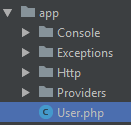
1. Models



**On créer tous nos models (singulier ac Majuscule)**

php artisan make:model Post -m



php artisan make:model Role -m



php artisan make:model Photo -m



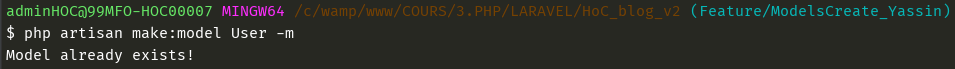
php artisan make:model Category -m

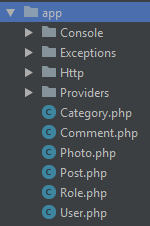


php artisan make:model Comment -m

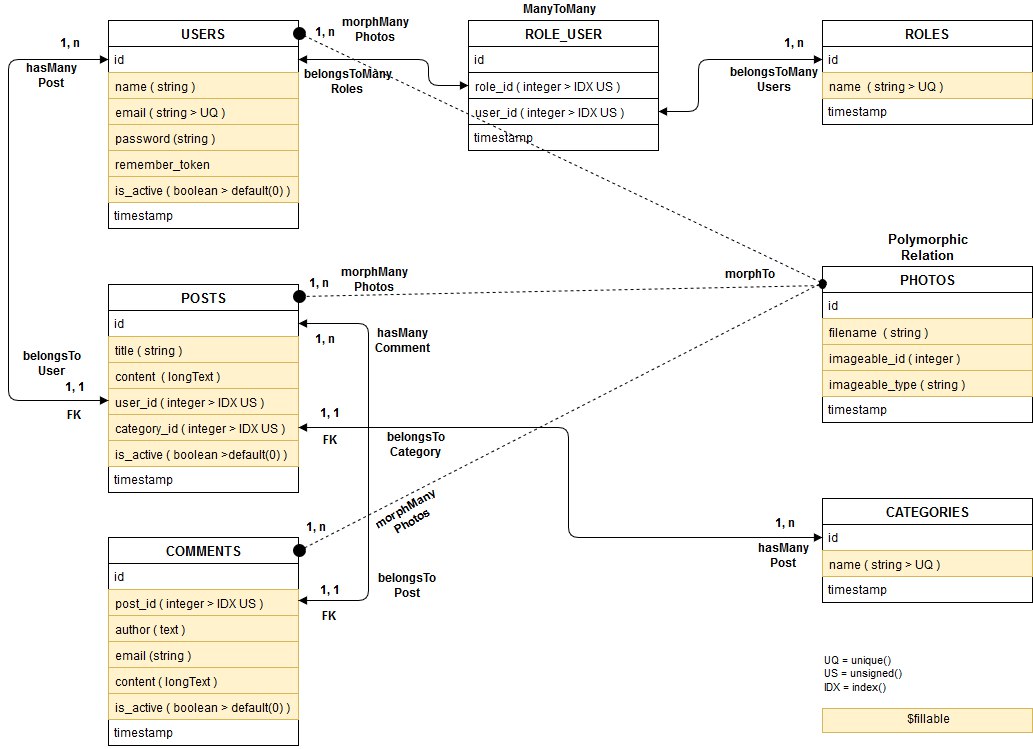


Le Model « User » existe déjà, on n’a pas besoin de le creer





**On crée nos relations (jointures)**



**On détermine nos jonctions en fonction des cardinalités de nos tables :**

**Users => Posts**

1 User **a (has)** 1 ou N posts (**has**)

1 Post n’**appartient** qu’**a (belongsTo)** 1 User (**belongsTo**)

* On a donc une relation **OneToMany** (lien le plus étendu)

**Posts => Comments**

1 Post peux avoir 1 ou N Comments (OneToMany)

1 Comment ne peux avoir qu’1 Posts (OneToOne)

* On a donc une relation **OneToMany** (lien le plus étendu)

**Posts=> Categories**

1 Post peux avoir 1 ou N Categories (OneToMany)

1 Category ne peux avoir qu’1 Posts (OneToOne)

* On a donc une relation **OneToMany** (lien le plus étendu)

**Users => Role\_User => Roles**

1 User **appartient à** 1 ou N Roles (**belongsToMany**)

1 Role **appartient à** 1 ou N Users (**belongsToMany**)

* On a donc une relation Many des deux côtés donc une relations **ManyToMany**

**Photos => Users, Posts, Comments**

On sait que :

* Un **user** peux avoir une **photo** ou plusieurs
  + Une photo peut appartenir à un user
* Un **post** peux avoir une ou plusieurs **photo**
  + Une photo ne peut être que dans un post
* Un **comments** peux avoir une ou plusieurs **photos**
  + Une photo ne peut être que dans un commentaire

On constate qu’on a une relation entre photos et 3 autres tables, on peut donc soit créer 3 relations Many to Many ou alors créer une Relation Polymorphique qui gère toutes ces liaisons automatiquement pour nous.

**Users => Posts**

|  |  |  |
| --- | --- | --- |
| **USERS** |  | **POSTS** |
| id |  | id |
| name |  | title |
| email |  | content |
| password |  | user\_id |
| remember\_token |  | photo\_id |
| is\_active |  | category\_id |
| timestamp |  | is\_active |
|  |  | timestamp |

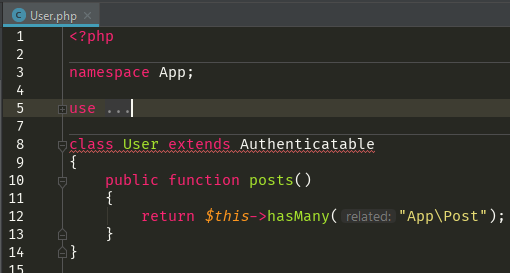
**On rend nos tables modifiables pr éviter l’erreur MassAssigment**

class User extends Authenticatable  
{  
 protected *$fillable* = [  
 'name', 'email', 'password', 'remember\_token', 'is\_active'  
 ];  
}

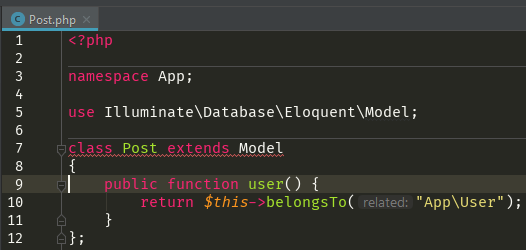
class Post extends Model  
{  
 protected *$fillable* = [  
 "title", "content", "user\_id", "photo\_id", "category\_id", "is\_active"  
 ];  
};

**On définit nos jointures (migrations)**

1 User **a (has)** 1 ou N posts



1 Post n’**appartient** qu’**a (belongsTo)** 1 User



**Posts => Comments**

|  |  |  |
| --- | --- | --- |
| **POSTS** |  | **COMMENTS** |
| id |  | id |
| title |  | post\_id |
| content |  | content |
| user\_id |  | author |
| photo\_id |  | email |
| category\_id |  | is\_active |
| is\_active |  | timestamp |
| timestamp |  |  |

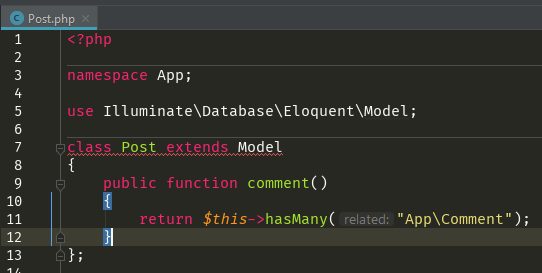
**On rend nos tables modifiables pr éviter l’erreur MassAssigment**

La classe Posts a déjà été faite ci-dessus

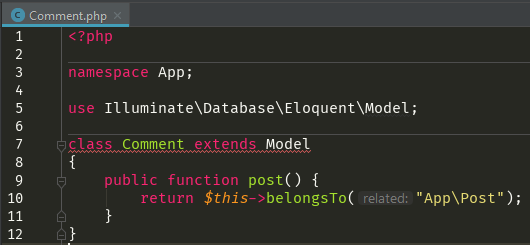
class Comment extends Model  
{  
 protected *$fillable* = [  
 'posts\_id', 'content', 'author', 'email', 'is\_active'  
 ];  
  
}

**On définit nos jointures (migrations)**

1 Post **a (has)** 1 ou N Comments



1 Comment n’**appartient** qu’**a (belongsTo)** 1 Post



**Categories => Posts**

|  |  |  |
| --- | --- | --- |
| **CATEGORIES** |  | **POSTS** |
| id |  | id |
| name |  | title |
| timestamp |  | content |
|  |  | user\_id |
|  |  | photo\_id |
|  |  | category\_id |
|  |  | is\_active |
|  |  | timestamp |

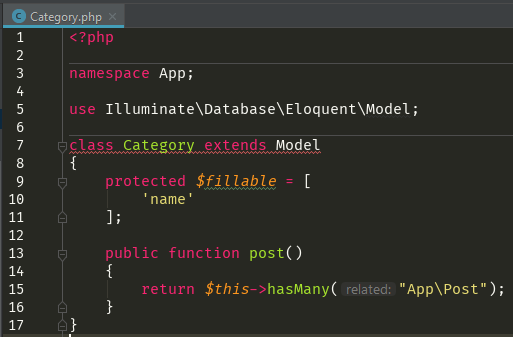
**On rend nos tables modifiables pr éviter l’erreur MassAssigment**

class Category extends Model  
{  
 protected *$fillable* = [  
 'name'  
 ];  
}

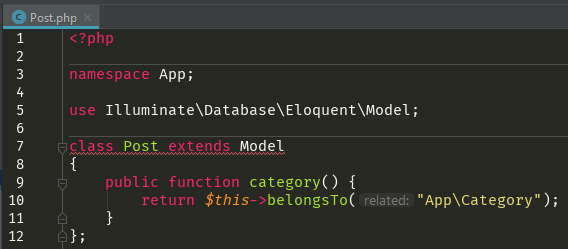
La classe Posts a déjà été faite ci-dessus

**On définit nos jointures (migrations)**

1 Category **a (has)** 1 ou N posts

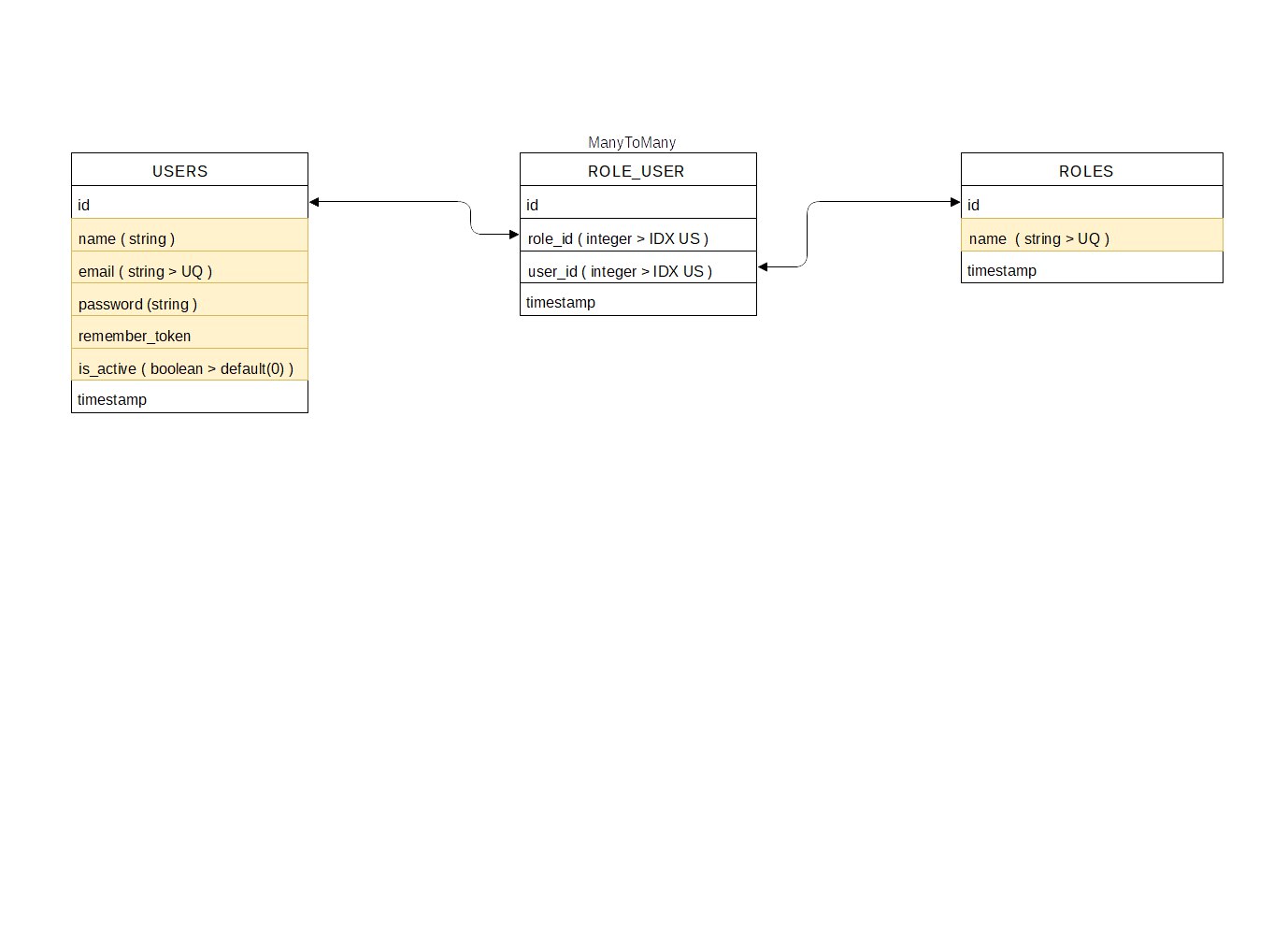


1 Post n’**appartient** qu’**a (belongsTo)** 1 Category

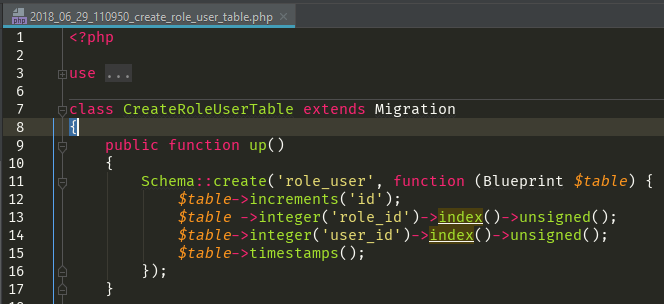


**Users => Role\_User => Roles**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USERS** |  | **ROLE\_USER** |  | **ROLE** |
| id |  | id |  | id |
| name |  | role\_id |  | name |
| email |  | user\_id |  | timestamp |
| password |  | timestamp |  |  |
| remember\_token |  |  |  |  |
| is\_active |  |  |  |  |
| timestamp |  |  |  |  |

****

**La migration** role\_user\_table que l’on a créée va permettre à Laravel de gerer notre relation ManyToMany



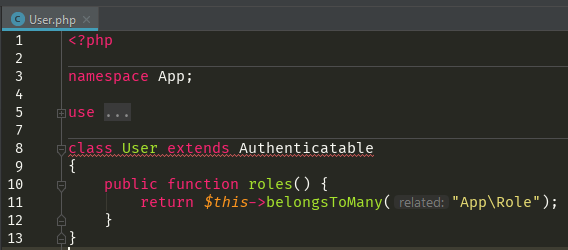
**On rend nos tables modifiables pr éviter l’erreur MassAssigment**

La classe User a déjà été faite ci-dessus

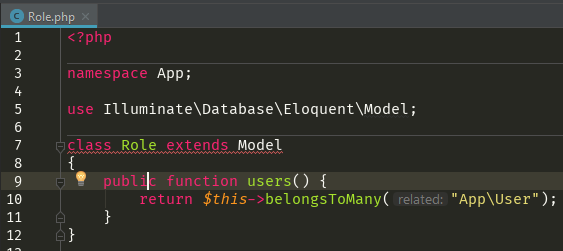
La classe Role a déjà été faite ci-dessus

**On définit nos jointures (migrations)**

1 User **appartient à** 1 ou N Roles (**belongsToMany**)



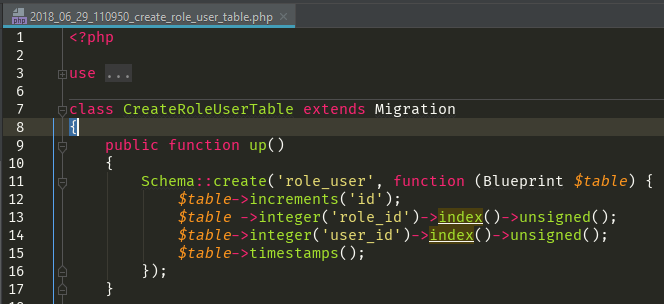
1 Role **appartient à** 1 ou N Users (**belongsToMany**)



**Photos => Users, Posts, Comments**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **PHOTO** |  |  |
|  |  | id |  |  |
|  |  | imageable\_id |  |  |
|  |  | imageable\_type |  |  |
|  |  | filename |  |  |
|  |  | timestamp |  |  |
|  |  |  |  |  |
| **COMMENTS** |  | **USERS** |  | **POSTS** |
| id |  | id |  | id |
| post\_id |  | name |  | title |
| content |  | email |  | content |
| author |  | password |  | user\_id |
| email |  | remember\_token |  | photo\_id |
| is\_active |  | is\_active |  | category\_id |
| timestamp |  | timestamp |  | is\_active |
|  |  |  |  | timestamp |

**La migration** role\_user\_table que l’on a créée va permettre à Laravel de gerer notre relation ManyToMany



**On rend nos tables modifiables pr éviter l’erreur MassAssigment**

La classe Comment a déjà été faite ci-dessus

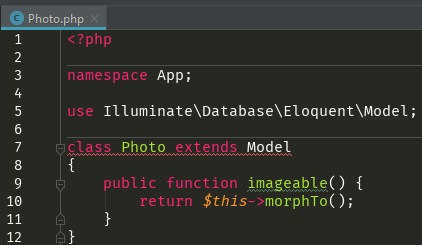
La classe User a déjà été faite ci-dessus

La classe Post a déjà été faite ci-dessus

class Photo extends Model  
{  
 protected *$fillable* = [  
 "filename", "imageable\_id", "imageable\_type"  
 ];  
}

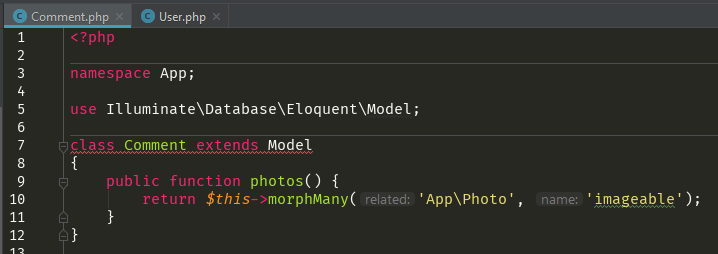
**On définit nos jointures (migrations)**

On détermine la fonction qui nous permettra d’associer une table X à une photo

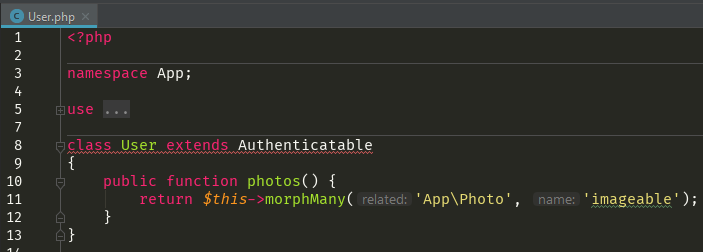


Et on dit quelles sont les classes qui peuvent se raccorder à nos photos (imageable)

**Comment**



**User**



**Post**

