

Documentation C++ (Qt)

Agenda étudiant

Projet de semestre

Jérôme Moret, Paul Ntawuruhunga,
Thibaud Duchoud, David Kunzmann, Mario Ferreira
Version 1.0

02 Juin 2015

Index hiérarchique

Hiérarchie des classes

Cette liste d'héritage est classée approximativement par ordre alphabétique :

Average	9
QAbstractTableModel	
CustomModel	25
QCalendarWidget	
CustomCalendarWidget	23
QDate	
WeekDate	103
QDialog	
gradedialogbox	28
QLabel	
WeekDate	103
QMainWindow	
MainWindow	39
QObject	
Course	20
Lesson	32
LessonRect	36
Task	58
QStandardItem	
TreeNode	83
QStandardItemModel	
TreeModel	81
QToolBar	
ToolBarCalendar	75
ToolBarSummary	77
ToolBarTask	78
QTreeView	
TreeView	84
QWidget	
CalendarColumn	10

ColumnDay	16
VerticalHeaderCalendar	92
CalendarWidget	13
DayWidget.....	26
HourCal	31
MiniTaskList.....	42
Priority	44
SideBarCalendar	46
SideBarSummary	49
SideBarTask.....	51
SideBarTreeView	54
TaskCheckBox.....	63
taskDisplay	67
TaskListItem	71
TermDate	73
TriStateButton.....	86
VCourseDetails.....	88
VCourseFill.....	90
VLessonFill.....	93
VModule	95
VModuleFill	96
VSemesterFill	98
VSummary.....	100
SummaryUtils	56
tooltipNote	80

Index des classes

Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

Average (Structure représentant une moyenne (nom et valeur))	9
CalendarColumn (Colonne de la grille du calendrier)	10
CalendarWidget (La class CalendarWidget gère l'affichage des leçons et des tâches dans un calendrier. Ces informations sont affichées par semaine. Il est possible de naviguer entre les semaines en choisissant un jour depuis le mini calendrier de la sidebar)	13
ColumnDay (Colonne d'un jour dans le calendrier)	16
Course (Modèle pour les cours)	20
CustomCalendarWidget (MiniCalendar modifié pour une amélioration du design La navigation entre les dates est géré manuellement pour plus de fonctionnalités)	23
CustomModel	25
DayWidget (Classe utilitaire permettant d'afficher le nom d'un jour ainsi sa date. Le numéro du jour est affiché en grand en dessous du nom du jour)	26
gradedialogbox	28
HourCal (HourCal rempli chaque ligne de d'une colonne. Les lignes représente des heure)	31
Lesson (Modèle pour les leçons)	32
LessonRect (LessonRect représente une leçon du calendrier)	36
MainWindow	39
MiniTaskList (MiniTaskList est une vue personnalisée d'une model Task Elle affiche une liste allégé de tâches)	42
Priority	44
SideBarCalendar (Sidebar de la vue Calendrier)	46
SideBarSummary (Classe représentant la side bar (barre à droite) dans la vue résumé)	49
SideBarTask	51
SideBarTreeView (Classe créant et contenant le TreeView)	54
SummaryUtils (Classe utilitaire)	56
Task (Modèle pour les tâches)	58
TaskCheckBox	63
taskDisplay	67
TaskListItem	71
TermDate	73
ToolBarCalendar (ToolBarCalendar est la barre d'outil de la vue Calendrier. Elle affiche le slider permettant de zoomer sur la vue calendrier)	75
ToolBarSummary (Classe représentant la tool bar (barre affiché au sommet de la vue))	77
ToolBarTask	78
tooltipNote	80
TreeModel (Classe héritée de QStandardItemModel et qui correspond au model pour le TreeView)	81
TreeNode	83
TreeView (Classe héritée de QTreeView représentant un TreeView)	84
TriStateButton	86
VCourseDetails (Classe représentant l'affichage des détails des cours)	88

VCourseFill (Classe representant la zone de modification des données d'un cours)	90
VerticalHeaderCalendar (HeaderVertical du Calendrier)	92
VLessonFill (Classe representant la zone de modification des données d'une leçon)	93
VModule (Classe représentant l'affichage des différents modules)	95
VModuleFill (Classe representant la zone de modification des données d'un module)	96
VSemesterFill (Classe representant la zone de modification des données d'un semestre)	98
VSummary (Classe représentant l'affichage lors du clique sur l'ongle résumé (vue résumé)) .	100
WeekDate (WeekDate un mélange de QLabel et de QDate)	103

Index des fichiers

Liste des fichiers

Liste de tous les fichiers avec une brève description :

Agenda/mainwindow.cpp	105
Agenda/mainwindow.h	106
Agenda/tasklistitem.cpp	113
Agenda/tasklistitem.h	114
Agenda/models/course.cpp	107
Agenda/models/course.h	108
Agenda/models/lesson.cpp	109
Agenda/models/lesson.h	110
Agenda/models/task.cpp	111
Agenda/models/task.h	112
Agenda/viewCalendar/calendarcolumn.cpp	115
Agenda/viewCalendar/calendarcolumn.h	116
Agenda/viewCalendar/calendarwidget.cpp	117
Agenda/viewCalendar/calendarwidget.h	118
Agenda/viewCalendar/columnnday.cpp	119
Agenda/viewCalendar/columnnday.h	120
Agenda/viewCalendar/customcalendarwidget.cpp	121
Agenda/viewCalendar/customcalendarwidget.h	122
Agenda/viewCalendar/daywidget.cpp	123
Agenda/viewCalendar/daywidget.h	124
Agenda/viewCalendar/hourcal.h	125
Agenda/viewCalendar/lessonrect.cpp	126
Agenda/viewCalendar/lessonrect.h	127
Agenda/viewCalendar/minitasklist.cpp	128
Agenda/viewCalendar/minitasklist.h	129
Agenda/viewCalendar/sidebarcalendar.cpp	130
Agenda/viewCalendar/sidebarcalendar.h	131
Agenda/viewCalendar/toolbarcalendar.cpp	132
Agenda/viewCalendar/toolbarcalendar.h	133
Agenda/viewCalendar/verticalheadercalendar.cpp	134
Agenda/viewCalendar/verticalheadercalendar.h	135
Agenda/viewCalendar/weekdate.cpp	136
Agenda/viewCalendar/weekdate.h	137
Agenda/viewSummary/average.h	138
Agenda/viewSummary/custommodel.h	139
Agenda/viewSummary/sidebarsummary.cpp	140
Agenda/viewSummary/sidebarsummary.h	141
Agenda/viewSummary/summaryutils.h	142
Agenda/viewSummary/testtreeview.cpp	143
Agenda/viewSummary/testtreeview.h	144

Agenda/viewSummary/toolbarsummary.cpp	145
Agenda/viewSummary/toolbarsummary.h	146
Agenda/viewSummary/treemodel.cpp	147
Agenda/viewSummary/treemodel.h	148
Agenda/viewSummary/treenode.h	149
Agenda/viewSummary/treeview.h	150
Agenda/viewSummary/vcoursedetails.cpp	151
Agenda/viewSummary/vcoursedetails.h	152
Agenda/viewSummary/vcoursefill.cpp	153
Agenda/viewSummary/vcoursefill.h	154
Agenda/viewSummary/vlessonfill.cpp	155
Agenda/viewSummary/vlessonfill.h	156
Agenda/viewSummary/vmodule.cpp	157
Agenda/viewSummary/vmodule.h	158
Agenda/viewSummary/vmodulefill.cpp	159
Agenda/viewSummary/vmodulefill.h	160
Agenda/viewSummary/vsemesterfill.cpp	161
Agenda/viewSummary/vsemesterfill.h	162
Agenda/viewSummary/vsummary.cpp	163
Agenda/viewSummary/vsummary.h	164
Agenda/viewTask/gradedialogbox.cpp	165
Agenda/viewTask/gradedialogbox.h	166
Agenda/viewTask/priority.h	167
Agenda/viewTask/sidebartask.cpp	168
Agenda/viewTask/sidebartask.h	169
Agenda/viewTask/taskcheckbox.cpp	170
Agenda/viewTask/taskcheckbox.h	171
Agenda/viewTask/taskdisplay.cpp	172
Agenda/viewTask/taskdisplay.h	173
Agenda/viewTask/tasktableview.h	174
Agenda/viewTask/termdate.cpp	175
Agenda/viewTask/termdate.h	176
Agenda/viewTask/toolbartask.cpp	177
Agenda/viewTask/toolbartask.h	178
Agenda/viewTask/tooltipnote.cpp	179
Agenda/viewTask/tooltipnote.h	180
Agenda/viewTask/tristatebutton.cpp	181
Agenda/viewTask/tristatebutton.h	182

Documentation des classes

Référence de la structure Average

Structure représentant une moyenne (nom et valeur)

```
#include <average.h>
```

Fonctions membres publiques

- **Average** (QString **name**, float **value**)

Attributs publics

- QString **name**
 - float **value**
-

Description détaillée

Structure représentant une moyenne (nom et valeur)

Utilisé dans le calcul des moyennes pour les cours et modules

Documentation des constructeurs et destructeur

```
Average::Average (QString name, float value) [inline]
```

Documentation des données membres

QString Average::name

float Average::value

La documentation de cette structure a été générée à partir du fichier suivant :

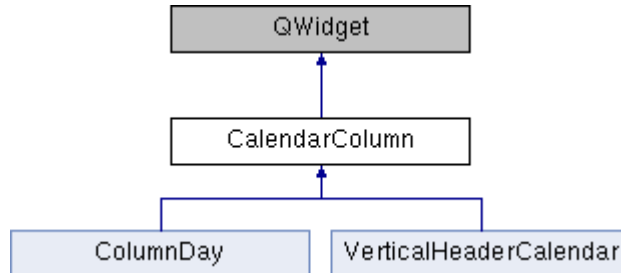
- Agenda/viewSummary/**average.h**

Référence de la classe CalendarColumn

Colonne de la grille du calendrier.

```
#include <calendarcolumn.h>
```

Graphe d'héritage de CalendarColumn:



Connecteurs publics

- void **setRowsMinimumHeight** (int height)
taille minimum des lignes

Fonctions membres publiques

- **CalendarColumn** (QWidget *parent=0)
Constructeur.
- void **addRow** (QWidget *widget, Qt::Alignment alignment=0)
Ajout d'une ligne.
- virtual int **getDayOfWeek** ()
numéro du jour de la semaine de la colonne
- **~CalendarColumn** ()

Attributs publics statiques

- static int **heightHour** = 50
- static int **nbDay** = 8
- static int **nbHours** = 24

Description détaillée

Colonne de la grille du calendrier.

Une colonne de la grille du calendrier peut contenir autant de widgets que de nombre d'heures dans une journée. Ces widgets sont ajoutés dans les lignes de chaque colonne.

Cette classe est utile si on l'hérite afin de créer un verticalHeader, ou pour créer la colonne d'un jour.

Documentation des constructeurs et destructeur

CalendarColumn::CalendarColumn (QWidget * *parent* = 0) [**explicit**]

Constructeur.

Paramètres:

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

CalendarColumn::~~CalendarColumn ()

Documentation des fonctions membres**void CalendarColumn::addRow (QWidget * *widget*, Qt::Alignment *alignment* = 0)**

Ajout d'une ligne.

Paramètres:

<i>widget</i>	le widget de la ligne
<i>alignment</i>	Alignement du widget dans la ligne

int CalendarColumn::getDayOfWeek () [virtual]

numéro du jour de la semaine de la colonne

Renvoie:

-1 si la colonne ne représente pas un jour de la semaine

void CalendarColumn::setRowsMinimumHeight (int *height*) [slot]

taille minimum des lignes

Paramètres:

<i>height</i>	nouvelle taille
---------------	-----------------

Documentation des données membres**int CalendarColumn::heightHour = 50 [static]****int CalendarColumn::nbDay = 8 [static]****int CalendarColumn::nbHours = 24 [static]**

La documentation de cette classe a été générée à partir des fichiers suivants :

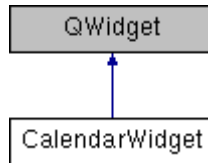
- Agenda/viewCalendar/calendarcolumn.h
- Agenda/viewCalendar/calendarcolumn.cpp

Référence de la classe CalendarWidget

La class **CalendarWidget** gère l'affichage des leçons et des tâches dans un calendrier. Ces informations sont affichées par semaine. Il est possible de naviguer entre les semaines en choisissant un jour depuis le mini calendrier de la sidebar.

```
#include <calendarwidget.h>
```

Graphe d'héritage de CalendarWidget:



Connecteurs publics

- void **changeHeightHours** (int height)
Change la taille du heure en pixel.
- void **updateCurrentWeek** (QDate currentDate)
Met à jour les dates affichée dans le calendrier.
- void **getTasksByDate** (QDate)
filtre le model tâche et récupère la liste de tâche dont l'échance correspond à une certaine date
- void **deselectOthers** (int dayOfWeek)
deselection tout les rectangles des lessons, sauf 1
- void **deselectAll** ()
déselectionne tous les rectangles de lesson.
- void **getTasksByLesson** (QDate date, QString courseName)
getTasksByLesson récupère les tâche d'une leçon précise

Signaux

- void **selectionUpdated** ()
indique que la selection d'un élément du calendrier (leçon/jour) à été sélectionné et que la base de donnée à été filtrée.

Fonctions membres publiques

- **CalendarWidget** (MainWindow *mainWindow)
CalendarWidget.
- QScrollBar * **getVerticalScrollbar** ()
Getter pour la barre de défilement.
- ~**CalendarWidget** ()

Description détaillée

La class **CalendarWidget** gère l'affichage des leçons et des tâches dans un calendrier. Ces informations sont affichées par semaine. Il est possible de naviguer entre les semaines en choisissant un jour depuis le mini calendrier de la sidebar.

Documentation des constructeurs et destructeur

CalendarWidget::CalendarWidget (MainWindow * *mainWindow*) [explicit]

CalendarWidget.

Paramètres:

<i>mainWindow</i>	la fenêtre principale
-------------------	-----------------------

CalendarWidget::~~CalendarWidget ()

Documentation des fonctions membres

void CalendarWidget::changeHeightHours (int *height*) [slot]

Change la taille du heure en pixel.

La taille est connectée à la valeur du slider situé dans la toolbar de la vue calendrier. Cela permet d'avoir un effet de zoom sur le calendrier

Paramètres:

<i>height</i>	nouvelle taille
---------------	-----------------

void CalendarWidget::deselectAll () [slot]

désélectionne tous les rectangles de leçon.

void CalendarWidget::deselectOthers (int *dayOfWeek*) [slot]

deselection tout les rectangles des leçons, sauf 1

Paramètres:

<i>dayOfWeek</i>	: numéro du jour où le rectangle à garder en selection se situe
------------------	---

void CalendarWidget::getTasksByDate (QDate *date*) [slot]

filtre le model tâche et récupère la liste de tâche dont l'échance correspond à une certaine date

void CalendarWidget::getTasksByLesson (QDate *date*, QString *courseName*) [slot]

getTasksByLesson récupère les tâche d'une leçon précise

Paramètres:

<i>date</i>	date de la leçon
-------------	------------------

<i>courseName</i>	nom du cours de la leçon
-------------------	--------------------------

QScrollBar * CalendarWidget::getVerticalScrollbar ()

Getter pour la barre de défilement.

void CalendarWidget::selectionUpdated () [signal]

indique que la selection d'un élément du calendrier (leçon/jour) à été sélectionné et que la base de donnée à été filtrée.

void CalendarWidget::updateCurrentWeek (QDate *currentDate*) [slot]

Met à jour les dates affichée dans le calendrier.

C'est la horizontalheader qui sera modifié afin de filter les leçons et tâches selon la date du jour.

Paramètres:

<i>currentDate</i>	La date qui doit être afficher dans le calendrier les autres date de la même semaine seront aussi affichée.
--------------------	---

La documentation de cette classe a été générée à partir des fichiers suivants :

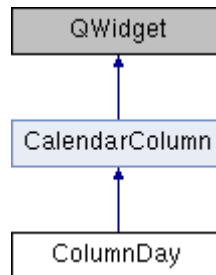
- Agenda/viewCalendar/**calendarwidget.h**
- Agenda/viewCalendar/**calendarwidget.cpp**

Référence de la classe ColumnDay

Colonne d'un jour dans le calendrier.

```
#include <columnDay.h>
```

Graphe d'héritage de ColumnDay:



Connecteurs publics

- **void updateValues ()**
récupère les éventuels modifications effectuées dans la base de donnée puis met à jour la vue calendrier
- **void deselectLesson ()**
désélectionne une leçon
- **void selectLesson (int id)**
sélectionne un leçon
- **int getDayOfWeek ()**
récupère le jour de la semaine courant pour cette colonne

Signaux

- **void lessonSelected (int dayOfWeek)**
indique qu'une leçon est sélectionnée
- **void lessonSelected (QDate date, QString courseName)**
indique qu'une leçon est sélectionnée
- **void columnDaySelected (int)**
indique qu'une la colonne à été sélectionnée. Ce signal est envoyé seulement quand uniquement une colonne est sélectionnée Par exemple quand on clique en dehors des rectangles des leçons.

Fonctions membres publiques

- **ColumnDay (QSqlRelationalTableModel *modelLesson, QDate *date, QWidget *parent=0)**
Constructeur.
- **~ColumnDay ()**
- **void paintEvent (QPaintEvent *event)**
paintEvent est redéfinie afin de pouvoir dessiner les rectangles des leçons
- **void selectLessonByHour (double hour)**
sélectionne une leçon affiché sur le calendrier selon une heure

Fonctions membres protégées

- **void mousePressEvent (QMouseEvent *)**
mousePressEvent redéfinie afin de capturer les événements de clics de souris à l'intérieur d'une colonne

Membres hérités additionnels

Description détaillée

Colonne d'un jour dans le calendrier.

Les colonnes des jours sont basées sur la classe **CalendarColumn**. Chaque ligne contient un widget vide servant uniquement à afficher une séparation au niveau de chaque heure.

Les rectangles des leçons sont dessinés par dessus la grille au moyen d'un painter. Ce painter est mis à jour à chaque repaint de la fenêtre en cherchant les leçons dans la base de données.

Chaque repaint ne fait pas un accès à la base de données ce qui serait trop lourd, mais recherche plutôt dans une liste stockée dans cette classe représentant le dernier état de la base de données. Cette liste est mise à jour seulement lors du mise à jour dans la base de données.

Documentation des constructeurs et destructeur

ColumnDay::ColumnDay (QSqlRelationalTableModel * *modelLesson*, QDate * *date*, QWidget * *parent* = 0) [explicit]

Constructeur.

Paramètres:

<i>modelLesson</i>	model observant la table Leçon
<i>date</i>	date courante de la colonne
<i>parent</i>	widget parent

ColumnDay::~~ColumnDay ()

Documentation des fonctions membres

void ColumnDay::columnDaySelected (int) [signal]

indique qu'une la colonne a été sélectionnée. Ce signal est envoyé seulement quand uniquement une colonne est sélectionnée. Par exemple quand on clique en dehors des rectangles des leçons.

void ColumnDay::deselectLesson () [slot]

désélectionne une leçon

Une seule leçon peut être sélectionnée par jour. Si une leçon est actuellement sélectionnée, elle sera désélectionnée.

int ColumnDay::getDayOfWeek () [slot]

recupère le jour de la semaine courant pour cette colonne

Renvoie:

void ColumnDay::lessonSelected (int *dayOfWeek*) [signal]

indique qu'une leçon est sélectionnée

Paramètres:

<i>dayOfWeek</i>	jour de la semaine. Sert à identifier dans quelle colonne la sélection à été effectuée.
------------------	---

void ColumnDay::lessonSelected (QDate *date*, QString *courseName*) [signal]

indique qu'une leçon est sélectionnée

Paramètres:

<i>date</i>	date de la leçon
<i>courseName</i>	nom du cours

void ColumnDay::mousePressEvent (QMouseEvent * *event*) [protected]

mousePressEvent redéfinie afin de capturer les événements de clics de souris à l'intérieur d'une colonne

void ColumnDay::paintEvent (QPaintEvent * *event*)

paintEvent est redéfinie afin de pouvoir dessiner les rectangles des leçons

Paramètres:

<i>event</i>	type de l'événement
--------------	---------------------

void ColumnDay::selectLesson (int *id*) [slot]

sélectionne un leçon

Paramètres:

<i>id</i>	id de la leçon à selectionner
-----------	-------------------------------

void ColumnDay::selectLessonByHour (double *hour*)

sélectionne une leçon affiché sur le calendrier selon une heure

Paramètres:

<i>hour</i>	heure de la leçon. L'heure cherchée doit se trouver entre l'heure de début et de fin de d'une leçon.
-------------	--

void ColumnDay::updateValues () [slot]

récupère les éventuels modifications effectuées dans la base de donnée puis met à jour la vue calendrier

La documentation de cette classe a été générée à partir des fichiers suivants :

- Agenda/viewCalendar/columnday.h
- Agenda/viewCalendar/columnday.cpp

Référence de la classe Course

Modèle pour les cours.

```
#include <course.h>
```

Graphe d'héritage de Course:



Fonctions membres publiques

- `void update ()`
Mettre à jour les données.
- `QSqlRelationalTableModel * getModel () const`
Récupérer le modèle de liaison.
- `int getModuleIndex () const`
Récupérer l'index du champ de la Foreign Key.
- `int getSemesterIndex () const`
Récupérer l'index du champ de la Foreign Key.
- `int rowCount () const`
Retourne le nombre de cours dans la base de données.
- `~Course ()`
Destructeur.

Fonctions membres publiques statiques

- `static Course * getInstance ()`
Récupérer l'instance.

Description détaillée

Modèle pour les cours.

Modèle qui est utilisé comme lien entre la table "Course" de la base de données et le programme.

Documentation des constructeurs et destructeur

`Course::~~Course ()`

Destructeur.

Documentation des fonctions membres

Course * Course::getInstance () [static]

Récupérer l'instance.

Renvoie:

L'instance unique (singleton)

QSqlRelationalTableModel * Course::getModel () const

Récupérer le modèle de liaison.

Renvoie:

Le modèle sur lequel effectuer les requêtes

int Course::getModuleIndex () const

Récupérer l'index du champ de la Foreign Key.

Renvoie:

L'index de la colonne "moduleId"

int Course::getSemesterIndex () const

Récupérer l'index du champ de la Foreign Key.

Renvoie:

L'index de la colonne "semesterId"

int Course::rowCount () const

Retourne le nombre de cours dans la base de données.

Renvoie:

nombre de cours dans la base de données

void Course::update ()

Mettre à jour les données.

La documentation de cette classe a été générée à partir des fichiers suivants :

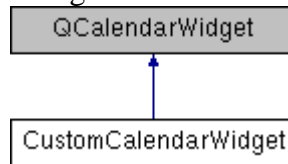
- `Agenda/models/course.h`
- `Agenda/models/course.cpp`

Référence de la classe CustomCalendarWidget

MiniCalendar modifié pour une amélioration du design La navigation entre les dates est géré manuellement pour plus de fonctionnalités.

```
#include <customcalendarwidget.h>
```

Graphe d'héritage de CustomCalendarWidget:



Fonctions membres publiques

- CustomCalendarWidget ()
- ~CustomCalendarWidget ()

Fonctions membres protégées

- void **paintCell** (QPainter *painter, const QRect &rect, const QDate &date) const
Méthode de QCalendarWidget reféfinie afin de gérer le dessin des cellules du calendrier.

Description détaillée

MiniCalendar modifié pour une amélioration du design La navigation entre les dates est géré manuellement pour plus de fonctionnalités.

Documentation des constructeurs et destructeur

CustomCalendarWidget::CustomCalendarWidget ()

CustomCalendarWidget::~~CustomCalendarWidget ()

Documentation des fonctions membres

void CustomCalendarWidget::paintCell (QPainter * *painter*, const QRect & *rect*, const QDate & *date*) const [protected]

Méthode de QCalendarWidget reféfinie afin de gérer le dessin des cellules du calendrier.

Paramètres:

<i>painter</i>	Painter permettant de dessiner des formes
<i>rect</i>	zone dans laquelle il est possible de dessiner (largeur et hauteur d'une cellule)
<i>date</i>	date de la cellule

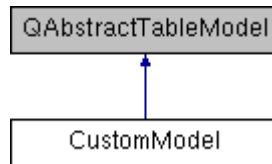
La documentation de cette classe a été générée à partir des fichiers suivants :

- Agenda/viewCalendar/customcalendarwidget.h
- Agenda/viewCalendar/customcalendarwidget.cpp

Référence de la classe CustomModel

```
#include <custommodel.h>
```

Graphe d'héritage de CustomModel:



Fonctions membres publiques

- **CustomModel ()**
- **~CustomModel ()**
- **int rowCount** (const QModelIndex &parent) const
- **int columnCount** (const QModelIndex &parent) const
- **QVariant data** (const QModelIndex &index, int role) const
- **bool setData** (const QModelIndex &index, const QVariant &value, int role)

Documentation des constructeurs et destructeur

CustomModel::CustomModel () [inline], [explicit]

CustomModel::~~CustomModel () [inline]

Documentation des fonctions membres

int CustomModel::columnCount (const QModelIndex & *parent*) const [inline]

QVariant CustomModel::data (const QModelIndex & *index*, int *role*) const [inline]

int CustomModel::rowCount (const QModelIndex & *parent*) const [inline]

bool CustomModel::setData (const QModelIndex & *index*, const QVariant & *value*, int *role*) [inline]

La documentation de cette classe a été générée à partir du fichier suivant :

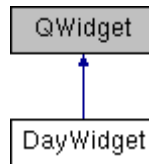
- Agenda/viewSummary/custommodel.h

Référence de la classe DayWidget

Classe utilitaire permettant d'afficher le nom d'un jour ainsi sa date. Le numéro du jour est affiché en grand en dessous du nom du jour.

```
#include <daywidget.h>
```

Graphe d'héritage de DayWidget:



Fonctions membres publiques

- **DayWidget** (QWidget *parent=0)
Constructeur.
- **~DayWidget** ()
- void **setDayNumber** (int noDay)
Modifier la numéro du jour.
- void **setDayName** (QString dayName)
Modification du nom du jour de la semaine.

Description détaillée

Classe utilitaire permettant d'afficher le nom d'un jour ainsi sa date. Le numéro du jour est affiché en grand en dessous du nom du jour.

Documentation des constructeurs et destructeur

DayWidget::DayWidget (QWidget * *parent* = 0) [explicit]

Constructeur.

Paramètres:

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

DayWidget::~DayWidget ()

Documentation des fonctions membres

void DayWidget::setDayName (QString *dayName*)

Modification du nom du jour de la semaine.

Paramètres:

<i>dayName</i>	nom du jour
----------------	-------------

void DayWidget::setDayNumber (int *noDay*)

Modifier la numéro du jour.

Paramètres:

<i>noDay</i>	numéro du jour
--------------	----------------

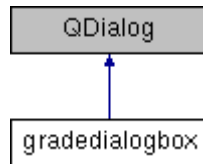
La documentation de cette classe a été générée à partir des fichiers suivants :

- Agenda/viewCalendar/**daywidget.h**
- Agenda/viewCalendar/**daywidget.cpp**

Référence de la classe gradedialogbox

```
#include <gradedialogbox.h>
```

Graphe d'héritage de gradedialogbox:



Connecteurs publics

- void **setGrade** (double grade)
slot récupérant la valeur de la note de la boîte de dialogue
- void **setCoeff** (double coeff)
slot récupérant la valeur du coefficient de la boîte de dialogue
- void **changedValue** ()
slot émettant un signal avertissant du changement des valeurs
- void **canceled** ()
slot émettant un signal avertissant qu'aucune valeur n'a été changée

Signaux

- void **updateValue** ()
signal émis lorsque les note ou coefficient sont modifiés
- void **canceledValue** ()
signal émis lorsque les note ou coefficient ne sont pas modifiés

Fonctions membres publiques

- **gradedialogbox** (QWidget *parent)
- **~gradedialogbox** ()
- double **getGrade** ()
retourne la valeur de la note
- double **getCoeff** ()
retourne la valeur du coefficient

Description détaillée

Classe créant une pop up pour ajouter une note et un coefficient à une tâche

Documentation des constructeurs et destructeur

gradedialogbox::gradedialogbox (QWidget * parent)

Constructeur de la classe gradedialogbox

Paramètres:

<i>parent</i>	:widget parent
---------------	----------------

gradedialogbox::~~gradedialogbox ()

Destructeur de la classe gradedialogbox

Documentation des fonctions membres

void gradedialogbox::canceled () [inline], [slot]

slot emettant un signal avertissant qu'aucune valeur n'a été changée

void gradedialogbox::canceledValue () [signal]

signal emis lorsque les note ou coefficient ne sont pas modifiés

void gradedialogbox::changedValue () [inline], [slot]

slot emettant un signal avertissant du changement des valeurs

double gradedialogbox::getCoeff () [inline]

retourne la valeur du coefficient

Renvoie:

la valeur du coefficient

double gradedialogbox::getGrade () [inline]

retourne la valeur de la note

Renvoie:

la valeur de la note

void gradedialogbox::setCoeff (double *coeff*) [inline], [slot]

slot récupérant la valeur du coefficient de la boîte de dialogue

Paramètres:

<i>coeff</i>	: valeur de la coefficient
--------------	----------------------------

void gradedialogbox::setGrade (double *grade*) [inline], [slot]

slot récupérant la valeur de la note de la boîte de dialogue

Paramètres:

<i>grade</i>	: valeur de la note
--------------	---------------------

void gradedialogbox::updateValue () [signal]

signal emis lorsque les note ou coefficient sont modifiés

La documentation de cette classe a été générée à partir des fichiers suivants :

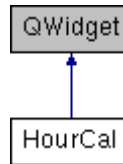
- Agenda/viewTask/**gradedialogbox.h**
- Agenda/viewTask/**gradedialogbox.cpp**

Référence de la classe HourCal

HourCal remplit chaque ligne de d'une colonne. Les lignes représente des heure.

```
#include <hourcal.h>
```

Graphe d'héritage de HourCal:



Fonctions membres publiques

- **HourCal** (QWidget *parent=0)
- **~HourCal** ()
- void **paintEvent** (QPaintEvent *event)
paintEvent redéfini afin de pouvoir dessiner une ligne en dessous de chaque ligne

Description détaillée

HourCal remplit chaque ligne de d'une colonne. Les lignes représente des heure.

Cette classe est utilitaire. Plusieurs instance de cette classe sont crée afin d'afficher une ligne de séparation au niveau des heures du calendrier.

Documentation des constructeurs et destructeur

HourCal::HourCal (QWidget * *parent* = 0)[inline], [explicit]

HourCal::~HourCal () [inline]

Documentation des fonctions membres

void HourCal::paintEvent (QPaintEvent * *event*)[inline]

paintEvent redéfini afin de pouvoir dessiner une ligne en dessous de chaque ligne

Paramètres:

<i>event</i>	type de l'événement
--------------	---------------------

La documentation de cette classe a été générée à partir du fichier suivant :

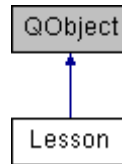
- Agenda/viewCalendar/hourcal.h

Référence de la classe Lesson

Modèle pour les leçons.

```
#include <lesson.h>
```

Graphe d'héritage de Lesson:



Types publics

- `enum Sorting { DAY_OF_WEEK }`

Fonctions membres publiques

- **Lesson** (const **Lesson** &l)
Constructeur de copie.
- `void update` (bool needClear=false)
Mettre à jour les données.
- `QSqlRelationalTableModel * getModel` () const
Récupérer le modèle de liaison.
- `int getCourseIndex` () const
Récupérer l'index du champ de la Foreign Key.
- `void sortBy` (**Sorting** sort, const Qt::SortOrder &order=Qt::AscendingOrder)
Trier les données.
- `Sorting getCurrentSort` () const
Récupérer l'ordre dans lequel les données sont actuellement triées.
- `void setCurrentSort` (const **Sorting** &value)
Configurer l'ordre dans lequel les données seront triées.
- `int rowCount` () const
Retourne le nombre de leçons dans la base de données.
- `~Lesson` ()
Destructeur.

Fonctions membres publiques statiques

- `static Lesson * getInstance` ()
Récupérer l'instance.

Attributs publics

- `bool needRefreshView`

Description détaillée

Modèle pour les leçons.

Modèle qui est utilisé comme lien entre la table "Lesson" de la base de données et le programme.

Documentation des énumérations membres

enum Lesson::Sorting

Valeurs énumérées
DAY_OF_WEEK

Documentation des constructeurs et destructeur

Lesson::Lesson (const Lesson & l)[inline]

Constructeur de copie.

Paramètres:

<i>l</i>	Leçon de base pour la copie
----------	-----------------------------

Lesson::~~Lesson ()

Destructeur.

Documentation des fonctions membres

int Lesson::getCOURSEIndex () const

Récupérer l'index du champ de la Foreign Key.

Renvoie:

L'index de la colonne courseId

Lesson::Sorting Lesson::getCurrentSort () const

Récupérer l'ordre dans lequel les données sont actuellement triés.

Renvoie:

L'ordre de tri

Lesson * Lesson::getInstance () [static]

Récupérer l'instance.

Renvoie:

L'instance unique (singleton)

QSqlRelationalTableModel * Lesson::getModel () const

Récupérer le modèle de liaison.

Renvoie:

Le modèle sur lequel effectuer les requêtes

int Lesson::rowCount () const

Retourne le nombre de leçons dans la base de données.

Renvoie:

nombre de leçons dans la base de données

void Lesson::setCurrentSort (const Sorting & value)

Configurer l'ordre dans lequel les données seront triées.

Paramètres:

<i>value</i>	L'ordre de tri voulu
--------------	----------------------

void Lesson::sortBy (Sorting sort, const Qt::SortOrder & order = Qt::AscendingOrder)

Trier les données.

Paramètres:

<i>sort</i>	Le type de tri à utiliser
<i>order</i>	Dans quel ordre trier

void Lesson::update (bool needClear = false)

Mettre à jour les données.

Paramètres:

<i>needClear</i>	Variable de contrôle pour savoir s'il faut réinitialiser
------------------	--

Documentation des données membres

bool Lesson::needRefreshView

La documentation de cette classe a été générée à partir des fichiers suivants :

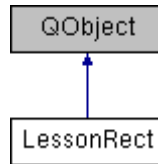
- `Agenda/models/lesson.h`
- `Agenda/models/lesson.cpp`

Référence de la classe LessonRect

LessonRect représente une leçon du calendrier.

```
#include <lessonrect.h>
```

Graphe d'héritage de LessonRect:



Fonctions membres publiques

- **LessonRect** (const QSqlRecord &, QObject *parent=0)
Constructeur.
- **~LessonRect** ()
- double **getStartTimeInHours** () const
récupère l'heure de début de la leçon en heure (décimale)
- double **getEndTimeInHours** () const
récupère l'heure de fin de la leçon en heure (décimale)
- double **getHeight** (double heightForHours) const
Calcule et retourne la taille en pixel de la leçon dans le calendrier.
- double **getY** (double heightForHours) const
Retourne la position y en pixel du rectangle de la leçon.
- bool **intersect** (double hour) const
Test si l'heure passé en paramètre se trouve dans entre l'heure de début de la leçon et son heure de fin.
- int **getId** () const
récupère l'id stocké dans la base de donnée de la leçon. Cette valeur est utilisée comme identifiant de leçon sélectionnée.
- QString **getName** () const
récupère le nom d'une leçon
- QString **getCourseName** ()
récupère le nom du cours auquel appartient la leçon

Description détaillée

LessonRect représente une leçon du calendrier.

Ce n'est pas cet objet qui est directement affiché. Il est utilisé pour stocker une leçon provenant de la base de donnée et de précalculer certaines valeurs qui serviront à dessiner le rectangle.

Documentation des constructeurs et destructeur

```
LessonRect::LessonRect (const QSqlRecord & record, QObject * parent = 0) [explicit]
```

Constructeur.

Paramètres:

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

LessonRect::~~LessonRect ()

Documentation des fonctions membres

QString LessonRect::getCourseName ()

récupère le nom du cours auquel appartient la leçon

double LessonRect::getEndTimeInHours () const

récupère l'heure de fin de la leçon en heure (décimale)

Renvoie:

double LessonRect::getHeight (double *heightForHours*) const

Calcule et retourne la taille en pixel de la leçon dans le calendrier.

Paramètres:

<i>heightForHours</i>	Taille en pixel que doit prendre un rectangle de 1 heure
-----------------------	--

int LessonRect::getId () const

récupère l'id stocké dans la base de donnée de la leçon. Cette valeur est utilisée comme identifiant de leçon sélectionnée.

Renvoie:

QString LessonRect::getName () const

récupère le nom d'une leçon

Renvoie:

double LessonRect::getStartTimeInHours () const

recupère l'heure de début de la leçon en heure (décimale)

double LessonRect::getY (double *heightForHours*) const

Retourne la position y en pixel du rectangle de la leçon.

Paramètres:

<i>heightForHours</i>	Taille en pixel que doit prendre un rectangle de 1 heure.
-----------------------	---

Renvoie:

bool LessonRect::intersect (double *hour*) const

Test si l'heure passé en paramètre se trouve dans entre l'heure de début de la leçon et son heure de fin.

Utile pour savoir quel leçon à été selectionné suite à un clic de souris dans le calendrier.

Paramètres:

<i>hour</i>	heure à tester
-------------	----------------

Renvoie:

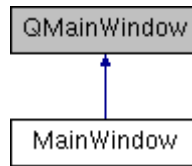
La documentation de cette classe a été générée à partir des fichiers suivants :

- Agenda/viewCalendar/lessonrect.h
- Agenda/viewCalendar/lessonrect.cpp

Référence de la classe MainWindow

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

Graphe d'héritage de MainWindow:



Signaux

- `void to_cal_triggered ()`
signal emis lorsque la vue Calendrier est appelée

Fonctions membres publiques

- **MainWindow** (QWidget *parent=0)
Constructeur.
- **~MainWindow** ()
Destructeur.
- **SideBarTask * getSideBarTask ()**
Récupère un pointeur sur une SideBarTask.
- **SideBarSummary * getSideBarSummary ()**
Récupère un pointeur sur une SideBarSummary.
- **SideBarCalendar * getSideBarCalendar ()**
Récupère un pointeur sur une SideBarCalendar.
- **ToolBarCalendar * getToolBarCalendar ()**
Récupère un pointeur sur une ToolBarCalendar.
- `void hideSidebar ()`
Cache la SideBar Lorsqu'il n'y a plus de tâche listée, la SideBar est cachée.
- `void showSidebar ()`
Montre la SideBar Tant qu'il y à des tâches affichée, la SideBar est visible.

Documentation des constructeurs et destructeur

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

Constructeur.

Paramètres:

<i>parent</i>	: widget parent
---------------	-----------------

MainWindow::~MainWindow ()

Destructeur.

Documentation des fonctions membres

SideBarCalendar * MainWindow::getSideBarCalendar ()

Récupère un pointeur sur une **SideBarCalendar**.

Renvoie:

le pointeur sur une **SideBarCalendar**

SideBarSummary * MainWindow::getSideBarSummary ()

Récupère un pointeur sur une **SideBarSummary**.

Renvoie:

le pointeur sur une **SideBarSummary**

SideBarTask * MainWindow::getSideBarTask ()

Récupère un pointeur sur une **SideBarTask**.

Renvoie:

le pointeur sur une **SideBarTask**

ToolBarCalendar * MainWindow::getToolBarCalendar ()

Récupère un pointeur sur une **ToolBarCalendar**.

Renvoie:

le pointeur sur une **ToolBarCalendar**

void MainWindow::hideSidebar ()

Cache la SideBar Lorsqu'il n'y a plus de tâche listée, la SideBar est cachée.

void MainWindow::showSidebar ()

Montre la SideBar Tant qu'il y a des tâches affichée, la SideBar est visible.

void MainWindow::to_cal_triggered () [signal]

signal emis lorsque la vue Calendrier est appelée

La documentation de cette classe a été générée à partir des fichiers suivants :

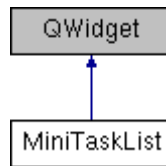
- Agenda/**mainwindow.h**
- Agenda/**mainwindow.cpp**

Référence de la classe MiniTaskList

MiniTaskList est une vue personnalisée d'une model **Task** Elle affiche une liste allégé de tâches.

```
#include <minitasklist.h>
```

Graphe d'héritage de MiniTaskList:



Connecteurs publics

- **void refreshTaskList ()**
rafraichissement de la liste des tâche Efface la liste des tâche est reconstruit la liste selon le nouvel état du model

Fonctions membres publiques

- **MiniTaskList (QWidget *parent=0)**
Constructeur.
- **~MiniTaskList ()**
- **void setUpTaskList ()**
Construction de la liste des tâche selon l'état courant du model.
- **void clearView ()**
Efface la liste des tâches.
- **TaskListItem * createTaskItem (int modelRow)**
Création d'un élément de la liste.

Description détaillée

MiniTaskList est une vue personnalisée d'une model **Task** Elle affiche une liste allégé de tâches.

Documentation des constructeurs et destructeur

MiniTaskList::MiniTaskList (QWidget * *parent* = 0) [explicit]

Constructeur.

Paramètres:

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

MiniTaskList::~MiniTaskList ()

Documentation des fonctions membres

void MiniTaskList::clearView ()

Efface la liste des tâches.

TaskListItem * MiniTaskList::createTaskItem (int *modelRow*)

Création d'un élément de la liste.

Paramètres:

<i>modelRow</i>	numéro de ligne du model correspondant à l'élément voulu.
-----------------	---

L'élément de la liste est mappé en fonction du numéro de ligne du model **Task** Ainsi chaque ligne se mettra à jour si le model est changé.

void MiniTaskList::refreshTaskList () [slot]

rafraichissement de la liste des tâche Efface la liste des tâche est reconstruit la liste selon le nouvel état du model

void MiniTaskList::setUpTaskList ()

Construction de la liste des tâche selon l'état courant du model.

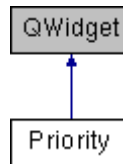
La documentation de cette classe a été générée à partir des fichiers suivants :

- Agenda/viewCalendar/**minitasklist.h**
- Agenda/viewCalendar/**minitasklist.cpp**

Référence de la classe Priority

```
#include <priority.h>
```

Graphe d'héritage de Priority:



Fonctions membres publiques

- **Priority** (QWidget *parent=0)
- **int getPriority () const**
retourne la priorité de la tâche
- **void setPriority (int value)**
modifie la priorité

Propriétés

- **int priority**

Description détaillée

Classe gérant la priorité d'une tâche

Documentation des constructeurs et destructeur

Priority::Priority (QWidget * *parent* = 0) [inline], [explicit]

Constructeur de la classe **Priority**

Paramètres:

<i>parent</i>	: widget parent
---------------	-----------------

Documentation des fonctions membres

int Priority::getPriority () const [inline]

retourne la priorité de la tâche

Renvoie:

la priorité

void Priority::setPriority (int *value*) [inline]

modifie la priorité

Documentation des propriétés

int Priority::priority[read], [write]

La documentation de cette classe a été générée à partir du fichier suivant :

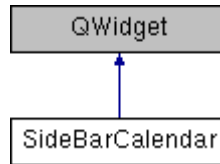
- Agenda/viewTask/priority.h

Référence de la classe SideBarCalendar

Sidebar de la vue Calendrier.

```
#include <sidebarcalendar.h>
```

Graphe d'héritage de SideBarCalendar:



Connecteurs publics

- void **pageChanged** (int, int)
Mise à jour des informations de la sidebar suite à un changement de mois depuis le miniCalendrier.
- void **onTodayClicked** ()
Aller à la date du jour.
- void **onDayClicked** ()
Lors d'un clique sur un jour, on met à jour les informations du numéro de jour et du nom de jour en cours.
- void **showMiniTaskList** ()
Affichage du miniCalendrier La Sidebar contient une pile de widget. Elle switch sur le widget voulue.
- void **showMiniCalendar** ()
showMiniCalendar La Sidebar contient une pile de widget. Elle switch sur le widget voulue.

Fonctions membres publiques

- **SideBarCalendar** (QWidget *parent=0)
Constructeur.
- **~SideBarCalendar** ()
- **CustomCalendarWidget * getMinicalendar** ()
Récupération du Widget MiniCalendar.
- **MiniTaskList * getMiniTaskList** ()
Récupération du widget MiniTaskList.

Description détaillée

Sidebar de la vue Calendrier.

La Sidebar fournit un miniCalendrier et peut switcher sur une liste de tâches si l'utilisateur sélectionne un jour dans l'entête du calendrier.

Le miniCalendrier sert à naviguer entre les semaines affichées dans le calendrier.

Documentation des constructeurs et destructeur

SideBarCalendar::SideBarCalendar (QWidget * parent = 0)[explicit]

Constructeur.

Paramètres:

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

SideBarCalendar::~~SideBarCalendar ()

Documentation des fonctions membres**CustomCalendarWidget * SideBarCalendar::getMinicalendar ()**

Récupération du Widget MiniCalendar.

MiniTaskList * SideBarCalendar::getMiniTaskList ()

Récupération du widget **MiniTaskList**.

Renvoie:**void SideBarCalendar::onDayClicked () [slot]**

Lors d'un clique sur un jour, on met à jour les informations du numéro de jour et du nom de jour en cours.

void SideBarCalendar::onTodayClicked () [slot]

Aller à la date du jour.

void SideBarCalendar::pageChanged (int *year*, int *month*) [slot]

Mise à jour des informations de la sidebar suite à un changement de mois depuis le miniCalendrier.

void SideBarCalendar::showMiniCalendar () [slot]

showMiniCalendar La Sidebar contient une pile de widget. Elle switch sur le widget voulue.

void SideBarCalendar::showMiniTaskList () [slot]

Affichage du miniCalendrier La Sidebar contient une pile de widget. Elle switch sur le widget voulue.

La documentation de cette classe a été générée à partir des fichiers suivants :

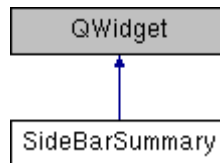
- Agenda/viewCalendar/sidebarcalendar.h
- Agenda/viewCalendar/sidebarcalendar.cpp

Référence de la classe SideBarSummary

Classe représentant la side bar (barre à droite) dans la vue résumé

```
#include <sidebarsummary.h>
```

Graphe d'héritage de SideBarSummary:



Fonctions membres publiques

- **SideBarSummary** (QWidget *parent=0)
Constructeur de la classe SideBarSummary.
- **~SideBarSummary** ()
Destructeur de la classe SideBarSummary.
- **QVBoxLayout * getLayout** ()
Getter pour le layout.

Description détaillée

Classe représentant la side bar (barre à droite) dans la vue résumé

Cette classe va contenir la **SideBarTreeView** et donc le **TreeView**

Documentation des constructeurs et destructeur

SideBarSummary::SideBarSummary (QWidget * *parent* = 0)[explicit]

Constructeur de la classe **SideBarSummary**.

Paramètres:

<i>parent</i>	: Le widget parent.
---------------	---------------------

SideBarSummary::~SideBarSummary ()

Destructeur de la classe **SideBarSummary**.

Documentation des fonctions membres

QVBoxLayout * SideBarSummary::getLayout ()

Getter pour le layout.

Renvoie:

Le layout correspondant

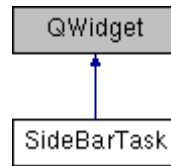
La documentation de cette classe a été générée à partir des fichiers suivants :

- Agenda/viewSummary/sidebarsummary.h
- Agenda/viewSummary/sidebarsummary.cpp

Référence de la classe SideBarTask

```
#include <sidebartask.h>
```

Grphe d'héritage de SideBarTask:



Connecteurs publics

- void **loadTask** (int)
chargement des tâches La tâche est chargée dans la SideBar
- void **submitTask** ()
sauvegarde des changement La tâche est enregistrée dans la base de donnée

Signaux

- void **isUpdated** ()
signal de modification d'une tâche
- void **needRefresh** ()
signal rafraichissement Lorsque la liste des tâches à besoin d'être rafraichie ce signal est envoyé
- void **changeSelection** (int)
signal de changement de sélection Lorsque la liste des tâches est modifié la ligne de la tâche est gardée en mémoire afin de rester sélectionnée après un tri par exemple

Fonctions membres publiques

- **SideBarTask** (QWidget *parent=0)
Constructeur Constructeur de la classe SideBarTask.
- **~SideBarTask** ()
Destructeur Destructeur de la classe SideBarTask.
- QDataWidgetMapper * **getMapper** ()
Récupère le pointeur sur le mapper de données.

Fonctions membres protégées

- void **keyReleaseEvent** (QKeyEvent *event)
Evenement lorsqu'une touche est relâchée Lorsque la touche entrée est relâchée les données de la tâche en édition sont sauvegardée.

Description détaillée

Classe gérant le widget de la bar latérale

Documentation des constructeurs et destructeur

SideBarTask::SideBarTask (QWidget * *parent* = 0) [explicit]

Constructeur Constructeur de la classe **SideBarTask**.

Paramètres:

<i>parent</i>	: widget parent
---------------	-----------------

SideBarTask::~~SideBarTask ()

Destructeur Destructeur de la classe **SideBarTask**.

Documentation des fonctions membres

void SideBarTask::changeSelection (int) [signal]

signal de changement de sélection Lorsque la liste des tâches est modifié la ligne de la tâche est gardée en mémoire afin de rester sélectionnée après un tri par exemple

Paramètres:

<i>int</i>	: la ligne de la tâche
------------	------------------------

QDataWidgetMapper * SideBarTask::getMapper ()

Récupère le pointeur sur le mapper de données.

Renvoie:

le pointeur sur le mapper

void SideBarTask::isUpdated () [signal]

signal de modification d'une tâche

void SideBarTask::keyReleaseEvent (QKeyEvent * *event*) [protected]

Evenement lorsqu'une touche est relâchée Lorsque la touche entrée est relâchée les données de la tâche en édition sont sauvegardée.

Paramètres:

<i>event</i>	: la touche relâchée
--------------	----------------------

void SideBarTask::loadTask (int *row*) [slot]

chargement des tâches La tâche est chargée dans la SideBar

Paramètres:

<i>int</i>	: la ligne de la tâche
------------	------------------------

void SideBarTask::needRefresh () [signal]

signal rafraichissement Lorsque la liste des tâches à besoin d'être rafraichie ce signal est envoyé

void SideBarTask::submitTask () [slot]

sauvegarde des changement La tâche est enregistrée dans la base de donnée

Paramètres:

<i>int</i>	: la ligne de la tâche
------------	------------------------

La documentation de cette classe a été générée à partir des fichiers suivants :

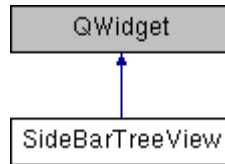
- Agenda/viewTask/**sidebartask.h**
- Agenda/viewTask/**sidebartask.cpp**

Référence de la classe SideBarTreeView

Classe créant et contenant le **TreeView**.

```
#include <testtreeview.h>
```

Graphe d'héritage de SideBarTreeView:



Fonctions membres publiques

- **SideBarTreeView** (QWidget *parent=0, VSummary *vSummary=0)
Constructeur de la classe SideBarTreeView.
- void **refresh** ()
Rafraîchit la SideBar et le TreeView.
- **TreeView** * **getTreeView** ()
Getter sur le TreeView.
- **TreeModel** * **getTreeModel** ()
Getter sur le TreeModel.

Description détaillée

Classe créant et contenant le **TreeView**.

Crée et contient le **TreeView**. Ce widget sera ajouté dans la **SideBarSummary**.

Documentation des constructeurs et destructeur

SideBarTreeView::SideBarTreeView (QWidget * *parent* = 0, VSummary * *vSummary* = 0)[**explicit**]

Constructeur de la classe **SideBarTreeView**.

Paramètres:

<i>parent</i>	: Le widget parent.
<i>summary</i>	: Fenêtre contenant le SideBarTreeView (vue résumé).

Documentation des fonctions membres

TreeModel* **SideBarTreeView::getTreeModel** () [**inline**]

Getter sur le **TreeModel**.

Renvoie:

Le **TreeModel**

TreeView* SideBarTreeView::getTreeView () [inline]

Getter sur le **TreeView**.

Renvoie:

Le **TreeView**

void SideBarTreeView::refresh ()

Rafraîchit la SideBar et le **TreeView**.

La documentation de cette classe a été générée à partir des fichiers suivants :

- Agenda/viewSummary/testtreeview.h
- Agenda/viewSummary/testtreeview.cpp

Référence de la classe SummaryUtils

Classe utilitaire.

```
#include <summaryutils.h>
```

Fonctions membres publiques statiques

- static void **execQueryOnDatabase** (QString query)
Exécute la requête SQL passée en paramètre.
- static int **messageBoxOkCancel** (QString text)
Affiche une message box contenant un message avec les boutons "OK" et "Cancel".
- static void **clearLayout** (QLayout *layout)
Vide le layout passé en paramètre.
- static QList< **Average** > **getAllAveragesInCourse** (int courseId)
Retourne la liste des moyennes des cours.

Description détaillée

Classe utilitaire.

Possède des fonctions statiques utilitaires.

Documentation des fonctions membres

static void SummaryUtils::clearLayout (QLayout * *layout*) [inline], [static]

Vide le layout passé en paramètre.

Supprime tous les widgets à l'intérieur du layout passé en paramètre.

Paramètres:

<i>layout</i>	: Le layout à vider
---------------	---------------------

static void SummaryUtils::execQueryOnDatabase (QString *query*) [inline], [static]

Exécute la requête SQL passée en paramètre.

Paramètres:

<i>query</i>	: Requête SQL à exécuter
--------------	--------------------------

static QList<Average> SummaryUtils::getAllAveragesInCourse (int *courseId*) [inline], [static]

Retourne la liste des moyennes des cours.

Retourne les moyennes des cours (moyenne des types de tâches (TE, Labo...)) et du cours

Paramètres:

<i>courseId</i>	: L'id du cours dans lequel calculer les moyennes
-----------------	---

Renvoie:

La liste des moyennes

static int SummaryUtils::messageBoxOkCancel (QString *text*)[inline], [static]

Affiche une message box contenant un message avec les boutons "OK" et "Cancel".

Paramètres:

<i>text</i>	: Le texte à afficher
-------------	-----------------------

Renvoie:

la valeur du bouton sélectionné par l'utilisateur (OK = QMessageBox::Ok, Cancel = QMessageBox::Cancel)

La documentation de cette classe a été générée à partir du fichier suivant :

- Agenda/viewSummary/summaryutils.h

Référence de la classe Task

Modèle pour les tâches.

```
#include <task.h>
```

Graphe d'héritage de Task:



Types publics

- enum **Sorting** { **BY_DATE**, **BY_COURSE** }
- enum **Filter** { **TODO**, **DONE** }

Signaux

- void **newTask** (int)
Signal pour une nouvelle tâche.
- void **modelUpdated** ()
Signal pour actualiser le modèle.
- void **modelFiltered** ()
Signal pour appliquer des filtres.

Fonctions membres publiques

- ~**Task** ()
Destructeur.
- void **update** (bool needClear=false)
Mettre à jour les données.
- QSqlRelationalTableModel * **getModel** () const
Récupérer le modèle de liaison.
- int **getCourseIndex** () const
Récupérer l'index du champ de la Foreign Key.
- int **getTypeIndex** () const
Récupérer l'index du champ de la Foreign Key.
- void **addTask** ()
Ajouter une tâche.
- void **removeTask** (const int row)
Supprimer une tâche.
- int **rowCount** () const
Nombre de tâches.
- void **sortBy** (**Sorting** sort, const Qt::SortOrder &order=Qt::AscendingOrder)
Trier les données.
- void **filterBy** (**Filter** filter)
Ajouter des filtres aux données.
- **Filter** **getCurrentFilter** () const
Récupérer le filtre actuellement utilisé

- void **setCurrentFilter** (const **Filter** &value)
Configurer le filtre à utiliser.
- **Sorting** **getCurrentSort** () const
Récupérer le tri actuellement utilisé
- void **setCurrentSort** (const **Sorting** &value)
Configurer le tri à utiliser.
- int **findRowById** (int id)
Trouver une tâche en utilisant l'index.

Fonctions membres publiques statiques

- static **Task** * **getInstance** ()
Récupérer l'instance unique (singleton)

Attributs publics

- bool **needRefreshView**

Description détaillée

Modèle pour les tâches.

Modèle qui est utilisé comme lien entre la table "Task" de la base de données et le programme.

Documentation des énumérations membres

enum Task::Filter

Valeurs énumérées

TODO

DONE

enum Task::Sorting

Valeurs énumérées

BY_DATE

BY_COURSE

Documentation des constructeurs et destructeur

Task::~Task ()

Destructeur.

Documentation des fonctions membres

void Task::addTask ()

Ajouter une tâche.

void Task::filterBy (Filter *filter*)

Ajouter des filtres aux données.

Paramètres:

<i>filter</i>	Le filtre à appliquer
---------------	-----------------------

int Task::findRowById (int *id*)

Trouver une tâche en utilisant l'index.

Paramètres:

<i>id</i>	L'index de la tâche
-----------	---------------------

Renvoie:

La ligne à laquelle la tâche se trouve

int Task::getCourseIndex () const

Récupérer l'index du champ de la Foreign Key.

Renvoie:

L'index de la colonne courseId

Task::Filter Task::getCurrentFilter () const

Récupérer le filtre actuellement utilisé

Renvoie:

filtre actuel

Task::Sorting Task::getCurrentSort () const

Récupérer le tri actuellement utilisé

Renvoie:

trie actuel

Task * Task::getInstance () [static]

Récupérer l'instance unique (singleton)

Renvoie:

instance unique

QSqlRelationalTableModel * Task::getModel () const

Récupérer le modèle de liaison.

Renvoie:

Le modèle sur lequel effectuer les requêtes

int Task::getTypeIndex () const

Récupérer l'index du champ de la Foreign Key.

Renvoie:

L'index de la colonne typeId

void Task::modelFiltered () [signal]

Signal pour appliquer des filtres.

void Task::modelUpdated () [signal]

Signal pour actualiser le modèle.

void Task::newTask (int) [signal]

Signal pour une nouvelle tâche.

void Task::removeTask (const int row)

Supprimer une tâche.

Paramètres:

<i>row</i>	tâche à supprimer
------------	-------------------

int Task::rowCount () const

Nombre de tâches.

Renvoie:

nombre de tâches

void Task::setCurrentFilter (const Filter & *value*)

Configurer le filtre à utiliser.

Paramètres:

<i>value</i>	Filtre à utiliser
--------------	-------------------

void Task::setCurrentSort (const Sorting & *value*)

Configurer le tri à utiliser.

Paramètres:

<i>value</i>	Tri à utiliser
--------------	----------------

void Task::sortBy (Sorting *sort*, const Qt::SortOrder & *order* = Qt::AscendingOrder)

Trier les données.

Paramètres:

<i>sort</i>	Selon quel champs les trier
<i>order</i>	Ordre croissant/décroissant

void Task::update (bool *needClear* = false)

Mettre à jour les données.

Paramètres:

<i>needClear</i>	Variable de contrôle pour savoir s'il faut réinitialiser
------------------	--

Documentation des données membres

bool Task::needRefreshView

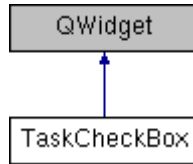
La documentation de cette classe a été générée à partir des fichiers suivants :

- Agenda/models/task.h
- Agenda/models/task.cpp

Référence de la classe TaskCheckBox

```
#include <taskcheckbox.h>
```

Graphe d'héritage de TaskCheckBox:



Connecteurs publics

- void **updateTaskWidget** ()
récupère la ligne de la tâche courante
- void **onRemoveTaskClicked** ()
suppression de la tâche courante
- void **checkboxChecked** (int)
Lorsque la checkbox est validée.
- void **checkBoxClicked** (bool)
Lorsque la checkbox est cliquée.
- void **getGradeAndCoeff** ()
Récupère les valeurs une fois la tâche finie Récupération de la note ainsi que du coefficient d'une tâche notée.
- void **changedCanceled** ()
Si la tâche ne doit pas être passée en mode terminé

Signaux

- void **editTask** (int)
signal lorsqu'une tâche est éditée
- void **removeTask** (int)
signal lorsqu'une tâche est supprimée
- void **needRefreshTask** ()
signal lorsque la liste de tâche doit être rafraichie

Fonctions membres publiques

- **TaskCheckBox** (SideBarTask *sidebar, int row, QWidget *parent=0)
*Constructeur Constructeur de la classe **TaskCheckBox**.*
- void **setSelected** (bool enable)
Changement de design Changement de design d'une tâche si elle est sélectionne.
- int **getRow** ()
retourne la ligne de la tâche
- void **emitEditTask** ()
Emet un signal lorsque la tâche est modifiée.

Fonctions membres protégées

- virtual void **mousePressEvent** (QMouseEvent *event)

Événement de souris Lorsque l'on clique sur la checkbox celle si change de style et la tâche passe en mode terminé

- **~TaskCheckBox ()**
Destructeur Destructeur de la classe TaskCheckBox.

Description détaillée

Cette classe a pour objectif de lier une checkbox signifiant la complétion d'une tâche à cette tâche

Documentation des constructeurs et destructeur

TaskCheckBox::TaskCheckBox (SideBarTask * sidebar, int row, QWidget * parent = 0) [explicit]

Constructeur Constructeur de la classe **TaskCheckBox**.

Paramètres:

<i>sidebar</i>	pointeur sur le widget de la barre latérale
<i>row</i>	: ligne de la tâche traitée
<i>parent</i>	: widget parent

TaskCheckBox::~TaskCheckBox () [protected]

Destructeur Destructeur de la classe **TaskCheckBox**.

Documentation des fonctions membres

void TaskCheckBox::changedCanceled () [slot]

Si la tâche ne doit pas être passée en mode terminé

void TaskCheckBox::checkboxChecked (int state) [slot]

Lorsque la checkbox est validée.

Paramètres:

<i>int</i>	: la ligne de la tâche
------------	------------------------

void TaskCheckBox::checkBoxClicked (bool checked) [slot]

Lorsque la checkbox est cliquée.

Paramètres:

<i>bool</i>	: true si la checkbox est sélectionnée, false si elle est désélectionnée
-------------	--

void TaskCheckBox::editTask (int) [signal]

signal lorsqu'une tâche est éditée

Paramètres:

<i>int</i>	: la ligne de la tâche
------------	------------------------

void TaskCheckBox::emitEditTask ()

Emet un signal lorsque la tâche est modifiée.

void TaskCheckBox::getGradeAndCoeff () [slot]

Récupère les valeurs une fois la tâche finie Récupération de la note ainsi que du coefficient d'une tâche notée.

int TaskCheckBox::getRow ()

retourne la ligne de la tâche

Renvoie:

la ligne de la tâche

void TaskCheckBox::mousePressEvent (QMouseEvent * event) [protected], [virtual]

Événement de souris Lorsque l'on clique sur la checkbox celle si change de style et la tâche passe en mode terminé

Paramètres:

<i>event</i>	non utilisé
--------------	-------------

void TaskCheckBox::needRefreshTask () [signal]

signal lorsque la liste de tâche doit être rafraichie

void TaskCheckBox::onRemoveTaskClicked () [slot]

suppression de la tâche courante

void TaskCheckBox::removeTask (int) [signal]

signal lorsqu'une tâche est supprimée

Paramètres:

<i>int</i>	: la ligne de la tâche
------------	------------------------

void TaskCheckBox::setSelected (bool *enable*)

Changement de design Changement de design d'une tâche si elle est sélectionnée.

Paramètres:

<i>enable</i>	true si la tâche est sélectionnée
---------------	-----------------------------------

void TaskCheckBox::updateTaskWidget () [slot]

récupère la ligne de la tâche courante

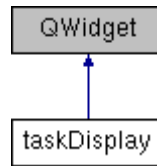
La documentation de cette classe a été générée à partir des fichiers suivants :

- Agenda/viewTask/**taskcheckbox.h**
- Agenda/viewTask/**taskcheckbox.cpp**

Référence de la classe taskDisplay

```
#include <taskdisplay.h>
```

Grphe d'héritage de taskDisplay:



Connecteurs publics

- void **refreshTaskList** (int selection=0)
Rafraichissement de la liste des tâches.
- void **updateTaskWidget** ()
Rafraichissement d'une tâche.
- void **addTask** (int)
Ajout d'une tâche.
- void **selectWidget** (int)
Sélection d'une tâche Changement du design de la tâche sélectionnée.
- void **removeTask** (int row)
Suppression d'une tâche.
- void **beforeInsert** (QSqlRecord &)
Avant insertion d'une tâche Mise des valeurs par défaut de la tâche.
- void **retrieveSelection** ()
Récupération de la tâche précédemment sélectionnée En cas de tri il se peut que la tâche sélectionnée soit perdu afin de garder la sélection retrieveSelection est appelée.

Signaux

- void **isUpdated** ()
Signal émit lorsqu'une tâche est modifiée.
- void **taskInserted** (int)
Signal émit lorsqu'une tâche est insérée.

Fonctions membres publiques

- **taskDisplay** (MainWindow *main_ui, QWidget *parent=0)
*Constructeur Constructeur de la classe **taskDisplay**.*
- void **setUpTaskList** ()
Création de la liste des tâches.
- ~**taskDisplay** ()
*Destructeur Destructeur de la classe **taskDisplay**.*
- void **clearView** ()
Destruction de la liste des tâches.

Fonctions membres protégées

- void **keyReleaseEvent** (QKeyEvent *event)

keyPressEvent

- void **mousePressEvent** (QMouseEvent *event)
mousePressEvent
- **TaskCheckBox** * **createTaskItem** (int modelRow)
*Création d'un objet **TaskCheckBox** TaskCheckBoxy est composé de la tâche ainsi que de la checkbox.*

Description détaillée

Classe gérant l'affichage d'une liste de tâche

Documentation des constructeurs et destructeur

taskDisplay::taskDisplay (MainWindow * *main_ui*, QWidget * *parent* = 0) [explicit]

Constructeur Constructeur de la classe **taskDisplay**.

Paramètres:

<i>main_ui</i>	: fenêtre principale
<i>parent</i>	: widget parent

taskDisplay::~~taskDisplay ()

Destructeur Destructeur de la classe **taskDisplay**.

Documentation des fonctions membres

void taskDisplay::addTask (int *row*) [slot]

Ajout d'une tâche.

Paramètres:

<i>int</i>	: la ligne de la tâche
------------	------------------------

void taskDisplay::beforeInsert (QSqlRecord & *record*) [slot]

Avant insertion d'une tâche Mise des valeurs par défaut de la tâche.

Paramètres:

<i>QSqlRecord</i>	: valeur de la tâche
-------------------	----------------------

void taskDisplay::clearView ()

Destruction de la liste des tâches.

taskDisplay::clearView efface la liste des éléments du layout

TaskCheckBox * taskDisplay::createTaskItem (int *modelRow*) [protected]

Création d'un objet **TaskCheckBox** TaskCheckBoxy est composé de la tâche ainsi que de la checkbox.

taskDisplay::createTaskItem crée un widget (TaskCheckbox).

Paramètres:

<i>modelRow</i>	: la ligne de la tâche
<i>modelRow</i>	No de ligne du model

Renvoie:

le widget créé

void taskDisplay::isUpdated () [signal]

Signal émit lorsqu'une tâche est modifiée.

void taskDisplay::keyReleaseEvent (QKeyEvent * *event*) [protected]

keyReleaseEvent

Gestion des raccourcis claviers Touche Up/Down pour parcourir les tâches

Paramètres:

<i>event</i>	
--------------	--

void taskDisplay::mouseReleaseEvent (QMouseEvent * *event*) [protected]

mouseReleaseEvent

Gestion du click sur la vue principale En cas d'un clique gauche on donne le focus à la vue Cela permet d'utiliser les raccourcis claviers

Paramètres:

<i>event</i>	
--------------	--

void taskDisplay::refreshTaskList (int *selection* = 0) [slot]

Rafraichissement de la liste des tâches.

Paramètres:

<i>selection</i>	: la ligne de la tâche
------------------	------------------------

void taskDisplay::removeTask (int *row*) [slot]

Suppression d'une tâche.

Paramètres:

<i>row</i>	: la ligne de la tâche
------------	------------------------

void taskDisplay::retrieveSelection () [slot]

Récupération de la tâche précédemment sélectionnée En cas de tri il se peut que la tâche sélectionnée soit perdu afin de garder la sélection retrieveSelection est appelée.

void taskDisplay::selectWidget (int row) [slot]

Sélection d'une tâche Changement du design de la tâche sélectionnée.

Paramètres:

<i>int</i>	: la ligne de la tâche
------------	------------------------

void taskDisplay::setUpTaskList ()

Création de la liste des tâches.

taskDisplay::setUpTaskList initialise/réinitialise la liste des tâches

La liste des tâches est représentée par un layout vertical contenant des widgets **TaskCheckBox**.

void taskDisplay::taskInserted (int) [signal]

Signal émit lorsqu'une tâche est insérée.

Paramètres:

<i>int</i>	: la ligne de la tâche
------------	------------------------

void taskDisplay::updateTaskWidget () [slot]

Rafraichissement d'une tâche.

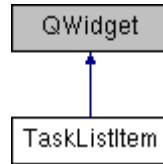
La documentation de cette classe a été générée à partir des fichiers suivants :

- Agenda/viewTask/taskdisplay.h
- Agenda/viewTask/taskdisplay.cpp

Référence de la classe TaskListItem

```
#include <tasklistitem.h>
```

Graphe d'héritage de TaskListItem:



Connecteurs publics

- void **updateTaskWidget** ()
- void **checkboxChecked** (int)
- void **checkBoxClicked** (bool)
- void **getGradeAndCoeff** ()
- void **changedCanceled** ()

Signaux

- void **editTask** (int)
- void **needRefreshTask** ()

Fonctions membres publiques

- **TaskListItem** (int row, QWidget *parent=0)
- **~TaskListItem** ()

Documentation des constructeurs et destructeur

TaskListItem::TaskListItem (int *row*, QWidget * *parent* = 0) [explicit]

TaskListItem::~~TaskListItem ()

Documentation des fonctions membres

void TaskListItem::changedCanceled () [slot]

void TaskListItem::checkboxChecked (int *state*) [slot]

void TaskListItem::checkBoxClicked (bool *checked*) [slot]

void TaskListItem::editTask (int) [signal]

void TaskListItem::getGradeAndCoeff () [slot]

void TaskListItem::needRefreshTask () [signal]

void TaskListItem::updateTaskWidget () [slot]

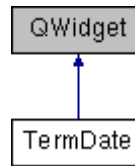
La documentation de cette classe a été générée à partir des fichiers suivants :

- Agenda/tasklistitem.h
- Agenda/tasklistitem.cpp

Référence de la classe TermDate

```
#include <termdate.h>
```

Grphe d'héritage de TermDate:



Fonctions membres publiques

- **TermDate** (QWidget *parent=0)
*Constructeur Constructeur de la classe **TermDate**.*
- **~TermDate** ()
*Destructeur Destructeur de la classe **TermDate**.*
- QString **getTermDate** () const
Récupération de la date d'échéance Récupération de la date d'échéance sous la forme demain, dans 3 jours etc...
- void **setTermDate** (QString value)
TermDate::setTermDate.

Attributs publics

- QLabel * **label**
- QHBoxLayout * **layout**
- QFont **font**
- const QString **STYLE_DEFAULT**
- const QString **STYLE_WARNING**

Propriétés

- QString **termDate**

Description détaillée

Indication de la date d'échance d'une tâche nb jours = 1 : demain nb jours = 2 : après-demain nb jours > 2 : lundi, mardi, etc..

classe gérant l'affichage de la date d'échéance d'une tâche

Documentation des constructeurs et destructeur

TermDate::TermDate (QWidget * *parent* = 0) [explicit]

Constructeur Constructeur de la classe **TermDate**.

Paramètres:

<i>parent</i>	: le widget parent
---------------	--------------------

TermDate::~~TermDate ()

Destructeur Destructeur de la classe **TermDate**.

Documentation des fonctions membres

QString TermDate::getTermDate () const

Récupération de la date d'échéance Récupération de la date d'échéance sous la forme demain, dans 3 jours etc...

Renvoie:

la date d'échéance

void TermDate::setTermDate (QString *value*)

TermDate::setTermDate.

Affichage friendly de la date d'échance dans la liste des tâches aujourd'hui, demain, après-demain ou dans x jours

Paramètres:

<i>value</i>	
--------------	--

Documentation des données membres

QFont TermDate::font

QLabel* TermDate::label

QHBoxLayout* TermDate::layout

const QString TermDate::STYLE_DEFAULT

const QString TermDate::STYLE_WARNING

Documentation des propriétés

QString TermDate::termDate [read], [write]

La documentation de cette classe a été générée à partir des fichiers suivants :

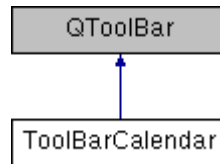
- Agenda/viewTask/**termdate.h**
- Agenda/viewTask/**termdate.cpp**

Référence de la classe ToolBarCalendar

ToolBarCalendar est la barre d'outil de la vue Calendrier. Elle affiche le slider permettant de zoomer sur la vue calendrier.

```
#include <toolbarcalendar.h>
```

Graphes d'héritage de ToolBarCalendar:



Fonctions membres publiques

- **ToolBarCalendar** (Ui::MainWindow *ui)
Constructeur.
- **~ToolBarCalendar** ()
- **QSlider * getSlider** ()
retourne le slider

Description détaillée

ToolBarCalendar est la barre d'outil de la vue Calendrier. Elle affiche le slider permettant de zoomer sur la vue calendrier.

Documentation des constructeurs et destructeur

ToolBarCalendar::ToolBarCalendar (Ui::MainWindow * *ui*) [**explicit**]

Constructeur.

Paramètres:

<i>ui</i>	: la MainWindow
-----------	------------------------

ToolBarCalendar::~~ToolBarCalendar ()

Documentation des fonctions membres

QSlider * ToolBarCalendar::getSlider ()

retourne le slider

La documentation de cette classe a été générée à partir des fichiers suivants :

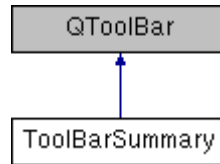
- Agenda/viewCalendar/toolbarcalendar.h
- Agenda/viewCalendar/toolbarcalendar.cpp

Référence de la classe **ToolBarSummary**

Classe représentant la tool bar (barre affichée au sommet de la vue)

```
#include <toolbarsummary.h>
```

Graphe d'héritage de **ToolBarSummary**:



Fonctions membres publiques

- **ToolBarSummary** (Ui::MainWindow *ui, VSummary *vSummary)
*Constructeur de la classe **ToolBarSummary**.*
- **~ToolBarSummary** ()
*Destructeur de la classe **VCourseFill**.*

Description détaillée

Classe représentant la tool bar (barre affichée au sommet de la vue)

Elle contient seulement le bouton d'ajout des cours et semestres.

Documentation des constructeurs et destructeur

ToolBarSummary::ToolBarSummary (Ui::MainWindow * *ui*, VSummary * *vSummary*) [**explicit**]

Constructeur de la classe **ToolBarSummary**.

Paramètres:

<i>ui</i>	: La fenêtre principale du programme
<i>vSummary</i>	: La fenêtre contenant le VCourseFill (vue résumé).

ToolBarSummary::~~ToolBarSummary ()

Destructeur de la classe **VCourseFill**.

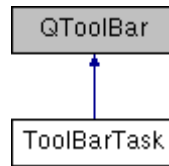
La documentation de cette classe a été générée à partir des fichiers suivants :

- Agenda/viewSummary/toolbarsummary.h
- Agenda/viewSummary/toolbarsummary.cpp

Référence de la classe **ToolBarTask**

```
#include <toolbartask.h>
```

Graphe d'héritage de **ToolBarTask**:



Connecteurs publics

- `void onAddtaskTriggered ()`
Clique sur le bouton d'ajout d'une tâche.
- `void onSortButtonDateClicked ()`
Clique sur le bouton de tri par date.
- `void onSortButtonCourseClicked ()`
Clique sur le bouton de tri par cours.
- `void onFilterButtonDoneClicked ()`
Clique sur le bouton de filtre des tâches terminées.
- `void onFilterButtonToDoClicked ()`
Clique sur le bouton de filtre des tâches à faire.

Signaux

- `void newTask ()`
Signal lorsqu'une nouvelle tâche doit être créée.

Fonctions membres publiques

- `ToolBarTask (Ui::MainWindow *ui)`
*Constructeur Constructeur de la classe **ToolBarTask**.*
- `~ToolBarTask ()`
*Destructeur Destructeur de la classe **ToolBarTask**.*

Description détaillée

Widget de la vue tâche permettant la création d'une nouvelle tâche

Documentation des constructeurs et destructeur

ToolBarTask::ToolBarTask (Ui::MainWindow * ui)[explicit]

Constructeur Constructeur de la classe **ToolBarTask**.

Paramètres:

<i>ui</i>	:Fenêtre principale
-----------	---------------------

ToolBarTask::~~ToolBarTask ()

Destructeur Destructeur de le classe **ToolBarTask**.

Documentation des fonctions membres**void ToolBarTask::newTask () [signal]**

Signal lorsqu'une nouvelle tâche doit être créée.

void ToolBarTask::onAddtaskTriggered () [slot]

Clique sur le bouton d'ajout d'une tâche.

void ToolBarTask::onFilterButtonDoneClicked () [slot]

Clique sur le bouton de filtre des tâches terminées.

void ToolBarTask::onFilterButtonToDoClicked () [slot]

Clique sur le bouton de filtre des tâches à faire.

void ToolBarTask::onSortButtonCourseClicked () [slot]

Clique sur le bouton de tri par cours.

void ToolBarTask::onSortButtonDateClicked () [slot]

Clique sur le bouton de tri par date.

La documentation de cette classe a été générée à partir des fichiers suivants :

- Agenda/viewTask/**toolbartask.h**
- Agenda/viewTask/**toolbartask.cpp**

Référence de la classe tooltipNote

```
#include <tooltipnote.h>
```

Fonctions membres publiques

- `tooltipNote ()`
- `~tooltipNote ()`

Documentation des constructeurs et destructeur

`tooltipNote::tooltipNote ()`

`tooltipNote::~~tooltipNote ()`

La documentation de cette classe a été générée à partir des fichiers suivants :

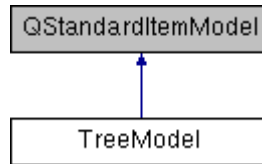
- `Agenda/viewTask/tooltipnote.h`
- `Agenda/viewTask/tooltipnote.cpp`

Référence de la classe **TreeModel**

Classe héritée de **QStandardItemModel** et qui correspond au model pour le **TreeView**.

```
#include <treemodel.h>
```

Graphe d'héritage de **TreeModel**:



Fonctions membres publiques

- **TreeModel** (const **QString** &**data**, **QObject** ***parent**=0)
*Constructeur de la classe **TreeModel**.*
- **TreeModel** (int **rows**, int **columns**, **QObject** ***parent**=0)
*Constructeur de la classe **TreeModel**.*
- **QVariant data** (const **QModelIndex** &**index**, int **role**) const
Retourne la donnée (stockée dans le modèle) à l'index "index" en fonction du rôle.
- **QModelIndexList getPersistentIndexList** ()
Retourne la liste des indices persistants.

Description détaillée

Classe héritée de **QStandardItemModel** et qui correspond au model pour le **TreeView**.

Correspond au model pour un **TreeView**.

Documentation des constructeurs et destructeur

TreeModel::TreeModel (const **QString** & **data**, **QObject** * **parent** = 0)

Constructeur de la classe **TreeModel**.

Paramètres:

<i>data</i>	: La valeur à affecter.
<i>parent</i>	: Le parent du TreeModel

TreeModel::TreeModel (int **rows**, int **columns**, **QObject** * **parent** = 0)

Constructeur de la classe **TreeModel**.

Paramètres:

<i>rows</i>	: Le nombre de lignes.
<i>columns</i>	: Le nombre de colonnes

<i>parent</i>	: Le parent du TreeModel
---------------	---------------------------------

Documentation des fonctions membres

QVariant TreeModel::data (const QModelIndex & *index*, int *role*) const

Retourne la donnée (stockée dans le modèle) à l'index "*index*" en fonction du rôle.

Paramètres:

<i>rows</i>	: Le nombre de lignes.
<i>columns</i>	: Le nombre de colonnes
<i>parent</i>	: Le parent du TreeModel

QModelIndexList TreeModel::getPersistentIndexList () [inline]

Retourne la liste des indices persistants.

La méthode `persistentIndexList()` dans `QStandardItemModel` est privée, il fallait la rendre accessible de l'exterieur.

Renvoie:

La liste des indices persistants.

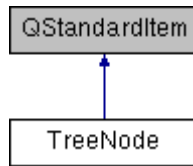
La documentation de cette classe a été générée à partir des fichiers suivants :

- Agenda/viewSummary/**treemodel.h**
- Agenda/viewSummary/**treemodel.cpp**

Référence de la classe `TreeNode`

```
#include <treenode.h>
```

Graphe d'héritage de `TreeNode`:



Fonctions membres publiques

- `TreeNode` (`QString` *text*, `int` *row*)
- `QVariant data` (`int` *role*) `const`

Documentation des constructeurs et destructeur

```
TreeNode::TreeNode (QString text, int row) [inline]
```

Documentation des fonctions membres

```
QVariant TreeNode::data (int role) const [inline]
```

La documentation de cette classe a été générée à partir du fichier suivant :

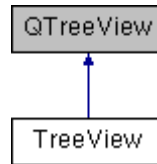
- `Agenda/viewSummary/treenode.h`

Référence de la classe **TreeView**

Classe héritée de **QTreeView** représentant un **TreeView**.

```
#include <treeview.h>
```

Graphe d'héritage de **TreeView**:



Fonctions membres publiques

- **TreeView** (**QWidget** *parent=0, **VSummary** *vSummary=0)
*Constructeur de la classe **TreeView**.*
- **~TreeView** ()
*Destructeur de la classe **TreeView**.*
- void **setSqlTableModel** (**QSqlTableModel** *sqlTableModel)
*Setter pour le champs *sqlTableModel*.*

Connecteurs protégés

- void **currentChanged** (const **QModelIndex** ¤t, const **QModelIndex** &previous)

Description détaillée

Classe héritée de **QTreeView** représentant un **TreeView**.

Représente l'affichage des jours et semestres dans la vue résumé.

Documentation des constructeurs et destructeur

```
TreeView::TreeView (QWidget * parent = 0, VSummary * vSummary = 0)[inline], [explicit]
```

Constructeur de la classe **TreeView**.

Paramètres:

<i>parent</i>	: Le widget parent.
<i>summary</i>	: Fenêtre contenant le TreeView (vue résumé).

```
TreeView::~TreeView () [inline]
```

Destructeur de la classe **TreeView**.

Documentation des fonctions membres

void TreeView::currentChanged (const QModelIndex & *current*, const QModelIndex & *previous*) [*inline*], [*protected*], [*slot*]

void TreeView::setSqlTableModel (QSqlTableModel * *sqlTableModel*) [*inline*]

Setter pour le champs sqlTableModel.

Paramètres:

<i>sqlTableModel</i>	: Nouvelle valeur
----------------------	-------------------

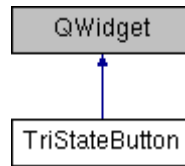
La documentation de cette classe a été générée à partir du fichier suivant :

- Agenda/viewSummary/treeview.h

Référence de la classe TriStateButton

```
#include <tristatebutton.h>
```

Graphe d'héritage de TriStateButton:



Fonctions membres publiques

- **TriStateButton** (QString label, QString text1, QString text2, QString text3)
*Constructeur Constructeur de la classe **TriStateButton**.*
- int **getChoice** ()
Récupère le bouton choisi.
- void **setChoice** (int value)
Sélection d'un bouton.
- **~TriStateButton** ()
*Destructeur Destructeur de la classe **TriStateButton**.*

Propriétés

- int **choice**

Description détaillée

Classe représentant un widget avec trois boutons dont seul un choix entre les trois boutons est permis

Documentation des constructeurs et destructeur

TriStateButton::TriStateButton (QString *label*, QString *text1*, QString *text2*, QString *text3*) [**explicit**]

Constructeur Constructeur de la classe **TriStateButton**.

Paramètres:

<i>label</i>	: nom du widget
<i>text1</i>	: texte du premier bouton
<i>text2</i>	: texte du deuxième bouton
<i>text3</i>	: texte du troisième bouton

TriStateButton::~~TriStateButton () [**inline**]

Destructeur Destructeur de la classe **TriStateButton**.

Documentation des fonctions membres

int TriStateButton::getChoice ()

Récupère le bouton choisi.

Renvoie:

la valeur du bouton choisi

void TriStateButton::setChoice (int *value*)

Sélection d'un bouton.

Paramètres:

<i>value</i>	: valeur du bouton chois
--------------	--------------------------

Documentation des propriétés

int TriStateButton::choice [read], [write]

La documentation de cette classe a été générée à partir des fichiers suivants :

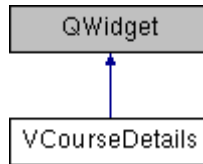
- Agenda/viewTask/**tristatebutton.h**
- Agenda/viewTask/**tristatebutton.cpp**

Référence de la classe VCourseDetails

Classe représentant l'affichage des détails des cours.

```
#include <vcoursedetails.h>
```

Graphe d'héritage de VCourseDetails:



Fonctions membres publiques

- **VCourseDetails** (QWidget *parent=0, VSummary *vSummary=0)
Constructeur de la classe VCourseDetails.
- **~VCourseDetails** ()
Destructeur de la classe VCourseFill.
- void **prepare** (QSqlRecord course)
Prépare les informations à afficher.
- void **refreshLessons** ()
Rafrâchit la liste des leçons.

Description détaillée

Classe représentant l'affichage des détails des cours.

Représente les 3 onglets du bas lors de la sélection d'un jour.

Documentation des constructeurs et destructeur

VCourseDetails::VCourseDetails (QWidget * *parent* = 0, VSummary * *vSummary* = 0) [explicit]

Constructeur de la classe **VCourseDetails**.

Paramètres:

<i>parent</i>	: Le widget parent.
<i>summary</i>	: Fenêtre contenant le VCourseDetails (vue résumé).

VCourseDetails::~VCourseDetails ()

Destructeur de la classe **VCourseFill**.

Documentation des fonctions membres

void VCourseDetails::prepare (QSqlRecord *course*)

Prépare les informations à afficher.

Rempli les différents onglet en fonction du cours passé en paramètre.

Paramètres:

<i>course</i>	: Le sql record correspondant au cours sélectionné.
---------------	---

void VCourseDetails::refreshLessons ()

Rafraîchit la liste des leçons.

Rafraîchit la liste des leçons lors de l'ajout/modification d'une leçon.

La documentation de cette classe a été générée à partir des fichiers suivants :

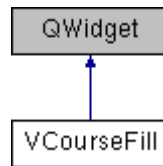
- Agenda/viewSummary/vcoursedetails.h
- Agenda/viewSummary/vcoursedetails.cpp

Référence de la classe VCourseFill

Classe représentant la zone de modification des données d'un cours.

```
#include <vcoursefill.h>
```

Graphe d'héritage de VCourseFill:



Fonctions membres publiques

- **VCourseFill** (QWidget *parent=0, VSummary *vSummary=0)
Constructeur de la classe VCourseFill.
- **~VCourseFill** ()
Destructeur de la classe VCourseFill.
- void **setFieldsValue** (int id, QString name, QString description, float threshold, float weighting, QString color, int semesterId, int moduleId)
Rempli les champs de l'interface.
- void **setEditionMode** (bool value)
Modifie l'état de la vue (ajout ou édition).

Description détaillée

Classe représentant la zone de modification des données d'un cours.

La classe affiche le cours sélectionné dans la SideBar et permet également de créer ainsi que de modifier un cours.

Documentation des constructeurs et destructeur

VCourseFill::VCourseFill (QWidget * *parent* = 0, VSummary * *vSummary* = 0) [explicit]

Constructeur de la classe **VCourseFill**.

Paramètres:

<i>parent</i>	: Le widget parent.
<i>summary</i>	: Fenêtre contenant le VCourseFill (vue résumé).

VCourseFill::~VCourseFill ()

Destructeur de la classe **VCourseFill**.

Documentation des fonctions membres

void VCourseFill::setEditionMode (bool *value*)

Modifie l'état de la vue (ajout ou édition).

Paramètres:

<i>value</i>	: Est-ce qu'on est en modification (true) ou en ajout (false).
--------------	--

void VCourseFill::setFieldsValue (int *id*, QString *name*, QString *description*, float *threshold*, float *weighting*, QString *color*, int *semesterId*, int *moduleId*)

Rempli les champs de l'interface.

Paramètres:

<i>id</i>	: Identifiant du cours
<i>name</i>	: Nom du cours
<i>description</i>	: Description du cours.
<i>threshold</i>	: Seuil du cours.
<i>weighting</i>	: Poids du cours.
<i>semesterId</i>	: Identifiant du semestre correspondant au cours.
<i>moduleId</i>	: Identifiant du module correspondant au cours.

La documentation de cette classe a été générée à partir des fichiers suivants :

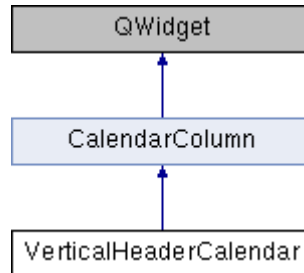
- Agenda/viewSummary/vcoursefill.h
- Agenda/viewSummary/vcoursefill.cpp

Référence de la classe VerticalHeaderCalendar

HeaderVertical du Calendrier.

```
#include <verticalheadercalendar.h>
```

Graphe d'héritage de VerticalHeaderCalendar:



Fonctions membres publiques

- **VerticalHeaderCalendar** (QWidget *parent=0)
Constructeur.
- **~VerticalHeaderCalendar** ()

Membres hérités additionnels

Description détaillée

HeaderVertical du Calendrier.

Affiche la première colonne de la grille du calendrier contenant une heure dans chaque ligne

Documentation des constructeurs et destructeur

VerticalHeaderCalendar::VerticalHeaderCalendar (QWidget * *parent* = 0)[explicit]

Constructeur.

Paramètres:

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

VerticalHeaderCalendar::~~VerticalHeaderCalendar ()

La documentation de cette classe a été générée à partir des fichiers suivants :

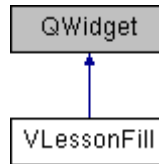
- Agenda/viewCalendar/verticalheadercalendar.h
- Agenda/viewCalendar/verticalheadercalendar.cpp

Référence de la classe VLessonFill

Classe représentant la zone de modification des données d'une leçon.

```
#include <vlessonfill.h>
```

Graphe d'héritage de VLessonFill:



Fonctions membres publiques

- **VLessonFill** (QWidget *parent=0, int courseId=0, VSummary *vSummary=0)
Constructeur de la classe VLessonFill.
- **~VLessonFill** ()
Destructeur de la classe VLessonFill.
- void **setFieldsValue** (int id, QString name, QString startHour, QString endHour, int dayOfWeek, int courseId)
Rempli les champs de l'interface.
- void **setEditionMode** (bool value)
Modifie l'état de la vue (ajout ou édition).

Description détaillée

Classe représentant la zone de modification des données d'une leçon.

La classe affiche la leçon courante et permet la modification de celle-ci ou l'ajout d'une nouvelle leçon.

Documentation des constructeurs et destructeur

VLessonFill::VLessonFill (QWidget * *parent* = 0, int *courseId* = 0, VSummary * *vSummary* = 0)[**explicit**]

Constructeur de la classe **VLessonFill**.

Paramètres:

<i>parent</i>	: Le widget parent.
<i>courseId</i>	: L'id du cours qui contient la leçon en cours.
<i>summary</i>	: Fenêtre contenant le VModuleFill (vue résumé).

VLessonFill::~VLessonFill ()

Destructeur de la classe **VLessonFill**.

Documentation des fonctions membres

void VLessonFill::setEditionMode (bool *value*)

Modifie l'état de la vue (ajout ou édition).

Paramètres:

<i>value</i>	: Est-ce qu'on est en modification (true) ou en ajout (false).
--------------	--

void VLessonFill::setFieldsValue (int *id*, QString *name*, QString *startHour*, QString *endHour*, int *dayOfWeek*, int *courseId*)

Rempli les champs de l'interface.

Paramètres:

<i>id</i>	: Identifiant de la leçon
<i>name</i>	: Nom de la leçon
<i>startHour</i>	: Heure de début de la leçon
<i>endHour</i>	: Heure de fin de la leçon
<i>dayOfWeek</i>	: Indice du jour de la semaine de la leçon
<i>courseId</i>	: Identifiant du cours correspondant à la leçon

La documentation de cette classe a été générée à partir des fichiers suivants :

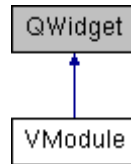
- Agenda/viewSummary/vlessonfill.h
- Agenda/viewSummary/vlessonfill.cpp

Référence de la classe VModule

Classe représentant l'affichage des différents modules.

```
#include <vmodule.h>
```

Graphe d'héritage de VModule:



Fonctions membres publiques

- **VModule** (QWidget *parent=0, VSummary *vSummary=0)
Constructeur de la classe VModule.
- **~VModule** ()
Destructeur de la classe VModule.

Description détaillée

Classe représentant l'affichage des différents modules.

La classe affiche la liste des modules (dans des **VModuleFill**).

Documentation des constructeurs et destructeur

VModule::VModule (QWidget * *parent* = 0, VSummary * *vSummary* = 0)[**explicit**]

Constructeur de la classe **VModule**.

Paramètres:

<i>parent</i>	: Le widget parent.
<i>summary</i>	: Fenêtre contenant le VModule (vue résumé).

VModule::~~VModule ()

Destructeur de la classe **VModule**.

La documentation de cette classe a été générée à partir des fichiers suivants :

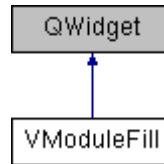
- Agenda/viewSummary/vmodule.h
- Agenda/viewSummary/vmodule.cpp

Référence de la classe VModuleFill

Classe représentant la zone de modification des données d'un module.

```
#include <vmodulefill.h>
```

Graphe d'héritage de VModuleFill:



Fonctions membres publiques

- **VModuleFill** (QWidget *parent=0, VSummary *vSummary=0)
Constructeur de la classe VModuleFill.
- **~VModuleFill** ()
Destructeur de la classe VModuleFill.
- void **setFieldsValue** (unsigned int id, QString name, QString description, float threshold)
Rempli les champs de l'interface.
- void **setEditionMode** (bool value)
Modifie l'état de la vue (ajout ou édition).

Description détaillée

Classe représentant la zone de modification des données d'un module.

La classe affiche le module courant et permet la modification de celui-ci ou l'ajout d'un nouveau module.

Documentation des constructeurs et destructeur

VModuleFill::VModuleFill (QWidget * *parent* = 0, VSummary * *vSummary* = 0) [**explicit**]

Constructeur de la classe **VModuleFill**.

Paramètres:

<i>parent</i>	: Le widget parent.
<i>summary</i>	: Fenêtre contenant le VModuleFill (vue résumé).

VModuleFill::~~VModuleFill ()

Destructeur de la classe **VModuleFill**.

Documentation des fonctions membres

void VModuleFill::setEditionMode (bool *value*)

Modifie l'état de la vue (ajout ou édition).

Paramètres:

<i>value</i>	: Est-ce qu'on est en modification (true) ou en ajout (false).
--------------	--

void VModuleFill::setFieldsValue (unsigned int *id*, QString *name*, QString *description*, float *threshold*)

Rempli les champs de l'interface.

Paramètres:

<i>id</i>	: Identifiant du module
<i>name</i>	: Nom du module
<i>description</i>	: Description du module
<i>threshold</i>	: Seuil du module

La documentation de cette classe a été générée à partir des fichiers suivants :

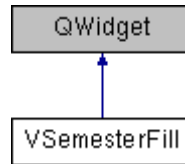
- Agenda/viewSummary/vmodulefill.h
- Agenda/viewSummary/vmodulefill.cpp

Référence de la classe VSemesterFill

Classe représentant la zone de modification des données d'un semestre.

```
#include <vsemesterfill.h>
```

Graphe d'héritage de VSemesterFill:



Fonctions membres publiques

- **VSemesterFill** (QWidget *parent=0, VSummary *vSummary=0)
Constructeur de la classe VSemestreFill.
- **~VSemesterFill** ()
Destructeur de la classe VSemestreFill.
- void **setFieldsValue** (int id, QString name, QDate startDate, QDate endDate)
Rempli les champs de l'interface.
- void **setEditionMode** (bool value)
Modifie l'état de la vue (ajout ou édition).

Description détaillée

Classe représentant la zone de modification des données d'un semestre.

La classe affiche le semestre sélectionné dans la SideBar et permet également de créer ainsi que de modifier un semestre

Documentation des constructeurs et destructeur

VSemesterFill::VSemesterFill (QWidget * *parent* = 0, VSummary * *vSummary* = 0)[explicit]

Constructeur de la classe VSemestreFill.

Paramètres:

<i>parent</i>	: Le widget parent.
<i>summary</i>	: Fenêtre contenant le VSemesterFill (vue résumé).

VSemesterFill::~VSemesterFill ()

Destructeur de la classe VSemestreFill.

Documentation des fonctions membres

void VSemesterFill::setEditionMode (bool *value*)

Modifie l'état de la vue (ajout ou édition).

Paramètres:

<i>value</i>	: Est-ce qu'on est en modification (true) ou en ajout (false).
--------------	--

void VSemesterFill::setFieldsValue (int *id*, QString *name*, QDate *startDate*, QDate *endDate*)

Rempli les champs de l'interface.

Paramètres:

<i>id</i>	: Identifiant du semestre
<i>name</i>	: Nom du semestre
<i>startDate</i>	: Date de début du semestre
<i>endDate</i>	: Date de fin du semestre

La documentation de cette classe a été générée à partir des fichiers suivants :

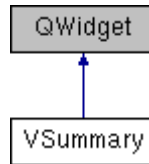
- Agenda/viewSummary/vsemesterfill.h
- Agenda/viewSummary/vsemesterfill.cpp

Référence de la classe VSummary

Classe représentant l'affichage lors du clique sur l'ongle résumé (vue résumé).

```
#include <vsummary.h>
```

Graphe d'héritage de VSummary:



Fonctions membres publiques

- **VSummary (MainWindow *mainWindow, QWidget *parent=0)**
Constructeur de la classe VSummary.
- **~VSummary ()**
Destructeur de la classe VSummary.
- **void switchViews (bool isSemesterView)**
Change la vue entre cours et semestre.
- **void refreshModules ()**
Rafraichissement de l'affichage des modules.
- **void refreshLessons ()**
Rafraichissement de l'affichage des leçons.
- **void refreshSideBar ()**
Rafraichissement de la side bar (TreeView).
- **void emptyVSemesterFill ()**
Réinitialise et vide la vue d'ajout/modification des semestres et le layout correspondant.
- **void emptyVCourseFill ()**
Réinitialise et vide la vue d'ajout/modification des cours et le layout correspondant.
- **void addCourse ()**
Modifie la vue cours pour l'ajout d'un cours et modifie le layout correspondant.
- **void addSemester ()**
Modifie la vue cours pour l'ajout d'un cours et le layout correspondant.

Attributs publics

- **VSemesterFill * vSemesterFill**
- **VCourseFill * vCourseFill**
- **VCourseDetails * vCourseDetails**
- **SideBarTreeView * ttv**

Description détaillée

Classe représentant l'affichage lors du clique sur l'ongle résumé (vue résumé).

Elle va afficher les informations correspondant à la vue résumé.

- La liste des semestres, des cours et des modules
- La possibilité de modifier divers informations (cours, semestres, modules et leçons)
- La présentation des informations pour les cours (notes ...)

Documentation des constructeurs et destructeur

VSummary::VSummary (MainWindow * *mainWindow*, QWidget * *parent* = 0) [explicit]

Constructeur de la classe **VSummary**.

VSummary::~~VSummary ()

Destructeur de la classe **VSummary**.

Documentation des fonctions membres

void VSummary::addCourse ()

Modifie la vue cours pour l'ajout d'un cours et modifie le layout correspondant.

void VSummary::addSemester ()

Modifie la vue cours pour l'ajout d'un cours et le layout correspondant.

void VSummary::emptyVCourseFill ()

Réinitialise et vide la vue d'ajout/modification des cours et le layout correspondant.

void VSummary::emptyVSemesterFill ()

Réinitialise et vide la vue d'ajout/modification des semestres et le layout correspondant.

void VSummary::refreshLessons ()

Rafraichissement de l'affichage des leçons.

Rafraichissement de l'affichage des leçon d'un cours dans la fenêtre résumé. Lors de l'ajout/modification d'une leçon

void VSummary::refreshModules ()

Rafraichissement de l'affichage des modules.

Rafraichissement de l'affichage des modules dans la fenêtre résumé. Lors de l'ajout/modification d'un module.

void VSummary::refreshSideBar ()

Rafraichissement de la side bar (**TreeView**).

Rafraichissement de la side bar (**TreeView**). Lors de l'ajout/modification d'un cours/semestre.

void VSummary::switchViews (bool isSemesterView)

Change la vue entre cours et semestre.

Lors d'un clique sur un cours ou un semestre on "switch" sur la vue correspondante (cours si clique sur un cours et semestre si clique sur un semestre).

Paramètres:

<i>isSemesterView</i>	: Est-ce qu'on affiche un cours (false) ou un semestre (true).
-----------------------	--

Documentation des données membres

SideBarTreeView* VSummary::ttv

VCourseDetails* VSummary::vCourseDetails

VCourseFill* VSummary::vCourseFill

VSemesterFill* VSummary::vSemesterFill

La documentation de cette classe a été générée à partir des fichiers suivants :

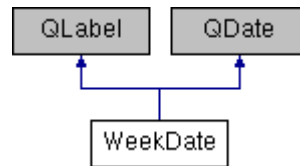
- Agenda/viewSummary/vsummary.h
- Agenda/viewSummary/vsummary.cpp

Référence de la classe WeekDate

WeekDate un mélange de QLabel et de QDate.

```
#include <weekdate.h>
```

Graphe d'héritage de WeekDate:



Signaux

- void **daySelected** (QDate)
signal que le widget à été sélectionné par un clic

Fonctions membres publiques

- **WeekDate** (QWidget *parent=0)
Constructeur.
- **~WeekDate** ()
- void **setDate** (QDate date)
setDate: modification de la date

Fonctions membres protégées

- void **mousePressEvent** (QMouseEvent *)
redéfini afin de capturer les événements de clic de souris sur ce widget

Description détaillée

WeekDate un mélange de QLabel et de QDate.

Il peut s'afficher en tant que QLabel dans l'entête du Calendrier et peut être vue en tant que QDate afin que la colonne du calendrier correspondante sache quelle date doit être affichée.

Documentation des constructeurs et destructeur

WeekDate::WeekDate (QWidget * *parent* = 0)

Constructeur.

Paramètres:

<i>parent</i>	: Widget parent
---------------	-----------------

WeekDate::~WeekDate ()

Documentation des fonctions membres

void WeekDate::daySelected (QDate) [signal]

signal que le widget à été selectionné par un clic

void WeekDate::mousePressEvent (QMouseEvent * event) [protected]

redéfini afin de capturer les événements de clic de souris sur ce widget

void WeekDate::setDate (QDate date)

setDate: modification de la date

La documentation de cette classe a été générée à partir des fichiers suivants :

- Agenda/viewCalendar/**weekdate.h**
- Agenda/viewCalendar/**weekdate.cpp**

Documentation des fichiers

Référence du fichier Agenda/mainwindow.cpp

```
#include "mainwindow.h"  
#include "ui_mainwindow.h"  
#include <QLabel>  
#include <QListView>  
#include <QListWidget>  
#include <QPushButton>  
#include <QStringListModel>  
#include <QPropertyAnimation>  
#include "viewSummary/summaryutils.h"
```

Référence du fichier Agenda/mainwindow.h

```
#include <QMainWindow>
#include <QDirModel>
#include <QTreeView>
#include <QWidget>
#include <QDebug>
#include <QToolButton>
#include <QStackedWidget>
#include <QFile>
#include "viewCalendar/calendarwidget.h"
#include "viewCalendar/toolbarcalendar.h"
#include "viewCalendar/sidebarcalendar.h"
#include "viewSummary/vsummary.h"
#include "viewSummary/sidebarsummary.h"
#include "viewSummary/toolbarsummary.h"
#include "viewTask/sidebartask.h"
#include "viewTask/taskdisplay.h"
#include "viewTask/toolbartask.h"
```

Classes

- class **MainWindow**

Espaces de nommage

- **Ui**

Référence du fichier Agenda/models/course.cpp

```
#include "course.h"
```

Référence du fichier Agenda/models/course.h

```
#include <QString>
#include "QDate"
#include <QObject>
#include <QSqlRecord>
#include <QSqlRelationalTableModel>
#include "sqlconnection.h"
```

Classes

- class **Course**

Modèle pour les cours.

Référence du fichier Agenda/models/lesson.cpp

```
#include "lesson.h"  
#include "course.h"
```

Référence du fichier Agenda/models/lesson.h

```
#include <QString>
#include <QTime>
#include <QObject>
#include <QSqlRecord>
#include <QSqlRelationalTableModel>
#include "sqlconnection.h"
```

Classes

- class **Lesson**

Modèle pour les leçons.

Référence du fichier Agenda/models/task.cpp

```
#include "task.h"  
#include "sqlconnection.h"  
#include <QMessageBox>  
#include "viewTask/taskdisplay.h"
```

Référence du fichier Agenda/models/task.h

```
#include <QObject>
#include <QDate>
#include <QString>
#include <QLayout>
#include <QSqlRecord>
#include <QSqlRelationalTableModel>
```

Classes

- class **Task**

Modèle pour les tâches.

Référence du fichier Agenda/tasklistitem.cpp

```
#include "tasklistitem.h"
```

Référence du fichier Agenda/tasklistitem.h

```
#include <QWidget>
#include <QPushButton>
#include <QCheckBox>
#include <QLabel>
#include <QDataWidgetMapper>
#include <QHBoxLayout>
#include <QVBoxLayout>
#include <QSqlRelationalTableModel>
#include "priority.h"
#include "sidebartask.h"
#include "termdate.h"
#include "gradedialogbox.h"
```

Classes

- class **TaskListItem**

Référence du fichier Agenda/viewCalendar/calendarcolumn.cpp

```
#include "calendarcolumn.h"
```

Référence du fichier Agenda/viewCalendar/calendarcolumn.h

```
#include <QWidget>
#include <QGridLayout>
#include "hourcal.h"
```

Classes

- class **CalendarColumn**
Colonne de la grille du calendrier.

Référence du fichier Agenda/viewCalendar/calendarwidget.cpp

```
#include "calendarwidget.h"  
#include "mainwindow.h"
```

Référence du fichier Agenda/viewCalendar/calendarwidget.h

```
#include <QLabel>
#include <QWidget>
#include <QScrollBar>
#include "hourcal.h"
#include "weekdate.h"
#include "columnnday.h"
#include "models/lesson.h"
#include "ui_calendarwidget.h"
#include "verticalheadercalendar.h"
#include "viewCalendar/minitasklist.h"
```

Classes

- class **CalendarWidget**

La class CalendarWidget gère l'affichage des leçons et des tâches dans un calendrier. Ces informations sont affichées par semaine. Il est possible de naviguer entre les semaines en choisissant un jour depuis le mini calendrier de la sidebar. Espaces de nommage

- Ui

Référence du fichier Agenda/viewCalendar/columnnday.cpp

```
#include "columnnday.h"  
#include <QSqlRelationalTableModel>  
#include <QSqlRecord>  
#include "hourcal.h"
```

Référence du fichier Agenda/viewCalendar/columnnday.h

```
#include <QLabel>
#include <QWidget>
#include <QVBoxLayout>
#include <QPainter>
#include <QPaintEvent>
#include <QDebug>
#include "lessonrect.h"
#include "models/lesson.h"
#include "calendarcolumn.h"
```

Classes

- class **ColumnDay**

Colonne d'un jour dans le calendrier.

Référence du fichier Agenda/viewCalendar/customcalendarwidget.cpp

```
#include "customcalendarwidget.h"  
#include <QDebug>  
#include <QFile>  
#include <QTextFormat>
```

Référence du fichier Agenda/viewCalendar/customcalendarwidget.h

```
#include <QCalendarWidget>
#include <QPainter>
#include <QPoint>
```

Classes

- class **CustomCalendarWidget**

MiniCalendar modifié pour une amélioration du design La navigation entre les dates est géré manuellement pour plus de fonctionnalités.

Référence du fichier Agenda/viewCalendar/daywidget.cpp

```
#include "daywidget.h"
```

Référence du fichier Agenda/viewCalendar/daywidget.h

```
#include <QLabel>
#include <QWidget>
#include <QVBoxLayout>
```

Classes

- class **DayWidget**

Classe utilitaire permettant d'afficher le nom d'un jour ainsi sa date. Le numéro du jour est affiché en grand en dessous du nom du jour.

Référence du fichier Agenda/viewCalendar/hourcal.h

```
#include <QLabel>
#include <QPaintEvent>
#include <QPainter>
#include <QWidget>
```

Classes

- class **HourCal**

HourCal remplit chaque ligne de d'une colonne. Les lignes représentent des heures.

Référence du fichier Agenda/viewCalendar/lessonrect.cpp

```
#include "lessonrect.h"
```

Référence du fichier Agenda/viewCalendar/lessonrect.h

```
#include <QObject>
#include <QSqlRecord>
#include <QVariant>
#include <QTime>
#include "models/lesson.h"
```

Classes

- class **LessonRect**

***LessonRect** représente une leçon du calendrier.*

Référence du fichier Agenda/viewCalendar/minitasklist.cpp

```
#include "minitasklist.h"
```


Référence du fichier Agenda/viewCalendar/minitasklist.h

```
#include <QWidget>
#include <QLabel>
#include <QList>
#include <QSqlRelationalTableModel>
#include "viewTask/tasklistitem.h"
```

Classes

- class **MiniTaskList**

***MiniTaskList** est une vue personnalisée d'une model **Task** Elle affiche une liste allégée de tâches.*

Référence du fichier Agenda/viewCalendar/sidebarcalendar.cpp

```
#include "sidebarcalendar.h"  
#include <QDebug>
```

Référence du fichier Agenda/viewCalendar/sidebarcalendar.h

```
#include <QWidget>
#include "customcalendarwidget.h"
#include <QLabel>
#include <QVBoxLayout>
#include <QPushButton>
#include <QStackedWidget>
#include <QToolButton>
#include "daywidget.h"
#include "viewCalendar/minitasklist.h"
```

Classes

- class **SideBarCalendar**
Sidebar de la vue Calendrier.

Référence du fichier Agenda/viewCalendar/toolbarcalendar.cpp

```
#include "toolbarcalendar.h"
```

Référence du fichier Agenda/viewCalendar/toolbarcalendar.h

```
#include <QObject>
#include <QToolBar>
#include <QLabel>
#include "utils/toolbutton.h"
#include "utils/spacer.h"
#include "models/task.h"
#include <QSlider>
```

Classes

- class **ToolBarCalendar**

ToolBarCalendar est la barre d'outil de la vue Calendrier. Elle affiche le slider permettant de zoomer sur la vue calendrier. Espaces de nommage

- Ui

Référence du fichier Agenda/viewCalendar/verticalheadercalendar.cpp

```
#include "verticalheadercalendar.h"
```

Référence du fichier Agenda/viewCalendar/verticalheadercalendar.h

```
#include "calendarcolumn.h"
```

Classes

- class **VerticalHeaderCalendar**
HeaderVertical du Calendrier.

Référence du fichier Agenda/viewCalendar/weekdate.cpp

```
#include "weekdate.h"
```


Référence du fichier Agenda/viewCalendar/weekdate.h

```
#include <QLabel>
#include <QDate>
#include <QMouseEvent>
```

Classes

- class **WeekDate**

***WeekDate** un mélange de *QLabel* et de *QDate*.*

Référence du fichier Agenda/viewSummary/average.h

```
#include <QString>
```

Classes

- struct **Average**

Structure représentant une moyenne (nom et valeur)

Référence du fichier Agenda/viewSummary/custommodel.h

```
#include <QAbstractTableModel>
#include <QSqlTableModel>
#include <QDebug>
```

Classes

- class **CustomModel**

Référence du fichier Agenda/viewSummary/sidebarsummary.cpp

```
#include "sidebarsummary.h"  
#include "ui_sidebarsummary.h"
```

Référence du fichier Agenda/viewSummary/sidebarsummary.h

```
#include <QVBoxLayout>
#include <QWidget>
```

Classes

- class **SideBarSummary**

Classe représentant la side bar (barre à droite) dans la vue résumé Espaces de nommage

- Ui

Référence du fichier Agenda/viewSummary/summaryutils.h

```
#include <cmath>
#include <QLayout>
#include <QLayoutItem>
#include <QMap>
#include <QMessageBox>
#include <QSqlQuery>
#include <QString>
#include <QTableWidget>
#include <QWidget>
#include "average.h"
```

Classes

- class **SummaryUtils**
Classe utilitaire.

Référence du fichier Agenda/viewSummary/testtreeview.cpp

```
#include "testtreeview.h"
```

Référence du fichier Agenda/viewSummary/testtreeview.h

```
#include <QSqlTableModel>
#include <QStandardItemModel>
#include <QVBoxLayout>
#include <QWidget>
#include "sqlconnection.h"
#include "treemodel.h"
#include "treeview.h"
#include "vsummary.h"
```

Classes

- class **SideBarTreeView**

*Classe créant et contenant le **TreeView**.*

Référence du fichier Agenda/viewSummary/toolbarsummary.cpp

```
#include "toolbarsummary.h"
```

Référence du fichier Agenda/viewSummary/toolbarsummary.h

```
#include <QMenu>
#include <QToolBar>
#include "utils/toolbutton.h"
#include "vsummary.h"
```

Classes

- class **ToolBarSummary**

Classe représentant la tool bar (barre affiché au sommet de la vue) Espaces de nommage

- Ui

Référence du fichier Agenda/viewSummary/treemodel.cpp

```
#include "treemodel.h"
```

Référence du fichier Agenda/viewSummary/treemodel.h

```
#include <QStandardItemModel>
```

Classes

- class **TreeModel**

*Classe héritée de `QStandardItemModel` et qui correspond au model pour le **TreeView**.*

Référence du fichier Agenda/viewSummary/treenode.h

```
#include <QStandardItem>
```

Classes

- class **TreeNode**

Référence du fichier Agenda/viewSummary/treeview.h

```
#include <QSqlRecord>
#include <QSqlTableModel>
#include <QTreeView>
#include "vsummary.h"
```

Classes

- class **TreeView**

*Classe héritée de QTreeView représentant un **TreeView**.*

Référence du fichier Agenda/viewSummary/vcoursedetails.cpp

```
#include "vcoursedetails.h"  
#include "ui_vcoursedetails.h"  
#include "vsummary.h"
```

Référence du fichier Agenda/viewSummary/vcoursedetails.h

```
#include <cmath>
#include <QLabel>
#include <QLayout>
#include <QLayoutItem>
#include <QListWidget>
#include <QSqlQuery>
#include <QScrollArea>
#include <QSqlRecord>
#include <QTableWidget>
#include <QWidget>
#include "vlessonfill.h"
```

Classes

- class **VCourseDetails**

Classe représentant l'affichage des détails des cours. Espaces de nommage

- Ui

Référence du fichier Agenda/viewSummary/vcoursefill.cpp

```
#include "vcoursefill.h"  
#include "ui_vcoursefill.h"  
#include "vsummary.h"
```

Référence du fichier Agenda/viewSummary/vcoursefill.h

```
#include <QColorDialog>
#include <QComboBox>
#include <QSqlQuery>
#include <QSqlTableModel>
#include <QSqlRecord>
#include <QWidget>
#include "summaryutils.h"
```

Classes

- class **VCourseFill**

Classe représentant la zone de modification des données d'un cours. Espaces de nommage

- Ui

Référence du fichier Agenda/viewSummary/vlessonfill.cpp

```
#include "vlessonfill.h"  
#include "ui_vlessonfill.h"  
#include "vsummary.h"
```

Référence du fichier Agenda/viewSummary/vlessonfill.h

```
#include <QDate>
#include <QSqlQuery>
#include <QSqlRecord>
#include <QWidget>
#include "summaryutils.h"
```

Classes

- class **VLessonFill**

Classe représentant la zone de modification des données d'une leçon. Espaces de nommage

- **Ui**

Référence du fichier Agenda/viewSummary/vmodule.cpp

```
#include "vmodule.h"  
#include "ui_vmodule.h"
```

Référence du fichier Agenda/viewSummary/vmodule.h

```
#include <QLayout>
#include <QScrollArea>
#include <QWidget>
#include "vmodulefill.h"
```

Classes

- class **VModule**

Classe représentant l'affichage des différents modules. Espaces de nommage

- **Ui**

Référence du fichier Agenda/viewSummary/vmodulefill.cpp

```
#include "vmodulefill.h"  
#include "ui_vmodulefill.h"  
#include "vsummary.h"
```

Référence du fichier Agenda/viewSummary/vmodulefill.h

```
#include <QSqlQuery>
#include <QSqlRecord>
#include <QWidget>
#include "summaryutils.h"
```

Classes

- class **VModuleFill**

Classe représentant la zone de modification des données d'un module. Espaces de nommage

- Ui

Référence du fichier Agenda/viewSummary/vsemesterfill.cpp

```
#include "vsemesterfill.h"  
#include "ui_vsemesterfill.h"  
#include "vsummary.h"
```

Référence du fichier Agenda/viewSummary/vsemesterfill.h

```
#include <QWidget>
#include "summaryutils.h"
```

Classes

- class **VSemesterFill**

Classe représentant la zone de modification des données d'un semestre. Espaces de nommage

- **Ui**

Référence du fichier Agenda/viewSummary/vsummary.cpp

```
#include "vsummary.h"  
#include "ui_vsummary.h"  
#include <QtSql/QtSql>  
#include "vsemesterfill.h"  
#include "vcoursefill.h"  
#include "sqlconnection.h"  
#include "utils.h"  
#include "mainwindow.h"  
#include "testtreeview.h"
```

Référence du fichier Agenda/viewSummary/vsummary.h

```
#include <QWidget>
#include "vsemesterfill.h"
#include "vcoursefill.h"
#include "vcoursedetails.h"
#include "vmodule.h"
```

Classes

- class **VSummary**

Classe représentant l'affichage lors du clique sur l'ongle résumé (vue résumé).

Espaces de nommage

- **Ui**

Référence du fichier Agenda/viewTask/gradedialogbox.cpp

```
#include "gradedialogbox.h"
```

Référence du fichier Agenda/viewTask/gradedialogbox.h

```
#include <QDialog>
#include <QDoubleSpinBox>
#include <QPushButton>
#include <QFormLayout>
#include <QDebug>
```

Classes

- class **gradedialogbox**

Référence du fichier Agenda/viewTask/priority.h

```
#include <QLabel>
#include <QHBoxLayout>
#include <QWidget>
```

Classes

- class **Priority**

Référence du fichier Agenda/viewTask/sidebartask.cpp

```
#include "sidebartask.h"  
#include "ui_sidebartask.h"  
#include "models/task.h"  
#include "QDebug"  
#include <QSqlQuery>  
#include <QMessageBox>  
#include <QDialogButtonBox>
```


Référence du fichier Agenda/viewTask/sidebartask.h

```
#include <QWidget>
#include <QComboBox>
#include <QGroupBox>
#include <QRadioButton>
#include <QLineEdit>
#include <QFormLayout>
#include <QTextEdit>
#include <QCheckBox>
#include <QDateTimeEdit>
#include <QKeyEvent>
#include "sqlconnection.h"
#include <QSqlRelationalTableModel>
#include <QDataWidgetMapper>
#include <QItemSelectionModel>
#include <QSqlRelationalDelegate>
#include "viewTask/tristatebutton.h"
```

Classes

- class **SideBarTask**

Espaces de nommage

- **Ui**

Référence du fichier Agenda/viewTask/taskcheckbox.cpp

```
#include "taskcheckbox.h"  
#include "ui_taskcheckbox.h"  
#include "sidebartask.h"  
#include <QInputDialog>
```

Référence du fichier Agenda/viewTask/taskcheckbox.h

```
#include <QLabel>
#include <QDebug>
#include <QWidget>
#include <QCheckBox>
#include <QTextEdit>
#include <QSqlQuery>
#include <QHBoxLayout>
#include <QMouseEvent>
#include <QSignalMapper>
#include <QDateTimeEdit>
#include "sqlconnection.h"
#include <QDataWidgetMapper>
#include <QItemSelectionModel>
#include <QPushButton>
#include <QSqlRelationalDelegate>
#include <QSqlRelationalTableModel>
#include "models/task.h"
#include "termdate.h"
#include "priority.h"
#include "gradedialogbox.h"
```

Classes

- class **TaskCheckBox**

Espaces de nommage

- **Ui**

Référence du fichier Agenda/viewTask/taskdisplay.cpp

```
#include "taskdisplay.h"  
#include "ui_taskdisplay.h"  
#include "mainwindow.h"  
#include "sqlconnection.h"  
#include <QMessageBox>
```

Référence du fichier Agenda/viewTask/taskdisplay.h

```
#include <QDebug>
#include <QWidget>
#include <QLayout>
#include <QStringListModel>
#include <QStandardItemModel>
#include <QTableView>
#include "taskcheckbox.h"
```

Classes

- class **taskDisplay**

Espaces de nommage

- **Ui**

Référence du fichier Agenda/viewTask/tasktableview.h

Référence du fichier Agenda/viewTask/termdate.cpp

```
#include "termdate.h"  
#include <QDebug>
```

Référence du fichier Agenda/viewTask/termdate.h

```
#include <QWidget>
#include <QLabel>
#include <QLayout>
#include <QHBoxLayout>
#include <QDateTimeEdit>
```

Classes

- class **TermDate**

Référence du fichier Agenda/viewTask/toolbartask.cpp

```
#include "ui_mainwindow.h"  
#include "toolbartask.h"  
#include "utils/spacer.h"  
#include <QDebug>  
#include <QFile>
```

Référence du fichier Agenda/viewTask/toolbartask.h

```
#include <QObject>
#include <QToolBar>
#include <QLabel>
#include <QPushButton>
#include "utils/toolbutton.h"
#include "models/task.h"
```

Classes

- class **ToolBarTask**

Espaces de nommage

- **Ui**

Référence du fichier Agenda/viewTask/tooltipnote.cpp

```
#include "tooltipnote.h"
```

Référence du fichier Agenda/viewTask/tooltipnote.h

Classes

- class `tooltipNote`

Référence du fichier Agenda/viewTask/tristatebutton.cpp

```
#include "tristatebutton.h"
```

Référence du fichier Agenda/viewTask/tristatebutton.h

```
#include <QWidget>
#include <QString>
#include <QHBoxLayout>
#include <QPushButton>
#include <QLabel>
```

Classes

- class **TriStateButton**