

# La relation 1-n c'est simple

Dans la définition d'une relation dans le schéma d'une base sous symfony - doctrine, certains paramètres restent obscures. Je vais donc tenter d'y mettre quelques lumières.

La documentation (en anglais) sur [les relations avec doctrine est disponible ici](#) .

Imaginons deux tables simples pour stocker des messages dans un blog : *categorie* et *message* . Voilà

a quoi pourrait ressembler notre shema.yml. Les relations *1-n*

ne doivent être définies que sur une des tables (pour éviter des paramètres en doublons qui causeraient des problèmes insolubles). La table idéal est la table *n*

de la relation, c'est celle qui vous causera le moins de tracas.

```
categorie:
  columns:
    nom:
      type: string(40)
```

```
message:
  columns:
    auteur:
      type: string(40)
      notnull: true
    titre:
      type: string(80)
      notnull: true
    text: clob
    categorie_id: integer
  relations:
    categorie:
      foreignAlias: messages
```

Il s'agit ici des informations minimales qui permettent de rendre active une relation *1-n* entre *message*

et *categorie*

Dans la documentation, il est indiqué que les paramètres *local* et *foreign* sont obligatoire, en pratique, et si on respecte les conventions de nommage (clef primaire nommée *id* et clef de liaison nommée *nomDeTable\_id*) alors il n'est pas nécessaire de renseigner les noms de champs, ils sont automatiquement reconnu.

Il ne reste que la mystérieuse propriété *foreignAlias* qui pose problème à tant de débutants. Dans les paramètres des relations les propriétés préfixées par *foreign* correspondent à ce qui est défini de l'autre côté de la relation. La propriété *foreign* permet de désigner la clef à utiliser de l'autre côté de la relation. La propriété *foreignAlias* permet de définir quel est le nom de la relation vu de l'autre côté...

Bon, quelques petits exemples de code.

### **Exemple : récupérer le nom d'une catégorie depuis le message d'id = 1**

```
$message = Doctrine_Core::getTable('message')->findOneById(1);  
echo $message->getCategorie()->getNom();
```

### **Exemple : récupérer la liste de tous les titres des messages pour une catégorie d'id = 1**

```
findOneById = 1 ?>
```

```
getMessages as $message: ?>
```

```
- getTitre() ?>
```

## Doctrine - La relation 1-n c'est vraiment simple

Écrit par Michel Rotta

Samedi, 04 Septembre 2010 17:50 - Mis à jour Samedi, 04 Septembre 2010 17:50

---

On voit donc que l'on peut récupérer les objets par un simple get suivi du nom de la relation. A noter que la relation du côté *categorie* est *messages* avec un "s" ce qui indique que la relation va nous retourner plusieurs enregistrements (dans tous les cas elle retourne ici un Doctrine\_Collection). Ceci n'est qu'une convention de nommage, cette relation aurait très bien pu s'appeler tutu et avoir le même résultat, mais avouer que c'est moins clair ainsi !

### Exemple : un fichier de *fixture* pour remplir notre table

categorie:

cat1:

nom: Première catégorie

cat1:

nom: Deuxième catégorie

message:

msg1:

auteur: Michel Rotta

titre: le premier message catégorie 1

texte: le texte de notre message

categorie: cat1

A noter que pour la liaison entre les tables dans la *fixture*, il convient d'utiliser le nom de la relation (ou le nom du *foreignAlias*) et non pas le nom de la table, cela sera souvent la même chose, mais parfois non.

Ceci n'est qu'un minuscule aperçu des possibilités de doctrine, mais c'est une des questions le plus souvent demandée.