

ShareCount

23

Généré par Doxygen 1.8.13

Table des matières

1	Index hiérarchique	1
1.1	Hiérarchie des classes	1
2	Index des classes	3
2.1	Liste des classes	3
3	Index des fichiers	5
3.1	Liste des fichiers	5
4	Documentation des classes	7
4.1	Référence de la classe Accueil	7
4.1.1	Description détaillée	7
4.1.2	Documentation des fonctions membres	8
4.1.2.1	connexion_utilisateur()	8
4.1.2.2	fenetre_creer_un_compte()	8
4.1.2.3	fenetre_se_connecter()	8
4.2	Référence de la classe Cagnotte	9
4.2.1	Description détaillée	9
4.2.2	Documentation des constructeurs et destructeur	9
4.2.2.1	Cagnotte() [1/2]	9
4.2.2.2	Cagnotte() [2/2]	10
4.3	Référence de la classe Compte	10
4.3.1	Description détaillée	10
4.3.2	Documentation des constructeurs et destructeur	11
4.3.2.1	Compte()	11

4.3.3	Documentation des fonctions membres	11
4.3.3.1	getType()	11
4.4	Référence de la classe CompteCommun	12
4.4.1	Description détaillée	12
4.4.2	Documentation des constructeurs et destructeur	12
4.4.2.1	CompteCommun() [1/2]	12
4.4.2.2	CompteCommun() [2/2]	13
4.5	Référence de la classe Connexion	13
4.5.1	Description détaillée	13
4.5.2	Documentation des constructeurs et destructeur	14
4.5.2.1	Connexion()	14
4.5.3	Documentation des fonctions membres	14
4.5.3.1	getDonnees()	14
4.6	Référence de la classe CreerCompte	15
4.6.1	Description détaillée	15
4.6.2	Documentation des fonctions membres	15
4.6.2.1	getDonnees()	15
4.7	Référence de la classe DepensesPartagees	16
4.7.1	Description détaillée	16
4.7.2	Documentation des constructeurs et destructeur	16
4.7.2.1	DepensesPartagees() [1/2]	16
4.7.2.2	DepensesPartagees() [2/2]	17
4.7.3	Documentation des fonctions membres	17
4.7.3.1	rafraichir_affichage()	17
4.8	Référence de la classe FenetreUtilisateur	17
4.8.1	Description détaillée	18
4.8.2	Documentation des constructeurs et destructeur	18
4.8.2.1	FenetreUtilisateur()	18
4.8.3	Documentation des fonctions membres	18
4.8.3.1	afficher_comptes()	19

4.8.3.2	fenetre_cree_un_accueil()	19
4.8.3.3	fenetre_cree_un_compte_argent()	19
4.9	Référence de la classe GestionnaireBDD	20
4.9.1	Description détaillée	20
4.9.2	Documentation des fonctions membres	20
4.9.2.1	ajouter_compte_bdd()	20
4.9.2.2	ajouter_transaction()	21
4.9.2.3	ajouter_utilisateur_compte()	21
4.9.2.4	ajouterUtilisateur()	22
4.9.2.5	recherche_compte()	22
4.9.2.6	recherche_createur()	23
4.9.2.7	recherche_utilisateur()	23
4.9.2.8	recuperer_compte_utilisateur()	23
4.9.2.9	recuperer_liste_membres()	24
4.9.2.10	recuperer_liste_transactions()	24
4.9.2.11	supprimer_compte()	25
4.10	Référence de la classe GestionnaireDialogue	25
4.10.1	Description détaillée	26
4.10.2	Documentation des fonctions membres	26
4.10.2.1	ajouter_compte()	26
4.10.2.2	ajouter_utilisateur()	26
4.10.2.3	get_utilisateur_connecte()	27
4.10.2.4	getGestionnaireBDD()	27
4.10.2.5	recherche_utilisateur()	27
4.10.2.6	set_utilisateur_connecte()	28
4.11	Référence de la classe Inscription	28
4.11.1	Description détaillée	28
4.11.2	Documentation des fonctions membres	29
4.11.2.1	getDonnees()	29
4.12	Référence de la classe Invitation	29

4.12.1	Description détaillée	30
4.12.2	Documentation des constructeurs et destructeur	30
4.12.2.1	Invitation() [1/2]	30
4.12.2.2	Invitation() [2/2]	30
4.13	Référence de la classe Utilisateur	31
4.13.1	Description détaillée	31
4.13.2	Documentation des constructeurs et destructeur	31
4.13.2.1	Utilisateur()	31
4.13.3	Documentation des fonctions membres	32
4.13.3.1	get_adresse_mail()	32
4.13.3.2	get_mot_de_passe()	32
4.13.3.3	get_nom_utilisateur()	32
4.13.3.4	set_adresse_mail()	32
4.13.3.5	set_mot_de_passe()	33
4.13.3.6	set_nom_utilisateur()	33
5	Documentation des fichiers	35
5.1	Référence du fichier accueil.hpp	35
5.2	Référence du fichier cagnotte.hpp	35
5.2.1	Description détaillée	35
5.3	Référence du fichier compte.hpp	36
5.3.1	Description détaillée	36
5.4	Référence du fichier comptecommun.hpp	36
5.4.1	Description détaillée	36
5.5	Référence du fichier connexion.hpp	37
5.5.1	Description détaillée	37
5.6	Référence du fichier creercompte.hpp	37
5.6.1	Description détaillée	37
5.7	Référence du fichier depensespartagees.hpp	38
5.7.1	Description détaillée	38
5.8	Référence du fichier gestionnairebdd.hpp	38
5.8.1	Description détaillée	39
5.9	Référence du fichier gestionnairdialogue.hpp	39
5.9.1	Description détaillée	39
5.10	Référence du fichier inscription.hpp	39
5.10.1	Description détaillée	40
5.11	Référence du fichier invitation.hpp	40
5.11.1	Description détaillée	40
5.12	Référence du fichier utilisateur.hpp	41
5.12.1	Description détaillée	41
Index		43

Chapitre 1

Index hiérarchique

1.1 Hiérarchie des classes

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

Compte	10
GestionnaireBDD	20
GestionnaireDialogue	25
QDialog	
Cagnotte	9
CompteCommun	12
Connexion	13
CreerCompte	15
DepensesPartagees	16
Inscription	28
Invitation	29
QMainWindow	
Accueil	7
FenetreUtilisateur	17
Utilisateur	31

Chapitre 2

Index des classes

2.1 Liste des classes

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

Accueil	
La classe Accueil	7
Cagnotte	
La classe Cagnotte	9
Compte	
La classe Compte	10
CompteCommun	
La classe CompteCommun gère l'UI du compte commun d'un utilisateur	12
Connexion	
La classe Connexion gère l'UI de la connexion de l'utilisateur	13
CreerCompte	
La classe Invitation gère l'UI de la creation d'un compte	15
DepensesPartagees	
La classe DepensePartagees gère l'UI du compte commun d'un utilisateur	16
FenetreUtilisateur	
La classe FenetreUtilisateur gère l'UI de la fenêtre d'un utilisateur	17
GestionnaireBDD	
La classe GestionnaireBDD	20
GestionnaireDialogue	
La classe GestionnaireDialogue gère les fenêtres d'alertes	25
Inscription	
La classe Inscription gère l'UI de création de compte et la validité des données sasisies	28
Invitation	
La classe Invitation gère l'UI de l'invitation d'un utilisateur	29
Utilisateur	
La classe Utilisateur est la classe représentant les données d'un utilisateur	31

Chapitre 3

Index des fichiers

3.1 Liste des fichiers

Liste de tous les fichiers documentés avec une brève description :

accueil.hpp	
Le fichier qui contient la classe Accueil	35
cagnotte.hpp	
Le fichier qui contient la classe Cagnotte	35
compte.hpp	
Le fichier qui contient la classe Compte	36
comptecommun.hpp	
Le fichier qui contient la classe CompteCommun	36
connexion.hpp	
Le fichier qui contient la classe Connexion	37
creercompte.hpp	
Le fichier qui contient la classe CreerCompte	37
depensespartagees.hpp	
Le fichier qui contient la classe DepensePartagees	38
fenetreutilisateur.hpp	??
gestionnairebdd.hpp	
Le fichier qui contient la classe GestionnaireBDD	38
gestionnairedialogue.hpp	
Le fichier qui contient la classe GestionnaireDialogue	39
inscription.hpp	
Le fichier qui contient la classe Inscription	39
invitation.hpp	
Le fichier qui contient la classe Invitation	40
utilisateur.hpp	
Le fichier contenant la description de la classe Utilisateur	41

Chapitre 4

Documentation des classes

4.1 Référence de la classe Accueil

La classe [Accueil](#).

```
#include <accueil.hpp>
```

Graphe d'héritage de Accueil :

Graphe de collaboration de Accueil :

Fonctions membres publiques

- [Accueil](#) (QWidget *parent=nullptr)
Le constructeur de la classe.
- [~Accueil](#) ()
Le destructeur de la classe.
- bool [connexion_utilisateur](#) (const std : :string &adresse_mail, const std : :string &mot_de_passe, bool &mdp_correct)
Fonction qui connecte un utilisateur à l'application en fonction de l'adresse mail et du mot de passe donné en paramètre.
- void [fenetre_creeer_un_compte](#) ()
Création d'une fenêtre de d'inscription depuis la mainwindows.
- void [fenetre_se_connecter](#) (const std : :string &adresse_mail="", const std : :string &mot_de_passe="", const bool utilisateur_existe=false)
Création d'une fenêtre de connexion depuis la mainwindows.

4.1.1 Description détaillée

La classe [Accueil](#).

Elle contient 1 constructeur, 1 destructeur et 6 fonctions.

Version

9

Auteur

H. Mathieu Steinbach, univ. Lorraine

4.1.2 Documentation des fonctions membres

4.1.2.1 connexion_utilisateur()

```
bool Accueil::connexion_utilisateur (
    const std::string & adresse_mail,
    const std::string & mot_de_passe,
    bool & mdp_correct )
```

Fonction qui connecte un utilisateur à l'application en fonction de l'adresse mail et du mot de passe donné en paramètre.

Paramètres

<i>adresse_mail</i>	l'adresse mail donnée
<i>mot_de_passe</i>	un pointeur contenant le mot de passe donné
<i>mdp_correct</i>	un pointeur vers un booléen qui indique si le mot de passe est correct

Renvoie

Un booléen qui indique si l'utilisateur s'est connecté.

Version

5

Auteur

L. Yvoz, univ. Lorraine

4.1.2.2 fenetre_creer_un_compte()

```
void Accueil::fenetre_creer_un_compte ( )
```

Création d'une fenêtre de d'inscription depuis la mainwindows.

Version

5

Auteur

L. Yvoz, univ. Lorraine

4.1.2.3 fenetre_se_connecter()

```
void Accueil::fenetre_se_connecter (
    const std::string & adresse_mail = "",
    const std::string & mot_de_passe = "",
    const bool utilisateur_existe = false )
```

Création d'une fenêtre de connexion depuis la mainwindows.

Paramètres

<i>adresse_mail</i>	l'adresse mail donnée
<i>mot_de_passe</i>	un pointeur contenant le mot de passe donné
<i>utilisateur_existe</i>	un booléen qui contient l'information de l'existence du compte

Version

5

Auteur

L. Yvoz, univ. Lorraine

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

— [accueil.hpp](#)
— [accueil.cpp](#)

4.2 Référence de la classe Cagnotte

La classe [Cagnotte](#).

```
#include <cagnotte.hpp>
```

Graphe d'héritage de Cagnotte :

Graphe de collaboration de Cagnotte :

Fonctions membres publiques

- [Cagnotte](#) (QWidget *parent=nullptr)
Constructeur de la classe [Cagnotte](#).
- [Cagnotte](#) (QWidget *parent, const std::string compte, const [GestionnaireDialogue](#) &gd)
Constructeur de la classe [Cagnotte](#).
- [~Cagnotte](#) ()
Destructeur de la classe, ne fait rien.

4.2.1 Description détaillée

La classe [Cagnotte](#).

Elle contient les fonctions nécessaires à la réactivité de la fenêtre graphique définie dans le fichier Cagnotte.ui.

Version

22

Auteur

L. Yvoz, univ. de Lorraine

4.2.2 Documentation des constructeurs et destructeur

4.2.2.1 Cagnotte() [1/2]

```
Cagnotte::Cagnotte (
    QWidget * parent = nullptr ) [explicit]
```

Constructeur de la classe [Cagnotte](#).

Paramètres

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

4.2.2.2 Cagnotte() [2/2]

```
Cagnotte::Cagnotte (
    QWidget * parent,
    const std::string compte,
    const GestionnaireDialogue & gd )
```

Constructeur de la classe [Cagnotte](#).

Paramètres

<i>QWidget</i>	parent
<i>Compte</i>	compte
<i>gd</i>	gestionnaire de dialogue

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

- [cagnotte.hpp](#)
- [cagnotte.cpp](#)

4.3 Référence de la classe Compte

La classe [Compte](#).

```
#include <compte.hpp>
```

Fonctions membres publiques

- [Compte](#) (const [Utilisateur](#) &createur, const std::string &nom, const int type)
Le constructeur permet de définir les champs de la classe en fonction des paramètres passés.
- [~Compte](#) ()
Le destructeur ne fait rien.
- int [getType](#) ()
Fonction qui retourne un entier correspondant au type du compte.

4.3.1 Description détaillée

La classe [Compte](#).

Elle contient les attributs nécessaires à un compte et définit les méthodes abstraites des comptes.

Version

7

Auteur

L. Yvoz, univ. de Lorraine

4.3.2 Documentation des constructeurs et destructeur

4.3.2.1 Compte()

```
Compte::Compte (
    const Utilisateur & createur,
    const std::string & nom,
    const int type )
```

Le constructeur permet de définir les champs de la classe en fonction des paramètres passés.

Paramètres

<i>createur</i>	l'utilisateur ayant créé le compte.
<i>nom</i>	le nom du compte.
<i>type</i>	le type du compte représenté par un entier (1 : compte de dépenses partagées, 2 : cagnotte, 3 : compte commun).

Version

17

Auteur

L. Yvoz, univ. lorraine

4.3.3 Documentation des fonctions membres

4.3.3.1 getType()

```
int Compte::getType ( )
```

Fonction qui retourne un entier correspondant au type du compte.

Renvoie

le type du compte

Version

17

Auteur

L. Yvoz, univ. lorraine

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

- [compte.hpp](#)
- [compte.cpp](#)

4.4 Référence de la classe CompteCommun

La classe [CompteCommun](#) gère l'UI du compte commun d'un utilisateur.

```
#include <comptecommun.hpp>
```

Graphe d'héritage de [CompteCommun](#) :

Graphe de collaboration de [CompteCommun](#) :

Fonctions membres publiques

- [CompteCommun](#) (QWidget *parent=nullptr)
Constructeur de la classe [CompteCommun](#).
- [CompteCommun](#) (QWidget *parent, const std::string compte, const [GestionnaireDialogue](#) &gd)
Constructeur de la classe [CompteCommun](#).
- [~CompteCommun](#) ()
Destructeur de la classe, ne fait rien.

4.4.1 Description détaillée

La classe [CompteCommun](#) gère l'UI du compte commun d'un utilisateur.

Elle contient 2 méthodes de construction, 1 méthode de destruction et 4 fonctions.

Version

18

Auteur

H. Iopeti, univ. Lorraine

4.4.2 Documentation des constructeurs et destructeur

4.4.2.1 [CompteCommun\(\)](#) [1/2]

```
CompteCommun::CompteCommun (
    QWidget * parent = nullptr ) [explicit]
```

Constructeur de la classe [CompteCommun](#).

Paramètres

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

4.4.2.2 CompteCommun() [2/2]

```
CompteCommun::CompteCommun (
    QWidget * parent,
    const std::string compte,
    const GestionnaireDialogue & gd )
```

Constructeur de la classe [CompteCommun](#).

Paramètres

QWidget	parent
Compte	compte

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

- [comptecommun.hpp](#)
- [comptecommun.cpp](#)

4.5 Référence de la classe Connexion

La classe [Connexion](#) gère l'UI de la connexion de l'utilisateur.

```
#include <connexion.hpp>
```

Graphe d'héritage de Connexion :

Graphe de collaboration de Connexion :

Fonctions membres publiques

- [Connexion](#) (QWidget *parent=nullptr)
Constructeur de la classe [Connexion](#).
- [Connexion](#) (QWidget *parent, const std::string &adresse_mail, const std::string &mot_de_passe, const bool utilisateur_existe)
Constructeur de la classe [Connexion](#).
- [~Connexion](#) ()
Destructeur de la classe, ne fait rien.
- void [getDonnees](#) (std::string &adresse_mail, std::string &mot_de_passe)
Fonction permettant de récupérer les informations rentrées par l'utilisateur.

4.5.1 Description détaillée

La classe [Connexion](#) gère l'UI de la connexion de l'utilisateur.

Elle contient 2 méthodes de construction, 1 méthode de destruction et 5 fonctions.

Version

4

Auteur

H. Iopeti, univ. Lorraine

4.5.2 Documentation des constructeurs et destructeur

4.5.2.1 Connexion()

```
Connexion::Connexion (
    QWidget * parent,
    const std::string & adresse_mail,
    const std::string & mot_de_passe,
    const bool utilisateur_existe )
```

Constructeur de la classe [Connexion](#).

Version

5

Auteur

L. Yvoz, univ. Lorraine

Paramètres

<i>parent</i>	Le parent graphique de la fenêtre
<i>adresse_mail</i>	L'adresse mail de l'utilisateur
<i>mot_de_passe</i>	Le mot de passe de l'utilisateur
<i>utilisateur_existe</i>	Existence de l'utilisateur

4.5.3 Documentation des fonctions membres

4.5.3.1 getDonnees()

```
void Connexion::getDonnees (
    std::string & adresse_mail,
    std::string & mot_de_passe )
```

Fonction permettant de récupérer les informations rentrée par l'utilisateur.

Version

5

Auteur

L. Yvoz, univ. Lorraine

Paramètres

<i>adresse_mail</i>	un pointeur contenant l'adresse vers le string de l'adresse mail donnée
<i>mot_de_passe</i>	un pointeur contenant l'adresse vers le string du mot de passe donné

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

- connexion.hpp
- connexion.cpp

4.6 Référence de la classe CreerCompte

La classe [Invitation](#) gère l'UI de la creation d'un compte.

```
#include <creercompte.hpp>
```

Graphe d'héritage de CreerCompte :

Graphe de collaboration de CreerCompte :

Fonctions membres publiques

- [CreerCompte](#) (QWidget *parent=nullptr)
Constructeur de la classe par défaut.
- [~CreerCompte](#) ()
Destructeur de la classe.
- void [getDonnees](#) (std::string &nom, int &choix)
Fonction permettant de récupérer les informations rentrée par l'utilisateur.

4.6.1 Description détaillée

La classe [Invitation](#) gère l'UI de la creation d'un compte.

Elle contient 1 méthode de construction, 1 méthode de destruction et 3 fonctions.

Version

10

Auteur

H. Mathieu Steinbach, univ. Lorraine

4.6.2 Documentation des fonctions membres

4.6.2.1 getDonnees()

```
void CreerCompte::getDonnees (
    std::string & nom,
    int & choix )
```

Fonction permettant de récupérer les informations rentrée par l'utilisateur.

Paramètres

<i>adresse_mail</i>	un pointeur contenant l'adresse vers le string du nom donné
<i>choix</i>	un pointeur contenant l'adresse vers le choix de type de compte

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

- [creercompte.hpp](#)
- [creercompte.cpp](#)

4.7 Référence de la classe DepensesPartagees

La classe `DepensePartagees` gère l'UI du compte commun d'un utilisateur.

```
#include <depensespartagees.hpp>
```

Graphe d'héritage de `DepensesPartagees` :

Graphe de collaboration de `DepensesPartagees` :

Fonctions membres publiques

- [DepensesPartagees](#) (`QWidget *parent=nullptr`)
Constructeur de la classe [CompteCommun](#).
- [DepensesPartagees](#) (`QWidget *parent`, `const std::string compte`, `const GestionnaireDialogue &gd`)
Constructeur de la classe [DepensesPartagees](#).
- `void rafraichir_affichage ()`
La fonction associée à un rafraichissement de la fenêtre du compte de dépenses partagées.

4.7.1 Description détaillée

La classe `DepensePartagees` gère l'UI du compte commun d'un utilisateur.

Elle contient 2 méthodes de construction, 1 méthode de destruction et 2 fonctions.

Version

21

Auteur

H. Mathieu Steinbach, univ. Lorraine

4.7.2 Documentation des constructeurs et destructeur

4.7.2.1 `DepensesPartagees()` [1/2]

```
DepensesPartagees::DepensesPartagees (
    QWidget * parent = nullptr ) [explicit]
```

Constructeur de la classe [CompteCommun](#).

Paramètres

<code>QWidget</code>	parent
----------------------	--------

4.7.2.2 `DepensesPartagees()` [2/2]

```
DepensesPartagees::DepensesPartagees (
    QWidget * parent,
    const std::string compte,
    const GestionnaireDialogue & gd )
```

Constructeur de la classe [DepensesPartagees](#).

Paramètres

<code>QWidget</code>	parent
<code>Compte</code>	compte

4.7.3 Documentation des fonctions membres

4.7.3.1 `rafraichir_affichage()`

```
void DepensesPartagees::rafraichir_affichage ( )
```

La fonction associée à un rafraichissement de la fenêtre du compte de dépenses partagées.

Version

20

Auteur

H. Mathieu Steinbahc, univ. Lorraine

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

- [depensespartagees.hpp](#)
- `depensespartagees.cpp`

4.8 Référence de la classe FenetreUtilisateur

La classe [FenetreUtilisateur](#) gère l'UI de la fenêtre d'un utilisateur.

```
#include <fenetreutilisateur.hpp>
```

Graphe d'héritage de FenetreUtilisateur :

Graphe de collaboration de FenetreUtilisateur :

Fonctions membres publiques

- [FenetreUtilisateur](#) (QWidget *parent=nullptr)
Constructeur de la classe par défaut.
- [~FenetreUtilisateur](#) ()
Destructeur de la classe, ne fait rien.
- [FenetreUtilisateur](#) (QWidget *parent, const [GestionnaireDialogue](#) &gd)
Constructeur de la classe [FenetreUtilisateur](#).
- void [fenetre_creer_un_compte_argent](#) ()
Création d'une fenêtre de création de compte.
- void [fenetre_creer_un_accueil](#) ()
Création d'une fenêtre d'accueil.
- void [afficher_comptes](#) ()
La fonction qui affiche les comptes dans une liste déroulante.

4.8.1 Description détaillée

La classe [FenetreUtilisateur](#) gère l'UI de la fenêtre d'un utilisateur.

Elle contient 1 méthode de construction, 1 méthode de destruction et 2 fonctions.

Version

10

Auteur

H. Mathieu Steinbach, univ. Lorraine

4.8.2 Documentation des constructeurs et destructeur

4.8.2.1 FenetreUtilisateur()

```
FenetreUtilisateur::FenetreUtilisateur (
    QWidget * parent,
    const GestionnaireDialogue & gd )
```

Constructeur de la classe [FenetreUtilisateur](#).

Paramètres

<i>parent</i>	Le parent graphique
<i>gd</i>	Le GestionnaireDialogue de l'application.

4.8.3 Documentation des fonctions membres

4.8.3.1 afficher_comptes()

```
void FenetreUtilisateur::afficher_comptes ( )
```

La fonction qui affiche les comptes dans une liste déroulante.

Version

15

Auteur

H. Mathieu Steinbach, univ. Lorraine

4.8.3.2 fenetre_creer_un_accueil()

```
void FenetreUtilisateur::fenetre_creer_un_accueil ( )
```

Création d'une fenêtre d'accueil.

Version

14

Auteur

H. Iopeti, univ. Lorraine

4.8.3.3 fenetre_creer_un_compte_argent()

```
void FenetreUtilisateur::fenetre_creer_un_compte_argent ( )
```

Création d'une fenêtre de création de compte.

Version

10

Auteur

H. Mathieu Steinbach, univ. Lorraine

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

- fenetreutilisateur.hpp
- fenetreutilisateur.cpp

4.9 Référence de la classe GestionnaireBDD

La classe [GestionnaireBDD](#).

```
#include <gestionnairebdd.hpp>
```

Fonctions membres publiques

- [GestionnaireBDD](#) ()
Constructeur de la classe.
- bool [ajouterUtilisateur](#) ([Utilisateur](#) utilisateur)
Fonction qui ajoute un utilisateur à la base de données.
- [Utilisateur recherche_utilisateur](#) (const QString &mail)
Fonction qui recherche un utilisateur dans la base de données à partir d'un email.
- void [ajouter_compte_bdd](#) (const std::string email, const std::string nom_compte, int type)
Procédure qui ajoute un compte à la base de données.
- [Compte recherche_compte](#) (const QString &compte)
Fonction qui recherche un compte dans la base de données à partir d'un nom de compte.
- std::vector< std::string > [recuperer_compte_utilisateur](#) (const std::string mail)
Fonction qui retourne la liste des comptes lié à un utilisateur.
- void [ajouter_transaction](#) (const int id, const std::string nom_compte, const std::string donneur, const std::string receveur, const int montant)
Procédure qui ajoute une transaction à la base de données.
- void [supprimer_compte](#) (const QString nom, const QString utilisateur_connecte)
Procédure qui supprimer un compte de la base de données.
- void [ajouter_utilisateur_compte](#) (const std::string compte, const std::string mail)
Procédure qui ajoute un utilisateur à un compte de la base de données.
- void [fermeturebdd](#) ()
Procédure qui ferme la connection à la base de données.
- std::vector< std::string > [recuperer_liste_membres](#) (const std::string compte)
Fonction qui récupère la liste des membres d'un compte.
- std::vector< std::string > [recuperer_liste_transactions](#) (const std::string compte)
Fonction qui récupère la liste des transactions d'un compte.
- [Utilisateur recherche_createur](#) (const QString &compte)
Fonction qui recherche le createur d'un compte dans la base de données à partir d'un nom de compte.

4.9.1 Description détaillée

La classe [GestionnaireBDD](#).

Elle contient 1 méthode de construction et 12 fonctions.

Version

11

Auteur

H. Iopeti, univ. Lorraine

4.9.2 Documentation des fonctions membres

4.9.2.1 ajouter_compte_bdd()

```
void GestionnaireBDD::ajouter_compte_bdd (
    const std::string email,
    const std::string nom_compte,
    int type )
```

Procédure qui ajoute un compte à la base de données.

Paramètres

<i>compte</i>	le compte à ajouter
---------------	---------------------

Version

13

Auteur

H. Mathieu Steinbach, univ. Lorraine

4.9.2.2 ajouter_transaction()

```
void GestionnaireBDD::ajouter_transaction (
    const int id,
    const std::string nom_compte,
    const std::string donneur,
    const std::string receveur,
    const int montant )
```

Procédure qui ajoute une transaction à la base de données.

Paramètres

<i>id</i>	l'identifiant de la transaction
<i>nom_compte</i>	le nom du compte sur lequel se déroule la transaction
<i>donneur</i>	l'adresse mail de l'utilisateur initiant la transaction
<i>receveur</i>	l'adresse mail de l'utilisateur initiant la transaction
<i>montant</i>	le montant de la transaction

4.9.2.3 ajouter_utilisateur_compte()

```
void GestionnaireBDD::ajouter_utilisateur_compte (
    const std::string compte,
    const std::string mail )
```

Procédure qui ajoute un utilisateur à un compte de la base de données.

Paramètres

<i>nom</i>	le nom du compte concerné
<i>l'email</i>	de l'utilisateur à ajouter

Version

16

Auteur

H. Mathieu Steinbach, univ. Lorraine

4.9.2.4 ajouterUtilisateur()

```
bool GestionnaireBDD::ajouterUtilisateur (
    Utilisateur utilisateur )
```

Fonction qui ajoute un utilisateur à la base de données.

Paramètres

<i>utilisateur</i>	l'utilisateur à ajouter
--------------------	-------------------------

Renvoie

un booléen valant faux si l'ajout ne s'est pas bien passé

4.9.2.5 recherche_compte()

```
Compte GestionnaireBDD::recherche_compte (
    const QString & compte )
```

Fonction qui recherche un compte dans la base de données à partir d'un nom de compte.

Paramètres

<i>compte</i>	le nom à partir duquel la recherche s'effectue
---------------	--

Renvoie

le compte trouvé ou un compte avec des données vides si aucun compte n'a été trouvé

Version

17

Auteur

L. Yvoz, univ. Lorraine

4.9.2.6 recherche_createur()

```
Utilisateur GestionnaireBDD::recherche_createur (
    const QString & compte )
```

Fonction qui recherche le createur d'un compte dans la base de données à partir d'un nom de compte.

Paramètres

<i>compte</i>	le nom à partir duquel la recherche s'effectue
---------------	--

Renvoie

l'utilisateur trouvé ou un utilisateur avec des données vides si aucun utilisateur n'a été trouvé

Version

22

Auteur

L. Yvoz, univ. Lorraine

4.9.2.7 recherche_utilisateur()

```
Utilisateur GestionnaireBDD::recherche_utilisateur (
    const QString & mail )
```

Fonction qui recherche un utilisateur dans la base de données à partir d'un email.

Paramètres

<i>mail</i>	l'adresse mail à partir de laquelle la recherche s'effectue
-------------	---

Renvoie

l'utilisateur trouvé ou un utilisateur avec des données vides si aucun utilisateur n'a été trouvé

4.9.2.8 recuperer_compte_utilisateur()

```
std::vector< std::string > GestionnaireBDD::recuperer_compte_utilisateur (
    const std::string mail )
```

Fonction qui retourne la liste des comptes lié à un utilisateur.

Paramètres

<i>mail</i>	l'adresse mail à partir de laquelle la recherche s'effectue
-------------	---

Renvoie

la liste des noms des comptes

Version

15

Auteur

H. Mathieu Steinbach, univ. Lorraine

4.9.2.9 recuperer_liste_membres()

```
std::vector< std::string > GestionnaireBDD::recuperer_liste_membres (
    const std::string compte )
```

Fonction qui récupère la liste des membres d'un compte.

Paramètres

<i>compte</i>	le compte de l'utilisateur concerné
---------------	-------------------------------------

Version

19

Auteur

H. Iopeti, univ. Lorraine

4.9.2.10 recuperer_liste_transactions()

```
std::vector< std::string > GestionnaireBDD::recuperer_liste_transactions (
    const std::string compte )
```

Fonction qui récupère la liste des transactions d'un compte.

Paramètres

<i>compte</i>	le compte de l'utilisateur concerné
---------------	-------------------------------------

Version

19

Auteur

H. Iopeti, univ. Lorraine

4.9.2.11 supprimer_compte()

```
void GestionnaireBDD::supprimer_compte (
    const QString nom,
    const QString utilisateur_connecte )
```

Procédure qui supprimer un compte de la base de données.

Paramètres

<i>nom</i>	le nom du compte à supprimer
------------	------------------------------

Version

16

Auteur

H. Mathieu Steinbach, univ. Lorraine

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

- [gestionnairebdd.hpp](#)
- [gestionnairebdd.cpp](#)

4.10 Référence de la classe GestionnaireDialogue

La classe [GestionnaireDialogue](#) gère les fenêtres d'alertes.

```
#include <gestionnairedialogue.hpp>
```

Fonctions membres publiques

- [GestionnaireDialogue](#) ()
Constructeur de la classe.
- [~GestionnaireDialogue](#) ()
Destructeur de la classe, ne fait rien.
- [Utilisateur](#) [get_utilisateur_connecte](#) ()
Fonction qui retourne l'utilisateur connecté.
- void [set_utilisateur_connecte](#) (const [Utilisateur](#) &utilisateur)
Fonction qui définit un utilisateur connecté.
- [Utilisateur](#) [recherche_utilisateur](#) (const std::string &adresse_mail)
Fonction qui trouve un utilisateur à l'aide de son mail.
- void [ajouter_utilisateur](#) (const [Utilisateur](#) &utilisateur)
Fonction qui ajoute un utilisateur à la liste des utilisateurs de l'application.
- void [ajouter_compte](#) (const std::string &nom, const int type_compte)
Fonction qui crée le bon compte à partir des paramètres et l'ajoute à la liste des comptes enregistrés.
- [GestionnaireBDD](#) [getGestionnaireBDD](#) ()
Fonction qui retourne le gestionnaire de la bdd.

4.10.1 Description détaillée

La classe `GestionnaireDialogue` gère les fenêtres d'alertes.

Elle contient 1 méthode de construction, 1 méthode de destruction et 3 fonctions.

Version

5

Auteur

H. Iopeti, univ. Lorraine

4.10.2 Documentation des fonctions membres

4.10.2.1 `ajouter_compte()`

```
void GestionnaireDialogue::ajouter_compte (
    const std::string & nom,
    const int type_compte )
```

Fonction qui crée le bon compte à partir des paramètres et l'ajoute à la liste des comptes enregistrés.

Version

7

Auteur

L. Yvoz, univ. Lorraine

Paramètres

<i>nom</i>	le nom du compte à créer
<i>type_compte</i>	un chiffre représentant le type du compte à créer, 1 : dépenses partagées, 2 : cagnotte, 3 : compte commun.

4.10.2.2 `ajouter_utilisateur()`

```
void GestionnaireDialogue::ajouter_utilisateur (
    const Utilisateur & utilisateur )
```

Fonction qui ajoute un utilisateur à la liste des utilisateurs de l'application.

Version

5

Auteur

L. Yvoz, univ. Lorraine

Paramètres

<i>utilisateur</i>	l'utilisateur à ajouter
--------------------	-------------------------

4.10.2.3 get_utilisateur_connecte()

```
Utilisateur GestionnaireDialogue::get_utilisateur_connecte ( )
```

Fonction qui retourne l'utilisateur connecté.

Renvoie

L'utilisateur qui est connecté.

4.10.2.4 getGestionnaireBDD()

```
GestionnaireBDD GestionnaireDialogue::getGestionnaireBDD ( )
```

Fonction qui retourne le gestionnaire de la bdd.

Renvoie

le gestionnaire de la bdd

4.10.2.5 recherche_utilisateur()

```
Utilisateur GestionnaireDialogue::recherche_utilisateur (
    const std::string & adresse_mail )
```

Fonction qui trouve un utilisateur à l'aide de son mail.

Paramètres

<i>adresse_mail</i>	L'adresse mail de l'utilisateur recherché
---------------------	---

Renvoie

Un utilisateur qui correspond à l'email donné.

4.10.2.6 set_utilisateur_connecte()

```
void GestionnaireDialogue::set_utilisateur_connecte (
    const Utilisateur & utilisateur )
```

Fonction qui définit un utilisateur connecté.

Paramètres

<i>utilisateur</i>	l'utilisateur à connecter
--------------------	---------------------------

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

- [gestionnairedialogue.hpp](#)
- [gestionnairedialogue.cpp](#)

4.11 Référence de la classe Inscription

La classe [Inscription](#) gère l'UI de création de compte et la validité des données saisies.

```
#include <inscription.hpp>
```

Graphe d'héritage de Inscription :

Graphe de collaboration de Inscription :

Fonctions membres publiques

- [Inscription](#) (QWidget *parent=nullptr)
Constructeur de la classe par défaut.
- [~Inscription](#) ()
Destructeur de la classe, ne fait rien.
- void [getDonnees](#) (std : :string &adresse_mail, std : :string &nom_utilisateur, std : :string &mot_de_passe)
Fonction permettant de récupérer les informations rentrée par l'utilisateur.

4.11.1 Description détaillée

La classe [Inscription](#) gère l'UI de création de compte et la validité des données saisies.

Elle contient 1 méthode de construction, 1 méthode de destruction et 2 fonctions.

Version

2

Auteur

H. Mathieu Steinbach, univ. Lorraine

4.11.2 Documentation des fonctions membres

4.11.2.1 getDonnees()

```
void Inscription::getDonnees (
    std::string & adresse_mail,
    std::string & nom_utilisateur,
    std::string & mot_de_passe )
```

Fonction permettant de récupérer les informations rentrée par l'utilisateur.

Version

5

Auteur

L. Yvoz, univ. Lorraine

Paramètres

<i>adresse_mail</i>	un pointeur contenant l'adresse vers le string de l'adresse mail donnée
<i>nom_utilisateur</i>	un pointeur contenant l'adresse vers le string du nom d'utilisateur donné
<i>mot_de_passe</i>	un pointeur contenant l'adresse vers le string du mot de passe donné

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

- [inscription.hpp](#)
- [inscription.cpp](#)

4.12 Référence de la classe Invitation

La classe [Invitation](#) gère l'UI de l'invitation d'un utilisateur.

```
#include <invitation.hpp>
```

Graphe d'héritage de Invitation :

Graphe de collaboration de Invitation :

Fonctions membres publiques

- [Invitation](#) (QWidget *parent=nullptr)
Constructeur de la classe [Invitation](#).
- [Invitation](#) (QWidget *parent, const std : :string compte, const [GestionnaireDialogue](#) &gd)
Constructeur de la classe [Invitation](#).
- [~Invitation](#) ()
Destructeur de la classe, ne fait rien.

4.12.1 Description détaillée

La classe [Invitation](#) gère l'UI de l'invitation d'un utilisateur.

Elle contient 2 méthodes de construction, 1 méthode de destruction et 2 fonctions.

Version

8

Auteur

H. Iopeti, univ. Lorraine

4.12.2 Documentation des constructeurs et destructeur

4.12.2.1 Invitation() [1/2]

```
Invitation::Invitation (
    QWidget * parent = nullptr ) [explicit]
```

Constructeur de la classe [Invitation](#).

Paramètres

QWidget	parent
-------------------------	--------

4.12.2.2 Invitation() [2/2]

```
Invitation::Invitation (
    QWidget * parent,
    const std::string compte,
    const GestionnaireDialogue & gd )
```

Constructeur de la classe [Invitation](#).

Paramètres

QWidget	parent
Compte	compte

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

- [invitation.hpp](#)
- [invitation.cpp](#)

4.13 Référence de la classe Utilisateur

La classe `Utilisateur` est la classe représentant les données d'un utilisateur.

```
#include <utilisateur.hpp>
```

Fonctions membres publiques

- `Utilisateur` (const std : :string &nom_utilisateur, const std : :string &adresse_mail, const std : :string &mot_de_passe)
Le constructeur, assigne les valeurs entrées en paramètre aux attributs de la classe.
- `~Utilisateur` ()
Le destructeur, ne fait rien.
- std : :string `get_nom_utilisateur` ()
Méthode d'obtention de l'attribut nom_utilisateur.
- void `set_nom_utilisateur` (const std : :string &nom_utilisateur)
Méthode de modification de l'attribut nom_utilisateur.
- std : :string `get_adresse_mail` ()
Méthode d'obtention de l'attribut adresse_mail.
- void `set_adresse_mail` (const std : :string &adresse_mail)
Méthode de modification de l'attribut adresse_mail.
- std : :string `get_mot_de_passe` ()
Méthode d'obtention de l'attribut mot_de_passe.
- void `set_mot_de_passe` (const std : :string &mdp)
Méthode de modification de l'attribut mot_de_passe.

4.13.1 Description détaillée

La classe `Utilisateur` est la classe représentant les données d'un utilisateur.

Elle contient 4 attributs, 1 méthode de construction, 1 méthode de destruction et 6 méthodes d'obtention et de modification des attributs cités précédemment.

Version

3

Auteur

L. Yvoz, univ. de Lorraine

4.13.2 Documentation des constructeurs et destructeur

4.13.2.1 Utilisateur()

```
Utilisateur::Utilisateur (
    const std::string & nom_utilisateur,
    const std::string & adresse_mail,
    const std::string & mot_de_passe )
```

Le constructeur, assigne les valeurs entrées en paramètre aux attributs de la classe.

Paramètres

<i>nom_utilisateur</i>	le nom d'utilisateur
<i>adresse_mail</i>	l'adresse mail
<i>mot_de_passe</i>	le mot de passe

4.13.3 Documentation des fonctions membres**4.13.3.1 get_adresse_mail()**

```
std::string Utilisateur::get_adresse_mail ( )
```

Méthode d'obtention de l'attribut `adresse_mail`.

Renvoie

La chaîne de caractères demandée.

4.13.3.2 get_mot_de_passe()

```
std::string Utilisateur::get_mot_de_passe ( )
```

Méthode d'obtention de l'attribut `mot_de_passe`.

Renvoie

La chaîne de caractères demandée.

4.13.3.3 get_nom_utilisateur()

```
std::string Utilisateur::get_nom_utilisateur ( )
```

Méthode d'obtention de l'attribut `nom_utilisateur`.

Renvoie

La chaîne de caractères demandée.

4.13.3.4 set_adresse_mail()

```
void Utilisateur::set_adresse_mail (
    const std::string & adresse_mail )
```

Méthode de modification de l'attribut `adresse_mail`.

Paramètres

<i>La</i>	chaîne de caractères à utiliser pour la modification.
-----------	---

4.13.3.5 set_mot_de_passe()

```
void Utilisateur::set_mot_de_passe (
    const std::string & mdp )
```

Méthode de modification de l'attribut mot_de_passe.

Paramètres

<i>adresse_mail</i>	La chaîne de caractères à utiliser pour la modification du mot de passe.
---------------------	--

4.13.3.6 set_nom_utilisateur()

```
void Utilisateur::set_nom_utilisateur (
    const std::string & nom_utilisateur )
```

Méthode de modification de l'attribut nom_utilisateur.

Paramètres

<i>La</i>	chaîne de caractères à utiliser pour la modification du nom d'utilisateur.
-----------	--

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

- [utilisateur.hpp](#)
- [utilisateur.cpp](#)

Chapitre 5

Documentation des fichiers

5.1 Référence du fichier accueil.hpp

Le fichier qui contient la classe [Accueil](#).

```
#include "connexion.hpp"
#include "fenetreutilisateur.hpp"
#include "gestionnairedialogue.hpp"
#include "inscription.hpp"
#include "ui_accueil.h"
#include <QMainWindow>
```

Graphe des dépendances par inclusion de accueil.hpp :

5.2 Référence du fichier cagnotte.hpp

Le fichier qui contient la classe [Cagnotte](#).

```
#include "gestionnairedialogue.hpp"
#include <QDialog>
```

Graphe des dépendances par inclusion de cagnotte.hpp : Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

— class [Cagnotte](#)
La classe [Cagnotte](#).

5.2.1 Description détaillée

Le fichier qui contient la classe [Cagnotte](#).

Version

22

Auteur

L. Yvoz, univ. Lorraine

5.3 Référence du fichier compte.hpp

Le fichier qui contient la classe [Compte](#).

```
#include "utilisateur.hpp"
#include <vector>
```

Graphe des dépendances par inclusion de compte.hpp : Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

- class [Compte](#)
La classe [Compte](#).

5.3.1 Description détaillée

Le fichier qui contient la classe [Compte](#).

Version

7

Auteur

L. Yvoz, univ. Lorraine

5.4 Référence du fichier comptecommun.hpp

Le fichier qui contient la classe [CompteCommun](#).

```
#include "gestionnairedialogue.hpp"
#include <QDialog>
```

Graphe des dépendances par inclusion de comptecommun.hpp : Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

- class [CompteCommun](#)
La classe [CompteCommun](#) gère l'UI du compte commun d'un utilisateur.

5.4.1 Description détaillée

Le fichier qui contient la classe [CompteCommun](#).

Version

18

Auteur

H. Iopeti, univ. Lorraine

5.5 Référence du fichier connexion.hpp

Le fichier qui contient la classe [Connexion](#).

```
#include "accueil.hpp"
#include "ui_connexion.h"
#include <QDialog>
#include <iostream>
#include <regex>
```

Graphe des dépendances par inclusion de connexion.hpp : Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

- class [Connexion](#)
La classe [Connexion](#) gère l'UI de la connexion de l'utilisateur.

5.5.1 Description détaillée

Le fichier qui contient la classe [Connexion](#).

Version

4

Auteur

H. Iopeti, univ. Lorraine

5.6 Référence du fichier creercompte.hpp

Le fichier qui contient la classe [CreerCompte](#).

```
#include <QDialog>
```

Graphe des dépendances par inclusion de creercompte.hpp : Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

- class [CreerCompte](#)
La classe [Invitation](#) gère l'UI de la creation d'un compte.

5.6.1 Description détaillée

Le fichier qui contient la classe [CreerCompte](#).

Version

10

Auteur

H. Mathieu Steinbach, univ. Lorraine

5.7 Référence du fichier depensespartagees.hpp

Le fichier qui contient la classe `DepensePartagees`.

```
#include <QDialog>
#include <gestionnairedialogue.hpp>
```

Graphe des dépendances par inclusion de `depensespartagees.hpp` : Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

- class `DepensesPartagees`
La classe `DepensePartagees` gère l'UI du compte commun d'un utilisateur.

5.7.1 Description détaillée

Le fichier qui contient la classe `DepensePartagees`.

Version

21

Auteur

H. Mathieu Steinbach, univ. Lorraine

5.8 Référence du fichier gestionnairebdd.hpp

Le fichier qui contient la classe `GestionnaireBDD`.

```
#include "compte.hpp"
#include "utilisateur.hpp"
#include <QFileDialog>
#include <QTableView>
#include <QtSql>
#include <cstring>
#include <iostream>
```

Graphe des dépendances par inclusion de `gestionnairebdd.hpp` : Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

- class `GestionnaireBDD`
La classe `GestionnaireBDD`.

5.8.1 Description détaillée

Le fichier qui contient la classe [GestionnaireBDD](#).

Version

11

Auteur

H. Iopeti, univ. Lorraine

5.9 Référence du fichier gestionnairedialogue.hpp

Le fichier qui contient la classe [GestionnaireDialogue](#).

```
#include "compte.hpp"
#include "gestionnairebdd.hpp"
#include "utilisateur.hpp"
#include <iostream>
#include <vector>
```

Graphe des dépendances par inclusion de gestionnairedialogue.hpp : Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

- class [GestionnaireDialogue](#)
La classe [GestionnaireDialogue](#) gère les fenêtres d'alertes.

5.9.1 Description détaillée

Le fichier qui contient la classe [GestionnaireDialogue](#).

Version

5

Auteur

H. Iopeti, univ. Lorraine

5.10 Référence du fichier inscription.hpp

Le fichier qui contient la classe [Inscription](#).

```
#include "ui_inscription.h"
#include <QDialog>
#include <iostream>
#include <regex>
#include <string>
```

Graphe des dépendances par inclusion de inscription.hpp : Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

- class [Inscription](#)

La classe [Inscription](#) gère l'UI de création de compte et la validité des données saisies.

5.10.1 Description détaillée

Le fichier qui contient la classe [Inscription](#).

Version

2

Auteur

H. Mathieu Steinbach, univ. Lorraine

5.11 Référence du fichier invitation.hpp

Le fichier qui contient la classe [Invitation](#).

```
#include "string.h"
#include "ui_invitation.h"
#include <QDialog>
#include <gestionnairedialogue.hpp>
#include <iostream>
#include <regex>
```

Graphe des dépendances par inclusion de invitation.hpp : Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

- class [Invitation](#)

La classe [Invitation](#) gère l'UI de l'invitation d'un utilisateur.

5.11.1 Description détaillée

Le fichier qui contient la classe [Invitation](#).

Version

8

Auteur

H. Iopeti, univ. Lorraine

5.12 Référence du fichier utilisateur.hpp

Le fichier contenant la description de la classe [Utilisateur](#).

```
#include <string>
```

Graphe des dépendances par inclusion de utilisateur.hpp : Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

— class [Utilisateur](#)

La classe [Utilisateur](#) est la classe représentant les données d'un utilisateur.

5.12.1 Description détaillée

Le fichier contenant la description de la classe [Utilisateur](#).

Version

3

Auteur

L. Yvoz, univ. de Lorraine

Index

Accueil, [7](#)
 connexion_utilisateur, [8](#)
 fenetre_creer_un_compte, [8](#)
 fenetre_se_connecter, [8](#)
accueil.hpp, [35](#)
afficher_comptes
 FenetreUtilisateur, [18](#)
ajouter_compte
 GestionnaireDialogue, [26](#)
ajouter_compte_bdd
 GestionnaireBDD, [20](#)
ajouter_transaction
 GestionnaireBDD, [21](#)
ajouter_utilisateur
 GestionnaireDialogue, [26](#)
ajouter_utilisateur_compte
 GestionnaireBDD, [21](#)
ajouterUtilisateur
 GestionnaireBDD, [22](#)

Cagnotte, [9](#)
 Cagnotte, [9, 10](#)
cagnotte.hpp, [35](#)
Compte, [10](#)
 Compte, [11](#)
 getType, [11](#)
compte.hpp, [36](#)
CompteCommun, [12](#)
 CompteCommun, [12](#)
comptecommun.hpp, [36](#)
Connexion, [13](#)
 Connexion, [14](#)
 getDonnees, [14](#)
connexion.hpp, [37](#)
connexion_utilisateur
 Accueil, [8](#)
CreerCompte, [15](#)
 getDonnees, [15](#)
creercompte.hpp, [37](#)

DepensesPartagees, [16](#)
 DepensesPartagees, [16, 17](#)
 rafraichir_affichage, [17](#)
depensespartagees.hpp, [38](#)

fenetre_creer_un_accueil
 FenetreUtilisateur, [19](#)
fenetre_creer_un_compte
 Accueil, [8](#)
fenetre_creer_un_compte_argent

 FenetreUtilisateur, [19](#)
fenetre_se_connecter
 Accueil, [8](#)
FenetreUtilisateur, [17](#)
 afficher_comptes, [18](#)
 fenetre_creer_un_accueil, [19](#)
 fenetre_creer_un_compte_argent, [19](#)
 FenetreUtilisateur, [18](#)

GestionnaireBDD, [20](#)
 ajouter_compte_bdd, [20](#)
 ajouter_transaction, [21](#)
 ajouter_utilisateur_compte, [21](#)
 ajouterUtilisateur, [22](#)
 recherche_compte, [22](#)
 recherche_createur, [22](#)
 recherche_utilisateur, [23](#)
 recuperer_compte_utilisateur, [23](#)
 recuperer_liste_membres, [24](#)
 recuperer_liste_transactions, [24](#)
 supprimer_compte, [25](#)
GestionnaireDialogue, [25](#)
 ajouter_compte, [26](#)
 ajouter_utilisateur, [26](#)
 get_utilisateur_connecte, [27](#)
 getGestionnaireBDD, [27](#)
 recherche_utilisateur, [27](#)
 set_utilisateur_connecte, [28](#)
gestionnairebdd.hpp, [38](#)
gestionnairdialogue.hpp, [39](#)
get_adresse_mail
 Utilisateur, [32](#)
get_mot_de_passe
 Utilisateur, [32](#)
get_nom_utilisateur
 Utilisateur, [32](#)
get_utilisateur_connecte
 GestionnaireDialogue, [27](#)
getDonnees
 Connexion, [14](#)
 CreerCompte, [15](#)
 Inscription, [29](#)
getGestionnaireBDD
 GestionnaireDialogue, [27](#)
getType
 Compte, [11](#)

Inscription, [28](#)
 getDonnees, [29](#)
inscription.hpp, [39](#)

Invitation, [29](#)
 Invitation, [30](#)
invitation.hpp, [40](#)

rafraichir_affichage
 DepensesPartagees, [17](#)

recherche_compte
 GestionnaireBDD, [22](#)

recherche_createur
 GestionnaireBDD, [22](#)

recherche_utilisateur
 GestionnaireBDD, [23](#)
 GestionnaireDialogue, [27](#)

recuperer_compte_utilisateur
 GestionnaireBDD, [23](#)

recuperer_liste_membres
 GestionnaireBDD, [24](#)

recuperer_liste_transactions
 GestionnaireBDD, [24](#)

set_adresse_mail
 Utilisateur, [32](#)

set_mot_de_passe
 Utilisateur, [33](#)

set_nom_utilisateur
 Utilisateur, [33](#)

set_utilisateur_connecte
 GestionnaireDialogue, [28](#)

supprimer_compte
 GestionnaireBDD, [25](#)

Utilisateur, [31](#)
 get_adresse_mail, [32](#)
 get_mot_de_passe, [32](#)
 get_nom_utilisateur, [32](#)
 set_adresse_mail, [32](#)
 set_mot_de_passe, [33](#)
 set_nom_utilisateur, [33](#)
 Utilisateur, [31](#)

utilisateur.hpp, [41](#)