

TD PHP MVC

Étape 15 bis : Classe mère Model pour les entity (Complément du TD 15)

Pour pouvoir instancier des objets des classes entity comme nous le faisons dans la méthode insertOne de MainRepository nous allons devoir créer un constructeur prenant un paramètre un tableau associatif (\$fields) pour chaque classe entity.

En effet, si l'insertion en base de données s'est bien passée nous renvoyons une instance de l'entity correspondant à la table dans laquelle nous venons d'ajouter une ligne (ligne 123/124 ci dessous)

```
repository > main.repository.php > MainRepository > insertOne
115     $sql = "INSERT INTO $this->table ($columns) VALUES ($values)";
116     $statement = $this->connect()->prepare($sql);
117     $result = $statement->execute($valuesToBind);
118     $test = $statement->rowCount() == 1;
119     if($result && $test){
120         $insertedId = $this->db->lastInsertId();
121         $fields['id'] = $insertedId;
122         $entityClass = $this->entity;
123         $entity = new $entityClass($fields);
124         return $entity;
125     }
126     return false;
127 }
```

Plutôt que de créer n fois le même constructeur dans chaque classe entity, nous allons créer une classe mère dont toutes ces entity hériteront, et c'est dans cette classe mère que nous écrivons le constructeur commun.

```
entity > model.entity.php > Model
1  <?php
2
3  class Model{
4
5      function __construct($fields = []){
6          foreach($fields as $k => $v){
7              $this->{$k} = $v;
8          }
9      }
10
11 }
```

Ne pas oublier l'héritage dans les classes entity Category et Product

(Voir page suivante ...)

```
entity > 🐘 category.entity.php > 📄 Category
1   <?php
2
3   class Category extends Model {}
4
```

```
entity > 🐘 product.entity.php > 📄 Product
1   <?php
2
3   class Product extends Model {}
4
```

Une dernière modification est nécessaire dans les fetchAll, pour indiquer à PDO que les constructeurs des entity ne sont plus « vide » et qu'ils prennent des paramètres.
Il faut ajouter l'option PDO::FETCH_PROPS_LATE

```
repository > 🐘 main.repository.php > 📄 MainRepository > 📄 getAll
40
41   function getAll($where = "1"){
42       $sql = "SELECT * FROM $this->table WHERE $where";
43       $resp = $this->connect()->query($sql);
44       $rows = $resp->fetchAll(PDO::FETCH_CLASS | PDO::FETCH_PROPS_LATE, $this->entity);
45       if(count($rows) == 0){
46           return $rows;
47       }
48   }
```

```
repository > 🐘 main.repository.php > 📄 MainRepository > 📄 getOne
72
73   function getOne($id){
74       if($id == null){
75           return null;
76       }
77       $sql = "SELECT * FROM $this->table WHERE id=$id";
78       $resp = $this->connect()->query($sql);
79       $rows = $resp->fetchAll(PDO::FETCH_CLASS | PDO::FETCH_PROPS_LATE, $this->entity);
80       $row = count($rows) == 1 ? $rows[0] : null;
81       if($row == null){
82           return null;
83       }
84   }
```