AJOUT D'UN CHAMP DANS GLPI

le mardi 10 juin 2014.

Thibault GEOFFROY



FORMULAIRE DE CRÉATION D'UN TICKET	3
Les menus déroulants	3
Les champs de saisi numérique	6
Les champs date	7
Les champs oui/non	9
FORMULAIRE DE CRÉATION D'UN SUIVI	11
Les champs de durée	11
Les menus déroulants	13
LISTE DES FONCTIONS AJOUTÉES	17

FORMULAIRE DE CRÉATION D'UN TICKET

Il est possible d'ajouter différents type de champs dans le formulaire de création d'un ticket.

LES MENUS DÉROULANTS



Un menu déroulant est un champ offrant plusieurs options. Ces options sont à ajouter dans le code de l'application. Nous allons prendre comme exemple, l'ajout d'un champ « Périmètre ».

• Ajouter l'attribut dans la base de données :

ALTER TABLE glpi_tickets ADD perimetre int(11) NOT NULL DEFAULT '1';

• Dans la classe : glpi/inc/commonITILObject.class.php Ajouter ces deux fonctions :



```
Fonction retournant le code correspondant au menu déroulant du Périmètre
   param les options(nom de l'attribut, mode l'affichage...)
Preturn le code
           static function dropdownPerimetre(array $options=array()) {
        $p['name'] = 'perimetr
$p['value'] = 0;
$p['showtype'] = 'normal';
$p['display'] = true;
                                                            = 'perimetre';
          if (is_array($options) && count($options)) {
   foreach ($options as $key ⇒ $val) {
     $p[$key] = $val;
                  }
        }
        $id = "select_".$p['name'].mt_rand();
$output = "<select id='$id' name='".$p['name']."'>";
      if ($p['showtype'] == 'normal') {
  $output .= "<option value='1' ".(($p['value'] == 1)?" selected ":"").">".
        __x('perimetre', '-----')."</option>";
  $output .= "<option value='2' ".(($p['value'] == 2)?" selected ":"").">".
        __x('perimetre', 'Construction - DIVERS')."</option>";
   $output .= "<option value='3' ".(($p['value'] == 3)?" selected ":"").">".
        __x('perimetre', 'Construction - DIVERS')."</option>";
   $output .= "<option value='3' ".(($p['value'] == 3)?" selected ":"").">".
        __x('perimetre', 'Construction - DYNAMICS')."</option>";
   $output .= "<option value='4' ".(($p['value'] == 4)?" selected ":"").">".
        __x('perimetre', 'Construction - SI BOBINE')."</option>";
   $output .= "<option value='5' ".(($p['value'] == 5)?" selected ":"").">".
        __x('perimetre', 'Construction - SOFIE')."</option>";
   $output .= "<option value='5' ".(($p['value'] == 5)?" selected ":"").">".
        __x('perimetre', 'Construction - SOFIE')."
                                      }
         $output .= "</select>";
          if ($p['display']) {
                   echo $output;
                    return $id;
           return $output;
```

Ajouter ces quelques lignes dans les trois fonctions ci-dessous :

function getSpecificValueToDisplay

function getSpecificValueToSelect

case 'perimetre' :
\$options['name'] = \$name;
\$options['value'] = \$values[\$field];
return self::dropdownPerimetre(\$options);

function prepareInputForAdd



Dans la classe : glpi/inc/ticket.class.php
 Ajouter ces quelques lignes dans les fonctions ci-dessous :

function prepareInputForUpdate

\$allowed_fields[] = 'perimetre';

function getSearchOptions

<pre>\$tab[126]['table']</pre>	<pre>= \$this->getTable();</pre>
<pre>\$tab[126]['field']</pre>	<pre>= 'perimetre';</pre>
<pre>\$tab[126]['name']</pre>	<pre>=('Périmètre');</pre>
<pre>\$tab[126]['searchtype']</pre>	<pre>= 'equals';</pre>
<pre>\$tab[126]['datatype']</pre>	= 'specific';

function getDefaultValues

'perimetre'

function showFormHelpdesk



function showForm



function commonListHeader

\$items[__('Périmètre')] = "glpi_tickets.perimetre";

Dans la classe : glpi/inc/tickettemplate.class.php
 Ajouter ces deux lignes dans la fonction getAllowedFields :

Pour ajouter une option au menu déroulant, il suffit simplement de se rendre dans le fichier commonITILObject.class.php, et d'ajouter des lignes aux fonctions dropdownPerimetre et getPerimetreName.

LES CHAMPS DE SAISI NUMÉRIQUE

Temps estimé	5,8

Il est possible d'ajouter un champ de saisi. Prenons l'exemple du champ « Temps estimé » permettant de renseigner un nombre décimal.

• Ajouter l'attribut dans la base de données :

ALTER TABLE glpi_tickets ADD temps_estime DECIMAL(5,2) NOT NULL DEFAULT '0.00';

Dans la classe : glpi/inc/ticket.class.php
 Ajouter ces quelques lignes dans les fonctions ci-dessous :

function prepareInputForUpdate



function getSearchOptions



function getDefaultValues



function showForm



function commonListHeader

\$items[__('Temps estimé')]

= "glpi_tickets.temps_estime";

Dans la classe : glpi/inc/tickettemplate.class.php
 Ajouter ces deux lignes dans la fonction getAllowedFields :

<pre>\$ticket->getSear</pre>	chOptionIDByField('field', 'te	emps_estime',
	'glpi_tickets')	<pre>=> 'temps_estime',</pre>

LES CHAMPS DATE



2014-06-06 🖪 09:04 🗙

Un champ date permet de stocker une date ainsi qu'une heure.

• Ajouter l'attribut dans la base de données :

ALTER TABLE glpi_tickets ADD delai_souhaite datetime DEFAULT NULL;

Dans la classe : glpi/inc/ticket.class.php
 Ajouter ces quelques lignes dans les fonctions ci-dessous :

function prepareInputForUpdate

\$allowed_fields[] = 'delai_souhaite';

function getSearchOptions

<pre>\$tab[106]['table']</pre>	<pre>= \$this->getTable();</pre>
<pre>\$tab[106]['field']</pre>	<pre>= 'delai_souhaite';</pre>
<pre>\$tab[106]['name']</pre>	<pre>=('Délai souhaité');</pre>
<pre>\$tab[106]['datatype']</pre>	<pre>= 'datetime';</pre>
<pre>\$tab[106]['massiveaction']</pre>	= false;

function getDefaultValues

function showFormHelpdesk

```
if (!$tt->isHiddenField('delai_souhaite')) {
    echo ";
    printf(__('%1$s%2$s'), __('Délai souhaité'), $tt->getMandatoryMark('delai_souhaite'));
    echo "

    #delai_souhaite = $velues["delai_souhaite"];
    Html::showDateTimeFormItem("delai_souhaite", $delai_souhaite, 1, false);
    echo "

    #delai_souhaite;;
```

function showForm



function commonListHeader

\$items[_('Délai souhaité')] = "glpi_tickets.delai_souhaite";

• Dans la classe : glpi/inc/tickettemplate.class.php

Ajouter ces deux lignes dans la fonction getAllowedFields :

Thibault GEOFFROY

LES CHAMPS OUI/NON



• Ajouter l'attribut dans la base de données :

ALTER TABLE glpi_tickets ADD conformite_sox int(1) NOT NULL DEFAULT '0';

Dans la classe : glpi/inc/ticket.class.php
 Ajouter ces quelques lignes dans les fonctions ci-dessous :

function prepareInputForUpdate

\$allowed_fields[] = 'conformite_sox';

function getSearchOptions

function getDefaultValues

```
'conformite_sox' >> 0,
```

function showForm

```
/* Conformité SOX */
echo "";
echo "".$tt->getBeginHiddenFieldText('conformite_sox');
printf(__('%1$s%2$s'), __('Conformité SOX'), $tt->getMandatoryMark('conformite_sox'));
echo $tt->getEndHiddenFieldText('conformite_sox')."";
echo $tt->getBeginHiddenFieldValue('conformite_sox');
if ($canupdate) {
    Dropdown::showYesNo('conformite_sox', $this->fields["conformite_sox"]);
} else {
    echo Dropdown::getYesNo($this->fields["conformite_sox"]);
}
echo $tt->getEndHiddenFieldValue('conformite_sox', $this);
echo *tt->getEndHiddenFieldValue('conformite_sox', $this);
echo *tt->
```

commonListHeader

\$items[_('Conformité SOX')] = "glpi_tickets.conformite_sox";

• Dans la classe : glpi/inc/tickettemplate.class.php Ajouter ces deux lignes dans la fonction getAllowedFields :

FORMULAIRE DE CRÉATION D'UN SUIVI

LES CHAMPS DE DURÉE

Durée

```
0 V Jour(s) 0 V Heure(s) 0 V Minute(s)
```

• Ajouter l'attribut dans la base de données :

ALTER TABLE glpi_ticketfollowups ADD day *int*(11) NOT NULL DEFAULT '0'; ALTER TABLE glpi_ticketfollowups ADD hour *int*(11) NOT NULL DEFAULT '0'; ALTER TABLE glpi_ticketfollowups ADD minute *int*(11) NOT NULL DEFAULT '0';

• Dans la classe : glpi/inc/ticketfollowup.class.php

Ajouter ces quelques lignes dans les fonctions ci-dessous :

function showForm

```
/* Durée */
echo "";
echo "";
echo "", sprintf(_('%1$s'), _('Durée'))."";
echo "";
sidday = "value_day".mt_rand();
echo "<input id='$idday' type='hidden' name='day' value='".$this->fields["day"]."'>";
Dropdown::dropdownInteger('day',$this->fields["day"],0,31);
echo sprintf(_('%1$s'), _('Jour(s)'));
$idhour = "value_hour".mt_rand();
echo "<input id='$idhour' type='hidden' name='hour' value='".$this->fields["hour"]."'>";
Dropdown::dropdownInteger('hour',$this->fields["hour"],0,23);
echo sprintf(_('%1$s'), _('Heure(s)'));
$idminute = "value_minute".mt_rand();
echo "<input id='$idminute".type='hidden' name='minute' value='".$this->fields["minute"]."'>";
Dropdown::dropdownInteger('hour',$this->fields["hour"],0,23);
echo sprintf(_('%1$s'), _('Heure(s)'));
$idminute = "value_minute".mt_rand();
echo "<input id='$idminute' type='hidden' name='minute' value='".$this->fields["minute"]."'>";
Dropdown::dropdownInteger('minute',$this->fields["minute"],0,59);
echo sprintf(_('%1$s'), _('Minute(s)'));
echo sprintf(_('%1$s'), _('Minute(s)'));
echo sprintf(_('%1$s'), _('Minute(s)'));
echo "
```

function getSearchOptions



function showFormMassiveAction



function showSummary

echo "" . _('Durée') . "";

- Il faut ajouter cette ligne pour ajouter une colonne
- « Durée »dans l'affichage des suivis d'un ticket.

function showInTicketSumnary



Dans la classe : glpi/inc/ticket.class.php

Ajouter ces 3 lignes dans le case « add_followup » de la fonction doSpecificMassiveActions

'day'	<pre>> \$input['day'],</pre>
'hour'	<pre>>> \$input['hour'],</pre>
'minute'	<pre>sinput['minute']);</pre>

Dans la classe : glpi/inc/dropdown.class.php

Ajouter cette fonction :

```
function dropdownInteger($myname,$value,$min=0,$max=100,$step=1,$toadd=array()){
    echo "<select name='$myname'>\n";
    if (count($toadd)){
        foreach ($toadd as $key => $val){
            echo "<option value='$key' ".($key==$value?" selected ":"").">$val</option>";
        }
        for ($i=$min;$i<=$max;$i+=$step){
        echo "<option value='$i' ".($i==$value?" selected ":"").">$i</option>";
        }
        echo "<option value='$i' ".($i==$value?" selected ":"").">$i</option>";
        }
        echo "<option value='$i' ".($i==$value?" selected ":"").">$i</option>";
        echo "</option value='$i' ".($i==$value?" selected ":"").">$i</option>";
        echo "<option value='$i' ".($i==$value?" selected ":"").">$i</option>";
        echo "
```

LES MENUS DÉROULANTS

✓ Ouvert
Approuvé
Approuvé business
Planifié
Etudié
En développement
Réalisé
Recetté
En production
Approuvé à postériori
Clôturé
Abandonné

Prenons l'exemple d'un menu déroulant proposant différent Statut afin qu'un suivi puisse renseigner cette information.

• Ajouter l'attribut dans la base de données :

ALTER TABLE glpi_ticketfollowups ADD status int(11) NOT NULL DEFAULT '0';

Dans la classe : glpi/inc/ticketfollowup.class.php
 Ajouter ces quelques lignes dans les fonctions ci-dessous :

function showForm



function getSearchOptions

<pre>\$tab[6]['table']</pre>	<pre>= \$this->getTable();</pre>
<pre>\$tab[6]['field']</pre>	= 'status';
<pre>\$tab[6]['name']</pre>	<pre>=('Statut SOX');</pre>
<pre>\$tab[6]['datatype']</pre>	<pre>= 'dropdown';</pre>

function showFormMassiveAction

```
echo "<br>&nbsp;".__('Statut SOX')."&nbsp;";
Dropdown::showAllStatus('status',1);
```

function showSummary

echo "" . _('Statut SOX') . ";

function showInTicketSumnary

echo "" . DropDown::getStatusName(\$this->fields["status"]) . "";

Dans la classe : glpi/inc/ticket.class.php

Ajouter cette ligne dans le case « add_followup » de la fonction doSpecificMassiveActions

'status' => \$input['status'],

Dans la classe : glpi/inc/dropdown.class.php

Ajouter ces fonctions :

```
Fonction permettant de récupérer le libellé du statut en fonction de son code */

static function getStatusName($value) {
   switch ($value) {
      case 0 :
    return _x('status', '-----');
      case 1 :
    return _x('status', 'Ouvert');
      case 2 :
    return _x('status', 'Approuvé');
        return _x('status', 'Approuvé business');
        return _x('status', 'Planifié');
         return _x('status', 'Etudié');
      case 6 :
        return _x('status', 'En développement');
         return _x('status', 'Réalisé');
      case 8 :
         return _x('status', 'Recetté');
         return _x('status', 'En production');
          return _x('status', 'Approuvé à postériori');
         return _x('status', 'Clôturé');
         return _x('status', 'Abandonné');
         return $value;
  }
 ì
```

```
static function showAllStatus($name, $value=0, $restrict_to=-1, $params=array()) {
    if ($restrict_to != 1) {
    $options[1] = __('Ouvert');
    }
    if ($restrict_to != 2) {
    $options[2] = __('Approuvé');
}
    }
    if ($restrict_to != 3) {
    $options[3] = __('Approuvé business');
    }
    if ($restrict_to != 4) {
    $options[4] = __('Planifié');
    }
    if ($restrict_to != 5) {
        $options[5] = __('Etudié');
    3
    if ($restrict_to != 6) {
        $options[6] = __('En développement');
    }
    if ($restrict_to != 11) {
        $options[11] = __('Clôturé');
    3
    if ($restrict_to != 12) {
        $options[12] = __('Abandonné');
    $params['value'] = $value;
    return self::showFromArray($name, $options, $params);
 }
```

LISTE DES FONCTIONS AJOUTÉES

- Dans la classe : glpi/inc/commonITILObject.class.php
- function getStatus_fuName
- function dropdownStatus_fu
- function getUsedStatus_fuBetween
- function getSiteName
- function dropdownSite
- function getUsedSiteBetween
- function getDomaineName
- function dropdownDomaine
- function getUsedDomaineBetween
- function getNatureName
- function dropdownNature
- function getUsedNatureBetween
- function getTypePACRName
- function dropdownTypePACR
- function getUsedTypePACRBetween
- function getPerimetreName
- function dropdownPerimetre
- function getUsedPerimetreBetween
- function getProjetName
- function dropdownProjet
- function getUsedProjetBetween
- Dans la classe : glpi/inc/dropdown.class.php

function showAllStatus function getStatusName function dropdownInteger