

# Module M2105 - RT web dyna - TD/P 4

Prise en main du projet RT-Cloud et du micro-framework

## -- Installation

- Télécharger ou cloner le projet <https://github.com/jcheron/RT-Cloud>
- Copier les fichiers dans le dossier **htdocs** de votre serveur.
- Renommer éventuellement le dossier **RT-Cloud-master** en **RT-Cloud**

## -- Exercices

### Exercice 1

1. Créer un contrôleur **Exemples**,
2. charger les services dans la méthode **index**.
3. Affichez les services directement depuis le contrôleur,
4. Testez le résultat à l'adresse **/Exemples**

```
class Exemples extends \BaseController {  
  
    public function index() {  
        $services=DAO::getAll("Service");  
        foreach ($services as $service){  
            echo $service->getNom()."<br>";  
        }  
    }  
}
```

### Exercice 2

1. Dans le contrôleur **Exemples**, charger les Utilisateurs dans la méthode **users**.
2. Créer une vue **users.php** dans le dossier **views/exemples/**.

```
class Exemples extends \BaseController {  
    ...  
  
    public function users(){  
        $users=DAO::getAll("Utilisateur");  
        $this->loadView("Exemples/users",array("users"=>$users));  
    }  
}
```

```
<h1>Utilisateurs</h1>
<?php
foreach ($users as $user){
    echo $user->getLogin()."<br>";
}
```

### Exercice 3

1. Dans le contrôleur **Exemples**, charger les Disques dans la méthode **disques**.
2. Créer une vue **disques.html** (utilisant Twig) dans le dossier **views/exemples/**.
3. Afficher le nom du disque et l'utilisateur associé.

```
class Exemples extends \BaseController {
    ...

    public function disques(){
        $disques=DAO::getAll("Disque");
        $this->loadView("Exemples/disques.html",array("disques"=>$disques));
    }
}
```

```
<h1>Disques</h1>
{% for disque in disques %}
<b>{{ disque.getNom() }}</b> {{ disque.getUtilisateur() }}<br>
{% endfor %}
```

### Exercice 4

1. Créer la méthode **sortedUsers** pour qu'elle prenne en paramètre le champ **field** sur lequel on effectuera un tri, et un deuxième déterminant l'ordre de tri (**order** ASC ou DESC) : le champ par défaut sera "login" et l'ordre "ASC".
2. Créer le template **sortedUsers.html** pour qu'il affiche le champ sur lequel s'effectue le tri, son ordre, et qu'il puisse le modifier.

```
class Exemples extends \BaseController {
    ...
    public function sortedUsers($field="login",$order="ASC"){
        $fields=["login","mail","tel"];
        if(!array_search($field,$fields))
            $field=$fields[0];
        $header="";
        foreach ($fields as $f){
            $header.="<th>";
            $icon="sort-by-alphabet";
            $url="Exemples/sortedUsers/" . $f;
            if($field==$f){
                if(strtoupper($order)=="ASC"){
                    $icon="sort-by-alphabet-alt";
                }
            }
        }
    }
}
```

```
        $url.="/DESC";
    }
    $header.="<a href='{ $url }'>{$f}&nbsp;<span class='glyphicon
glyphicon-{$icon}' aria-hidden='true'></span></a>";
    }else{
        $header.="<a href='{ $url }'>{$f}</a>";
    }
    $header.="</th>";
}
$users=DAO::getAll("Utilisateur","1=1 ORDER BY ".$field." ".$order);

$this->loadView("Exemples/sortedUsers.html",array("users"=>$users,"header"=>$header
));
}
```

```
<table class='table table-striped'>
  <thead>
    <tr>
      {{header|raw}}
    </tr>
  </thead>
  <tbody>
    {% for user in users %}
      <tr>
        <td>{{user.getLogin()}}</td>
        <td>{{user.getMail()}}</td>
        <td>{{user.getTel()}}</td>
      </tr>
    {% endfor %}
  </tbody>
</table>
```

From:  
<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**

Permanent link:  
<http://slamwiki2.kobject.net/php-rt/tp4-corr?rev=1459099032>

Last update: **2019/08/31 14:25**

