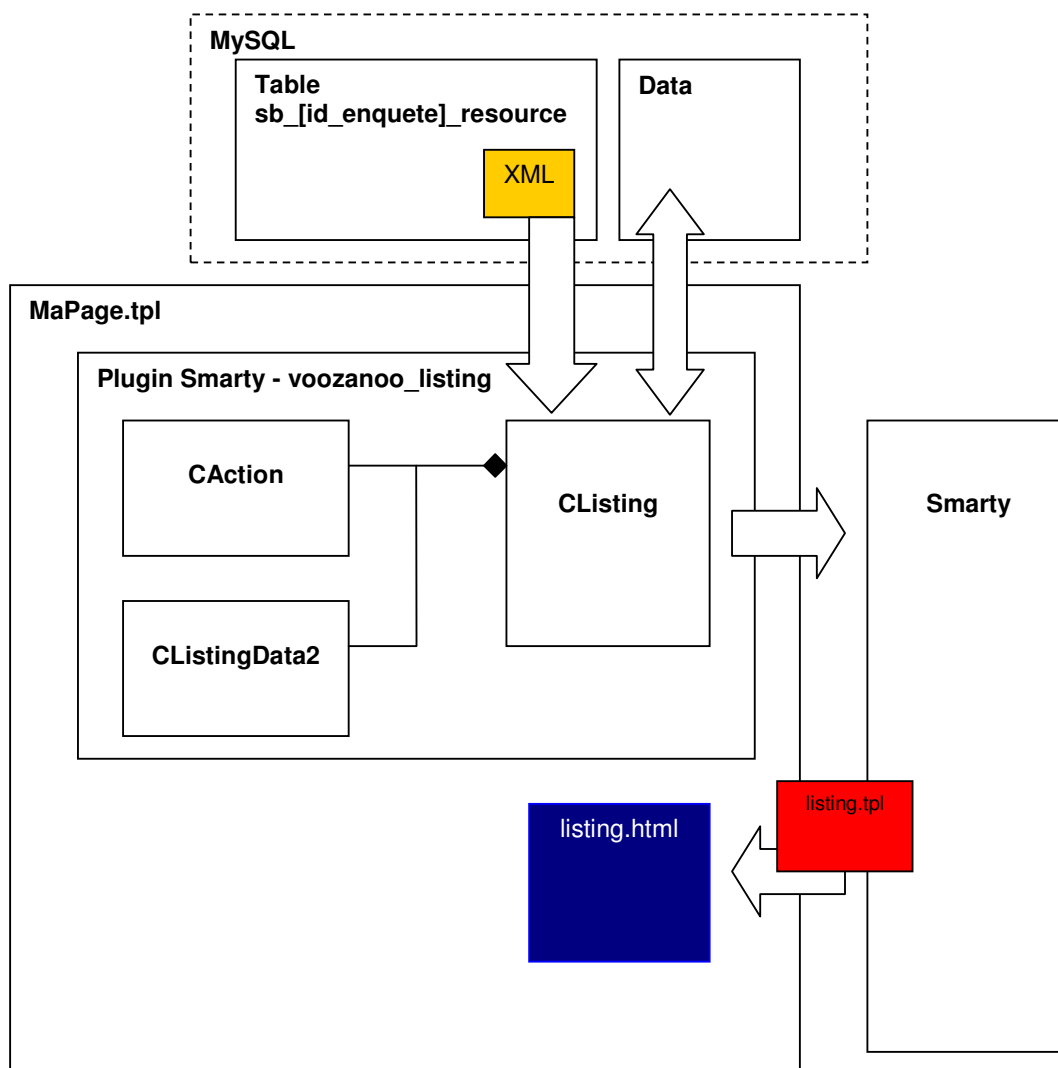


XML listing

Fonctionnement du module Listing	2
Utilisation du module via le plugin Smarty voozanoo_listing	2
Utilisation du module dans un script PHP	2
Structure du XML	4
Définition du format du XML	4
Titre du listing	5
Déclaration des colonnes	6
Déclaration d'une colonne	6
Exemple 1, affichage d'une variable simple	6
Exemple 2, affichage d'une variable à réponses multiples	6
Exemple 3, affichage de deux colonnes	6
Exemple 4, utilisation d'une requête SQL pour transformer la date au format européen	6
Exemple 5, utilisation de deux variables dans une requête SQL	6
Exemple 6, utilisation de la balise dico	6
Déclaration des conditions	6
Fin du XML	6
Jointure avec des tables annexes	6
Exemple d'un fichier XML complet	6
Format du tableau {\$listing} dans le template	6

Fonctionnement du module Listing

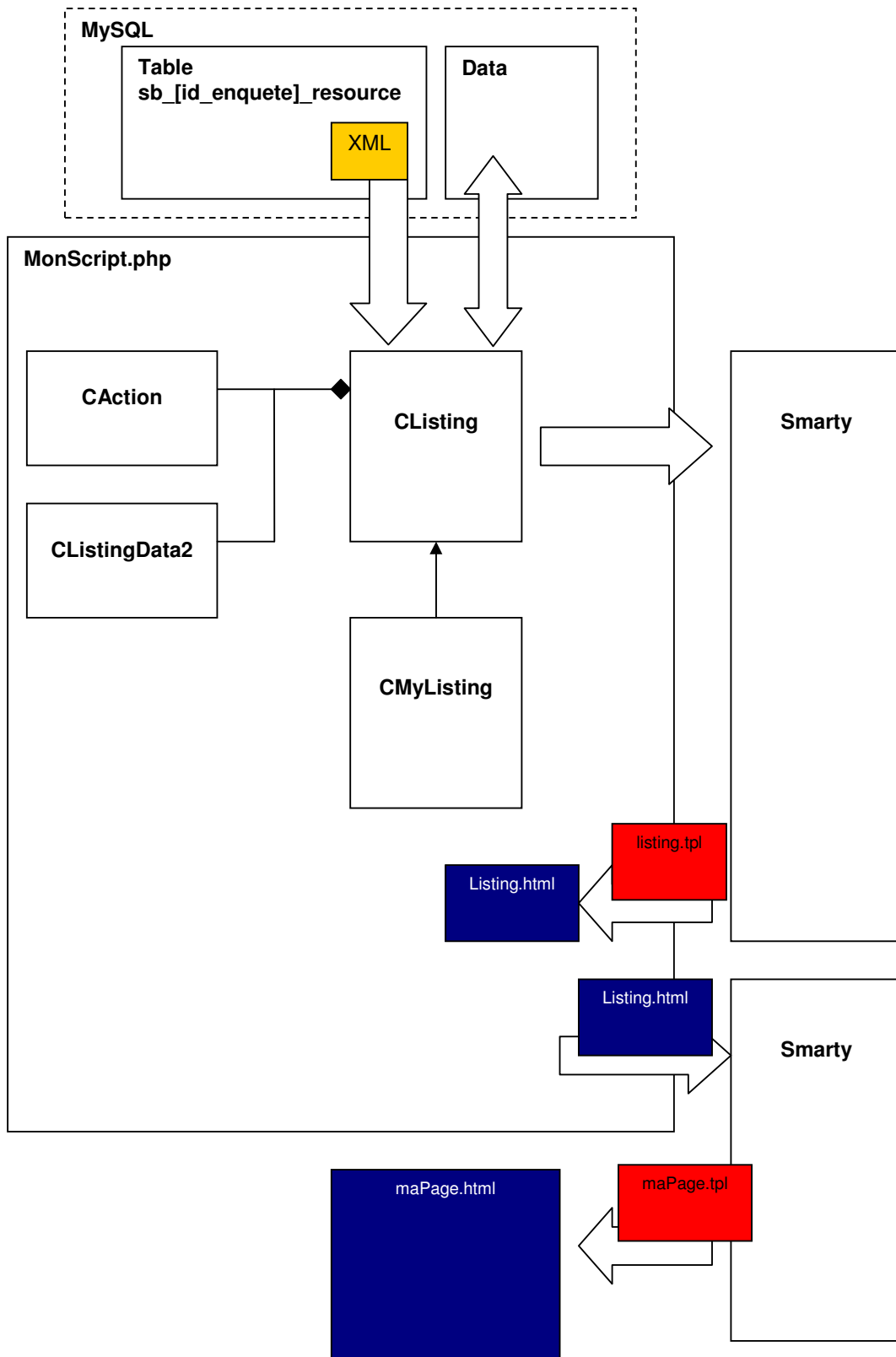
Utilisation du module via le plugin Smarty voozanoo_listing



Utilisation du module dans un script PHP

L'exemple suivant montre qu'il est possible de dériver une class depuis la class `CListing` pour modifier son comportement. Dans ce cas, nous n'utilisons plus le plugin Smarty, mais un script PHP.

NB : il est possible d'utiliser la nouvelle class dans un plugin Smarty qui sera spécifique au projet.



Structure du XML

Définition du format du XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

Entête du XML

```
<listing id="465" version="1">
```

Id : identifiant du listing. Il doit être unique. L'identifiant permet de retrouver les paramètres du listing dans la session.

Liste des paramètres :

- 1- Numéro de la page
- 2- Nombre de lignes à afficher
- 3- Variable utilisée pour faire le tri
- 4- Tri par ordre croissant ou décroissant
- 5- Valeurs entrées dans le moteur de recherche (si un moteur de recherche est lié au listing).

Ces paramètres, qu'ils proviennent du XML, ou qu'ils aient été changés par l'utilisateur, sont enregistrés dans la base de données.

Version : version du XML. Si la version du XML change, alors toutes les données du listing se trouvant dans la session sont remises à leur valeur par défaut (valeurs provenant du XML).

Paramètres

```
<default parameters>  
  <begin>0</begin>  
  <range>20</range>  
  <id_condition>10</id_condition>  
  <order desc="false">name</order>  
  <group_by>id_patient</group_by>  
</default>
```

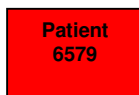
- Begin* : numéro de la page
- Range* : nombre de ligne à afficher
- Id_condition : identifiant du filtre servant de moteur de recherche. Optionnel
- Order* : colonne sur laquelle est fait le tri (voir chapitre déclaration d'une colonne).
 - Desc = « true » : tri croissant
 - Desc = « false » : tri décroissant
- Group_by : permet de regrouper les lignes en fonction de la valeur d'une variable. Dans cet exemple, tous les patients avec le même identifiant seront regroupés sur une seule ligne. Cela permet d'utiliser les fonctions mathématiques de MySQL (somme, moyenne, etc.).

* Ces valeurs peuvent être changées par l'utilisateur via l'interface. Ce sont donc des valeurs par défaut.

```
<master>1125205142</master>
```

Master : identifiant du questionnaire principal. Il s'agit, parmi tous les questionnaires qui sont affichés dans le listing, du questionnaire qui se situe au sommet de l'arbre relationnel.

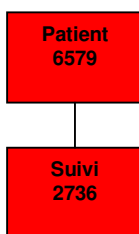
Exemple 1 :



Affichage uniquement des variables du questionnaire patient.

Identifiant master : 6579

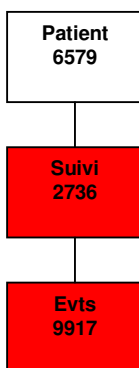
Exemple 2 :



Affichage des variables du questionnaire patient et du questionnaire suivi.

Identifiant master : 6579

Exemple 3 :



Affichage des variables du questionnaire suivi et du questionnaire evts.

Identifiant master : 2736

Titre du listing

```
<title lang="fr">Listing des patients</title>  
<title lang="en">Patient listing</title>
```

Si l'enquête gère plusieurs langues, il faut alors mettre un titre par langue. L'identification de la langue se fait via la variable « lang ».

Actuellement les deux langues gérées sont :

Anglais : en

Français : fr

Déclaration des colonnes

```
<columns>
```

Balise de départ pour la déclaration des colonnes. Toutes les colonnes doivent se situer à l'intérieur de cette balise.

Déclaration d'une colonne

Un listing peut contenir un nombre indéfini de colonnes, provenant de questionnaires différents. Si plusieurs questionnaires sont utilisés, il est nécessaires qu'ils soient tous de type principal. Impossible d'afficher un même temps un questionnaire principal, un questionnaire annexe et un questionnaire de niveau.

```
<column id="col name" pos="1">
```

Id : identifiant de la colonne. Il doit être unique et ne peut contenir de caractères spéciaux (accents, espaces, virgules, etc.)

Pos : Position de la colonne dans le listing

```
<title lang="fr">Nom</title>
```

Titre de la colonne. Une déclaration de titre par langue gérée dans l'enquête.

```
<dico>etablissements</dico>
```

Optionnel: remplace la valeur renvoyée par sa correspondance dans le dictionnaire indiqué.

```
<sql>
```

Le contenu d'une colonne est défini par le biais d'une requête SQL.

```
<expr>substring('%s', 1, 5)</expr>
```

Expression de la colonne au format SQL. A l'intérieur de l'expression, les variables sont indiquées par les caractères « %s ».

```
<"params">
```

Définition des variables passées à la requête SQL.

```
<param type="var" pos="1">
```

Type : **deprecated**

Pos : position de la variable dans la requête SQL.

```
<tabletype>data</tabletype>
```

Table source.

Valeurs possibles :

- 1- Data : contient les variables du formulaire, hors variables à réponses multiples

- 2- Lookup : contient les variables du formulaire à réponse multiple
- 3- Group : contient le nom du groupe propriétaire de la fiche.
- 4- User : Contient des informations relatives au propriétaire de la fiche.

```
<varname>nom idt</varname>
```

Nom de la variable à récupérer.

Si la variable vient de la table data ou de la table lookup, entrer les noms du formulaire (exemple : nom, sexe, ddn, etc.)

Pour les tables group et user, liste des noms possibles :

Group :

- 1- name

User :

- 1- username

```
<id questionnaire>576301158</id questionnaire>
```

Identifiant du questionnaire d'où proviennent les données.

```
<value>aspirine</value>
```

Dans le cas où la variable est contenue dans une table de type lookup, il faut alors indiquer la valeur que va contenir la colonne. En effet, il faut créer une colonne par valeur possible si l'on souhaite afficher toutes les réponses possibles.

```

                </param>
            </params>
        </sql>
    </column>
</columns>

```

Exemple 1, affichage d'une variable simple

```

<column id="nom_pat" pos="1">
  <title lang="fr">Nom du patient</title>
  <sql>
    <expr>%s</expr>
    <params>
      <param type="var" pos="1">
        <tabletype>data</tabletype>
        <varname>nom</varname>
        <id_questionnaire>6579</id_questionnaire>
      </param>
    </params>
  </sql>
</column>

```

Exemple 2, affichage d'une variable à réponses multiples

Dans cet exemple, nous affichons une colonne indiquant si la valeur aspirine a été cochée ou non dans le formulaire.

```
<column id="ttt_asp" pos="1">
  <title lang="fr">Traitement : aspirine</title>
  <sql>
    <expr>%s</expr>
    <params>
      <param type="var" pos="1">
        <tabletype>lookup</tabletype>
        <varname>ttt</varname>
        <value>aspirine</value>
        <id_questionnaire>6579</id_questionnaire>
      </param>
    </params>
  </sql>
</column>
```

Exemple 3, affichage de deux colonnes

```
<column id="ttt_asp" pos="1">
  <title lang="fr">Traitement : aspirine</title>
  <sql>
    <expr>%s</expr>
    <params>
      <param type="var" pos="1">
        <tabletype>lookup</tabletype>
        <varname>ttt</varname>
        <value>aspirine</value>
        <id_questionnaire>6579</id_questionnaire>
      </param>
    </params>
  </sql>
</column>
<column id="ttt_antibio" pos="2">
  <title lang="fr">Traitement : antibiotiques</title>
  <sql>
    <expr>%s</expr>
    <params>
      <param type="var" pos="1">
        <tabletype>lookup</tabletype>
        <varname>ttt</varname>
        <value>antibiotiques</value>
        <id_questionnaire>6579</id_questionnaire>
      </param>
    </params>
  </sql>
</column>
```


Exemple 4, utilisation d'une requête SQL pour transformer la date au format européen

```

<column id="date_de_debut" pos="5">
  <title lang="fr">Entrée</title>
  <sql>
    <expr>DATE_FORMAT(%s,'%d/%m/%Y')</expr>
    <params>
      <param type="var" pos="1">
        <tabletype>data</tabletype>
        <varname>date_debut_mvt</varname>
        <id_questionnaire>1154447495</id_questionnaire>
      </param>
    </params>
  </sql>
</column>

```

NB : la fonction DATE_FORMAT de MySQL utilise le caractère « % » pour indiquer le format de sortie. Pour des raisons de compatibilité, il faut doubler ce caractère, auquel cas la requête ne fonctionnera pas.

NB2 : attention, avec date_format, le tri ne fonctionne plus correctement, car le résultat de date_format est un chaîne de caractère et non plus une donnée de type date. Pour que le tri fonctionne sur une sortie au format européen, il faut faire la transformation au niveau du template.

Exemple 5, utilisation de deux variables dans une requête SQL

```

<column id="age" pos="1">
  <title lang="fr">Age lors de l'événement</title>
  <sql>
    <expr>ROUND(datediff(%s, %s) / 365.25) </expr>
    <params>
      <param type="var" pos="1">
        <tabletype>data</tabletype>
        <varname>date_evts</varname>
        <id_questionnaire>1154447495</id_questionnaire>
      </param>
      <param type="var" pos="2">
        <tabletype>data</tabletype>
        <varname>date_naiss</varname>
        <id_questionnaire>45464457</id_questionnaire>
      </param>
    </params>
  </sql>
</column>

```

Exemple 6, utilisation de la balise dico

```

<column id="date_de_debut" pos="5">
  <title lang="fr">Entrée</title>
  <dico>sexe</dico>
  <sql>
    <expr>%s</expr>
    <params>
      <param type="var" pos="1">
        <tabletype>data</tabletype>
        <varname>sexe</varname>
        <id_questionnaire>1154447495</id_questionnaire>
      </param>
    </params>
  </sql>
</column>

```

Déclaration des conditions

```
<conditions>
```

Les conditions sont déclarées comme les colonnes, mais elles ne sont pas affichées à l'écran. Leur intérêt est d'être utilisé directement dans le template notamment comme condition pour certains effets graphiques.

Par exemple, il est possible d'utiliser une condition pour mettre les lignes d'une certaine couleur en fonction des réponses données sur les formulaires.

```

<condition id="warning_date_prel">
  <sql>
    <expr>if(%s="O",TRUE,FALSE)</expr>
    <params>
      <param type="var" pos="1">
        <tabletype>data</tabletype>
        <varname>sitop_prel</varname>
        <id_questionnaire>6457</id_questionnaire>
      </param>
    </params>
  </sql>
</condition>
</conditions>

```

Syntaxe pour récupérer le résultat d'une condition dans un template :

\$listing.table.row[i].cond.warning_date_prel

Fin du XML

```
</listing>
```

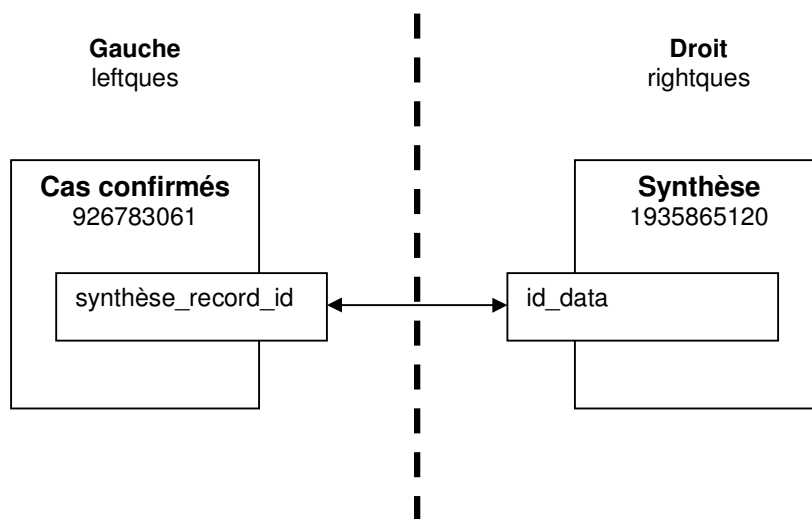
Jointure avec des tables annexes

Par défaut, le module listing permet de faire des jointures entre des questionnaires de type « principal ».

Il est possible en utilisant la balise « <joins> » de définir des jointures avec des tables annexes.

NB : il est également possible de faire un listing qui n'est composé que de tables annexes.

Définition d'une jointure



```

...
<title>Cas confirmés</title>
<joins>
  <join>
    <leftques>
      <id>926783061</id>
      <join_var>synthese_record_id</join_var>
    </leftques>
    <rightques alias="synt">
      <id>1935865120</id>
      <join_var>id_data</join_var>
    </rightques>
  </join>
</joins>
<columns>
...

```

La balise « <id> » correspond à l'identifiant du questionnaire, et la balise « <join_var> » à la variable qui servira de jointure.

Insertion de variables issues d'une table jointe

```

<column id="synt_num_ref" pos="61">
  <title lang="fr">N° enquête</title>
  <sql>
    <expr>%s</expr>
    <params>
      <param type='var' pos='1'>
        <alias>synt</alias>
        <varname>fich_num_ref</varname>
      </param>
    </params>
  </sql>
</column>...

```

Contrairement aux colonnes standards, on n'utilise pas les balises « <tabletype> » et « <id_questionnaire> » pour définir l'origine de la variable. A la place on utilise la balise « <alias> » qui a été définie dans les jointures.

Définition de plusieurs jointures

```

<joins>
  <join>
    <leftques>
      <id>926783061</id>
      <join_var>synthese_record_id</join_var>
    </leftques>
    <rightques alias="synt">
      <id>1935865120</id>
      <join_var>id_data</join_var>
    </rightques>
  </join>
  <join>
    <leftques>
      <id>926783061</id>
      <join_var>commune</join_var>
    </leftques>
    <rightques alias="com">
      <id>1426722264</id>
      <join_var>id_data</join_var>
    </rightques>
  </join>
  <join>
    <leftques>
      <id>926783061</id>
      <join_var>synthese_record_id</join_var>
    </leftques>
    <rightques alias="synt2">
      <id>2142100718</id>
      <join_var>id_data</join_var>
    </rightques>
  </join>
</joins>

```

Définition de jointures multiples

```
<joins>
  <join>
    <leftques>
      <id>926783061</id>
      <join_var>synthese_record_id</join_var>
    </leftques>
    <rightques alias="synt">
      <id>1935865120</id>
      <join_var>id_data</join_var>
    </rightques>
  </join>
  <join>
    <leftques alias="synt">
      <id>1935865120</id>
      <join_var>commune</join_var>
    </leftques>
    <rightques alias="com">
      <id>1426722264</id>
      <join_var>id_data</join_var>
    </rightques>
  </join>
</joins>
```

Dans cet exemple, la table « cas confirmés » (id 926783061), est jointe à la table « synthèse » (id 1935865120) qui elle-même est liée à la table des communes (id 1426722264).

Dans la seconde jointure, il est nécessaire d'indiquer l'alias défini avant : **<leftques alias="synt">**

Exemple d'un fichier XML complet

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<listing id="prelevements_ressource_07" version="8">
  <default_parameters>
    <begin>0</begin>
    <id_condition>7</id_condition>
    <range>20</range>
    <order desc="false">IPP</order>
  </default_parameters>
  <master>2072042111</master>
  <title lang="fr">Liste des prélèvements</title>
  <conditions>
    <condition id="warning_date_prel">
      <sql>
      <expr>if(%s="O",TRUE,FALSE)</expr>
      <params>
        <param type="var" pos="1">
          <tabletype>data</tabletype>
          <varname>sitop_prel</varname>
          <id_questionnaire>3215</id_questionnaire>
        </param>
      </params>
      </sql>
    </condition>
  </conditions>
  <columns>
    <column id="nom" pos="1">
      <title lang="fr">Nom</title>
      <sql>
      <expr>%s</expr>
      <params>
        <param type="var" pos="1">
          <tabletype>data</tabletype>
          <varname>nom_idt</varname>
          <id_questionnaire>21654</id_questionnaire>
        </param>
      </params>
      </sql>
    </column>
    <column id="date_prelevement" pos="2">
      <title lang="fr">Date prel.</title>
      <sql>
      <expr>CAST(%s as DATE)</expr>
      <params>
        <param type="var" pos="1">
          <tabletype>data</tabletype>
          <varname>date_prel</varname>
          <id_questionnaire>3215</id_questionnaire>
        </param>
      </params>
      </sql>
    </column>
  </columns>
</listing>

```

Format du tableau {\$listing} dans le template

```
Array
(
  [data] => Array
    (
      [nb_rows] => 3
      [id_questionnaire] => 2072042111
      [nb_fields] => 11
    )
  [table] => Array
    (
      [title] => Array
        (
          [0] => Array
            (
              [title] => Nom
              [name] => nom
              [type] => string
            )
          [1] => Array
            (
              [title] => Date prel.
              [name] => date_prelevement
              [type] => date //Par défaut string, il faut caster la colonne pour obtenir le type date
            )
        )
      [row] => Array
        (
          [0] => Array
            (
              [data] => Array
                (
                  [0] => DUVAL
                  [1] => 31/05/2007
                )
              [cond] => Array
                (
                  [warning_date_prel] => 1
                )
              [id_data] => 1044
              [locked] => N
            )
          [1] => Array
            (
              [data] => Array
                (
                  [0] => DUPONT
                  [1] => 04/07/2007
                )
              [cond] => Array
                (
                  [warning_date_prel] => 0
                )
              [id_data] => 1045
              [locked] => N
            )
        )
    )
)
```