

# **symfony** **“TAG”**

**2007-8-22(wed.) TAKAHARA Yoshihiro**

# 今日のテーマ

**“TAG”**

**TAG?**

# 合唱組曲『ニコニコ動画』 J・いさじ・ゼブラ・RE・内緒妹

伴奏を派手にしたライブ仕様、是非極大音量で10分間駆け抜けて下さい。作成にあたってsm682897、sm607197、sm609360、sm625064(消えてしまったようです…)、sm564475、sm724740、sm564674様方を使用させて頂きました、深くお礼申し上げます作者自身のオリジナリティは少しも有りませんがお楽しみ頂ければ幸いです。mp3は…二万コメいった行きましょうw◇「/hanten」コメ大杉だろw自重しろw

登録タグ [歌ってみた](#) [組曲『ニコニコ動画』](#) [才能の無駄遣い](#) [/hanten](#) [プレミアム専用](#) [ヘッドホ](#)  
[リーダーダム](#) [【タグ編集】](#)

マイメモリー 現在のコメントを保存 **マイリスト** おもしろい (35) に 登録

ニワニュース情報局 朝・昼・最新ニュース メール

niwa news ニワニュース 8月22日 民主が続々改

**/hontai**

↑ニコニコ動画

はてなブックマーク→

web2.0 webmail willcom  
 wordpress work xen xgl xul  
 yaml youtube Zend  
 Framework zk お役立ち きた  
 みりゅうじ これはひどい これ  
 は便利 はてな アイコン アル  
 ゴリズム エンジニアリング オ  
 ンラインストレージ オープン  
 ソース ゲーム システム運用  
**システム開発**  
 スポーツ セキュリティ ソフト  
 タグ チューニング ツール テ  
 ンプレート データベース ドラ  
 えもん ネタ ファイル転送 プ  
 レゼンテーション プログラミ  
 ング ホノルルマラソン ホーム  
 ページ マンガ メカ ライブ  
 ドア ワイン 上野 仕事 会社  
 住居 便利 北海道 同人誌 宇宙

# あまりにも典型的な コンテンツ分類方法

なんとなく  
Web2.0っぽい

# タグ実装の パターンを覚えよう

→今日のテーマ

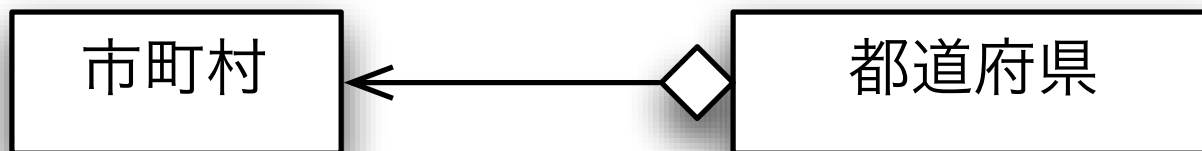
**お勉強の時間です**

**Tagの実装には  
多対多の関係を使います**

**多对多？**

そのまえに  
一対多

# 一对多



id	name
1	札幌
2	釧路
3	盛岡

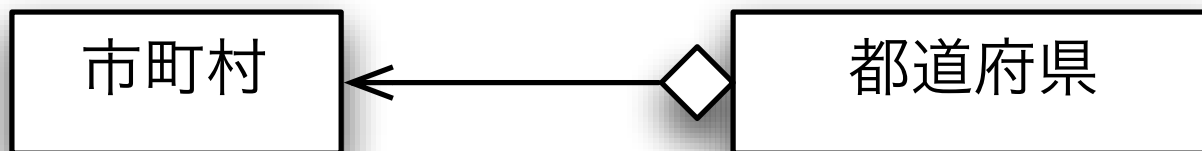
id	name
1	北海道
2	青森県
3	岩手県

多

一

親一つに対し  
子が複数

# 一对多



id	name	prefecture_id
1	札幌	1
2	釧路	1
3	盛岡	3

id	name
1	北海道
2	青森県
3	岩手県

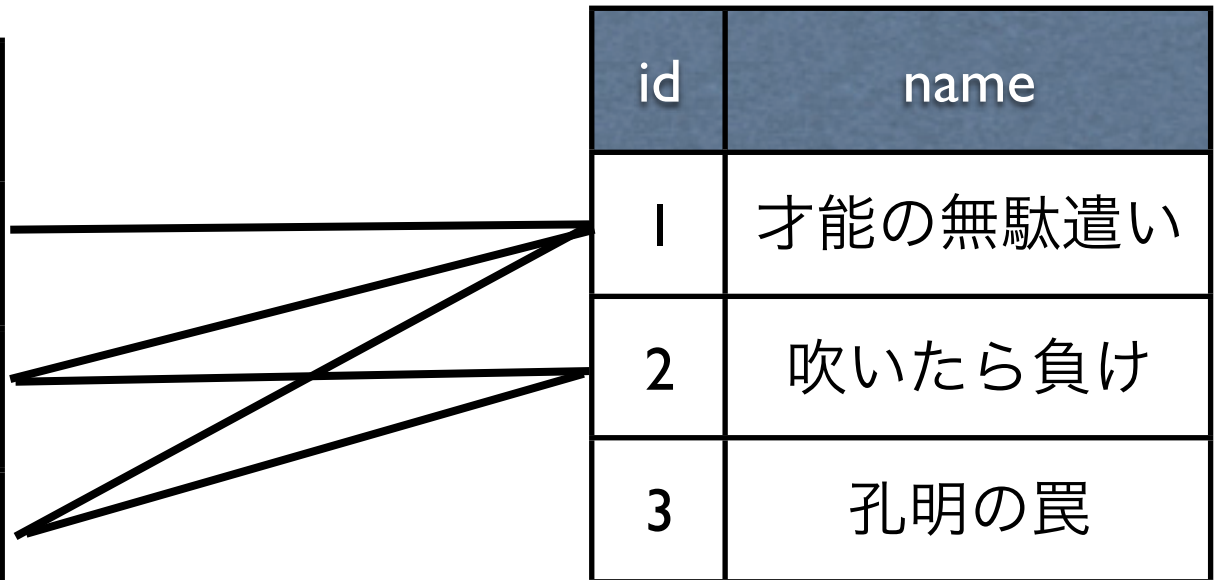
多

一

**多対多は？**

# 多対多

id	name
1	組曲「ニコニコ」
2	PHP以外みんな不正解
3	symfonyのソースコード の値段

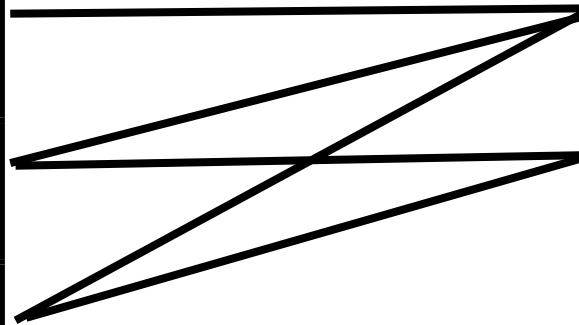


# 多対多



id	name
1	組曲「ニコニコ」
2	PHP以外みんな不正解
3	symfonyのソースコードの値段

id	name
1	才能の無駄遣い
2	吹いたら負け
3	孔明の罫



# 多対多



id	name
1	組曲「ニコニコ」
2	PHP以外みんな不正解
3	symfonyのソースコードの値段

content_id	tag_id
1	1
2	1
2	2
3	1
3	2

id	name
1	才能の無駄遣い
2	吹いたら負け
3	孔明の罫

# スキーマ定義

**schema.yml**

```
propel:  
  content:  
    id:  
    name: varchar(255)  
    body: varchar(1024)  
    created_at:  
    updated_at:  
    _uniques:  
      name_uniq: [name]  
  tag:  
    id:  
    name: varchar(255)  
    _uniques:  
      name_uniq: [name]  
  content_tag:  
    id: ← !最近の流行  
    content_id:  
    tag_id:  
    _uniques:  
      combination_uniq: [content_id, tag_id]
```

→ symfony propel-convert-yml-schema

```

<database name="propel" defaultIdMethod="native" noXsd="true" package="lib.model">
  <table name="content">
    <column name="id" type="integer" required="true" primaryKey="true" autoincrement="true" />
  />
  <column name="name" type="varchar" size="255" />
  <column name="body" type="varchar" size="1024" />
  <unique name="name_uniq">
    <unique-column name="name" />
  </unique>
</table>

  <table name="tag">
    <column name="id" type="integer" required="true" primaryKey="true" autoincrement="true" />
  />
  <column name="name" type="varchar" size="255" />
  <unique name="name_uniq">
    <unique-column name="name" />
  </unique>
</table>

  <table name="content_tag">
    <column name="id" type="integer" required="true" primaryKey="true" autoincrement="true" />
  />
  <column name="content_id" type="integer" />
  <foreign-key foreignTable="content">
    <reference local="content_id" foreign="id" />
  </foreign-key>
  <column name="tag_id" type="integer" />
  <foreign-key foreignTable="tag">
    <reference local="tag_id" foreign="id" />
  </foreign-key>
  <unique name="combination_uniq">
    <unique-column name="content_id" />
    <unique-column name="tag_id" />
  </unique>
</table>

</database>

```

# サンプルデータ投入

# **data/fixtures/ contents.yml**

→ symfony propel-load-data frontend

```
Content:
  nico:
    name: "組曲「ニコニコ」"
  hnw:
    name: "PHP以外みんな不正解"
  arasho:
    name: "symfonyのソースコードの値段"
Tag:
  tag_1:
    name: "才能の無駄遣い"
  tag_2:
    name: "書きにくい"
  tag_3:
    name: "孔明の罠"
```

```
ContentTag:
  tagging_1:
    content_id: nico
    tag_id: tag_1
  tagging_2:
    content_id: hnw
    tag_id: tag_1
  tagging_3:
    content_id: hnw
    tag_id: tag_2
  tagging_4:
    content_id: arasho
    tag_id: tag_1
  tagging_5:
    content_id: arasho
    tag_id: tag_2
```

□

```
Content:
  nico:
    name: "組曲「ニコニコ」"
    tag: ["才能の無駄遣い"]
  hnw:
    name: "PHP以外みんな不正解"
    tag: ["才能の無駄遣い", "吹いたら負け"]
  arasho:
    name: "symfonyのソースコードの値段"
    tag: ["才能の無駄遣い", "吹いたら負け"]
```

こんな風には書けたらなあ～

**書けます！**

```

class Content extends BaseContent
{
    public function setTag($tags) {
        if (is_array($tags)) {
            foreach($tags as $tag) {
                $this->setTag($tag);
            }
        } else {
            $tag_name = $tags;
            $crit = new Criteria;
            $crit->add(TagPeer::NAME, $tag_name);
            $tag = TagPeer::doSelectOne($crit);
            if (!$tag) {
                $tag = new Tag;
                $tag->setName($tag_name);
            }
            $tagged = new ContentTag;
            $tagged->setTag($tag);
            $this->addContentTag($tagged);
        }
    }
}

```

```

Content:
  nico:
    name: "組曲「ニコニコ」"
    tag: ["才能の無駄遣い"]
  hnw:
    name: "PHP以外みんな不正"
    tag: ["才能の無駄遣い"]
  arasho:
    name: "symfonyのソース"
    tag: ["才能の無駄遣い"]

```

# 多対多テーブルの 扱い方

**「吹いたら負け」  
全部持ってきて～**

```
$contents = ContentPeer::getTagged("吹いたら負け");
```

```

class ContentPeer extends BaseContentPeer
{
    public static function getTagged($name){
        $c = new Criteria;
        $c->add(TagPeer::NAME, $name);
        $tag = TagPeer::doSelectOne($c);
        if($tag){
            $contents = null;
            $content_tags = $tag->getContentTags();
            foreach($content_tags as $content_tag){
                $contents[] = $content_tag->getContent();
            }
            return $contents;[]
        }
        return null;
    }
}

```

# sfPropelManyToMany でもっと簡単にできるよ

[symfonyで開発日記] 多対多テーブルを簡単に扱う

<http://blog.symfony.jp/2006/12/29/sfpropelmanytomany/>

```
class ContentPeer extends BaseContentPeer
{
    public static function getTagged($name){
        $c = new Criteria;
        $c->add(TagPeer::NAME, $name);
        $tag = TagPeer::doSelectOne($c);
        if($tag){
            return sfPropelManyToMany::getRelatedObjects($tag, "ContentTag");
        }
        return null;
    }
}
```

```

class ContentPeer extends BaseContentPeer
{
    public static function getTagged($name){
        $c = new Criteria;
        $c->add(TagPeer::NAME,$name);
        $tag = TagPeer::doSelectOne($c);
        if($tag){
            $contents = null;
            $content_tags = $tag->getContentTags();
            foreach($content_tags as $content_tag){
                $contents[] = $content_tag->getContent();
            }
            return $contents;[]
        }
        return null;
    }
}

```



```

class ContentPeer extends BaseContentPeer
{
    public static function getTagged($name){
        $c = new Criteria;
        $c->add(TagPeer::NAME,$name);
        $tag = TagPeer::doSelectOne($c);
        if($tag){
            return sfPropelManyToMany::getRelatedObjects($tag,"ContentTag");
        }
        return null;
    }
}

```

**「才能の無駄遣い」**

**新しいの2つ持ってきて**

**て～**

```
$crit = new Criteria;  
$crit->addDescendingOrderByColumn(ContentPeer::CREATED_AT);  
$crit->setLimit(2);  
$contents = ContentPeer::getTagged("才能の無駄遣い", $crit);
```

```
class ContentPeer extends BaseContentPeer
{
    public static function getTagged($name, $crit=null){
        $c = new Criteria;
        $c->add(TagPeer::NAME, $name);
        $tag = TagPeer::doSelectOne($c);
        if($tag){
            return sfPropelManyToMany::getRelatedObjects($tag, "ContentTag", $crit);
        }
        return null;[]
    }
}
```

```
SELECT content_tag.ID, content_tag.CONTENT_ID, content_tag.TAG_ID, content.ID, content.NAME, content.BODY, content.CREATED_AT, content.UPDATED_AT FROM content_tag, content WHERE content_tag.TAG_ID=1 AND content_tag.CONTENT_ID=content.ID ORDER BY content.CREATED_AT DESC LIMIT 2
```

# **sfPropelManyToMany** 便利なので使ってください

<http://svn.symfony-project.com/trunk/lib/addon/propel/sfPropelManyToMany.class.php>

みんな大好き

**Admin Generator**

モジュール名



**symfony propel-init-admin frontend content Content**

アプリケーション名



モデル名



## edit content

Name:	<input type="text" value="組曲「ニコニコ」"/>
Body:	<input type="text"/>
Created at:	<input type="text" value="2007-08-22 6:59"/> 
Updated at:	<input type="text" value="2007-08-22 6:59"/> 

[list](#)[save](#)[save and add](#)[delete](#)

**タグ付けしたい！**

```
generator:  
  class:          sfPropelAdminGenerator  
  param:  
    model_class: Content  
    theme:        default
```

`apps/frontend/modules/content/config/generator.yml`

```
generator:
  class:          sfPropelAdminGenerator
  param:
    model_class:  Content
    theme:        default

  edit:
    display:
      "Edit": [name, body, content_tag]
    fields:
      content_tag: { type: admin_double_list, params: through_class=ContentTag }
```

apps/frontend/modules/content/config/generator.yml

※generator.ymlを変更したらsymfony cc

# edit content

**Edit**

Name:

Body:

Content tag:

Unassociated	Associated
2	1

← 惜しい！

delete

list save save and add

```
class Tag extends BaseTag
{
    function __toString() {
        return $this->getName();
    }
}
```

# edit content

**Edit**

Name:

Body:

Content tag:

Unassociated	Associated
<input type="text" value="吹いたら負け"/>	<input type="text" value="才能の無駄遣い"/>

delete

list save save and add

# おさらい

- 多対多スキーマの定義
- サンプルデータ投入方法
- **sfPropelManyToMany**
- **admin\_double\_list**

**ご清聴ありがとうございました**