

Phileas_Gestion

Généré par Doxygen 1.8.9.1

Jeudi 21 Mai 2015 10 :00 :29

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	Documentation des classes	5
3.1	Référence de la classe AppDelegate	5
3.1.1	Documentation des méthodes	5
3.1.1.1	chargerUtilisateur()	5
3.1.1.2	estConnecte()	5
3.2	Référence de la catégorie AppDelegate()	6
3.3	Référence de la classe BaremeAuto	6
3.3.1	Documentation des méthodes	6
3.3.1.1	initWithName :trancheBasse :trancheMoyenne :trancheHaute :fixe :andContext : (NSString *cylindre,[trancheBasse] NSNumber *basse,[trancheMoyenne] NSNumber *moyenne,[trancheHaute] NSNumber *haute,[fixe] NSNumber *fixe,[andContext] NSManagedObjectContext *context)	6
3.3.1.2	selectBaremeAuto :andContext : (NSString *cylindre,[andContext] NSManagedObjectContext *context)	7
3.4	Référence de la catégorie BaremeAuto(CoreDataGeneratedAccessors)	7
3.5	Référence de la catégorie BaremeAuto(DataModel)	7
3.5.1	Documentation des méthodes	7
3.5.1.1	initWithName :trancheBasse :trancheMoyenne :trancheHaute :fixe :andContext : (NSString *cylindre,[trancheBasse] NSNumber *basse,[trancheMoyenne] NSNumber *moyenne,[trancheHaute] NSNumber *haute,[fixe] NSNumber *fixe,[andContext] NSManagedObjectContext *context)	8
3.5.1.2	selectBaremeAuto :andContext : (NSString *cylindre,[andContext] NSManagedObjectContext *context)	9
3.6	Référence de la classe ConnexionViewController	9
3.6.1	Documentation des méthodes	10
3.6.1.1	Connexion : (id sender)	10
3.7	Référence de la classe Draft	10
3.7.1	Description détaillée	10

3.7.2	Documentation des méthodes	10
3.7.2.1	mapping()	10
3.8	Référence de la classe Expense_def	11
3.8.1	Description détaillée	11
3.9	Référence de la classe Frais	11
3.9.1	Documentation des méthodes	12
3.9.1.1	addIndemnitéK :(IndemnitéK *indemnité)	12
3.9.1.2	initWithDate :localisation :type :image :montant :commentaire :andContext :(← NSDate *date,[localisation] NSString *loc,[type] NSString *type,[image] NSData *img,[montant] NSNumber *montant,[commentaire] NSString *com,[andContext] NSManagedObjectContext *context)	12
3.9.1.3	updateFrais :localisation :type :image :montant :commentaire :andContext :(N← NSDate *date,[localisation] NSString *loc,[type] NSString *type,[image] NSData *img,[montant] NSNumber *montant,[commentaire] NSString *com,[andContext] NSManagedObjectContext *context)	12
3.10	Référence de la catégorie Frais(DataModel)	13
3.10.1	Documentation des méthodes	13
3.10.1.1	addIndemnitéK :(IndemnitéK *indemnité)	13
3.10.1.2	initWithDate :localisation :type :image :montant :commentaire :andContext :(← NSDate *date,[localisation] NSString *loc,[type] NSString *type,[image] NSData *img,[montant] NSNumber *montant,[commentaire] NSString *com,[andContext] NSManagedObjectContext *context)	13
3.10.1.3	updateFrais :localisation :type :image :montant :commentaire :andContext :(N← NSDate *date,[localisation] NSString *loc,[type] NSString *type,[image] NSData *img,[montant] NSNumber *montant,[commentaire] NSString *com,[andContext] NSManagedObjectContext *context)	13
3.11	Référence de la classe FraisTableViewController	14
3.11.1	Documentation des méthodes	15
3.11.1.1	compterNotifs()	15
3.11.1.2	creerBaremesAuto()	15
3.11.1.3	creerTypesFrais()	15
3.11.1.4	envoyer :(id sender)	15
3.12	Référence de la classe HistoriqueTableViewController	15
3.13	Référence de la classe IndemnitéK	16
3.13.1	Documentation des méthodes	17
3.13.1.1	updateIndemnitéK :villeArrivée :allezR :baremeAuto :andDistance :(NSString *villeD,[villeArrivée] NSString *villeA,[allezR] NSNumber *AR,[baremeAuto] BaremeAuto *cylindree,[andDistance] NSNumber *distance)	17
3.14	Référence de la catégorie IndemnitéK(DataModel)	18
3.14.1	Documentation des méthodes	18
3.14.1.1	updateIndemnitéK :villeArrivée :allezR :baremeAuto :andDistance :(NSString *villeD,[villeArrivée] NSString *villeA,[allezR] NSNumber *AR,[baremeAuto] BaremeAuto *cylindree,[andDistance] NSNumber *distance)	18
3.15	Référence de la classe IndemnitésTableViewController	18
3.16	Référence de la classe MapTableViewController	19

3.17	Référence de la classe Message	19
3.17.1	Description détaillée	20
3.18	Référence de la classe NotificationCellule	20
3.18.1	Description détaillée	20
3.19	Référence de la classe NotificationTableViewController	20
3.19.1	Documentation des méthodes	21
3.19.1.1	chargerMessages()	21
3.19.1.2	loadFrais()	21
3.20	Référence de la classe OptionsViewController	21
3.20.1	Documentation des méthodes	22
3.20.1.1	Deconnexion :(id sender)	22
3.21	Référence de la classe Sheet	22
3.21.1	Description détaillée	23
3.21.2	Documentation des méthodes	23
3.21.2.1	getLabelFromStatusId :(NSNumber *idEtat)	23
3.21.3	Documentation des données membres	23
3.21.3.1	draft	23
3.22	Référence de la classe Type	23
3.22.1	Documentation des méthodes	24
3.22.1.1	initWithName :andID :andContext :(NSString *lib,[andID] NSNumber *id↔ Type,[andContext] NSManagedObjectContext *context)	24
3.22.1.2	selectTypeFrais :andContext :(NSString *lib,[andContext] NSManagedObjectContext *context)	24
3.23	Référence de la catégorie Type(CoreDataGeneratedAccessors)	24
3.24	Référence de la catégorie Type(DataModel)	24
3.24.1	Documentation des méthodes	25
3.24.1.1	initWithName :andID :andContext :(NSString *lib,[andID] NSNumber *id↔ Type,[andContext] NSManagedObjectContext *context)	25
3.24.1.2	selectTypeFrais :andContext :(NSString *lib,[andContext] NSManagedObjectContext *context)	25
3.25	Référence de la classe User	25
3.25.1	Documentation des méthodes	26
3.25.1.1	initWithPseudo :motDePasse :andContext :(NSString *pseudo,[motDePasse] NSString *mdp,[andContext] NSManagedObjectContext *context)	26
3.26	Référence de la catégorie User(CoreDataGeneratedAccessors)	26
3.27	Référence de la catégorie User(DataModel)	26
3.27.1	Documentation des méthodes	26
3.27.1.1	initWithPseudo :motDePasse :andContext :(NSString *pseudo,[motDePasse] NSString *mdp,[andContext] NSManagedObjectContext *context)	26

Chapitre 1

Index hiérarchique

1.1 Hiérarchie des classes

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

AppDelegate()	6
BaremeAuto(CoreDataGeneratedAccessors)	7
BaremeAuto(DataModel)	7
<CLLocationManagerDelegate>	
FraisTableViewController	14
Frais(DataModel)	13
IndemnitéK(DataModel)	18
<MKMapViewDelegate>	
IndemnitésTableViewController	18
<NSFetchedResultsControllerDelegate>	
FraisTableViewController	14
HistoriqueTableViewController	15
NSObject	
BaremeAuto	6
Frais	11
IndemnitéK	16
Type	23
User	25
NSObject	
Draft	10
Expense_def	11
Message	19
NSNumber	
Sheet	22
Type(CoreDataGeneratedAccessors)	24
Type(DataModel)	24
<UIActionSheetDelegate>	
HistoriqueTableViewController	15
<UIAlertViewDelegate>	
FraisTableViewController	14
<UIApplicationDelegate>	
AppDelegate	5
<UIImagePickerControllerDelegate>	
FraisTableViewController	14
<UINavigationControllerDelegate>	
FraisTableViewController	14
UIResponder	
AppDelegate	5

<UISearchBarDelegate>	
MapTableViewController	19
<UISearchDisplayDelegate>	
MapTableViewController	19
UITableViewCell	
NotificationCellule	20
UITableViewController	
FraisTableViewController	14
HistoriqueTableViewController	15
IndemnitesTableViewController	18
MapTableViewController	19
NotificationTableViewController	20
UIViewController	
ConnexionViewController	9
OptionsViewController	21
User(CoreDataGeneratedAccessors)	26
User(DataModel)	26

Chapitre 2

Index des classes

2.1 Liste des classes

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

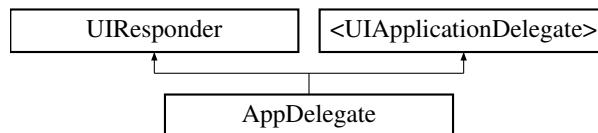
AppDelegate	5
AppDelegate()	6
BaremeAuto	6
BaremeAuto(CoreDataGeneratedAccessors)	7
BaremeAuto(DataModel)	7
ConnexionViewController	9
Draft	
Classe servant à la communication avec l'api Phileas via RestKit	10
Expense_def	
Classe servant à la communication avec l'api Phileas via RestKit	11
Frais	11
Frais(DataModel)	13
FraisTableViewController	14
HistoriqueTableViewController	15
IndemnitéK	16
IndemnitéK(DataModel)	18
IndemnitésTableViewController	18
MapTableViewController	19
Message	
Classe servant à la communication avec l'api Phileas via RestKit	19
NotificationCellule	
Cette classe sert à remplir le contenu présent sur chaque ligne du tableView notifications	20
NotificationTableViewController	20
OptionsViewController	21
Sheet	
Classe servant à la communication avec l'api Phileas via RestKit	22
Type	23
Type(CoreDataGeneratedAccessors)	24
Type(DataModel)	24
User	25
User(CoreDataGeneratedAccessors)	26
User(DataModel)	26

Chapitre 3

Documentation des classes

3.1 Référence de la classe AppDelegate

Graphe d'héritage de AppDelegate :



Méthodes d'instance

- (void) - **saveContext**
- (NSURL *) - **applicationDocumentsDirectory**
- (BOOL) - **estConnecte**
Cette fonction permet de savoir si un utilisateur est déjà connecté.
- (User *) - **chargerUtilisateur**
Cette fonction permet de charger les informations concernant l'utilisateur connecté.

Propriétés

- UIWindow * **window**
- NSManagedObjectContext * **managedObjectContext**
- NSManagedObjectModel * **managedObjectModel**
- NSPersistentStoreCoordinator * **persistentStoreCoordinator**

3.1.1 Documentation des méthodes

3.1.1.1 - (User *) chargerUtilisateur

Cette fonction permet de charger les informations concernant l'utilisateur connecté.

Renvoie

l'utilisateur connecté

3.1.1.2 - (BOOL) estConnecte

Cette fonction permet de savoir si un utilisateur est déjà connecté.

Si un utilisateur est sauvé en local alors nous considérons qu'il est connecté et l'application passera la page de connexion.

Renvoie

true si un utilisateur existe, false sinon.

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

- ProjetIndus/Phileas_Gestion/AppDelegate.h
- ProjetIndus/Phileas_Gestion/AppDelegate.m

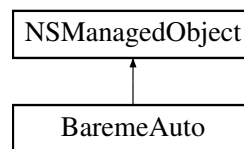
3.2 Référence de la catégorie AppDelegate()

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

- ProjetIndus/Phileas_Gestion/AppDelegate.m

3.3 Référence de la classe BaremeAuto

Graphe d'héritage de BaremeAuto :



Méthodes d'instance

- (void) - **addIndemnitesBaremeObject** :
- (void) - **removeIndemnitesBaremeObject** :
- (void) - **addIndemnitesBareme** :
- (void) - **removeIndemnitesBareme** :
- (void) - **initWithName :trancheBasse :trancheMoyenne :trancheHaute :fixe :andContext** :
Fonction permettant de créer un barème automobile servant pour calculer les indemnités kilométriques.

Méthodes de classe

- (**BaremeAuto** *) + **selectBaremeAuto :andContext** :
Fonction permettant de sélectionner un barème automobile.

Propriétés

- NSNumber * **basse**
- NSString * **cylindre**
- NSNumber * **fixe**
- NSNumber * **haute**
- NSNumber * **moyenne**
- NSSet * **indemnitesBareme**

3.3.1 Documentation des méthodes

- 3.3.1.1 - (void) initWithName : (NSString*) *cylindre* trancheBasse :(NSNumber*) *basse* trancheMoyenne :(NSNumber*) *moyenne* trancheHaute :(NSNumber*) *haute* fixe :(NSNumber*) *fixe* andContext :(NSObject*) *context*

Fonction permettant de créer un barème automobile servant pour calculer les indemnités kilométriques.

Paramètres

<i>cylindre</i>	le nombre de CV du véhicule
<i>basse</i>	la tranche basse du barème pour ce nombre de CV, i.e le nombre de km parcouru < 5000
<i>moyenne</i>	la tranche moyenne du barème pour ce nombre de CV, i.e le nombre de km parcouru compris entre 5000 et 20000
<i>haute</i>	la tranche haute du barème pour ce nombre de CV, i.e le nombre de km parcouru > 20000
<i>fixe</i>	la part fixe de l'indemnité pour ce nombre de CV pour la tranche moyenne
<i>context</i>	le contexte de l'application (pour sauvegarder en local)

Déclarée dans la catégorie [BaremeAuto\(DataModel\)](#).

3.3.1.2 + (BaremeAuto *) selectBaremeAuto : (NSString*) cylindre andContext : (NSManagedObjectContext*) context

Fonction permettant de sélectionner un barème automobile.

Paramètres

<i>cylindre</i>	le nombre de CV du véhicule
<i>context</i>	le contexte de l'application (pour sauvegarder en local)

Renvoie

le barème automobile correspondant à cette cylindrée

Déclarée dans la catégorie [BaremeAuto\(DataModel\)](#).

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

— ProjetIndus/Phileas_Gestion/BaremeAuto.h

3.4 Référence de la catégorie BaremeAuto(CoreDataGeneratedAccessors)

Méthodes d'instance

- (void) - **addIndemnitesBaremeObject** :
- (void) - **removeIndemnitesBaremeObject** :
- (void) - **addIndemnitesBareme** :
- (void) - **removeIndemnitesBareme** :

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

— ProjetIndus/Phileas_Gestion/BaremeAuto.h

3.5 Référence de la catégorie BaremeAuto(DataModel)

Méthodes d'instance

- (void) - **initWithName :trancheBasse :trancheMoyenne :trancheHaute :fixe :andContext** :
Fonction permettant de créer un barème automobile servant pour calculer les indemnités kilométriques.

Méthodes de classe

- ([BaremeAuto](#) *) + **selectBaremeAuto :andContext** :
Fonction permettant de sélectionner un barème automobile.

3.5.1 Documentation des méthodes

3.5.1.1 - (void) initWithName : (NSString*) *cylindre* trancheBasse :(NSNumber*) *basse* trancheMoyenne :(NSNumber*)
moyenne trancheHaute :(NSNumber*) *haute* fixe :(NSNumber*) *fixe* andContext :(NSManagedObjectContext*) *context*

Fonction permettant de créer un barème automobile servant pour calculer les indemnités kilométriques.

Paramètres

<i>cylindre</i>	le nombre de CV du véhicule
<i>basse</i>	la tranche basse du barème pour ce nombre de CV, i.e le nombre de km parcouru < 5000
<i>moyenne</i>	la tranche moyenne du barème pour ce nombre de CV, i.e le nombre de km parcouru compris entre 5000 et 20000
<i>haute</i>	la tranche haute du barème pour ce nombre de CV, i.e le nombre de km parcouru > 20000
<i>fixe</i>	la part fixe de l'indemnité pour ce nombre de CV pour la tranche moyenne
<i>context</i>	le contexte de l'application (pour sauvegarder en local)

Dérive la classe [BaremeAuto](#).

3.5.1.2 + (BaremeAuto *) selectBaremeAuto : (NSString*) cylindre andContext :(NSManagedObjectContext*) context

Fonction permettant de sélectionner un barème automobile.

Paramètres

<i>cylindre</i>	le nombre de CV du véhicule
<i>context</i>	le contexte de l'application (pour sauvegarder en local)

Renvoi

le barème automobile correspondant à cette cylindrée

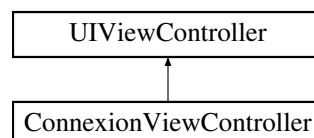
Dérive la classe [BaremeAuto](#).

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

- ProjetIndus/Phileas_Gestion/BaremeAuto+DataModel.h
- ProjetIndus/Phileas_Gestion/BaremeAuto+DataModel.m

3.6 Référence de la classe ConnexionViewController

Graphe d'héritage de ConnexionViewController :



Méthodes d'instance

- (IBAction) - [Connexion](#) :
Cette fonction sert à se connecter à l'application.
- (void) - [SauverUtilisateur](#)
Cette fonction permet de sauvegarder l'utilisateur qui vient de se connecter dans le CoreData afin qu'il reste connecté.
- (void) - [alerteIdentifiants](#)
Cette fonction sert à afficher une alerte si les identifiants de l'utilisateur ne sont pas les bons.

Propriétés

- [AppDelegate](#) * **appDelegate**
- NSManagedObjectContext * **context**
- [User](#) * **utilisateur**
L'utilisateur connecté.
- IBOutlet UITextField * **pseudoTextField**
- IBOutlet UITextField * **mdpTextField**

3.6.1 Documentation des méthodes

3.6.1.1 -(IBAction) Connexion : (id) sender

Cette fonction sert à se connecter à l'application.

Pour cela, il faut être connecté à internet et au moyen de la fonction user de l'api nous allons vérifier que les identifiants fourni correspondent à ceux d'un utilisateur enregistré sur l'application web.

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

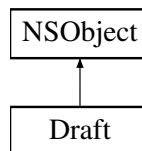
- ProjetIndus/Phileas_Gestion/ConnexionViewController.h
- ProjetIndus/Phileas_Gestion/ConnexionViewController.m

3.7 Référence de la classe Draft

Classe servant à la communication avec l'api Phileas via RestKit.

```
#import <Draft.h>
```

Graphe d'héritage de Draft :



Méthodes de classe

- (RKObjectMapping *) + [mapping](#)
Cette fonction sert à effectuer rapidement une map RestKit pour la fonction draft.

Propriétés

- NSNumber * **def_id**
- NSString * **amount**
- NSString * **date**
- NSString * **km**
- NSData * **receipt**
- NSString * **com**
- NSString * **from**
- NSString * **to**
- NSString * **localisation**

3.7.1 Description détaillée

Classe servant à la communication avec l'api Phileas via RestKit.

Cette classe sert à envoyer des données relatives à un frais via la commande draft.

3.7.2 Documentation des méthodes

3.7.2.1 + (RKObjectMapping *) mapping

Cette fonction sert à effectuer rapidement une map RestKit pour la fonction draft.

Cela permet de connecter la class draft de l'application à la fonction draft de l'api.

Renvoie

mapping le RKObjectMapping pour la fonction draft.

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

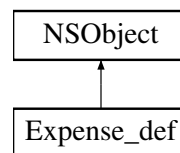
- ProjetIndus/Draft.h
- ProjetIndus/Draft.m

3.8 Référence de la classe Expense_def

Classe servant à la communication avec l'api Phileas via RestKit.

```
#import <Expense-def.h>
```

Graphe d'héritage de Expense_def :

**Propriétés**

- NSNumber * **expense_def_id**
- NSString * **name_fr**

3.8.1 Description détaillée

Classe servant à la communication avec l'api Phileas via RestKit.

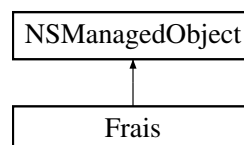
Cette classe sert à récupérer les type de frais via la commande expense-def.

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

- ProjetIndus/Expense-def.h

3.9 Référence de la classe Frais

Graphe d'héritage de Frais :

**Méthodes d'instance**

- (Frais *) - initWithDate :localisation :type :image :montant :commentaire :andContext :
Fonction permettant de créer un frais.
- (void) - updateFrais :localisation :type :image :montant :commentaire :andContext :
Fonction permettant de modifier un frais.
- (void) - addIndemniteK :
Fonction permettant d'ajouter une indemnité kilométrique à un frais (Utilisé si le frais est de type indemnité kilométrique).

Propriétés

- NSString * **commentaire**
- NSDate * **date**
- NSData * **image**
- NSString * **localisation**
- NSNumber * **montant**
- IndemniteK * **indemniteKFrais**
- User * **owner**
- Type * **typeFrais**

3.9.1 Documentation des méthodes

3.9.1.1 - (void) addIndemniteK : (IndemniteK*) *indemnite*

Fonction permettant d'ajouter une indemnité kilométrique à un frais (Utilisé si le frais est de type indemnité kilométrique).

Paramètres

<i>indemnite</i>	une indemnité kilométrique à ajouter au frais
------------------	---

Déclarée dans la catégorie [Frais\(DataModel\)](#).

3.9.1.2 - (Frais *) initWithDate : (NSDate*) *date* localisation :(NSString*) *loc* type :(NSString *) *type* image :(NSData*) *img* montant :(NSNumber*) *montant* commentaire :(NSString*) *com* andContext :(NSManagedObjectContext*) *context*

Fonction permettant de créer un frais.

Paramètres

<i>date</i>	la date du frais
<i>loc</i>	la localisation de l'utilisateur
<i>type</i>	le type de frais, ce paramètre est obligatoire
<i>img</i>	le justificatif
<i>montant</i>	le montant du frais
<i>com</i>	le commentaire associé à ce frais
<i>context</i>	le contexte de l'application (pour sauvegarder en local)

Renvoie

le frais créé

Déclarée dans la catégorie [Frais\(DataModel\)](#).

3.9.1.3 - (void) updateFrais : (NSDate*) *date* localisation :(NSString*) *loc* type :(NSString *) *type* image :(NSData*) *img* montant :(NSNumber*) *montant* commentaire :(NSString*) *com* andContext :(NSManagedObjectContext*) *context*

Fonction permettant de modifier un frais.

Paramètres

<i>date</i>	la date du frais
<i>loc</i>	la localisation de l'utilisateur
<i>type</i>	le type de frais, ce paramètre est obligatoire
<i>img</i>	le justificatif

<i>montant</i>	le montant du frais
<i>com</i>	le commentaire associé à ce frais
<i>context</i>	le contexte de l'application (pour sauvegarder en local)

Déclarée dans la catégorie [Frais\(DataModel\)](#).

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

— `ProjetIndus/Phileas_Gestion/Frais.h`

3.10 Référence de la catégorie Frais(DataModel)

Méthodes d'instance

- ([Frais *](#)) - [initWithDate :localisation :type :image :montant :commentaire :andContext :](#)
Fonction permettant de créer un frais.
- (void) - [updateFrais :localisation :type :image :montant :commentaire :andContext :](#)
Fonction permettant de modifier un frais.
- (void) - [addIndemnitéK :](#)
Fonction permettant d'ajouter une indemnité kilométrique à un frais (Utilisé si le frais est de type indemnité kilométrique).

3.10.1 Documentation des méthodes

3.10.1.1 - (void) addIndemnitéK : (IndemnitéK*) indemnité

Fonction permettant d'ajouter une indemnité kilométrique à un frais (Utilisé si le frais est de type indemnité kilométrique).

Paramètres

<i>indemnité</i>	une indemnité kilométrique à ajouter au frais
------------------	---

Dérive la classe [Frais](#).

3.10.1.2 - ([Frais *](#)) initWithDate : (NSDate*) date localisation : (NSString*) loc type : (NSString*) type image : (NSData*) img montant : (NSNumber*) montant commentaire : (NSString*) com andContext : (NSManagedObjectContext*) context

Fonction permettant de créer un frais.

Paramètres

<i>date</i>	la date du frais
<i>loc</i>	la localisation de l'utilisateur
<i>type</i>	le type de frais, ce paramètre est obligatoire
<i>img</i>	le justificatif
<i>montant</i>	le montant du frais
<i>com</i>	le commentaire associé à ce frais
<i>context</i>	le contexte de l'application (pour sauvegarder en local)

Renvoie

le frais créé

Dérive la classe [Frais](#).

3.10.1.3 - (void) updateFrais : (NSDate*) date localisation : (NSString*) loc type : (NSString*) type image : (NSData*) img montant : (NSNumber*) montant commentaire : (NSString*) com andContext : (NSManagedObjectContext*) context

Fonction permettant de modifier un frais.

Paramètres

<i>date</i>	la date du frais
<i>loc</i>	la localisation de l'utilisateur
<i>type</i>	le type de frais, ce paramètre est obligatoire
<i>img</i>	le justificatif
<i>montant</i>	le montant du frais
<i>com</i>	le commentaire associé à ce frais
<i>context</i>	le contexte de l'application (pour sauvegarder en local)

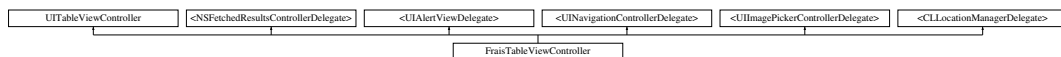
Dérive la classe [Frais](#).

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

- [ProjetIndus/Phileas_Gestion/Frais+DataModel.h](#)
- [ProjetIndus/Phileas_Gestion/Frais+DataModel.m](#)

3.11 Référence de la classe FraisTableViewController

Graphe d'héritage de FraisTableViewController :



Méthodes d'instance

- (IBAction) - [changerType](#) :
Cette fonction permet de lancer une alerte au clic sur le bouton [changerType](#) afin de choisir le type du frais parmi la liste de tous les types.
- (IBAction) - [choisirImage](#) :
Cette fonction permet de lancer une alerte au clic sur le bouton justificatif pour choisir de prendre une photo ou de sélectionner une image dans la librairie.
- (IBAction) - [saisir](#) :
Cette fonction sert à sauvegarder le brouillon en local au clic sur le bouton saisir ou à modifier un brouillon au clic sur le bouton modifier.
- (IBAction) - [envoyer](#) :
Cette fonction permet d'envoyer le brouillon à l'application web de phileas.
- (void) - [creerTypesFrais](#)
Cette fonction sert à récupérer les différents types de frais possibles depuis la BD phileas et à les stocker sur l'appareil.
- (void) - [creerBaremesAuto](#)
Cette fonction sert à créer les différents barèmes kilométriques et à les stocker sur l'appareil.
- (void) - [chargerListeTypesFrais](#)
Cette fonction permet de charger depuis le CoreData tous les types de frais existant.
- (void) - [changerDeDate](#) :
Cette fonction permet de changer la date du frais après l'avoir sélectionnée dans un datePicker.
- (void) - [compterNotifs](#)
Cette fonction sert à récupérer le nombre de messages récents afin d'afficher un badge sur les notifications.

Propriétés

- AppDelegate * [appDelegate](#)
- NSManagedObjectContext * [context](#)
- CLLocationManager * [locationManager](#)
- CLGeocoder * [geocoder](#)
- CLPlacemark * [placemark](#)
- NSArray * [listeType](#)
Un tableau qui va contenir la liste des type de frais.
- NSMutableString * [montant](#)
Une chaîne de caractère contenant le montant du frais.

- `IndemniteK * indemniteK`
L'indemnité kilométrique du frais si elle existe.
- `Frais * fraisChoisi`
Le frais qui va être créé ou modifié.
- `User * utilisateur`
L'utilisateur connecté.
- `NSArray * listeExpenseDef`
Un tableau qui va contenir la liste des expense-def récupérée depuis la BD.
- `IBOutlet UINavigationController * navBar`
- `IBOutlet UITextField * dateTexte`
- `IBOutlet UILabel * localisationLbl`
- `IBOutlet UIButton * typeF`
- `IBOutlet UITextField * montantTextField`
- `IBOutlet UIButton * modifierIndemnite`
- `IBOutlet UITextView * comTextArea`
- `IBOutlet UIImageView * image`
- `IBOutlet UIButton * saisirButton`
- `UIDatePicker * pickerViewDate`
- `NSDate * dateToChoose`

3.11.1 Documentation des méthodes

3.11.1.1 - (void) compterNotifs

Cette fonction sert à récupérer le nombre de messages récents afin d'afficher un badge sur les notifications.

Elle utilise la class message et la fonction message de l'api.

3.11.1.2 - (void) creerBaremesAuto

Cette fonction sert à créer les différents barèmes kilométriques et à les stocker sur l'appareil.

Les barèmes kilométriques sont créés uniquement s'il n'existe pas déjà sur l'appareil, donc au premier lancement de l'application. Il sera nécessaire de modifier les différents chiffres chaque année.

3.11.1.3 - (void) creerTypesFrais

Cette fonction sert à récupérer les différents types de frais possibles depuis la BD phileas et à les stocker sur l'appareil.

Les types de frais sont créés uniquement s'il n'existe pas déjà sur l'appareil, donc au premier lancement de l'application.

3.11.1.4 - (IBAction) envoyer : (id) sender

Cette fonction permet d'envoyer le brouillon à l'application web de phileas.

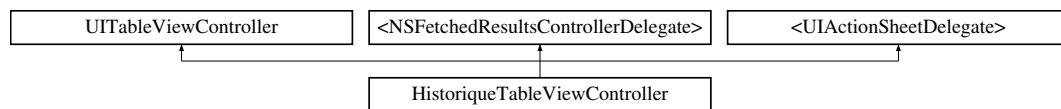
Cette fonction utilise le framework RestKit et la fonction draft de l'api de phileas.

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

- `ProjetIndus/Phileas_Gestion/FraisTableViewController.h`
- `ProjetIndus/Phileas_Gestion/FraisTableViewController.m`

3.12 Référence de la classe HistoriqueTableViewController

Graphe d'héritage de HistoriqueTableViewController :



Méthodes d'instance

- (IBAction) - [editer](#) :
Cette fonction permet de passer en mode édition pour pouvoir supprimer plusieurs brouillons en une seule fois.
- (IBAction) - [annuler](#) :
Cette fonction permet de quitter le mode édition et pouvoir de nouveau sélectionner un frais à modifier.
- (IBAction) - [supprimer](#) :
Cette fonction permet de lancer un actionSheet afin de supprimer un ou plusieurs brouillons.
- (void) - [chargerListeFrais](#)
Cette fonction permet de charger la liste des brouillons enregistrés dans le CoreData.

Propriétés

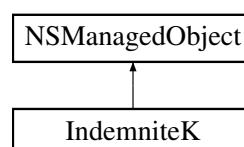
- AppDelegate * [appDelegate](#)
- NSManagedObjectContext * [context](#)
- NSFetchedResultsController * [fetchedResultsController](#)
- NSIndexPath * [selectedFactions](#)
- IBOutlet UINavigationController * [navBar](#)
- IBOutlet UIBarButtonItem * [editerBouton](#)
- IBOutlet UIBarButtonItem * [annulerBouton](#)
- IBOutlet UIBarButtonItem * [supprimerBouton](#)

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

- ProjetIndus/Phileas_Gestion/HistoriqueTableViewController.h
- ProjetIndus/Phileas_Gestion/HistoriqueTableViewController.m

3.13 Référence de la classe IndemnitéK

Graphe d'héritage de IndemnitéK :



Méthodes d'instance

- (void) - [updateIndemnitéK :villeArrivée :allezR :baremeAuto :andDistance](#) :
Fonction permettant de modifier une indemnité kilométrique.

Propriétés

- NSNumber * [allezRetour](#)
- NSNumber * [distance](#)
- NSString * [villeArrivée](#)
- NSString * [villeDepart](#)
- [BaremeAuto](#) * [cylindree](#)
- [Frais](#) * [fraisIndemnitéK](#)

3.13.1 Documentation des méthodes

3.13.1.1 - (void) updateIndemnitéK : (NSString*) *villeD* villeArrivée :(NSString*) *villeA* allerR :(NSNumber*) *AR*
baremeAuto :(BaremeAuto*) *cyllndree* andDistance :(NSNumber *) *distance*

Fonction permettant de modifier une indemnité kilométrique.

Paramètres

<i>villeD</i>	la ville de départ, paramètre requis
<i>villeA</i>	la ville d'arrivée, paramètre requis
<i>cylindree</i>	le nombre de CV,paramètre requis
<i>distance</i>	la distance parcouru en km, paramètre requis

Déclarée dans la catégorie [IndemnitéK\(DataModel\)](#).

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

— `ProjetIndus/Phileas_Gestion/IndemnitéK.h`

3.14 Référence de la catégorie IndemnitéK(DataModel)

Méthodes d'instance

— (void) - [updateIndemnitéK :villeArrivée :allezR :baremeAuto :andDistance :](#)
Fonction permettant de modifier une indemnité kilométrique.

3.14.1 Documentation des méthodes

3.14.1.1 - (void) updateIndemnitéK : (NSString*) *villeD* villeArrivée :(NSString*) *villeA* allezR :(NSNumber*) *AR*
baremeAuto :(BaremeAuto*) *cylindree* andDistance :(NSNumber *) *distance*

Fonction permettant de modifier une indemnité kilométrique.

Paramètres

<i>villeD</i>	la ville de départ, paramètre requis
<i>villeA</i>	la ville d'arrivée, paramètre requis
<i>cylindree</i>	le nombre de CV,paramètre requis
<i>distance</i>	la distance parcouru en km, paramètre requis

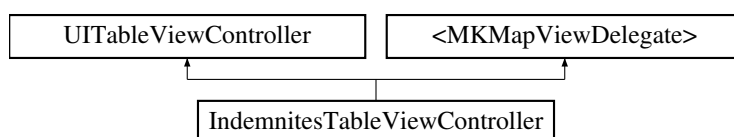
Dérive la classe [IndemnitéK](#).

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

— `ProjetIndus/Phileas_Gestion/IndemnitéK+DataModel.h`
— `ProjetIndus/Phileas_Gestion/IndemnitéK+DataModel.m`

3.15 Référence de la classe IndemnitésTableViewController

Graphe d'héritage de IndemnitésTableViewController :



Méthodes d'instance

— (IBAction) - [choisirPuissance :](#)
Cette fonction permet de lancer une alerte au clic sur le bouton choisirCylindree pour choisir le nombre de CV du véhicule.

— (IBAction) - [changementTypeTrajet :](#)

Cette fonction permet de recharger la vue au changement d'état du switch pour recalculer directement la distance et le coût de l'indemnité.

- (void) - [chargementListeBaremes](#)

Cette fonction permet de charger la liste des barèmes automobiles enregistrés dans le CoreData.

- (void) - [calculDistance](#)

Cette fonction permet de calculer la distance entre deux emplacements géographiques.

- (void) - [calculMontant](#)

Cette fonction permet de calculer le coût d'une indemnité kilométrique en fonction de la cylindrée et de la distance.

Propriétés

- [AppDelegate](#) * **appDelegate**
- NSObject * **context**
- NSArray * **resultat**
- NSMutableString * **montant**

Une chaîne de caractère contenant le montant du frais.

- [IndemniteK](#) * **indemnitéK**

L'indemnité kilométrique du frais.

- [BaremeAuto](#) * **ba**

Le barème automobile associé à ce frais.

- NSMutableArray * **depart**

Un tableau contenant la liste des villes possibles répondant à la recherche pour celle de départ.

- NSMutableArray * **arrivee**

Un tableau contenant la liste des villes possibles répondant à la recherche pour celle d'arrivée.

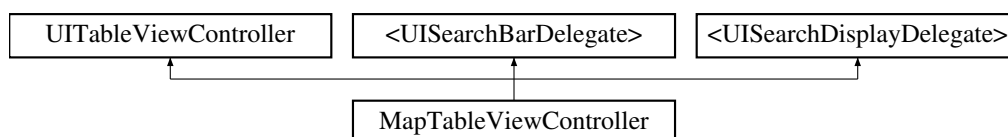
- IBOutlet UITableViewCell * **departCell**
- IBOutlet UITableViewCell * **arriveeCell**
- IBOutlet UITextField * **distanceTextField**
- IBOutlet UIButton * **cvButton**
- IBOutlet UISwitch * **arSwitch**
- IBOutlet UITextField * **montantTextField**

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

- `ProjetIndus/Phileas_Gestion/IndemnitesTableViewController.h`
- `ProjetIndus/Phileas_Gestion/IndemnitesTableViewController.m`

3.16 Référence de la classe MapTableViewController

Graphique d'héritage de MapTableViewController :



Propriétés

- NSMutableArray * **lieu**

Un tableau contenant la liste des villes possibles répondant à la recherche.

- IBOutlet UISearchBar * **barreRecherche**

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

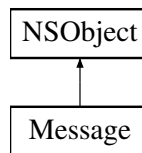
- `ProjetIndus/Phileas_Gestion/MapTableViewController.h`

3.17 Référence de la classe Message

Classe servant à la communication avec l'api Phileas via RestKit.

```
#import <Message.h>
```

Graphe d'héritage de Message :



Propriétés

- NSString * **message_title**
- NSString * **message_body**
- NSString * **created**

3.17.1 Description détaillée

Classe servant à la communication avec l'api Phileas via RestKit.

Cette classe sert à récupérer les messages via la commande message.

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

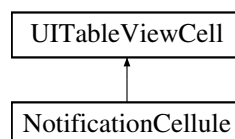
- ProjetIndus/Message.h

3.18 Référence de la classe NotificationCellule

Cette classe sert à remplir le contenu présent sur chaque ligne du tableView notifications.

```
#import <NotificationCellule.h>
```

Graphe d'héritage de NotificationCellule :



Propriétés

- IBOutlet UILabel * **titreLabel**
- IBOutlet UILabel * **dateLabel**
- IBOutlet UILabel * **msgLabel**

3.18.1 Description détaillée

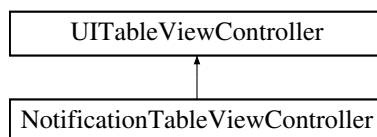
Cette classe sert à remplir le contenu présent sur chaque ligne du tableView notifications.

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

- ProjetIndus/Phileas_Gestion/NotificationCellule.h

3.19 Référence de la classe NotificationTableViewController

Graphe d'héritage de NotificationTableViewController :



Méthodes d'instance

- (void) - [loadFrais](#)
Cette fonction sert à charger tous les frais disponible sur l'application web.
- (void) - [chargerMessages](#)
Cette fonction sert à charger tous les messages récents récupérée depuis l'application web.

Propriétés

- [AppDelegate](#) * **appDelegate**
- [NSManagedObjectContext](#) * **context**
- [User](#) * **utilisateur**
L'utilisateur connecté.
- [NSArray](#) * **listeFrais**
Un tableau contenant la liste des frais de l'utilisateur récupérée depuis l'application Web.
- [NSArray](#) * **listeMessages**
Un tableau contenant la liste des messages de l'utilisateur récupérée depuis l'application Web.
- [IBOutlet UINavigationController](#) * **navBar**

3.19.1 Documentation des méthodes

3.19.1.1 - (void) chargerMessages

Cette fonction sert à charger tous les messages récents récupérée depuis l'application web.
Elle utilise la class message et la fonction message de l'api.

3.19.1.2 - (void) loadFrais

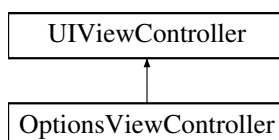
Cette fonction sert à charger tous les frais disponible sur l'application web.
Elle utilise la class sheet et la fonction sheet de l'api.

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

- [ProjetIndus/Phileas_Gestion/NotificationTableViewController.h](#)
- [ProjetIndus/Phileas_Gestion/NotificationTableViewController.m](#)

3.20 Référence de la classe OptionsViewController

Graphe d'héritage de OptionsViewController :



Méthodes d'instance

- (IBAction) - [Deconnexion](#) :
Cette fonction permet de se déconnecter.

Propriétés

- [AppDelegate](#) * **appDelegate**
- [NSManagedObjectContext](#) * **context**
- [User](#) * **utilisateur**
L'utilisateur connecté.
- IBOutlet UILabel * **pseudoLbl**

3.20.1 Documentation des méthodes

3.20.1.1 - (IBAction) Deconnexion : (id) sender

Cette fonction permet de se déconnecter.

Pour cela, elle supprime totalement l'utilisateur enregistré en local ainsi que ses brouillons locaux.

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

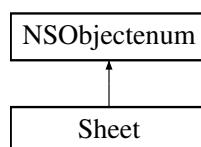
- `ProjetIndus/Phileas_Gestion/OptionsViewController.h`
- `ProjetIndus/Phileas_Gestion/OptionsViewController.m`

3.21 Référence de la classe Sheet

Classe servant à la communication avec l'api Phileas via RestKit.

```
#import <Sheet.h>
```

Graphe d'héritage de Sheet :



Méthodes d'instance

- (NSString *) - [getLabelFromStatusId](#) :
Cette fonction permet de convertir l'id relatif à l'état d'un frais en une chaîne de caractère correspondant à cette id.

Types protégés

- typedef NSInteger * **status**

Attributs protégés

- **validated** = 2
- **rejected** = 3
- **exportable** = 4
- **draft**

Propriétés

- NSString * **creation_date**
- NSNumber * **latest_status_id**
- NSString * **object**

3.21.1 Description détaillée

Classe servant à la communication avec l'api Phileas via RestKit.

Cette classe sert pour récupérer des données d'un frais via la commande sheet.

3.21.2 Documentation des méthodes

3.21.2.1 - (NSString *) getLabelFromStatusId : (NSNumber*) idEtat

Cette fonction permet de convertir l'id relatif à l'état d'un frais en une chaîne de caractère correspondant à cette id.

Paramètres

<i>idEtat</i>	l'identifiant relatif à l'état d'un frais
---------------	---

Renvoie

label le nom de cet état

3.21.3 Documentation des données membres

3.21.3.1 - draft [protected]

Valeur initiale :

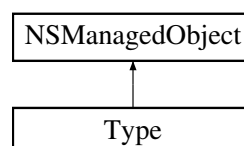
```
= 5
}
```

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

- ProjetIndus/Sheet.h
- ProjetIndus/Sheet.m

3.22 Référence de la classe Type

Graphe d'héritage de Type :



Méthodes d'instance

- (void) - **addFraisTypeObject** :
 - (void) - **removeFraisTypeObject** :
 - (void) - **addFraisType** :
 - (void) - **removeFraisType** :
 - (void) - **initWithName :andID :andContext** :
- Fonction permettant de créer un type de frais.*

Méthodes de classe

- (Type *) + **selectTypeFrais :andContext :**
Fonction permettant de sélectionner un type de frais.

Propriétés

- NSString * **lib**
- NSNumber * **idType**
- NSSet * **fraisType**

3.22.1 Documentation des méthodes

3.22.1.1 - (void) initWithName : (NSString*) lib andID : (NSNumber*) idType andContext : (NSManagedObjectContext*) context

Fonction permettant de créer un type de frais.

Paramètres

<i>lib</i>	le libelé du type de frais, paramètre obligatoire
<i>idType</i>	l'id du type de frais, paramètre obligatoire
<i>context</i>	le contexte de l'application (pour sauvegarder en local)

Déclarée dans la catégorie [Type\(DataModel\)](#).

3.22.1.2 + (Type *) selectTypeFrais : (NSString*) lib andContext : (NSManagedObjectContext*) context

Fonction permettant de sélectionner un type de frais.

Paramètres

<i>lib</i>	le libelé du type de frais, ce paramètre est obligatoire
<i>context</i>	le contexte de l'application (pour sauvegarder en local)

Renvoie

le type de frais associé au libelé

Déclarée dans la catégorie [Type\(DataModel\)](#).

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

- ProjetIndus/Phileas_Gestion/Type.h

3.23 Référence de la catégorie Type(CoreDataGeneratedAccessors)

Méthodes d'instance

- (void) - **addFraisTypeObject :**
- (void) - **removeFraisTypeObject :**
- (void) - **addFraisType :**
- (void) - **removeFraisType :**

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

- ProjetIndus/Phileas_Gestion/Type.h

3.24 Référence de la catégorie Type(DataModel)

Méthodes d'instance

- (void) - `initWithName :andID :andContext :`
Fonction permettant de créer un type de frais.

Méthodes de classe

- (Type *) + `selectTypeFrais :andContext :`
Fonction permettant de sélectionner un tyoe de frais.

3.24.1 Documentation des méthodes

3.24.1.1 - (void) `initWithName : (NSString*) lib andID :(NSNumber*) idType andContext :(NSManagedObjectContext*) context`

Fonction permettant de créer un type de frais.

Paramètres

<i>lib</i>	le libelé du type de frais, paramètre obligatoire
<i>idType</i>	l'id du type de frais, paramètre obligatoire
<i>context</i>	le contexte de l'application (pour sauvegarder en local)

Dérive la classe `Type`.

3.24.1.2 + (Type *) `selectTypeFrais : (NSString*) lib andContext :(NSManagedObjectContext*) context`

Fonction permettant de sélectionner un tyoe de frais.

Paramètres

<i>lib</i>	le libelé du type de frais, ce paramètre est obligatoire
<i>context</i>	le contexte de l'application (pour sauvegarder en local)

Renvoi

le type de frais associé au libelé

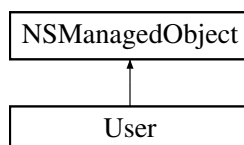
Dérive la classe `Type`.

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

- `ProjetIndus/Phileas_Gestion/Type+DataModel.h`
- `ProjetIndus/Phileas_Gestion/Type+DataModel.m`

3.25 Référence de la classe User

Graphe d'héritage de User :



Méthodes d'instance

- (void) - `addFraisUserObject :`
- (void) - `removeFraisUserObject :`

- (void) - **addFraisUser** :
- (void) - **removeFraisUser** :
- (User *) - **initWithPseudo :motDePasse :andContext :**
Fonction permettant de créer un utilisateur.

Propriétés

- NSString * **mdp**
- NSString * **pseudo**
- NSSet * **fraisUser**

3.25.1 Documentation des méthodes

- 3.25.1.1 - (User *) **initWithPseudo :** (NSString*) *pseudo* **motDePasse :** (NSString*) *mdp*
andContext : (NSManagedObjectContext*) *context*

Fonction permettant de créer un utilisateur.

Paramètres

<i>pseudo</i>	le pseudo de l'utilisateur
<i>mdp</i>	le mot de passe de l'utilisateur
<i>context</i>	le contexte de l'application (pour sauvegarder en local)

Renvoie

l'utilisateur créé

Déclarée dans la catégorie [User\(DataModel\)](#).

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

- ProjetIndus/Phileas_Gestion/User.h

3.26 Référence de la catégorie User(CoreDataGeneratedAccessors)

Méthodes d'instance

- (void) - **addFraisUserObject** :
- (void) - **removeFraisUserObject** :
- (void) - **addFraisUser** :
- (void) - **removeFraisUser** :

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

- ProjetIndus/Phileas_Gestion/User.h

3.27 Référence de la catégorie User(DataModel)

Méthodes d'instance

- (User *) - **initWithPseudo :motDePasse :andContext :**
Fonction permettant de créer un utilisateur.

3.27.1 Documentation des méthodes

- 3.27.1.1 - (User *) **initWithPseudo :** (NSString*) *pseudo* **motDePasse :** (NSString*) *mdp*
andContext : (NSManagedObjectContext*) *context*

Fonction permettant de créer un utilisateur.

Paramètres

<i>pseudo</i>	le pseudo de l'utilisateur
<i>mdp</i>	le mot de passe de l'utilisateur
<i>context</i>	le contexte de l'application (pour sauvegarder en local)

Renvoie

l'utilisateur créé

Dérive la classe [User](#).

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

- `ProjetIndus/Phileas_Gestion/User+DataModel.h`
- `ProjetIndus/Phileas_Gestion/User+DataModel.m`

Index

addIndemnitéK :
 Frais, 12
 Frais(DataModel), 13
AppDelegate, 5
 chargerUtilisateur, 5
 estConnecté, 5
AppDelegate(), 6

BaremeAuto, 6
 initWithName :trancheBasse :trancheMoyenne↔
 :trancheHaute :fixe :andContext :, 6
 selectBaremeAuto :andContext :, 7
BaremeAuto(CoreDataGeneratedAccessors), 7
BaremeAuto(DataModel), 7
 initWithName :trancheBasse :trancheMoyenne↔
 :trancheHaute :fixe :andContext :, 7
 selectBaremeAuto :andContext :, 9

chargerMessages
 NotificationTableViewController, 21
chargerUtilisateur
 AppDelegate, 5
compterNotifs
 FraisTableViewController, 15
Connexion :
 ConnexionViewController, 10
ConnexionViewController, 9
 Connexion :, 10
créerBaremesAuto
 FraisTableViewController, 15
créerTypesFrais
 FraisTableViewController, 15

Deconnexion :
 OptionsViewController, 22
Draft, 10
 mapping, 10
draft
 Sheet, 23

envoyer :
 FraisTableViewController, 15
estConnecté
 AppDelegate, 5
Expense_def, 11

Frais, 11
 addIndemnitéK :, 12
 initWithDate :localisation :type :image :montant↔
 :commentaire :andContext :, 12

updateFrais :localisation :type :image :montant↔
 :commentaire :andContext :, 12
Frais(DataModel), 13
 addIndemnitéK :, 13
 initWithDate :localisation :type :image :montant↔
 :commentaire :andContext :, 13
 updateFrais :localisation :type :image :montant↔
 :commentaire :andContext :, 13
FraisTableViewController, 14
 compterNotifs, 15
 créerBaremesAuto, 15
 créerTypesFrais, 15
 envoyer :, 15

getLabelFromStatusId :
 Sheet, 23

HistoriqueTableViewController, 15

IndemnitéK, 16
 updateIndemnitéK :villeArrivée :allezR :bareme↔
 Auto :andDistance :, 17
IndemnitéK(DataModel), 18
 updateIndemnitéK :villeArrivée :allezR :bareme↔
 Auto :andDistance :, 18
IndemnitésTableViewController, 18
initWithDate :localisation :type :image :montant↔
 :commentaire :andContext :
 Frais, 12
 Frais(DataModel), 13
initWithName :andID :andContext :
 Type, 24
 Type(DataModel), 25
initWithName :trancheBasse :trancheMoyenne↔
 :trancheHaute :fixe :andContext :
 BaremeAuto, 6
 BaremeAuto(DataModel), 7
initWithPseudo :motDePasse :andContext :
 User, 26
 User(DataModel), 26

loadFrais
 NotificationTableViewController, 21

MapTableViewController, 19
mapping
 Draft, 10
Message, 19

NotificationCellule, 20
NotificationTableViewController, 20

- chargerMessages, [21](#)
- loadFrais, [21](#)
- OptionsViewController, [21](#)
 - Deconnexion :, [22](#)
- selectBaremeAuto :andContext :
 - BaremeAuto, [7](#)
 - BaremeAuto(DataModel), [9](#)
- selectTypeFrais :andContext :
 - Type, [24](#)
 - Type(DataModel), [25](#)
- Sheet, [22](#)
 - draft, [23](#)
 - getLabelFromStatusId :, [23](#)
- Type, [23](#)
 - initWithName :andID :andContext :, [24](#)
 - selectTypeFrais :andContext :, [24](#)
- Type(CoreDataGeneratedAccessors), [24](#)
- Type(DataModel), [24](#)
 - initWithName :andID :andContext :, [25](#)
 - selectTypeFrais :andContext :, [25](#)
- updateFrais :localisation :type :image :montant↵
 - :commentaire :andContext :
 - Frais, [12](#)
 - Frais(DataModel), [13](#)
- updateIndemnitéK :villeArrivée :allezR :baremeAuto↵
 - :andDistance :
 - IndemnitéK, [17](#)
 - IndemnitéK(DataModel), [18](#)
- User, [25](#)
 - initWithPseudo :motDePasse :andContext :, [26](#)
- User(CoreDataGeneratedAccessors), [26](#)
- User(DataModel), [26](#)
 - initWithPseudo :motDePasse :andContext :, [26](#)