

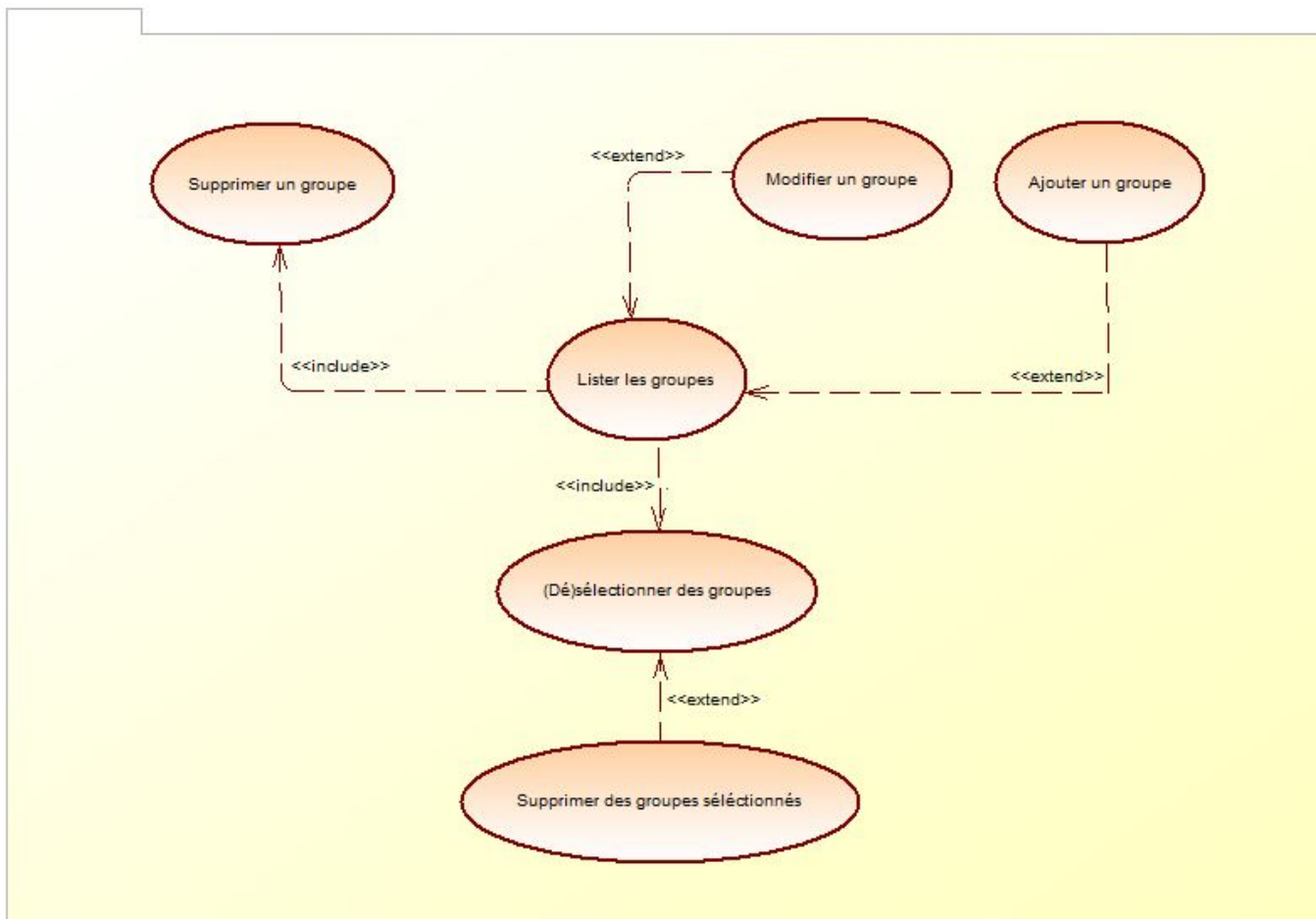
# BugReport : plateforme de remontée d'incidents

**Documentation technique - Sommaire :**

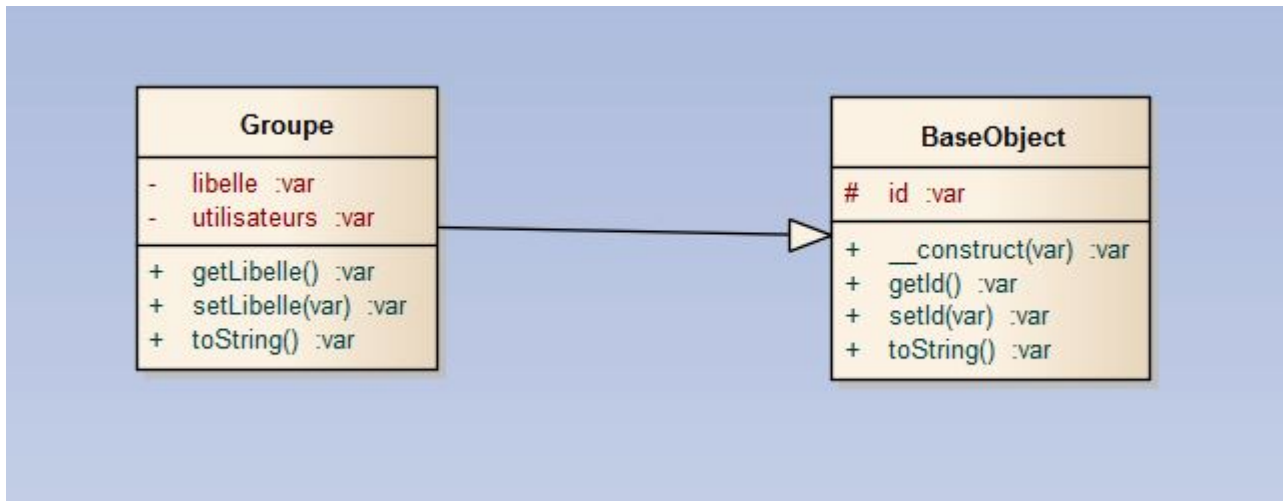
- 1. [Gestion des utilisateurs](#)
- 2. [Gestion des applications](#)
- 3. [Gestion des groupes](#)
- 4. [Soumettre un report](#)
- 5. [Gestion des Droits](#)

## Gestion des groupes

### 1- Fonctionnalités



### 2- Classes métiers



### 3- Variables de session (\$\_SESSION)

Nom	Type	Descriptif
\$_SESSION["ckAll"]	Booléen	Etat de la case à cocher permettant de sélectionner ou de désélectionner tous les groupes
\$_SESSION["selectedGroups"]	Tableau d'entiers	Ids des groupes sélectionnés

### 4- Vues

#### 4.1 vGroups

```
<div id="groups">
  <fieldset class="cadre">
    <legend><input type="checkbox" id="ckAll" title="(DÃ©)sÃ©lectionner les
groupes">&nbsp;Groupes :</legend>
    <table id="usersTable">
      <?php Gui::show($data["groups"],"addDelete")?>
    </table>
    <div class="buttons">
      <div class="btn" id="btAdd" name="btAdd" title="Ajouter un nouveau
groupe"><span class="add">Ajouter un groupe...</span></div>
      <div id="multi"><b>Pour la sÃ©lection :</b>&nbsp;
        <a href="#" class="delete" id="btMultiDelete" name="btMultiDelete"
title="Supprimer les groupes sÃ©lectionnÃ©s">Supprimer</a>&nbsp;
        </div>
      </div>
    </fieldset>
  </div>
<?php echo $data["js"];?>
```

Groupes :

- Utilisateurs 
- Administrateurs 
- Chefs de projet 
- Développeurs 
- Tech. Réseaux 

## 4.2 vAddGroup

```
<?php $group=$data["group"];?>
<form id="frmAddGroup" name="frmAddGroup" onsubmit="return false;">
<fieldset>
<legend><?php echo $data["title"];?></legend>
  <input type="hidden" id="id" name="id" value="<?php echo $group->getId();?>">
  <label class="label" for="libelle">Libellé <font color="red">*</font>
: </label><input title="Libellé obligatoire" placeholder="libellé" type="text"
id="libelle" name="libelle" value="<?php echo $group->getLibelle();?>"
maxlength="30" required><br>
  <div class="buttons">
    <div id="btValidAddGroup" class="btn">Valider</div>
    <div id="btCancelAddGroup" class="btn">Annuler</div>
  </div>
</fieldset>
</form>
<?php echo $data["js"];?>
```

Ajouter un groupe :

Libellé \* :

## 5- Contrôleurs

### 5.1 cGroupe/index

Fonctionnalité	Descriptif
Gestion des groupes	<b>Contrôleur de gestion des groupes</b> Appel du contrôleur <a href="#">refresh</a>

```
public function index(){
    $this->refresh();
}
```

### 5.2 cGroupe/refresh

Fonctionnalité	Descriptif
Liste des groupes	<b>Utilisé pour l'affichage initial et le rafraîchissement après une modification</b>
	Chargement des groupes \$groups
	Application de la sélection des groupes
	Mise en place logique client #btAdd -> add
	.delete -> confirmDelete .update -> update .ck -> select #ckAll -> selectAll #deleteMulti -> confirmDelete
	Passage de variables(\$data[]) et affichage de la vue vGroups

```

public function refresh(){
    //Chargement des groupes $groups
    $groups=DAO::getAll("Groupe");
    //Application de la sélection des groupes
    $selectedGroups=SessionUtils::getArray("selectedGroups");
    $js=JsUtils::getAndBindTo("#btAdd", "click",
    "cGroupe/add","{}","#operation");
    $js.=JsUtils::getAndBindTo(".delete", "click",
    "cGroupe/confirmDelete","{}","#operation");
    $js.=JsUtils::getAndBindTo(".update", "click",
    "cGroupe/update","{}","#operation");
    $js.=JsUtils::getAndBindTo(".ck", "click",
    "cGroupe/select","{}","#message");
    $js.=JsUtils::setChecked("ck", $selectedGroups);
    $js.=JsUtils::setChecked("ckAll", SessionUtils::getBoolean("ckAll"));
    $js.=JsUtils::getAndBindTo("#deleteMulti", "click",
    "cGroupe/confirmDelete/multi","{}","#operation");
    $js.=JsUtils::postAndBindTo("#ckAll", "click", "cGroupe/selectAll/","{ids:
    $("input:checkbox[class=ck]").map(function(){return
    this.value;}).get().join(",")}'","#message");
    $nb=sizeof($selectedGroups);
    if($nb>0)
        $js.=JsUtils::doSomethingOn("#multi", "show");
    else
        $js.=JsUtils::doSomethingOn("#multi", "hide");
    $data=array("groups"=>$groups, "js"=>$js);
    $this->loadView("vGroups", $data);
}
    
```

### 5.3 cGroupe/add

**Remarque :**

Le contrôleur **add** gère l'appel de l'affichage du formulaire d'ajout (GET) et la soumission de ce même formulaire (POST).

Functionalité	Descriptif
Ajout	<b>Ajout d'un groupe</b>
	Si le formulaire est posté (POST)
	Instanciation d'un nouveau groupe Affectation des variables du POST aux membres du groupe Ajout dans la base de données Appel du contrôleur <u>refresh</u> Affichage du message de mise à jour
	Sinon (GET)
	Mise en place logique client #btValidAddGroup -> add #btCancelAddGroup -> annulation Passage des variables et chargement de la vue <u>vAddGroup</u>

```

public function add(){
    if($_SERVER['REQUEST_METHOD']=='POST'){
        $group=new Groupe();
        RequestUtils::setValuesToObject($group,RequestUtils::getPost());
        if(DAO::insert($group)){
            Gui::showOne($group);
            echo " ajoutÃ©";
            echo JsUtils::get("cGroupe/refresh","{}","#groups");
        }else
            echo "ajout impossible";
        echo JsUtils::doSomethingOn("#operation", "hide",200);
    }else{
        $js=JsUtils::postFormAndBindTo("#btValidAddGroup", "click",
        "cGroupe/add", "frmAddGroup","#message",true);
        $js.=JsUtils::doSomethingAndBindTo("#btCancelAddGroup", "click",
        "#operation","hide",200);
        $js.=JsUtils::setHtmlAndBindTo("#btCancelAddGroup", "click",
        "#message","OpÃ©ration d\'ajout annulÃ©e");
        $js.=JsUtils::doSomethingOn("#operation", "show",400);
        $group=new Groupe();
        $this->loadView("vAddGroup",array("group"=>$group,"js"=>$js,"title"=>"Ajouter un
        groupe :"));
    }
}

```

## 5.4 cGroupe/update

Scénario pratiquement identique au contrôleur **cGroupe/add**

```

public function update($id){
    if($_SERVER['REQUEST_METHOD']=='POST'){
        $group=new Groupe();
        RequestUtils::setValuesToObject($group,RequestUtils::getPost());
        if(DAO::update($group)){
            Gui::showOne($group);
            echo " modifiÃ©";
            echo JsUtils::get("cGroupe/refresh","{}","#groups");
        }
    }
}

```

```
    }else
      echo "modification impossible";
      echo JsUtils::doSomethingOn("#operation", "hide",200);
    }else{
      $id=str_replace("update", "", $id[0]);
      if(is_numeric($id)){
        $group=DAO::getOne("Groupe", "id=".$id);
        $js=JsUtils::postFormAndBindTo("#btValidAddGroup", "click",
"CGroupe/update", "frmAddGroup", "#message", true);
        $js.=JsUtils::doSomethingAndBindTo("#btCancelAddGroup", "click",
"#operation", "hide", 200);
        $js.=JsUtils::setHtmlAndBindTo("#btCancelAddGroup", "click",
"#message", "Opération de mise à jour annulée");
        $js.=JsUtils::doSomethingOn("#operation", "show", 400);
$this->loadView("vAddGroup", array("group"=>$group, "js"=>$js, "title"=>"Modifier le
groupe :"));
      }
    }
  }
}
```

### 5.5 cGroupe/confirmDelete

Fonctionnalité	Descriptif
Confirmation	<b>Confirmation pour suppression de groupe(s)</b>
	Test pour savoir si la sélection est unique ou multiple
	Affichage du message de confirmation et des options possibles
	Mise en place logique client #confirmDelete -> delete ou deleteMulti #cancelDelete -> annulation

```
public function confirmDelete($id){
  $id=str_replace("delete", "", $id[0]);
  if(is_numeric($id)){
    echo "<div class='cadre'>Souhaitez vous supprimer le groupe ? <a
id='confirmDelete".$id."' href='#' class='accept'>Confirmer la suppression</a>";
    echo "<a id='cancelDelete' href='#' class='cancel'>Annuler</a>";
    echo JsUtils::getAndBindTo("#confirmDelete".$id, "click",
"CGroupe/delete/".$id, "{}", "#message");
  }else{
    echo "<div class='cadre'>Souhaitez vous supprimer les groupes
sélectionnés ? <a id='confirmDelete".$id."' href='#' class='accept'>Confirmer la
suppression</a>";
    echo "<a id='cancelDelete' href='#' class='cancel'>Annuler</a>";
    echo JsUtils::getAndBindTo("#confirmDelete".$id, "click",
"CGroupe/deleteMulti/".$id, "{}", "#message");
  }
  echo JsUtils::doSomethingOn("#operation", "show", 400);
  echo JsUtils::doSomethingAndBindTo("#cancelDelete", "click", "#operation",
"hide", 200, JsUtils::_setHtml("#message", "Suppression annulée"));
}
```

## 5.6 cGroupe/deleteMulti

Fonctionnalité	Descriptif
Suppression	<b>Supprimer 1 sélection de groupes</b>
	Récupération (dans 1 tableau stocké en session) des ids de(s) groupe(s) à supprimer
	Parcours des groupes et suppression dans la BDD
	Mise à jour de la sélection (en session)
	Affichage du message post-suppression

```
public function deleteMulti(){
    $selectedGroupId=SessionUtils::getArray("selectedGroups");
    $nb=0;
    foreach ($selectedGroupId as $groupId){
        $group=new Groupe();
        $group->setId($groupId);
        if(DAO::delete($group)){
            SessionUtils::removeValueFromArray("selectedGroups", $groupId);
            $nb++;
        }
    }
    echo JsUtils::get("cGroupe/refresh","{}","#groups");
    echo Gui::pluriel("groupe supprimé", "groupes supprimés", $nb);
    echo JsUtils::doSomethingOn("#operation", "hide",200);
}
```

## 5.7 cGroupe/delete

Fonctionnalité	Descriptif
Suppression	<b>Supprimer 1 groupe</b>
	Récupération (GET) de l'id du groupe à supprimer
	Suppression dans la BDD
	Mise à jour de la sélection (en session)
	Affichage du message post-suppression

```
public function delete($id){
    $id=str_replace("delete", "", $id[0]);
    if(is_numeric($id)){
        $group=new Groupe();
        $group->setId($id);
        if(DAO::delete($group)){
            SessionUtils::removeValueFromArray("selectedGroups", $id);
            Gui::showOne($group);
            echo " supprimé";
            echo JsUtils::get("cGroupe/refresh","{}","#groups");
        }else {
            echo "Suppression impossible";
        }
    }
    echo JsUtils::doSomethingOn("#operation", "hide",200);
}
```

## 5.8 cGroupe/select

Fonctionnalité	Descriptif
Sélection	<b>(Dé)Sélectionner 1 groupe</b>
	Récupération (GET) de l'id du groupe à (dé)sélectionner
	Ajout ou suppression de l'id du groupe dans le tableau \$_SESSION["selectedGroups"]
	Affichage du message post-(dé)sélection

```
public function select($id){
    $id=str_replace("ck", "", $id[0]);
    if(is_numeric($id)){
        SessionUtils::addOrRemoveValueFromArray("selectedGroups", $id);
        $nb=sizeof($_SESSION["selectedGroups"]);
        if($nb>0)
            echo JsUtils::doSomethingOn("#multi", "show");
        else
            echo JsUtils::doSomethingOn("#multi", "hide");
        echo Gui::pluriel("groupe s'activer", "groupes s'activer",
$nb);
    }
}
```

## 5.9 cGroupe/selectAll

Fonctionnalité	Descriptif
Sélection	<b>(Dé)Sélectionner tous les groupes</b>
	Récupération de l'opération à effectuer : sélection ou désélection
	Mise à jour dans le tableau \$_SESSION["selectedGroups"]
	Affichage du message post-(dé)sélection

```
public function selectAll(){
    $nb=0;
    $ckAll=SessionUtils::checkBoolean("ckAll");
    if(!$ckAll){
        $_SESSION["selectedGroups"]=array();
    }else{
        $_SESSION["selectedGroups"]=explode(",", $_POST["ids"]);
        $nb=sizeof($_SESSION["selectedGroups"]);
    }
    echo JsUtils::get("cGroupe/refresh/", "{", "#groups");
    echo Gui::pluriel("groupe s'activer", "groupes s'activer", $nb);
}
```

### Documentation technique - Sommaire :

1. [Gestion des utilisateurs](#)
2. [Gestion des applications](#)
3. [Gestion des groupes](#)
4. [Soumettre un report](#)



## 5. Gestion des Droits

From:

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

Permanent link:

<http://slamwiki2.kobject.net/etudiants/2014/bertrand/bugreport3?rev=1386543270>

Last update: **2019/08/31 14:32**

