbProjectsForACustomer, 132
  - getProject, 133
  - getProjectsOfCustomer, 133
  - getProjectsTable, 133
  - instance, 134
  - removeProject, 134
  - updateProject, 134
- Databases::RateDatabase, 142
  - instance, 142
- Databases::UserDatabase, 167
  - getUser, 167
  - instance, 167
- DbException
  - Exceptions::DbException, 100
- DialogAddCustomer



- Gui::Dialogs::DialogAddCustomer, 102
- editUser
  - Gui::MainWindow, 115
- error
  - Mustache::Renderer, 145
- errorPartial
  - Mustache::Renderer, 145
- errorPos
  - Mustache::Renderer, 145
- eval
  - CounterContext, 75
  - Mustache::Context, 59
  - Mustache::QtVariantContext, 140
- Exceptions::DbException, 100
  - DbException, 100
  - popupMessage, 101
- Exceptions::FileException, 104
  - FileException, 104
  - popupMessage, 104
- executeFile
  - Databases::Database, 95
- FileException
  - Exceptions::FileException, 104
- fillQuoteBilling
  - Gui::Dialogs::AddQuoteDialog, 23
- filterOnBillsOrQuotes
  - Models::Search, 147
- filterOnCompany
  - Models::Search, 147
- filterOnContributories
  - Models::Search, 147
- filterOnCustomersWithoutProject
  - Models::Search, 148
- filterOnNumberElements
  - Models::Search, 148
- filterOnProjects
  - Models::Search, 148
- filterOnReferentLastname
  - Models::Search, 148
- filterOnVarcharElements
  - Models::Search, 148
- firstLetterToUpper
  - Utils::String, 155
- flags
  - Gui::Widgets::WdgModels::BillingsTableModel, 37
  - Gui::Widgets::WdgModels::ContributoriesTable-Model, 64
  - Gui::Widgets::WdgModels::CustomersTableModel, 92
  - Gui::Widgets::WdgModels::ProjectContributories-TableModel, 130
  - Gui::Widgets::WdgModels::ProjectsTableModel, 137
- fn\_t
  - Mustache::QtVariantContext, 140
- Generation, 105
- Generator::PdfGenerator, 121
- Generator::TexGenerator, 158
- getAddress
  - Models::Customer, 78
  - Models::User, 161
- getAllBillingsOfProject
  - Databases::BillingDatabase, 32
- getAllProjects
  - Databases::ProjectDatabase, 132
- getBeginDate
  - Models::Project, 125
- getBilling
  - Databases::BillingDatabase, 32
- getBillings
  - Databases::BillingDatabase, 32
  - Gui::Widgets::WdgModels::BillingsTableModel, 37
- getBillingsTable
  - Databases::BillingDatabase, 33
- getBtnValid
  - Gui::Widgets::CheckFields::CheckQLineEdit, 54
- getCity
  - Models::Customer, 78
  - Models::User, 161
- getCompany
  - Models::Customer, 78
  - Models::User, 161
- getContributories
  - Gui::Widgets::ContributoriesWidget, 66
  - Gui::Widgets::WdgModels::ContributoriesTable-Model, 64
  - Models::Billing, 26
- getContributoriesByBilling
  - Databases::ContributoryDatabase, 70
- getContributoriesByBillingAndProject
  - Databases::ContributoryDatabase, 72
- getContributory
  - Databases::ContributoryDatabase, 72
- getCopy
  - Gui::Dialogs::AddQuoteDialog, 23
- getCost
  - Models::Project, 125
- getCountry
  - Gui::Widgets::CheckFields::CheckPhone, 49
  - Gui::Widgets::CheckFields::CheckPostalCode, 52
  - Models::Customer, 78
- getCurrentCustomerId
  - Gui::MainWindow, 115
  - Gui::Widgets::searchWidget, 153
- getCurrentCustomerName
  - Gui::MainWindow, 116
- getCurrentProjectId
  - Gui::MainWindow, 116
- getCurrentProjectName
  - Gui::MainWindow, 116
- getCurrentQuoteld
  - Gui::MainWindow, 116
- getCustomer
  - Databases::CustomerDatabase, 86

- Models::Project, 125
- getCustomers
  - Gui::Widgets::WdgModels::CustomersTableModel, 92
  - Utils::HierarchicalSystem, 106
- getCustomersTable
  - Databases::CustomerDatabase, 86
- getDailyRate
  - Gui::Widgets::RateWidget, 144
  - Models::Project, 125
- getDataMap
  - Models::Billing, 26
  - Models::Contributory, 68
  - Models::Customer, 78
  - Models::IModel, 108
  - Models::Project, 125
  - Models::User, 161
- getDatabaseName
  - Gui::Widgets::DatabaseSettingsWidget, 99
- getDate
  - Models::Billing, 27
- getDescription
  - Models::Billing, 27
  - Models::Contributory, 68
  - Models::Project, 125
- getDomainNameOrIP
  - Gui::Widgets::DatabaseSettingsWidget, 99
- getEmail
  - Models::Customer, 79
  - Models::User, 161
- getEndDate
  - Models::Project, 126
- getFax
  - Models::Customer, 79
- getFilename
  - Models::Billing, 27
- getFilter
  - Models::Search, 148
- getFirstname
  - Models::User, 161
- getFirstnameReferent
  - Models::Customer, 79
- getFolder
  - Models::Billing, 27
- getGroupFilter
  - Models::Search, 149
- getHourlyRate
  - Gui::Widgets::RateWidget, 144
- getId
  - Models::IModel, 108
- getIdCustomer
  - Gui::Dialogs::AddQuoteDialog, 23
- getImage
  - Gui::Dialogs::StartedWindowsDialog, 154
- getItemBillQuote
  - Databases::CustomerDatabase, 86
- getItemCustomer
  - Databases::CustomerDatabase, 87
- getItemProject
  - Databases::CustomerDatabase, 87
- getItemRoot
  - Databases::CustomerDatabase, 87
- getLastname
  - Models::User, 161
- getLastnameReferent
  - Models::Customer, 79
- getLogin
  - Gui::Widgets::DatabaseSettingsWidget, 99
- getMaxBillingNumber
  - Databases::BillingDatabase, 33
- getMaxBillingNumberOfCustomer
  - Databases::BillingDatabase, 33
- getMaxQuoteNumber
  - Databases::BillingDatabase, 33
- getMaxQuoteNumberOfCustomer
  - Databases::BillingDatabase, 33
- getMobilePhone
  - Models::Customer, 79
  - Models::User, 162
- getModel
  - Utils::ItemType, 110
- getName
  - Models::Project, 126
  - Utils::ItemType, 110
- getNameFolder
  - Models::Customer, 79
- getNbCustomers
  - Databases::CustomerDatabase, 87
- getNbHours
  - Models::Contributory, 68
- getNbProjects
  - Databases::ProjectDatabase, 132
- getNbProjectsForACustomer
  - Databases::ProjectDatabase, 132
- getNoSiret
  - Models::User, 162
- getNumber
  - Gui::Dialogs::AddQuoteDialog, 24
  - Models::Billing, 27
- getPartial
  - Mustache::PartialFileLoader, 119
  - Mustache::PartialMap, 120
  - Mustache::PartialResolver, 121
- getPassword
  - Gui::Widgets::DatabaseSettingsWidget, 99
- getPath
  - Models::Billing, 27
  - Models::Customer, 80
- getPhone
  - Models::Customer, 80
  - Models::User, 162
- getPort
  - Gui::Widgets::DatabaseSettingsWidget, 99
- getPostalCode
  - Models::Customer, 80
  - Models::User, 162

- getProject
  - Databases::ProjectDatabase, 133
  - Models::Contributory, 68
- getProjects
  - Gui::Widgets::WdgModels::ProjectsTableModel, 137
  - Utils::HierarchicalSystem, 106
- getProjectsOfCustomer
  - Databases::ProjectDatabase, 133
- getProjectsTable
  - Databases::ProjectDatabase, 133
- getSearchInBillsQuotes
  - Models::Search, 149
- getSearchInCompanies
  - Models::Search, 149
- getSearchInProjects
  - Models::Search, 149
- getSearchInReferentLastname
  - Models::Search, 149
- getSumQuantity
  - Models::Billing, 28
- getSumRate
  - Models::Billing, 28
- getText
  - Models::Search, 149
- getTitle
  - Models::Billing, 28
  - Models::User, 162
- getTree
  - Databases::CustomerDatabase, 87
- getType
  - Utils::ItemType, 110
- getUser
  - Databases::UserDatabase, 167
- getWorkspaceName
  - Models::User, 162
- getWorkspacePath
  - Models::User, 162
- Gui, 17
- Gui::Dialogs::AddProjectDialog, 21
  - AddProjectDialog, 22
- Gui::Dialogs::AddQuoteDialog, 22
  - AddQuoteDialog, 23
  - fillQuoteBilling, 23
  - getCopy, 23
  - getCustomerId, 23
  - getNumber, 24
  - setCopy, 24
  - setQuoteIdNumber, 24
- Gui::Dialogs::DialogAddCustomer, 102
  - DialogAddCustomer, 102
- Gui::Dialogs::MessageBox, 117
  - MessageBox, 118
  - setImage, 118
  - setText, 118
- Gui::Dialogs::StartedWindowsDialog, 153
  - databaseTypeChanged, 154
  - getImage, 154
  - isDatabaseCentralized, 155
  - isDatabaseTypeValid, 155
- Gui::Dialogs::UserDataDialog, 168
  - UserDataDialog, 169
- Gui::Docks::SearchDock, 151
- Gui::MainWindow, 113
  - addBill, 115
  - addDoc, 115
  - addProject, 115
  - addQuote, 115
  - editUser, 115
  - getCurrentCustomerId, 115
  - getCurrentCustomerName, 116
  - getCurrentProjectId, 116
  - getCurrentProjectName, 116
  - getCurrentQuoteId, 116
  - MainWindow, 114
  - rootTree, 116
  - search, 116
  - treeLevel, 117
- Gui::Widgets, 18
- Gui::Widgets::CheckFields::CheckCity, 39
  - CheckCity, 40
- Gui::Widgets::CheckFields::CheckCountry, 40
  - CheckCountry, 41
- Gui::Widgets::CheckFields::CheckEmail, 41
  - check, 42
  - CheckEmail, 42
- Gui::Widgets::CheckFields::CheckFieldsLetters, 43
  - check, 43
  - CheckFieldsLetters, 43
- Gui::Widgets::CheckFields::CheckFieldsNumbers, 44
  - check, 45
  - CheckFieldsNumbers, 44
- Gui::Widgets::CheckFields::CheckIpAddress, 45
  - check, 46
  - CheckIpAddress, 46
- Gui::Widgets::CheckFields::CheckLogin, 46
  - check, 47
  - CheckLogin, 47
- Gui::Widgets::CheckFields::CheckName, 48
  - CheckName, 48
- Gui::Widgets::CheckFields::CheckPhone, 48
  - check, 49
  - CheckPhone, 49
  - getCountry, 49
  - setCountry, 50
- Gui::Widgets::CheckFields::CheckPortNumber, 50
  - check, 51
  - CheckPortNumber, 51
- Gui::Widgets::CheckFields::CheckPostalCode, 51
  - check, 52
  - CheckPostalCode, 52
  - getCountry, 52
  - setCountry, 52
- Gui::Widgets::CheckFields::CheckQLineEdit, 53
  - CheckQLineEdit, 53
  - getBtnValid, 54

- isValid, 54
  - setBtnValid, 54
- Gui::Widgets::CheckFields::CheckSiretNumber, 54
  - check, 55
  - CheckSiretNumber, 55
- Gui::Widgets::CheckFields::CheckUntilField, 55
  - check, 56
  - CheckUntilField, 56
- Gui::Widgets::CheckFields::CheckValidField, 56
  - check, 57
  - CheckValidField, 57
- Gui::Widgets::CheckFields::ICheckField, 107
  - check, 107
- Gui::Widgets::ComboBoxModelWidget, 58
- Gui::Widgets::ContributoriesWidget, 65
  - add, 66
  - getContributories, 66
- Gui::Widgets::CustomerContextualMenu, 83
  - CustomerContextualMenu, 84
- Gui::Widgets::CustomerDataWidget, 89
  - printInformations, 89
- Gui::Widgets::DatabaseSettingsWidget, 96
  - DatabaseSettingsWidget, 98
  - getDatabaseName, 99
  - getDomainNameOrIP, 99
  - getLogin, 99
  - getPassword, 99
  - getPort, 99
  - isValid, 99
- Gui::Widgets::Delegates::ComboBoxDelegate, 58
- Gui::Widgets::Delegates::DoubleSpinBoxDelegate, 103
- Gui::Widgets::Delegates::ProjectComboDelegate, 128
- Gui::Widgets::Delegates::UnitComboDelegate, 158
- Gui::Widgets::Popup, 121
- Gui::Widgets::ProjectsWidget, 138
  - ProjectsWidget, 139
  - updateBtn, 139
- Gui::Widgets::RateWidget, 143
  - getDailyRate, 144
  - getHourlyRate, 144
  - RateWidget, 144
  - setDailyRate, 144
  - setHourlyRate, 144
  - setWidgetDailyRateValue, 144
  - setWidgetHourlyRateValue, 145
- Gui::Widgets::WdgModels::BillingsTableModel, 35
  - append, 36
  - BillingsTableModel, 36
  - columnCount, 37
  - count, 37
  - data, 37
  - flags, 37
  - getBillings, 37
  - headerData, 37
  - remove, 39
  - rowCount, 39
  - setData, 39
- Gui::Widgets::WdgModels::ContributoriesTableModel, 62
  - append, 63
  - columnCount, 63
  - ContributoriesTableModel, 63
  - count, 63
  - data, 63
  - flags, 64
  - getContributories, 64
  - headerData, 64
  - remove, 64
  - rowCount, 64
  - setData, 65
- Gui::Widgets::WdgModels::CustomersTableModel, 90
  - append, 91
  - columnCount, 91
  - count, 91
  - CustomersTableModel, 91
  - data, 91
  - flags, 92
  - getCustomers, 92
  - headerData, 92
  - remove, 92
  - rowCount, 92
  - setData, 93
- Gui::Widgets::WdgModels::ProjectContributoriesTableModel, 129
  - columnCount, 129
  - data, 130
  - flags, 130
  - headerData, 130
  - ProjectContributoriesTableModel, 129
  - rowCount, 130
  - setData, 130
- Gui::Widgets::WdgModels::ProjectsTableModel, 135
  - append, 136
  - columnCount, 136
  - count, 136
  - data, 136
  - flags, 137
  - getProjects, 137
  - headerData, 137
  - ProjectsTableModel, 136
  - remove, 137
  - rowCount, 137
  - setData, 138
- Gui::Widgets::searchWidget, 152
  - getCurrentCustomerId, 153
  - isCustomerSelected, 153
  - search, 153
  - searchWidget, 153
- headerData
  - Gui::Widgets::WdgModels::BillingsTableModel, 37
  - Gui::Widgets::WdgModels::ContributoriesTableModel, 64
  - Gui::Widgets::WdgModels::CustomersTableModel, 92

- Gui::Widgets::WdgModels::ProjectContributoriesTableModel, 130
  - Gui::Widgets::WdgModels::ProjectsTableModel, 137
- hydrat
  - Models::Billing, 28
  - Models::Contributory, 68
  - Models::Customer, 80
  - Models::IModel, 108
  - Models::Project, 126
  - Models::User, 163
- instance
  - Databases::BillingDatabase, 34
  - Databases::ContributoryDatabase, 72
  - Databases::CustomerDatabase, 88
  - Databases::Database, 96
  - Databases::ProjectDatabase, 134
  - Databases::RateDatabase, 142
  - Databases::UserDatabase, 167
  - Utils::Log, 112
- InvertedSectionStart
  - Mustache::Tag, 157
- isBilling
  - Models::Billing, 28
- isBillingPaid
  - Databases::BillingDatabase, 34
- isCustomerSelected
  - Gui::Widgets::searchWidget, 153
- isDatabaseCentralized
  - Gui::Dialogs::StartedWindowsDialog, 155
- isDatabaseTypeValid
  - Gui::Dialogs::StartedWindowsDialog, 155
- isFalse
  - Mustache::Context, 60
  - Mustache::QtVariantContext, 141
- isPaid
  - Models::Billing, 28
- isToRemoved
  - Models::IModel, 109
- isValid
  - Gui::Widgets::CheckFields::CheckQLineEdit, 54
  - Gui::Widgets::DatabaseSettingsWidget, 99
- ItemType
  - Utils::ItemType, 110
- ItemTypeTest, 111
- lastError
  - Databases::Database, 96
- listCount
  - Mustache::Context, 60
  - Mustache::QtVariantContext, 141
- MainWindow
  - Gui::MainWindow, 114
- makeDirectory
  - Utils::Directories, 103
- MessageBox
  - Gui::Dialogs::MessageBox, 118
- Models, 18
- Models::Billing, 24
  - addContributory, 26
  - Billing, 26
  - getContributories, 26
  - getDataMap, 26
  - getDate, 27
  - getDescription, 27
  - getFilename, 27
  - getFolder, 27
  - getNumber, 27
  - getPath, 27
  - getSumQuantity, 28
  - getSumRate, 28
  - getTitle, 28
  - hydrat, 28
  - isBilling, 28
  - isPaid, 28
  - operator<, 29
  - operator==, 29
  - setDate, 29
  - setDescription, 29
  - setIsBilling, 29
  - setIsPaid, 30
  - setNumber, 30
  - setTitle, 30
- Models::ContributoriesList, 61
  - setAllIdContributories, 61
- Models::Contributory, 66
  - Contributory, 67
  - getDataMap, 68
  - getDescription, 68
  - getNbHours, 68
  - getProject, 68
  - hydrat, 68
  - operator==, 69
  - setDescription, 69
  - setNbHours, 69
  - setProject, 69
- Models::Customer, 76
  - Customer, 78
  - getAddress, 78
  - getCity, 78
  - getCompany, 78
  - getCountry, 78
  - getDataMap, 78
  - getEmail, 79
  - getFax, 79
  - getFirstnameReferent, 79
  - getLastnameReferent, 79
  - getMobilePhone, 79
  - getNameFolder, 79
  - getPath, 80
  - getPhone, 80
  - getPostalCode, 80
  - hydrat, 80
  - operator==, 80
  - setAddress, 82

- setCity, [82](#)
- setCompany, [82](#)
- setCountry, [82](#)
- setEmail, [82](#)
- setFax, [82](#)
- setFirstnameReferent, [83](#)
- setLastnameReferent, [83](#)
- setMobilePhone, [83](#)
- setPhone, [83](#)
- setPostalCode, [83](#)
- Models::IModel, [107](#)
  - getDataMap, [108](#)
  - getId, [108](#)
  - hydrat, [108](#)
  - isToRemoved, [109](#)
  - setId, [109](#)
  - setToRemoved, [109](#)
- Models::Project, [122](#)
  - costCompute, [125](#)
  - getBeginDate, [125](#)
  - getCost, [125](#)
  - getCustomer, [125](#)
  - getDailyRate, [125](#)
  - getDataMap, [125](#)
  - getDescription, [125](#)
  - getEndDate, [126](#)
  - getName, [126](#)
  - hydrat, [126](#)
  - operator<, [126](#)
  - operator==, [127](#)
  - Project, [124](#)
  - setBeginDate, [127](#)
  - setCost, [127](#)
  - setCustomer, [127](#)
  - setDailyRate, [127](#)
  - setDescription, [127](#)
  - setEndDate, [128](#)
  - setName, [128](#)
- Models::Rate, [141](#)
- Models::Search, [146](#)
  - filterOnBillsOrQuotes, [147](#)
  - filterOnCompany, [147](#)
  - filterOnContributories, [147](#)
  - filterOnCustomersWithoutProject, [148](#)
  - filterOnNumberElements, [148](#)
  - filterOnProjects, [148](#)
  - filterOnReferentLastname, [148](#)
  - filterOnVarcharElements, [148](#)
  - getFilter, [148](#)
  - getGroupFilter, [149](#)
  - getSearchInBillsQuotes, [149](#)
  - getSearchInCompanies, [149](#)
  - getSearchInProjects, [149](#)
  - getSearchInReferentLastname, [149](#)
  - getText, [149](#)
  - searchInContributories, [149](#)
  - setGroupFilter, [150](#)
  - setSearchInBillsQuotes, [150](#)
  - setSearchInCompanies, [150](#)
  - setSearchInContributories, [150](#)
  - setSearchInProjects, [150](#)
  - setSearchInReferentLastname, [150](#)
  - setText, [151](#)
- Models::User, [159](#)
  - getAddress, [161](#)
  - getCity, [161](#)
  - getCompany, [161](#)
  - getDataMap, [161](#)
  - getEmail, [161](#)
  - getFirstname, [161](#)
  - getLastname, [161](#)
  - getMobilePhone, [162](#)
  - getNoSiret, [162](#)
  - getPhone, [162](#)
  - getPostalCode, [162](#)
  - getTitle, [162](#)
  - getWorkspaceName, [162](#)
  - getWorkspacePath, [162](#)
  - hydrat, [163](#)
  - operator==, [163](#)
  - setAddress, [163](#)
  - setCity, [163](#)
  - setCompany, [164](#)
  - setEmail, [164](#)
  - setFirstname, [164](#)
  - setLastname, [164](#)
  - setMobilePhone, [164](#)
  - setNoSiret, [164](#)
  - setPhone, [164](#)
  - setPostalCode, [166](#)
  - setTitle, [166](#)
  - setWorkspaceName, [166](#)
  - setWorkspacePath, [166](#)
  - updateFolders, [166](#)
  - User, [160](#)
- Mustache::Tag
  - Comment, [157](#)
  - InvertedSectionStart, [157](#)
  - Partial, [157](#)
  - SectionEnd, [157](#)
  - SectionStart, [157](#)
  - SetDelimiter, [157](#)
- Mustache::Context, [59](#)
  - canEval, [59](#)
  - Context, [59](#)
  - eval, [59](#)
  - isFalse, [60](#)
  - listCount, [60](#)
  - partialResolver, [60](#)
  - partialValue, [60](#)
  - pop, [60](#)
  - push, [60](#)
  - stringValue, [60](#)
- Mustache::PartialFileLoader, [119](#)
  - getPartial, [119](#)
- Mustache::PartialMap, [120](#)

- getPartial, 120
- Mustache::PartialResolver, 120
  - getPartial, 121
- Mustache::QtVariantContext, 139
  - canEval, 140
  - eval, 140
  - fn\_t, 140
  - isFalse, 141
  - listCount, 141
  - pop, 141
  - push, 141
  - stringValue, 141
- Mustache::Renderer, 145
  - error, 145
  - errorPartial, 145
  - errorPos, 145
  - render, 145
  - setTagMarkers, 146
- Mustache::Tag, 156
  - Type, 157
- operator<
  - Models::Billing, 29
  - Models::Project, 126
- operator<<
  - Utils::Log, 112
- operator==
  - Models::Billing, 29
  - Models::Contributory, 69
  - Models::Customer, 80
  - Models::Project, 127
  - Models::User, 163
- Parameters, 118
  - DB\_FILENAME, 119
  - VERSION, 119
- Partial
  - Mustache::Tag, 157
- partialResolver
  - Mustache::Context, 60
- partialValue
  - Mustache::Context, 60
- pop
  - Mustache::Context, 60
  - Mustache::QtVariantContext, 141
- popupMessage
  - Exceptions::DbException, 101
  - Exceptions::FileException, 104
- printInformations
  - Gui::Widgets::CustomerDataWidget, 89
- Project
  - Models::Project, 124
- ProjectContributoriesTableModel
  - Gui::Widgets::WdgModels::ProjectContributories-  
TableModel, 129
- ProjectDatabaseTest, 134
- ProjectModelTest, 135
- ProjectsTableModel
  - Gui::Widgets::WdgModels::ProjectsTableModel, 136
- ProjectsWidget
  - Gui::Widgets::ProjectsWidget, 139
- push
  - Mustache::Context, 60
  - Mustache::QtVariantContext, 141
- RateModelTest, 143
- RateWidget
  - Gui::Widgets::RateWidget, 144
- remove
  - Gui::Widgets::WdgModels::BillingsTableModel, 39
  - Gui::Widgets::WdgModels::ContributoriesTable-  
Model, 64
  - Gui::Widgets::WdgModels::CustomersTableModel, 92
  - Gui::Widgets::WdgModels::ProjectsTableModel, 137
- removeBilling
  - Databases::BillingDatabase, 34
- removeBillingProject
  - Databases::BillingDatabase, 34
- removeContributory
  - Databases::ContributoryDatabase, 72
- removeCustomer
  - Databases::CustomerDatabase, 88
- removeProject
  - Databases::ProjectDatabase, 134
- render
  - Mustache::Renderer, 145
- rootTree
  - Gui::MainWindow, 116
- rowCount
  - Gui::Widgets::WdgModels::BillingsTableModel, 39
  - Gui::Widgets::WdgModels::ContributoriesTable-  
Model, 64
  - Gui::Widgets::WdgModels::CustomersTableModel, 92
  - Gui::Widgets::WdgModels::ProjectContributories-  
TableModel, 130
  - Gui::Widgets::WdgModels::ProjectsTableModel, 137
- search
  - Gui::MainWindow, 116
  - Gui::Widgets::searchWidget, 153
- searchInContributories
  - Models::Search, 149
- searchTest, 151
- searchWidget
  - Gui::Widgets::searchWidget, 153
- SectionEnd
  - Mustache::Tag, 157
- SectionStart
  - Mustache::Tag, 157
- SetDelimiter
  - Mustache::Tag, 157
- setAddress



- Models::Customer, 82
- Models::User, 163
- setAllIdContributories
  - Models::ContributoriesList, 61
- setBeginDate
  - Models::Project, 127
- setBtnValid
  - Gui::Widgets::CheckFields::CheckQLineEdit, 54
- setCity
  - Models::Customer, 82
  - Models::User, 163
- setCompany
  - Models::Customer, 82
  - Models::User, 164
- setCopy
  - Gui::Dialogs::AddQuoteDialog, 24
- setCost
  - Models::Project, 127
- setCountry
  - Gui::Widgets::CheckFields::CheckPhone, 50
  - Gui::Widgets::CheckFields::CheckPostalCode, 52
  - Models::Customer, 82
- setCustomer
  - Models::Project, 127
- setDailyRate
  - Gui::Widgets::RateWidget, 144
  - Models::Project, 127
- setData
  - Gui::Widgets::WdgModels::BillingsTableModel, 39
  - Gui::Widgets::WdgModels::ContributoriesTable-Model, 65
  - Gui::Widgets::WdgModels::CustomersTableModel, 93
  - Gui::Widgets::WdgModels::ProjectContributories-TableModel, 130
  - Gui::Widgets::WdgModels::ProjectsTableModel, 138
- setDatabase
  - Databases::Database, 96
- setDate
  - Models::Billing, 29
- setDescription
  - Models::Billing, 29
  - Models::Contributory, 69
  - Models::Project, 127
- setEmail
  - Models::Customer, 82
  - Models::User, 164
- setEndDate
  - Models::Project, 128
- setFax
  - Models::Customer, 82
- setFirstname
  - Models::User, 164
- setFirstnameReferent
  - Models::Customer, 83
- setGroupFilter
  - Models::Search, 150
- setHourlyRate
  - Gui::Widgets::RateWidget, 144
- setId
  - Models::IModel, 109
- setImage
  - Gui::Dialogs::MessageBox, 118
- setIsBilling
  - Models::Billing, 29
- setIsPaid
  - Models::Billing, 30
- setLastname
  - Models::User, 164
- setLastnameReferent
  - Models::Customer, 83
- setMobilePhone
  - Models::Customer, 83
  - Models::User, 164
- setName
  - Models::Project, 128
  - Utils::ItemType, 111
- setNbHours
  - Models::Contributory, 69
- setNoSiret
  - Models::User, 164
- setNumber
  - Models::Billing, 30
- setPhone
  - Models::Customer, 83
  - Models::User, 164
- setPostalCode
  - Models::Customer, 83
  - Models::User, 166
- setProject
  - Models::Contributory, 69
- setQuoteIdNumber
  - Gui::Dialogs::AddQuoteDialog, 24
- setSearchInBillsQuotes
  - Models::Search, 150
- setSearchInCompanies
  - Models::Search, 150
- setSearchInContributories
  - Models::Search, 150
- setSearchInProjects
  - Models::Search, 150
- setSearchInReferentLastname
  - Models::Search, 150
- setTagMarkers
  - Mustache::Renderer, 146
- setText
  - Gui::Dialogs::MessageBox, 118
  - Models::Search, 151
- setTitle
  - Models::Billing, 30
  - Models::User, 166
- setToRemoved
  - Models::IModel, 109
- setType
  - Utils::ItemType, 111



- setWidgetDailyRateValue
  - Gui::Widgets::RateWidget, [144](#)
- setWidgetHourlyRateValue
  - Gui::Widgets::RateWidget, [145](#)
- setWorkspaceName
  - Models::User, [166](#)
- setWorkspacePath
  - Models::User, [166](#)
- StringTest, [156](#)
- stringValue
  - CounterContext, [75](#), [76](#)
  - Mustache::Context, [60](#)
  - Mustache::QtVariantContext, [141](#)
- TestAdder< T >, [157](#)
- TestMustache, [157](#)
- TestRunner, [158](#)
- testadder, [157](#)
- treeLevel
  - Gui::MainWindow, [117](#)
- Type
  - Mustache::Tag, [157](#)
- updateBtn
  - Gui::Widgets::ProjectsWidget, [139](#)
- updateCustomer
  - Databases::CustomerDatabase, [88](#)
- updateFolders
  - Models::User, [166](#)
- updateProject
  - Databases::ProjectDatabase, [134](#)
- User
  - Models::User, [160](#)
- UserDataDialog
  - Gui::Dialogs::UserDataDialog, [169](#)
- UserDatabaseTest, [168](#)
- UserModelTest, [169](#)
- Utils::Directories, [103](#)
  - makeDirectory, [103](#)
- Utils::HierarchicalSystem, [105](#)
  - addBillingToProject, [106](#)
  - addProjectToCustomer, [106](#)
  - getCustomers, [106](#)
  - getProjects, [106](#)
- Utils::ItemType, [109](#)
  - getModel, [110](#)
  - getName, [110](#)
  - getType, [110](#)
  - ItemType, [110](#)
  - setName, [111](#)
  - setType, [111](#)
- Utils::Log, [111](#)
  - instance, [112](#)
  - operator<<, [112](#)
  - write, [112](#)
- Utils::String, [155](#)
  - firstLetterToUpper, [155](#)
- Utils::pointers, [121](#)
- VERSION
  - Parameters, [119](#)
- value
  - Databases::Database, [96](#)
- write
  - Utils::Log, [112](#)