GESTION IFNTI

Version v1.0

Malia TCHABANA, TEOURI Toure Ydaou, Abdoul Malik KONDI, Wa

janv. 15, 2024

Table des matières:

CHAPITRE 1

Documentation de l'application principale : main

1.1 Modèles

class main.models.**AnneeUniversitaire**(*id*, *annee*, *annee_courante*)

exception DoesNotExist

exception MultipleObjectsReturned

annee

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

annee_courante

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

charge_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

compteetudiant_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`conge_set`

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```python
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`disable()`

`fichedepaie_set`

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```python
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`fournisseur_set`

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```python
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`frais_set`

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```python
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`generateSemeste()`

`static getNiveau`(*semestre_libelle*)

`get_semestres()`

`id`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`objects = <django.db.models.manager.Manager object>`

paiement_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

salaire_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

save(**args*, ***kwargs*)

Save the current instance. Override this in a subclass if you want to control the saving process.

The "force_insert" and "force_update" parameters can be used to insist that the « save » must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

semestre_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

static static_get_current_annee_universitaire()

class main.models.**Charge**(*id*, *dateDebut*, *dateFin*, *personnel*, *frais_de_vie*, *frais_nourriture*, *montant*, *montantEnLettre*, *annee_universitaire*, *compte_bancaire*)

exception DoesNotExist

exception MultipleObjectsReturned

annee_universitaire

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

annee_universitaire_id

compte_bancaire

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

compte_bancaire_id

dateDebut

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

dateFin

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

frais_de_vie

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

frais_nourriture

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

montant

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

montantEnLettre

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

personnel

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

personnel_id

save(**args*, ***kwargs*)

Save the current instance. Override this in a subclass if you want to control the saving process.

The "force_insert" and "force_update" parameters can be used to insist that the « save » must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

class main.models.**Competence**(*id*, *code*, *libelle*, *ue*, *matiere*)

exception DoesNotExist

exception MultipleObjectsReturned

code

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

libelle

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

matiere

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

matiere_id

objects = <django.db.models.manager.Manager object>

ue

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

ue_id

class main.models.**Comptable**(*nom*, *prenom*, *sexe*, *datenaissance*, *lieunaissance*, *contact*, *email*, *adresse*, *prefecture*, *is_active*, *carte_identity*, *nationalite*, *user*, *photo_passport*, *id*, *salaireBrut*, *dernierdiplome*, *nbreJrsCongesRestant*, *nbreJrsConsomme*, *personnel_ptr*)

exception DoesNotExist

exception MultipleObjectsReturned

paiement_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

personnel_ptr

Accessor to the related object on the forward side of a one-to-one relation.

In the example :

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`Restaurant.place` is a `ForwardOneToOneDescriptor` instance.

`personnel_ptr_id`

`save`(*force_insert=False*, *force_update=False*, *using=None*)

Save the current instance. Override this in a subclass if you want to control the saving process.

The "force_insert" and "force_update" parameters can be used to insist that the « save » must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

`class main.models.`**`CompteBancaire`**(*id*, *numero*, *solde_bancaire*, *frais_tenue_de_compte*)

`exception DoesNotExist`

`exception MultipleObjectsReturned`

`charge_set`

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`fichedepaie_set`

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`fournisseur_set`

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`frais_tenue_de_compte`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`id`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`numero`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

paiement_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

salaire_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

solde_bancaire

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class main.models.**CompteEtudiant**(*id*, *etudiant*, *annee_universitaire*, *solde*)

exception DoesNotExist

exception MultipleObjectsReturned

annee_universitaire

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

annee_universitaire_id

etudiant

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

etudiant_id

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

solde

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class main.models.**Conge**(*id*, *nature*, *autre_nature*, *date_et_heure_debut*, *date_et_heure_fin*, *personnel*, *motif_refus*, *valider*, *nombre_de_jours_de_conge*, *annee_universitaire*)

exception DoesNotExist

exception MultipleObjectsReturned

NATURE_CHOICES = [('Congé annuel', 'Congé annuel'), ('Congé de maternité', 'Congé de maternité'), ('Congé de paternité', 'Congé de paternité'), ('Autres', 'Autres')]

VALIDATION_CHOICES = [('Actif', 'Actif'), ('Inactif', 'Inactif'), ('Inconnu', 'Inconnu')]

annee_universitaire

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

annee_universitaire_id

autre_nature

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

date_et_heure_debut

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

date_et_heure_fin

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_nature_display(*, *field=<django.db.models.fields.CharField : nature>*)

get_next_by_date_et_heure_debut(*, *field=<django.db.models.fields.DateField : date_et_heure_debut>*, *is_next=True*, ***kwargs*)

get_next_by_date_et_heure_fin(*, *field=<django.db.models.fields.DateField : date_et_heure_fin>*, *is_next=True*, ***kwargs*)

get_previous_by_date_et_heure_debut(*, *field=<django.db.models.fields.DateField : date_et_heure_debut>*, *is_next=False*, ***kwargs*)

get_previous_by_date_et_heure_fin(*, *field=<django.db.models.fields.DateField : date_et_heure_fin>*, *is_next=False*, ***kwargs*)

get_valider_display(*, *field=<django.db.models.fields.CharField : valider>*)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

motif_refus

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nature

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nombre_de_jours_de_conge

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

personnel

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

personnel_id

save(**args*, ***kwargs*)

Save the current instance. Override this in a subclass if you want to control the saving process.

The "force_insert" and "force_update" parameters can be used to insist that the « save » must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

valider

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class main.models.**CorrespondanceMaquette**(*id*, *nature*, *ancienne*, *nouvelle*)

exception DoesNotExist

exception MultipleObjectsReturned

class Nature(*value*)

An enumeration.

MATIERE = 'M'

UE = 'U'

afficher_nature()

ancienne

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_ancienne()

get_nature_display(***, *field=<django.db.models.fields.CharField : nature>*)

get_nouvelle()

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nature

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nouvelle

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

save()

Save the current instance. Override this in a subclass if you want to control the saving process.

The "force_insert" and "force_update" parameters can be used to insist that the « save » must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

class main.models.DirecteurDesEtudes(nom, prenom, sexe, datenaissance, lieunaissance, contact, email, adresse, prefecture, is_active, carte_identity, nationalite, user, photo_passport, id, salaireBrut, dernierdiplome, nbreJrsCongesRestant, nbreJrsConsomme, personnel_ptr)

exception DoesNotExist

exception MultipleObjectsReturned

delete(*args, **kwargs)

information_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

personnel_ptr

Accessor to the related object on the forward side of a one-to-one relation.

In the example :

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

Restaurant.place is a ForwardOneToOneDescriptor instance.

personnel_ptr_id

save(*args, **kwargs)

Save the current instance. Override this in a subclass if you want to control the saving process.

The "force_insert" and "force_update" parameters can be used to insist that the « save » must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

class main.models.Domaine(id, nom, description)

exception DoesNotExist

exception MultipleObjectsReturned

description

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

generate_code()

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nom

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

parcours_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

class main.models.**Enseignant**(*nom*, *prenom*, *sexe*, *datenaissance*, *lieunaissance*, *contact*, *email*, *adresse*, *prefecture*, *is_active*, *carte_identity*, *nationalite*, *user*, *photo_passport*, *id*, *salaireBrut*, *dernierdiplome*, *nbreJrsCongesRestant*, *nbreJrsConsomme*, *personnel_ptr*, *type*, *specialite*)

CHOIX_TYPE = (('Vacataire', 'Vacataire'), ('Permanent', 'Permanent'))

exception DoesNotExist

exception MultipleObjectsReturned

fichedepaie_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

get_type_display(**, field=<django.db.models.fields.CharField : type>*)

information_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

matiere_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

niveaux()

personnel_ptr

Accessor to the related object on the forward side of a one-to-one relation.

In the example :

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`Restaurant.place` is a `ForwardOneToOneDescriptor` instance.

personnel_ptr_id

save(*force_insert=False*, *force_update=False*, *using=None*)

Save the current instance. Override this in a subclass if you want to control the saving process.

The "force_insert" and "force_update" parameters can be used to insist that the « save » must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

seance_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

seanceplannifier_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

specialite

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

ue_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

class main.models.Etudiant(*nom*, *prenom*, *sexe*, *datenaissance*, *lieunaissance*, *contact*, *email*, *adresse*, *prefecture*, *is_active*, *carte_identity*, *nationalite*, *user*, *photo_passport*, *id*, *seriebac1*, *seriebac2*, *anneeentree*, *anneebac1*, *anneebac2*, *etablissementSeconde*, *francaisSeconde*, *anglaisSeconde*, *mathematiqueSeconde*, *etablissementPremiere*, *francaisPremiere*, *anglaisPremiere*, *mathematiquePremiere*, *etablissementTerminale*, *francaisTerminale*, *anglaisTerminale*, *mathematiqueTerminale*, *delegue*, *passer_semestre_suivant*, *decision_conseil*, *profil*)

CHOIX_SERIE = [('A', 'A'), ('C', 'C'), ('D', 'D'), ('E', 'E'), ('F1', 'F1'), ('F2', 'F2'), ('F3', 'F3'), ('F4', 'F4'), ('G2', 'G2')]

exception DoesNotExist

exception MultipleObjectsReturned

adresse

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

anglaisPremiere

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

anglaisSeconde

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

anglaisTerminale

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

anneebac1

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

anneebac2

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

anneeentree

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

carte_identity

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

compteetudiant_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

contact

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`create_compte_etudiant``()`

`credits_obtenus_semestre``(`*semestre*`)`

`datenaissance`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`decision_conseil`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`delegue`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`email`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`etablissementPremiere`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`etablissementSeconde`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`etablissementTerminale`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`evaluation_set`

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example :

```python
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`francaisPremiere`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`francaisSeconde`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`francaisTerminale`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`generate_email``()`

`static` **`get_Ln`**`(`*semestres*, *annee_universitaire=None*`)`

semestre : liste de chaîne de caractère annee_universitaire : instance de la classe AnneUniversitaire

`static` **`get_etudiants_semestre`**`(`*semestre*, *id_annee_selectionnee=None*`)`

semestre : chaîne de caractère id_annee_selectionnee : entier

`get_niveau_annee``(`*annee_universitaire*`)`

get_semestre_courant()

get_semestres(*type='courant'*, *annee=None*)

get_seriebac1_display(*, *field=<django.db.models.fields.CharField : seriebac1>*)

get_seriebac2_display(*, *field=<django.db.models.fields.CharField : seriebac2>*)

get_sexe_display(*, *field=<django.db.models.fields.CharField : sexe>*)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

is_active

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

lieunaissance

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

mathematiquePremiere

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

mathematiqueSeconde

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

mathematiqueTerminale

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

moyenne_etudiant_matiere(*matiere*, *semestre*)

summary Returns :

type : _description_

moyenne_etudiant_matieres(*semestre*)

moyenne_etudiant_ue(*ue*, *semestre*)

nationalite

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nom

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

note_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

notes_etudiant_matiere(*matiere*, *semestre*)

summary Returns :

type : _description_

objects = <django.db.models.manager.Manager object>

paiement_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

passer_semestre_suivant

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

photo_passport

Just like the FileDescriptor, but for ImageFields. The only difference is assigning the width/height to the width_field/height_field, if appropriate.

prefecture

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

prenom

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

profil

Just like the FileDescriptor, but for ImageFields. The only difference is assigning the width/height to the width_field/height_field, if appropriate.

save(*force_insert=False*, *force_update=False*, *using=None*)

Save the current instance. Override this in a subclass if you want to control the saving process.

The "force_insert" and "force_update" parameters can be used to insist that the « save » must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

seance_auteur

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

seances_presents

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example :

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

semestres

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example :

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

seriebac1

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

seriebac2

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

sexe

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

static static_get_L1(*annee=None*, *semestres=['S1', 'S2']*)

static static_get_L2(*annee=None*, *semestres=['S3', 'S4']*)

static static_get_L3(*annee=None*, *semestres=['S5', 'S6']*)

tuteurs

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example :

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

user

Accessor to the related object on the forward side of a one-to-one relation.

In the example :

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`Restaurant.place` is a `ForwardOneToOneDescriptor` instance.

user_id

class `main.models.`**Evaluation**(*id*, *libelle*, *ponderation*, *date*, *matiere*, *semestre*, *rattrapage*)

exception DoesNotExist

exception MultipleObjectsReturned

afficher_rattrapage()

date

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`etudiants`

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example :

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`get_next_by_date`(**, field=<django.db.models.fields.DateField : date>, is_next=True, **kwargs*)

`get_previous_by_date`(**, field=<django.db.models.fields.DateField : date>, is_next=False, **kwargs*)

`id`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`libelle`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`matiere`

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

`matiere_id`

`note_set`

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`objects = <django.db.models.manager.Manager object>`

`ponderation`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`rattrapage`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`save`(**args, **kwargs*)

Save the current instance. Override this in a subclass if you want to control the saving process.

The "force_insert" and "force_update" parameters can be used to insist that the « save » must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

semestre

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

semestre_id

class main.models.**FicheDePaie**(*id*, *dateDebut*, *dateFin*, *matiere*, *enseignant*, *nombreHeureL1*, *nombreHeureL2*, *nombreHeureL3*, *nombreHeure*, *prixUnitaire*, *montantL1*, *montantL2*, *montantL3*, *montant*, *difference*, *acomptes*, *montantEnLettre*, *compte_bancaire*, *annee_universitaire*)

exception DoesNotExist

exception MultipleObjectsReturned

acomptes

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

annee_universitaire

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

annee_universitaire_id

compte_bancaire

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

compte_bancaire_id

dateDebut

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

dateFin

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

difference

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enseignant

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

enseignant_id

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

matiere

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

matiere_id

montant

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

montantEnLettre

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

montantL1

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

montantL2

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

montantL3

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nombreHeure

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nombreHeureL1

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nombreHeureL2

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nombreHeureL3

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

prixUnitaire

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

save(**args*, ***kwargs*)

Save the current instance. Override this in a subclass if you want to control the saving process.

The "force_insert" and "force_update" parameters can be used to insist that the « save » must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

class main.models.**Fournisseur**(*id*, *type*, *montant*, *dateversement*, *le_mois*, *compte_bancaire*, *annee_universitaire*)

exception DoesNotExist

exception MultipleObjectsReturned

TYPE = [('TDE', 'TDE'), ('CEET', 'CEET'), ('Espoir+', 'Espoir+'), ('Autres', 'Autres')]

TYPE_MOIS = [('Janvier', 'Janvier'), ('Février', 'Février'), ('Mars', 'Mars'), ('Avril', 'Avril'), ('Mai', 'Mai'), ('Juin', 'Juin'), ('Juillet', 'Juillet'), ('Août', 'Août'), ('Septembre', 'Septembre'), ('Octobre', 'Octobre'), ('Novembre', 'Novembre'), ('Décembre', 'Décembre')]

annee_universitaire

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

annee_universitaire_id

compte_bancaire

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

compte_bancaire_id

dateversement

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_le_mois_display(*, *field=<django.db.models.fields.CharField : le_mois>*)

get_next_by_dateversement(*, *field=<django.db.models.fields.DateField : dateversement>*, *is_next=True*, ***kwargs*)

get_previous_by_dateversement(*, *field=<django.db.models.fields.DateField : dateversement>*, *is_next=False*, ***kwargs*)

get_type_display(*, *field=<django.db.models.fields.CharField : type>*)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

le_mois

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

montant

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

save(**args*, ***kwargs*)

Save the current instance. Override this in a subclass if you want to control the saving process.

The "force_insert" and "force_update" parameters can be used to insist that the « save » must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class main.models.**Frais**(*id*, *annee_universitaire*, *montant_inscription*, *montant_scolarite*)

exception DoesNotExist

exception MultipleObjectsReturned

annee_universitaire

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

annee_universitaire_id

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

montant_inscription

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

montant_scolarite

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

class main.models.**Information**(*id*, *enseignant*, *directeur*, *numeroSecurite*, *discipline*, *niveau*, *dateDebut*, *dateFin*, *duree*)

exception DoesNotExist

exception MultipleObjectsReturned

TYPE_CHOISE = [('Premier', 'Niveau 1'), ('Deuxième', 'Niveau 2'), ('Troisième', 'Niveau 3')]

dateDebut

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

dateFin

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

directeur

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

directeur_id

discipline

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

discipline_id

duree

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

enseignant

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

enseignant_id

get_next_by_dateDebut(*, *field=<django.db.models.fields.DateField : dateDebut>*, *is_next=True*, ***kwargs*)

get_next_by_dateFin(*, *field=<django.db.models.fields.DateField : dateFin>*, *is_next=True*, ***kwargs*)

get_niveau_display(*, *field=<django.db.models.fields.CharField : niveau>*)

get_previous_by_dateDebut(*, *field=<django.db.models.fields.DateField : dateDebut>*, *is_next=False*, ***kwargs*)

get_previous_by_dateFin(*, *field=<django.db.models.fields.DateField : dateFin>*, *is_next=False*, ***kwargs*)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

niveau

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

numeroSecurite

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

save(**args*, ***kwargs*)

Save the current instance. Override this in a subclass if you want to control the saving process.

The "force_insert" and "force_update" parameters can be used to insist that the « save » must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

class main.models.**Matiere**(*id*, *codematiere*, *libelle*, *coefficient*, *minValue*, *heures*, *abbreviation*, *enseignant*, *ue*, *is_active*)

exception DoesNotExist

exception MultipleObjectsReturned

abbreviation

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

codematiere

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

coefficient

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

competence_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```python
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

count_evaluations(*annee*, *semestres*)

dans_semestre(*semestre*)

enseignant

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```python
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

enseignant_id

`evaluation_set`

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```python
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`fichedepaie_set`

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```python
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`get_etudiant_semestre`(*semestre*)

`get_etudiants_en_rattrapage`()

`get_semestres`(*annee_selectionnee*, *type*)

Cette méthode retourne les semestres d'une matiere type : __current__| __all__ annee_selectionnee : annee_selectionnee | __all__ # Passer plus tard le parcours

`heures`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`id`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`information_set`

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```python
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`is_active`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`is_available_to_add_evaluation`(*semestre*)

`libelle`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`minValue`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

ponderation_restante(*semestre*)

reactiver()

save(**args*, ***kwargs*)

Save the current instance. Override this in a subclass if you want to control the saving process.

The "force_insert" and "force_update" parameters can be used to insist that the « save » must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

seance_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```python
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

seanceplannifier_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```python
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

suspendre()

ue

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```python
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

ue_id

class main.models.**Note**(**args*, ***kwargs*)

Ce modèle représente la note d'un étudiant dans un semestre et une matière donnée.

Attributes :

valeurNote (decimal) : La valeur de la note. etudiant (Etudiant) : L'étudiant à qui cette note appartient. matiere (Matiere) : La matière dans laquelle l'étudiant a eu cette note.

Methods :

__str__() -> str : Renvoie une représentation en chaîne de caractères de l'objet Note.

exception **DoesNotExist**

exception **MultipleObjectsReturned**

etudiant

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

etudiant_id

evaluation

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

evaluation_id

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

valeurNote

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class main.models.**Paiement**(*id*, *type*, *montant*, *dateversement*, *etudiant*, *comptable*, *compte_bancaire*, *numerobordereau*, *annee_universitaire*)

exception DoesNotExist

exception MultipleObjectsReturned

TYPE_CHOICES = [('Frais de scolarité', 'Frais de scolarité'), ("Frais d'inscription", "Frais d'inscription")]

annee_universitaire

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

annee_universitaire_id

comptable

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

comptable_id

compte_bancaire

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

compte_bancaire_id

dateversement

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

etudiant

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

etudiant_id

get_next_by_dateversement(**, field=<django.db.models.fields.DateField : dateversement>, is_next=True, **kwargs*)

get_previous_by_dateversement(**, field=<django.db.models.fields.DateField : dateversement>, is_next=False, **kwargs*)

get_type_display(**, field=<django.db.models.fields.CharField : type>*)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

montant

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

numerobordereau

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class main.models.**Parcours**(*id, nom, domaine, description*)

exception DoesNotExist

exception MultipleObjectsReturned

description

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

domaine

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

domaine_id

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nom

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

programme_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

class main.models.**Personnel**(*nom*, *prenom*, *sexe*, *datenaissance*, *lieunaissance*, *contact*, *email*, *adresse*, *prefecture*, *is_active*, *carte_identity*, *nationalite*, *user*, *photo_passport*, *id*, *salaireBrut*, *dernierdiplome*, *nbreJrsCongesRestant*, *nbreJrsConsomme*)

exception DoesNotExist

exception MultipleObjectsReturned

adresse

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

carte_identity

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

charge_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`comptable`

Accessor to the related object on the reverse side of a one-to-one relation.

In the example :

```python
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`Place.restaurant` is a `ReverseOneToOneDescriptor` instance.

`conge_set`

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```python
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`contact`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`datenaissance`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`dernierdiplome`

Just like the FileDescriptor, but for ImageFields. The only difference is assigning the width/height to the width_field/height_field, if appropriate.

`directeurdesetudes`

Accessor to the related object on the reverse side of a one-to-one relation.

In the example :

```python
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`Place.restaurant` is a `ReverseOneToOneDescriptor` instance.

`email`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`enseignant`

Accessor to the related object on the reverse side of a one-to-one relation.

In the example :

```python
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`Place.restaurant` is a `ReverseOneToOneDescriptor` instance.

`get_sexe_display`(*, *field=<django.db.models.fields.CharField : sexe>*)

`id`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

is_active

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

lieunaissance

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nationalite

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nbreJrsCongesRestant

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nbreJrsConsomme

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nom

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

photo_passport

Just like the FileDescriptor, but for ImageFields. The only difference is assigning the width/height to the width_field/height_field, if appropriate.

prefecture

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

prenom

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

salaireBrut

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

salaire_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

save()

Save the current instance. Override this in a subclass if you want to control the saving process.

The “force_insert” and “force_update” parameters can be used to insist that the « save » must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

sexe

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

update_conge_counts()

user

Accessor to the related object on the forward side of a one-to-one relation.

In the example :

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`Restaurant.place` is a `ForwardOneToOneDescriptor` instance.

user_id

class main.models.**Programme**(*id*, *parcours*, *semestre*)

exception DoesNotExist

exception MultipleObjectsReturned

generate_code()

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

parcours

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

parcours_id

save(**args*, ***kwargs*)

Save the current instance. Override this in a subclass if you want to control the saving process.

The "force_insert" and "force_update" parameters can be used to insist that the « save » must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

semestre

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

semestre_id

ues

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example :

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`class main.models.Salaire`(*id*, *date_debut*, *date_fin*, *personnel*, *numero_cnss*, *qualification_professionnel*, *prime_efficacite*, *prime_qualite*, *frais_travaux_complementaires*, *prime_anciennete*, *frais_prestations_familiales*, *frais_risques_professionnel*, *frais_pension_vieillesse_emsalaire*, *frais_prestations_familiale_salsalaire*, *tcs*, *irpp*, *is_tcs*, *is_irpp*, *prime_forfaitaire*, *acomptes*, *salaire_net_a_payer*, *compte_bancaire*, *annee_universitaire*)

`exception DoesNotExist`

`exception MultipleObjectsReturned`

`TYPE_CHOICES = [('Enseignant', 'Enseignant'), ('Comptable', 'Comptable'), ('Directeur des études', 'Directeur des études'), ('Gardien', 'Gardien'), ("Agent d'entretien", "Agent d'entretien")]`

`acomptes`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`annee_universitaire`

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

`annee_universitaire_id`

`compte_bancaire`

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

`compte_bancaire_id`

`date_debut`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`date_fin`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`frais_pension_vieillesse_emsalaire`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`frais_prestations_familiale_salsalaire`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`frais_prestations_familiales`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`frais_risques_professionnel`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`frais_travaux_complementaires`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`get_qualification_professionnel_display`(*, *field=<django.db.models.fields.CharField : qualification_professionnel>*)

`id`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`irpp`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`is_irpp`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`is_tcs`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`numero_cnss`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`objects = <django.db.models.manager.Manager object>`

`personnel`

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

`personnel_id`

`prime_anciennete`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`prime_efficacite`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`prime_forfaitaire`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`prime_qualite`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`qualification_professionnel`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

salaire_net_a_payer

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

save(**args*, ***kwargs*)

Save the current instance. Override this in a subclass if you want to control the saving process.

The "force_insert" and "force_update" parameters can be used to insist that the « save » must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

tcs

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class main.models.**Semestre**(*id*, *libelle*, *credits*, *courant*, *annee_universitaire*)

CHOIX_SEMESTRE = [('S1', 'Semestre1'), ('S2', 'Semestre2'), ('S3', 'Semestre3'), ('S4', 'Semestre4'), ('S5', 'Semestre5'), ('S6', 'Semestre6')]

exception DoesNotExist

exception MultipleObjectsReturned

annee_universitaire

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

annee_universitaire_id

code_semestre()

courant

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

credits

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

etudiant_set

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example :

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

evaluation_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`get_all_ues``()`

`get_libelle_display``(*, `*field=<django.db.models.fields.CharField : libelle>*`)`

`id`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`libelle`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`objects = <django.db.models.manager.Manager object>`

`planning_set`

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`programme_set`

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`save``()`

Save the current instance. Override this in a subclass if you want to control the saving process.

The "force_insert" and "force_update" parameters can be used to insist that the « save » must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

`seance_set`

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`static_get_current_semestre``()`

`class` `main.models.`**`Tuteur`**`(`*id*, *nom*, *prenom*, *sexe*, *adresse*, *contact*, *profession*, *type*`)`

`CHOIX_SEX = [('F', 'Féminin'), ('M', 'Masculin')]`

`CHOIX_TYPE = [('pere', 'Père'), ('mere', 'Mère'), ('tuteur', 'Tuteur')]`

`exception DoesNotExist`

`exception MultipleObjectsReturned`

`Tuteurs`

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example :

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

`adresse`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`contact`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`get_sexe_display`(**, field=<django.db.models.fields.CharField : sexe>*)

`get_type_display`(**, field=<django.db.models.fields.CharField : type>*)

`id`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`nom`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`objects = <django.db.models.manager.Manager object>`

`prenom`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`profession`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`save`()

Save the current instance. Override this in a subclass if you want to control the saving process.

The "force_insert" and "force_update" parameters can be used to insist that the « save » must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

`sexe`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`type`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`class` `main.models.`**`Ue`**(*id, codeUE, libelle, niveau, type, nbreCredits, heures, enseignant*)

`exception DoesNotExist`

`exception MultipleObjectsReturned`

TYPES = [('Technologie', 'Technologie'), ('Communication', 'Communication'), ('Anglais', 'Anglais'), ('Maths', 'Maths')]

TYPES_NIVEAU = [('1', 'Licence'), ('2', 'Master'), ('3', 'Doctorat')]

codeUE

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

competence_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```python
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

enseignant

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example :

```python
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

enseignant_id

get_niveau_display(**, field=<django.db.models.fields.CharField : niveau>*)

get_type_display(**, field=<django.db.models.fields.CharField : type>*)

heures

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

libelle

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

matiere_principacle()

matiere_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example :

```python
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

nbreCredits

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

niveau

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

programme_set

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example :

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class main.models.**Utilisateur**(**args*, ***kwargs*)

class Meta

abstract = False

SEXE_CHOISE = [('F', 'Feminin'), ('M', 'Masculin')]

adresse

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

carte_identity

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

contact

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

datenaissance

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

email

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

full_name()

get_sexe_display(**, field=<django.db.models.fields.CharField : sexe>*)

getrole()

is_active

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

lieunaissance

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nationalite

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

nom

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

photo_passport

Just like the FileDescriptor, but for ImageFields. The only difference is assigning the width/height to the width_field/height_field, if appropriate.

prefecture

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

prenom

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

reactiver()

sexe

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

suspendre()

user

Accessor to the related object on the forward side of a one-to-one relation.

In the example :

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`Restaurant.place` is a `ForwardOneToOneDescriptor` instance.

user_id

main.models.**create_compte_etudiant**(*sender*, *instance*, *created*, ***kwargs*)

main.models.**generate_ue_code**(*sender*, *instance*, *created*, ***kwargs*)

1.2 Vues

1.3 Urls

Code source 1 – Exemple de code Python

```
def hello_world():
    print("Hello, world!")
```

CHAPITRE 2

Indices et tables

- genindex

Index des modules Python