

# PPE BugReport : application de remontée d'incidents

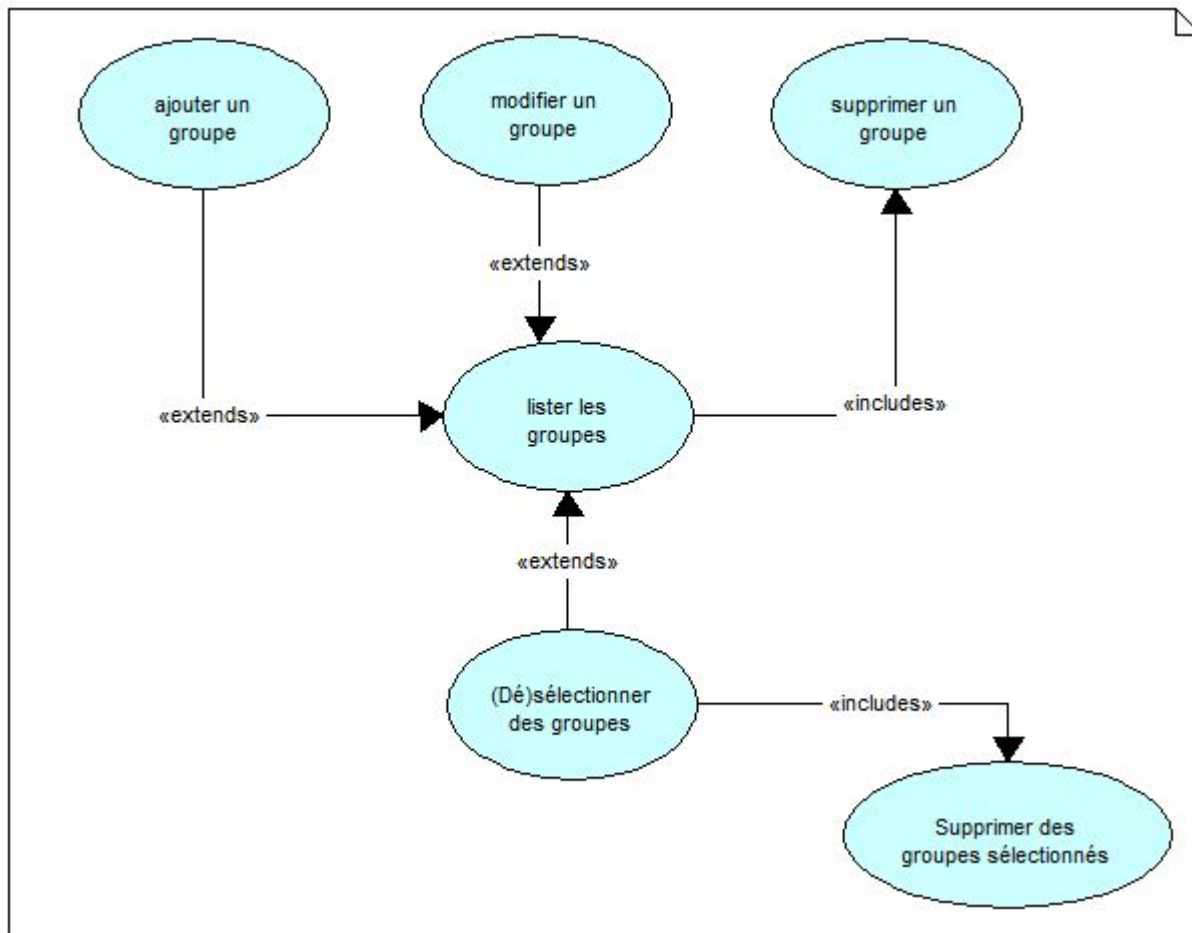
## -- Gestion des utilisateurs

Voir [BugReport : Application de remontée d'incidents](#)

## -- Gestion des groupes

### --Fonctionnalités :

Diagramme des cas d'utilisation pour l'ensemble des fonctionnalités relatives à la gestion des groupes



### --Classes métier

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

Nom	Type	Descriptif
\$_SESSION["ckAllGroups"]	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

## -- Vues

### vHeader

Page en-tête : il s'agit du même fichier que celui pour la gestion des utilisateurs

### vGroups

Liste des groupes (affichage initial et rafraîchissement)

Groupes :  

<input type="checkbox"/> Utilisateurs (5)	✘
<input type="checkbox"/> Administrateurs (2)	✘
<input type="checkbox"/> Chefs de projet (2)	✘
<input type="checkbox"/> Développeurs (6)	✘
<input type="checkbox"/> Tech. Réseaux (5)	✘

Gestion des groupes (Ajout, modification, suppression)

```
<div id="groups">
  <fieldset class="cadre">
    <legend><input type="checkbox" id="ckAll" title="(Dé)sélectionner les
groupes">&nbsp;Groupes :</legend>
    <table id="usersTable">
      <?php foreach ($data['groups'] as $aGroupe) {
        DAO::getOneToMany($aGroupe, "utilisateurs");
      }
      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 sélection :</b>&nbsp;
        <a href="#" class="deleteOther" id="btMultiDelete"
name="btMultiDelete" title="Supprimer les groupes sélectionnés">Supprimer</a>&nbsp;
      </div>
    </div>
  </fieldset>
</div>
<?php echo $data["js"];?>
```

### vFooter

Page pied-de-page : il s'agit du même fichier que celui pour la gestion des utilisateurs

### vAddGroup

[Ajout d'un nouveau groupe ou modification d'un groupe existant](#)



```

<?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* :</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"];?>

```

**--Contrôleurs**

**--cGroupe/index**

Il s'agit du contrôleur de la gestion des groupes.

Fonctionnalité	Descriptif
Gestion des groupes	Affichage de la vue <u>vHeader</u> (en-tête de la page, avec les balises de script notamment)
	Appel du contrôleur <u>refresh</u> (rafraîchissement de la page)
	Affichage de la vue <u>vFooter</u> (pied-de-page)

```

public function index(){
  $this->loadView("vHeader");
  $this->refresh();
  $this->loadView("vFooter", array("message"=>"Gestion des groupes (Ajout,
modification, suppression)"));
}

```

**-- cGroupe/refresh**

Cette méthode permet le rafraîchissement de la page.

Fonctionnalité	Descriptif
Liste des groupes	Chargement de tous les groupes dans la variable <code>\$groups</code>
	Application de la sélection de groupes
	Mise en place logique client <code>#btAdd</code> -> add (ajout du groupe) <code>.deleteOther</code> -> confirmDelete (confirmation de la suppression d'un groupe) <code>.update</code> -> update (mise à jour du groupe) <code>.ck</code> -> select (sélection d'un groupe) <code>#ckAll</code> -> selectAll (sélection de tous les groupes) <code>#deleteMulti</code> -> confirmDelete (confirmation de la suppression simultanée de plusieurs groupes)
	Passage de variables, ensemble des groupes + code JavaScript, dans <code>\$data[]</code> et affichage de la vue <code>vGroups</code>

```
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(".deleteOther", "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("ckAllGroups"));
    $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);
}
```

**-- cGroupe/add**

Cette méthode permet l'ajout d'un nouveau groupe dans la base de données.

Functionalité	Descriptif
Ajout d'un groupe	Si le formulaire est posté (POST)
	Instanciation d'un nouveau groupe Affectation des variables du POST (données saisies par l'utilisateur)aux membres du groupe Ajout dans la base de données Appel du contrôleur <u>refresh</u> pour le rafraîchissement de la page Affichage du message de mise à jour
	Sinon (GET)
	Mise en place logique client #btValidAddGroup -> add (ajout du nouveau groupe) #btCancelAddGroup -> annulation (annulation de l'ajout du groupe) 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 " ajouté";
            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","Opération d\'ajout annulée");
        $js.=JsUtils::doSomethingOn("#operation", "show",400);
        $group=new Groupe();
        $this->loadView("vAddGroup",
        array("group"=>$group,"js"=>$js,"title"=>"Ajouter un groupe :"));
    }
}

```

-- cGroupe/update

Cette méthode permet la mise à jour d'un groupe (modification du libellé du groupe).

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

```

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 " modifié";
            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 :"));
        }
    }
}
    
```

-- cGroupe/confirmDelete

Cette méthode correspond à la confirmation de la suppression d'un ou plusieurs groupes.

Fonctionnalité	Descriptif
Confirmation de la suppression d'un ou plusieurs groupes	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 (suppression d'un ou plusieurs groupes) #cancelDelete -> annulation (annulation de la suppression d'un groupe)

```

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"));
}

```

-- cGroupe/delete

Cette méthode permet de supprimer un groupe de la base de données.

Fonctionnalité	Descriptif
Suppression d'un groupe	Récupération (GET) du \$id du groupe à supprimer
	Suppression du groupe
	Mise à jour de la sélection des groupes
	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);
}

```

-- **cGroupe/deleteMulti**

Cette méthode permet la suppression de plusieurs groupes sélectionnés.

Fonctionnalité	Descriptif
Suppression de plusieurs groupes	Récupération (dans 1 tableau stocké en session) des ids de(s) groupe(s) à supprimer
	Parcours des groupes et suppression
	Mise à jour de la sélection des groupes
	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);
}
```

--**cGroupe/select**

Cette méthode permet de sélectionner ou de désélectionner un groupe.

Fonctionnalité	Descriptif
Sélection d'un groupe	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");
        Gui::pluriel("groupe sélectionné", "groupes sélectionnés", $nb);
    }
}
```

### -- cGroupe/selectAll

Cette méthode permet de sélectionner ou de désélectionner tous les groupes.

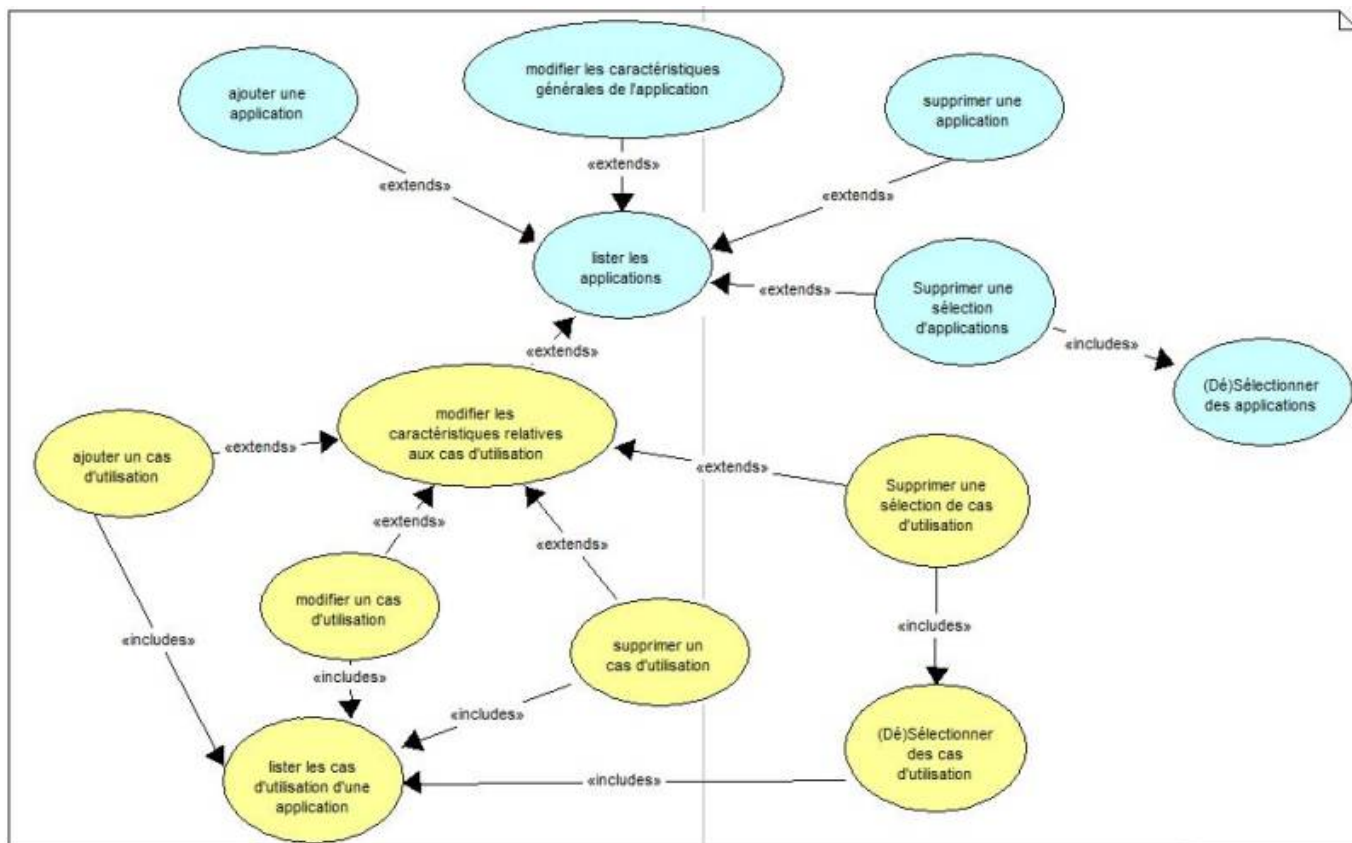
Fonctionnalité	Descriptif
Sélection	Récupération de l'opération à effectuer : sélection ou désélection
	Mise à jour dans le tableau \$_SESSION["selectedGroups"]
	Affichage du message post-(dé)sélection

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

### --Gestion des applications

#### --Fonctionnalités :

Diagramme des cas d'utilisation pour l'ensemble des fonctionnalités relatives à la gestion des applications et des cas d'utilisation



**--Classes métier**

**--Variables de session (\$\_SESSION)**

Nom	Type	Descriptif
\$_SESSION["ckAllApps"]	Booléen	Etat de la case à cocher permettant de sélectionner ou de désélectionner toutes les applications
\$_SESSION["selectedApps"]	Tableau d'entiers	Ids des applications sélectionnées

**-- Vues**

**vHeader**

Page en-tête : il s'agit du même fichier que celui pour la gestion des utilisateurs

**vApps**

Liste des applications (affichage initial et rafraîchissement)

Applications :

<input type="checkbox"/> Essai d'application	✘
<input type="checkbox"/> GSB	✘
<input type="checkbox"/> GSB 2	✘
<input type="checkbox"/> BugReport 2	✘

Ajouter une application...

**Gestion des applications (Ajout, modification, suppression)**

```
<div id="apps">
  <fieldset class="cadre">
    <legend><input type="checkbox" id="ckAll" title="(Dé)sélectionner les
applications">&nbsp;Applications :</legend>
    <table id="appsTable">
      <?php Gui::show($data["apps"],"addDelete")?>
    </table>
    <div class="buttons">
      <div class="btn" id="btAdd" name="btAdd" title="Ajouter une nouvelle
application"><span class="">Ajouter une application...</span></div>
      <div id="multi"><b>Pour la sélection :</b>&nbsp;
        <a href="#" class="delete" id="btMultiDelete" name="btMultiDelete"
title="Supprimer les applications sélectionnées">Supprimer</a>&nbsp;
      </div>
    </div>
  </fieldset>
</div>
<?php echo $data["js"];?>
```

**vFooter**

Page pied-de-page : il s'agit du même fichier que celui pour la gestion des utilisateurs

**--Gestion des applications**

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

Permanent link:  
<http://slamwiki2.kobject.net/etudiants/2014/giroad/bugreport?rev=1383328015>

Last update: **2019/08/31 14:33**

