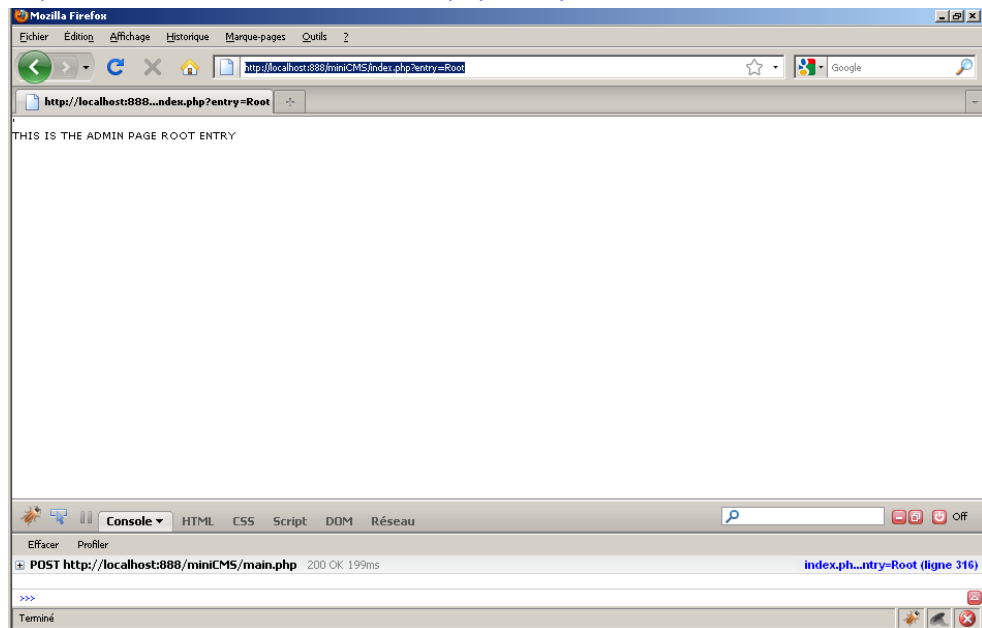


Il y a trois points d'entrée dans la solution :

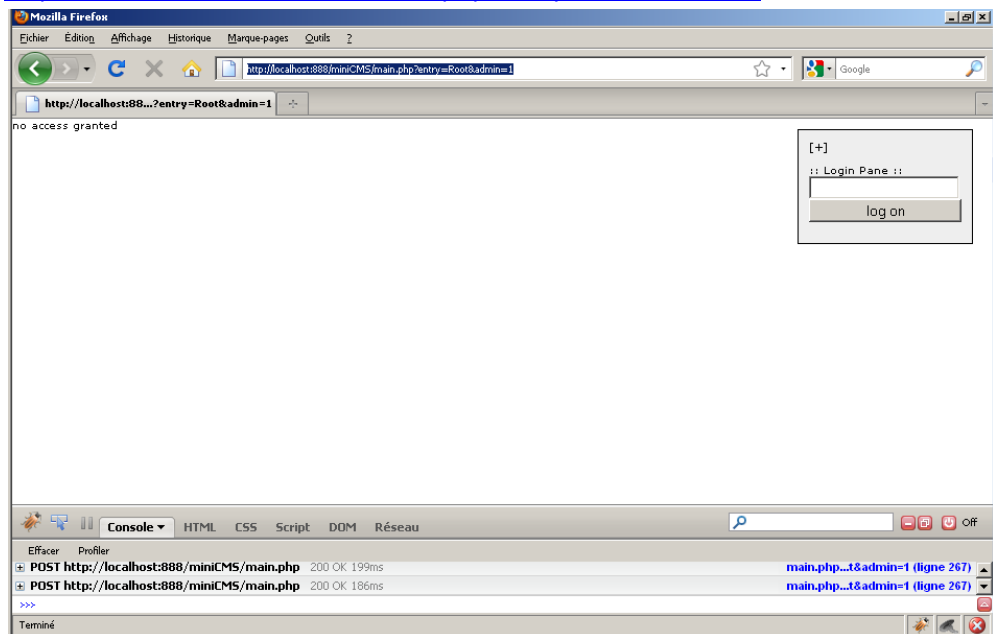
- La sortie extranet pour les internautes :

- <http://localhost:888/miniCMS/index.php?entry=Root>



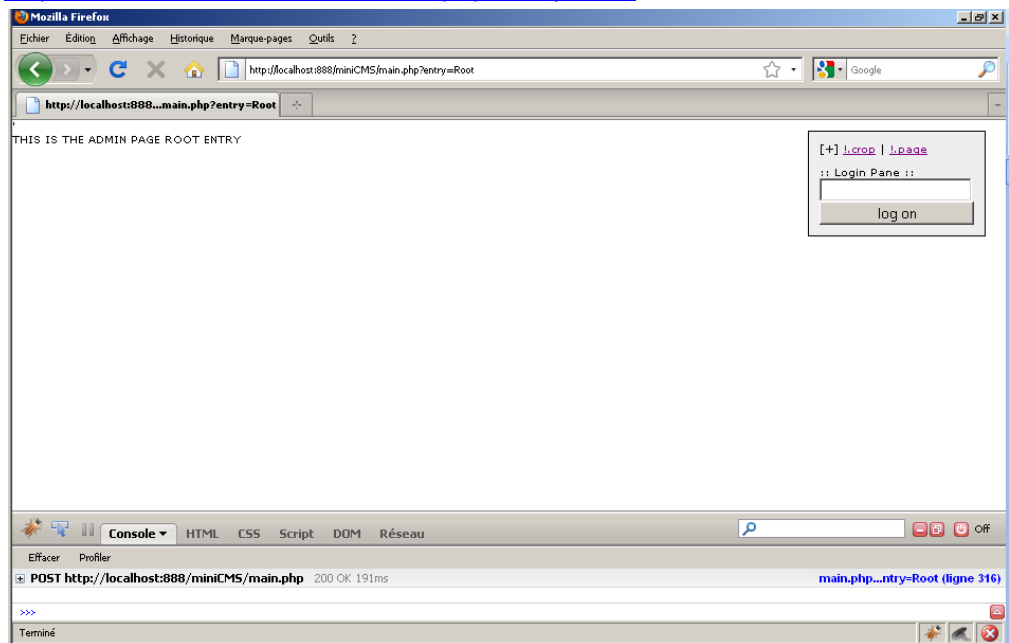
- L'interface d'administration des comptes

- <http://localhost:888/miniCMS/main.php?entry=Root&admin=1>



- L'interface du CMS

- <http://localhost:888/miniCMS/main.php?entry=Root>



On voit qu'il y a deux portes :

main.php       => entrée du cms  
index.php      => sortie pour internautes

On voit qu'il y a des paramètres dans l'adresse

:

entry   => est le groupe client  
admin   => appelle l'interface des admins

Les trois suivantes sont gérées par les menus :

page   => appelle la page  
user   => appelle les users  
crop   => appelle les tables de données xml

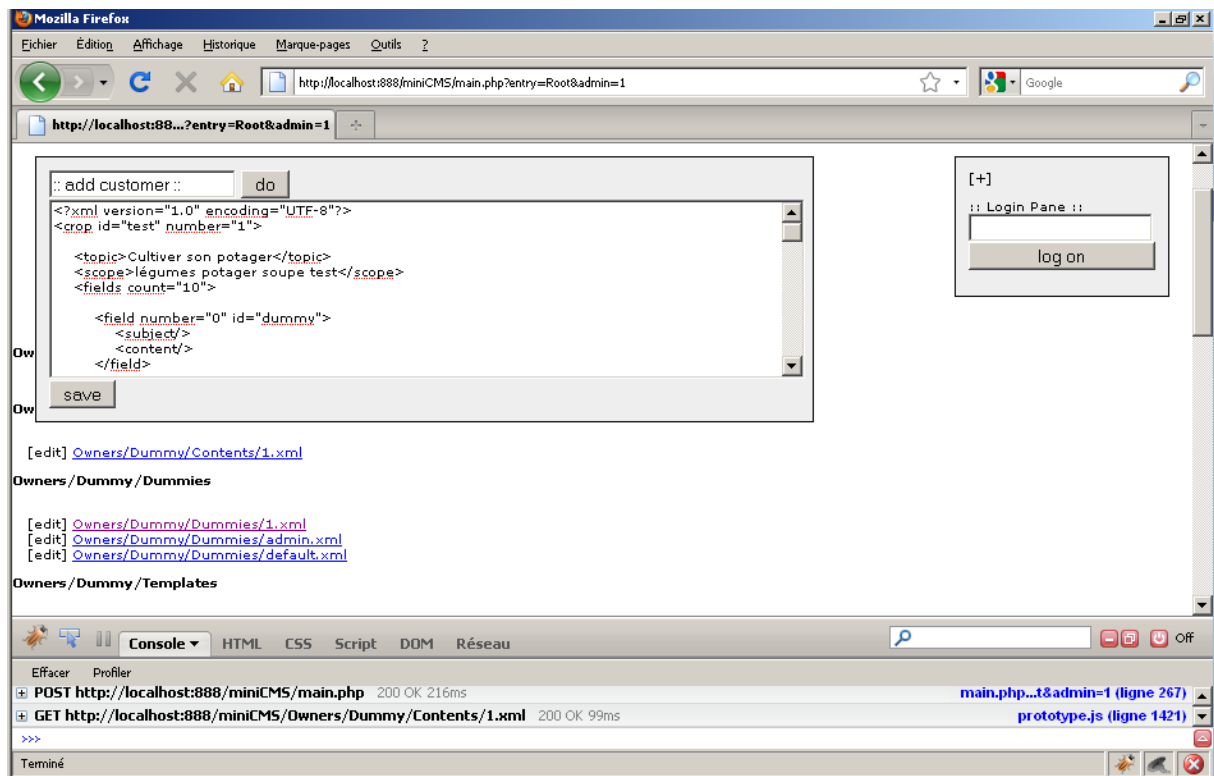
Le mot de pass admin est :

Pass

Le point d'entrée par défaut est un répertoire caché :

Dummy => point d'entrée de protection contre les hackers et robots

## L'interface d'admin



Après avoir introduit le mot de passe admin on voit :

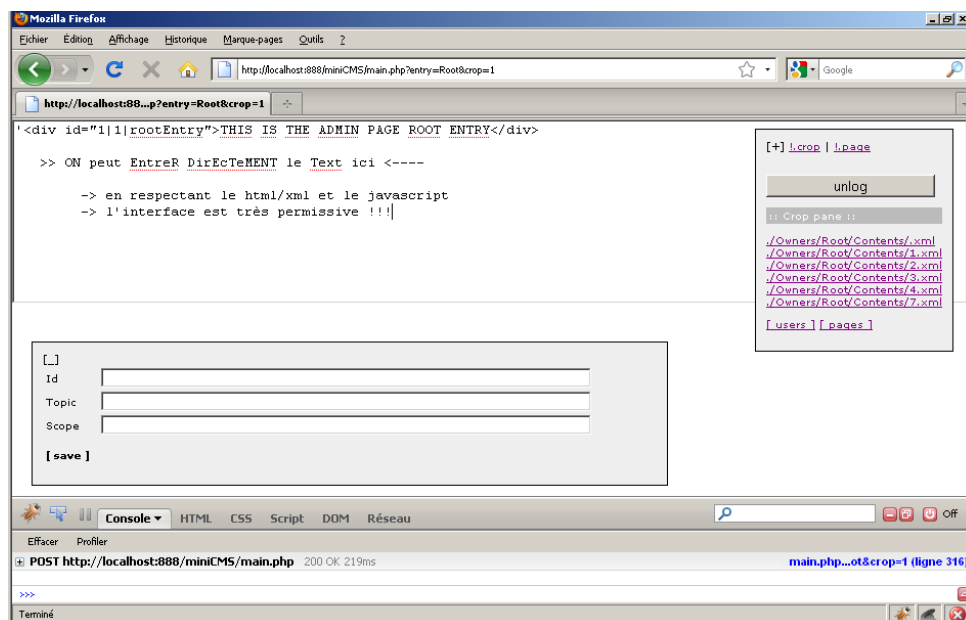
- [edit] => édite les champs de données
- :: add customer :: => permet de rajouter un customer
- ::Login pane :: => champs d'authentification

## L'interface du CMS

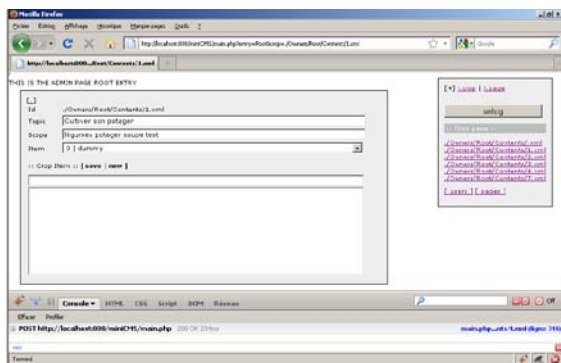
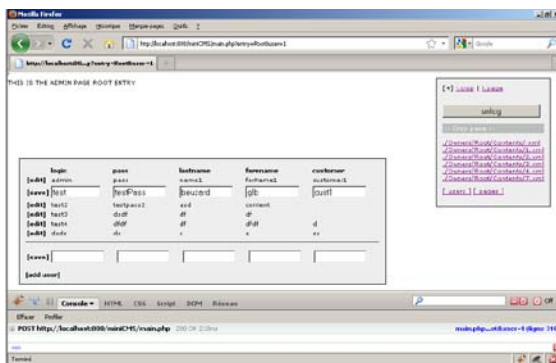
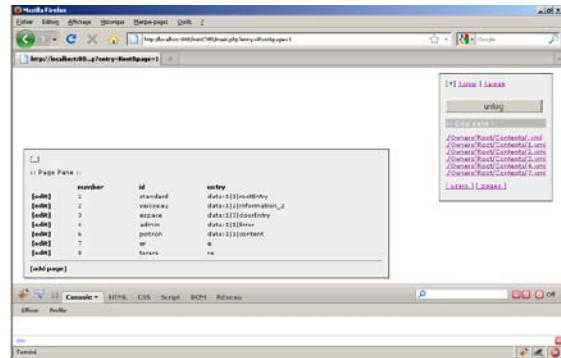
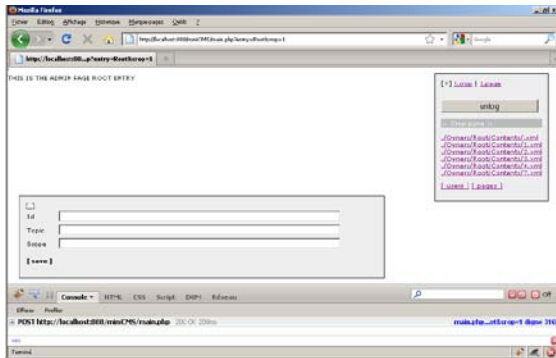
Il y a deux moyens d'encoder le contenu html ou xml :

1. Via les boites grisée
2. En double cliquant sur le texte

L'interface est prévue pour monitorer directement le contenu à l'écran durant l'encodage.



1. Nouveau champ de données
2. Gestion des pages
3. Gestion des users
4. Gestion des crops ou champs de données



## Gerer et travailler par le template :

Lorsque vous rajouter un nouveau customer un répertoire lui est créé.

Exemple FakeCustomer :

```
./Owners/FakeCustomer/  
    Contents/  
    Templates/  
    .../
```

Sous Templates prenons le default.xml

Le layout de la page est sous le nœud template/frameset/content

```
<content>  
<!--  
  
-->  
<![CDATA[  
  
    <?xml version="1.0" encoding="UTF-8" ?>  
    {{inc:token[xhtmlTransitional]}}  
    <html>  
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />  
        <head>  
  
            <style>{{inc:style[standard]}}</style>  
  
            <script>{{inc:script[topScript]}}</script>  
            <script>{{inc:script[pagerScript]}}</script>  
  
            {{inc:script[prototype]}}  
  
        </head>  
  
        <body onload="feedContent()">  
  
            {{inc:customer}}  
  
            {{inc:menuPane}}  
            {{inc:bottomPane}}  
  
            <!-- Page entry -->  
            <div id='mainContent'>  
  
                {{inc:page[standard]}}  
  
            </div>  
  
        </body>  
  
        <script>  
  
            {{inc:script[endScript]}}  
  
        </script>  
  
    </html>  
  
    ]>  
</content>
```

Les segments de pages sont placés dans default.xml de la manière suivante :

**{{inc:style[standard]}}** désigne l'inclusion style portant l'id standard soit dans le fichier :  
../content/<style id='standard'>...</style>

```
{{inc:page[standard]}}
```

```
= <page id="standard">
- <![CDATA[

    <input type="hidden" id="iCounter" name="nCounter" value="0" />
    <div id="lay:frameContent" class="toFeed" >> mainEntry container</div>

]]>
</page>
```

Ici, l'id lay :frameContent demande dynamiquement son homologue dans le template pour sortir le contenu au browser soit :

```
= <page id="frameContent">
- <![CDATA[

    <div id="lay:topContent"></div>
    <div id="lay:mainContent"></div>
    <div id="lay:bottomContent"></div>

]]>
</page>
```

Pour Conclure ,

On voit que frameContent appels trois containers lay :...

Il s'agit des données avec les id : topContent, mainContent, bottomContent de default.xml

Pour appeler du contenu depuis le template il faut :

Mettre le préfix **lay:idDelItem**

Pour appeler du contenu depuis les données il faut :

Mettre le préfix **data:nomDelItem**

Il est donc possible de construire le backend du site via le layout en respectant les appels par id , il en va de même pour les scripts.

Certaines inclusions sont injectés directement par la couche php, mais je déconseil de manipuler le code source du CMS. Via l'html il est possible de concevoir son site entièrement.

LES INCLUSIONS DEPUIS PHP SONT :

```
{{inc:customer}}    ← code client
{{inc:menuPane}}    ← menu supérieur droit
{{inc:bottomPane}}  ← panneau inferieur gauche
```

Il est possible à un confirmé d'injecter d'autres données sans trop de difficultés.

## Gerer et travailler avec les données :

Les données sont les items regroupés dans les champs de données ou « crop » en anglais.

Ils ont

- Un number
- Un id

### Ils se distinguent dans la table des crops par

IdDeCrop|numeroItem|IdItem

### Ils peuvent s'injecter dans les nœuds html par :

```
<div id='data:1|1|rootEntry'></div>
```

La balise sera remplie par la donnée numéro 1 du crop 1, rootEntry étant l'id de l'item

Si cette donnée vaut :

Ceci est le contenu de l'item rootEntry

Le container deviendra :

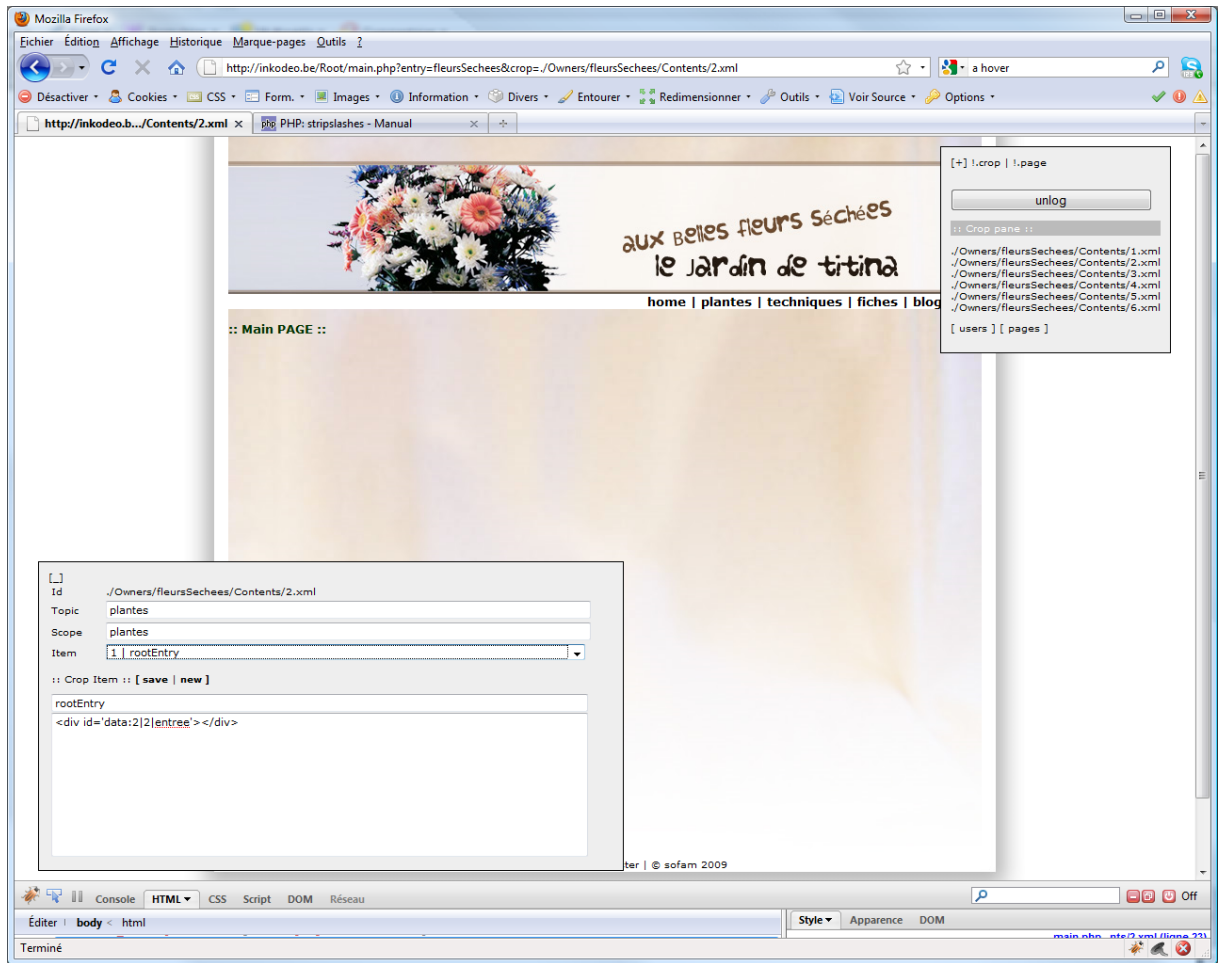
```
<div id='1|1|rootEntry'>
  Ceci est le contenu de l'item rootEntry
</div>
```

On remarque que le préfix data a disparu. La donnée a été injectée.

Un segment html comme une table peut être découpé en sous données :

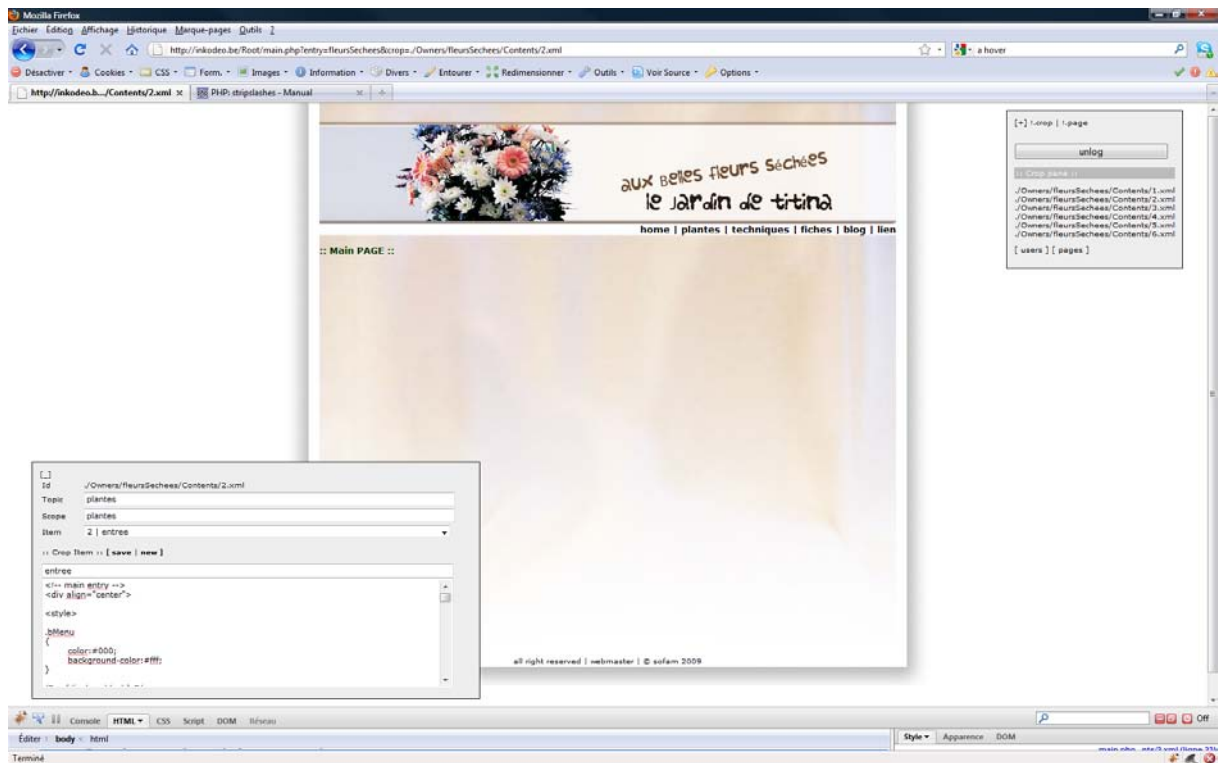
```
<table>
  <tr>
    <td>
      <p id='data :2|1|colonie1'></p>
    </td>
    <td>
      <p id='data :2|1|colonie2'></p>
    </td>
  </tr>
</table>
```

L'exemple en page suivante montre comment c'est en pratique :





La donnée 2 | entree étant :



```
<!-- main entry -->
<div align="center">

<style>

.bMenu
{
    color:#000;
    background-color:#fff;
}

/* a {display: block} */
a:link, a:visited {text-decoration: none; color: #000}
a:hover {text-decoration: none; color: #aaa; background: #fff}

</style>

<table border="0" cellspacing="0" cellpadding="0" width="800">

    <tr valign="top" height="10">
        <td style="background-image:url('http://inkodeo.be/Root/Owners/fleursSechees/Objects/ico/t_l.jpg'); background-
repeat:repeat-y;"></td>
        <td colspan="2">
            
        </td>
        <td style="background-image:url('http://inkodeo.be/Root/Owners/fleursSechees/Objects/ico/t_r.jpg'); background-
repeat:repeat-y;"></td>
    </tr>

    <tr valign="top" height="600">
        <td style="background-image:url('http://inkodeo.be/Root/Owners/fleursSechees/Objects/ico/t_l.jpg'); background-
repeat:repeat-y;">
        </td>
        <td colspan="2" style="background-image:url('http://inkodeo.be/Root/Owners/fleursSechees/Objects/bck_003.jpg');
background-repeat:no-repeat; font-family: Verdana, Geneva, sans-serif; font-weight: bold; font-size: small; color: #030;">

        <!-- MENU -->
```

```

<div class="bMenu" align="right">

    <a href="http://inkodeo.be/Root/index.php?entry=fleursSechees&page=standard">home</a>
|
    <a href="http://inkodeo.be/Root/index.php?entry=fleursSechees&page=plantes">plantes</a>
|
    <a
href="http://inkodeo.be/Root/index.php?entry=fleursSechees&page=techniques">techniques</a> |
    <a href="http://inkodeo.be/Root/index.php?entry=fleursSechees&page=fiches">fiches</a> |
    <a href="http://inkodeo.be/Root/index.php?entry=fleursSechees&page=blog">blog</a> |
    <a href="http://inkodeo.be/Root/index.php?entry=fleursSechees&page=lien">lien</a></div>

<p/>
<!-- Content -->
<div id="mainContent">

    <span id="data:2|3|intro"></span>

</div>

</td>
<td style="background-image:url('http://inkodeo.be/Root/Owners/fleursSechees/Objects/ico/t_r.jpg'); background-
repeat:repeat-y;"></td>
</tr>

<tr height="100" valign="top">
    <td style="background-image:url('http://inkodeo.be/Root/Owners/fleursSechees/Objects/ico/b_l.jpg'); background-
repeat:no-repeat;"></td>
    <td style="background-image:url('http://inkodeo.be/Root/Owners/fleursSechees/Objects/ico/b.jpg'); background-
repeat:repeat-x;" colspan="2">

        <div align="center">all right reserved | webmaster | © sofam 2009</div>

    </td>
    <td style="background-image:url('http://inkodeo.be/Root/Owners/fleursSechees/Objects/ico/b_r.jpg'); background-
repeat:no-repeat;"></td>
</tr>

</table>

</div>

```

## Travailler et gérer les pages :

Les pages sont appelées par la variable page dans l'adresse :

<http://inkodeo.be/Root/main.php?entry=fleursSechees&page=standard>

En pratique, pour une page on a un point d'entrée dans les données ou le layout

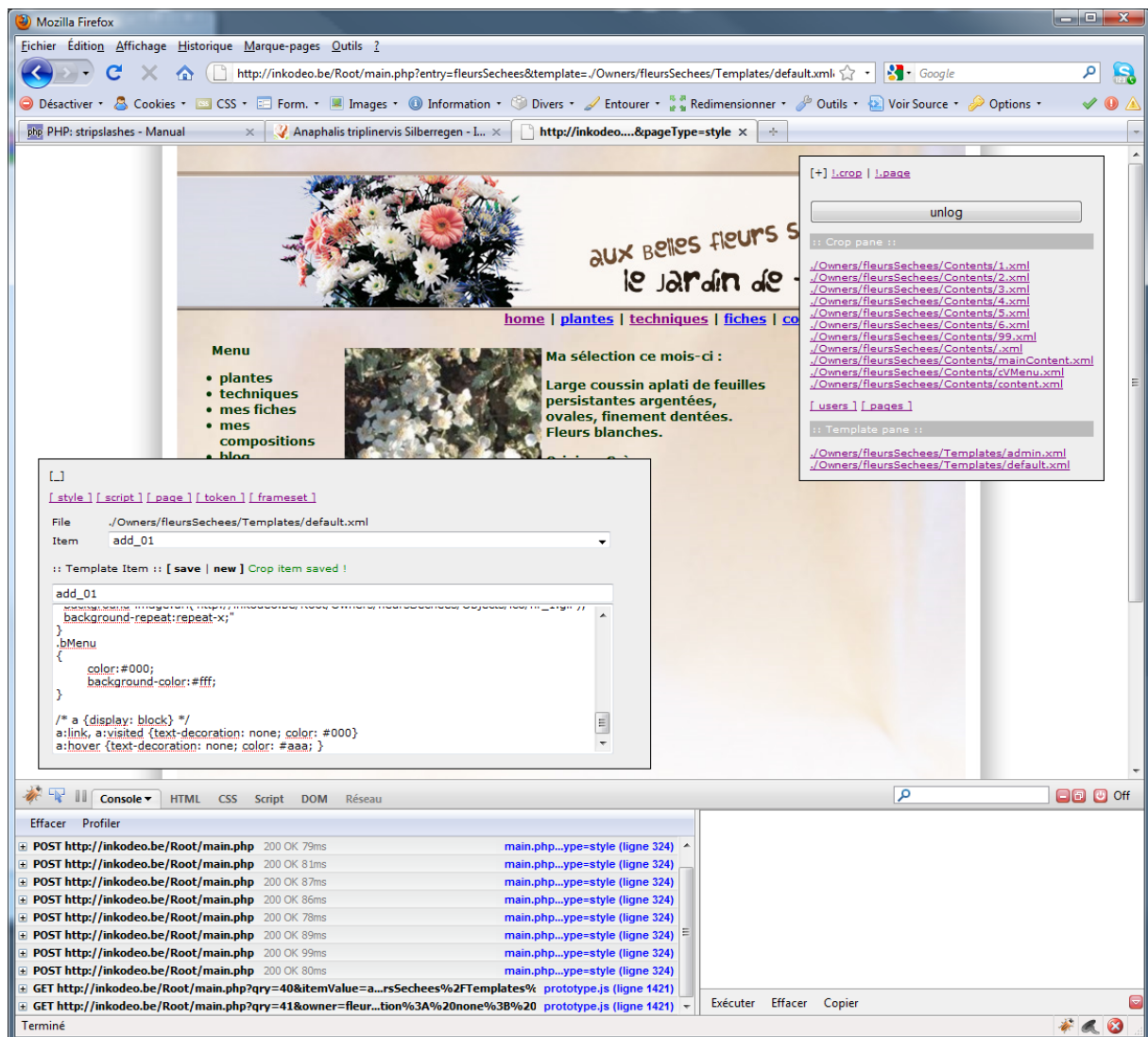
[ ]			
:: Page Pane ::			
	<b>number</b>	<b>id</b>	<b>entry</b>
[edit]	1	standard	data:1 1 rootEntry
[edit]	2	plantes	data:2 1 rootEntry
[edit]	3	techniques	data:3 1 rootEntry
[edit]	4	fiches	data:4 1 rootEntry
[edit]	5	blog	data:5 1 rootEntry
[edit]	6	lien	data:6 1 rootEntry
[add page]			

Soit pour la page 'plantes' la donnée du crop 2, item 1 qui a pour nom rootEntry.

Par simplicité j'ai créé un crop par page 1,2,...,6.

## Travailler avec les templates

Cette page n'a pas de style sauf ceux par défaut. Nous allons lui mettre un style personnalisé :



Nous sélectionnons le template default.xml et ajoutons un nœud add\_01 et lui mettons du style.

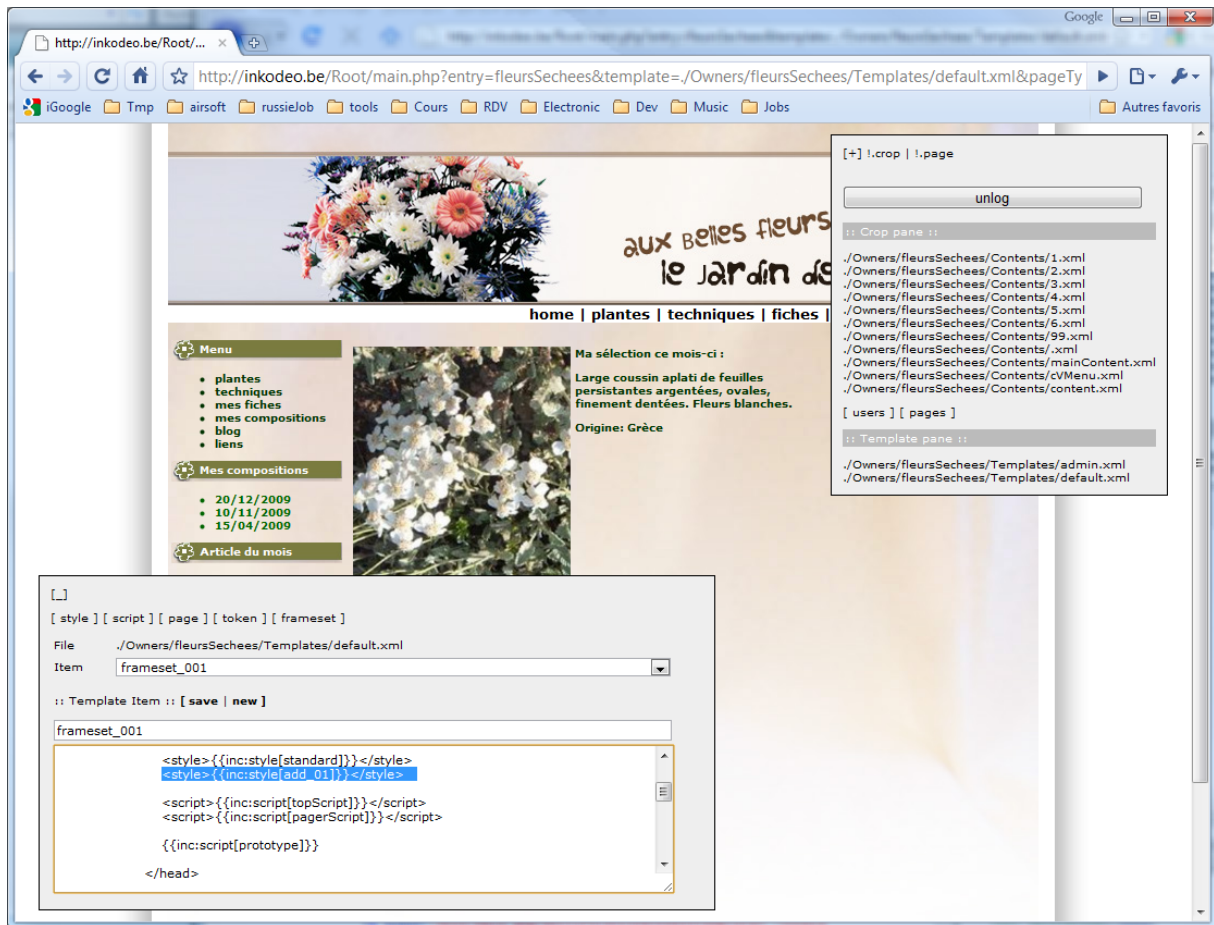
L'inclusion directement au sein du template default.xml ou admin.xml se fait par point d'encrage sur le champs comme dans l'image suivante.

```
{{inc :nomDeLaBalise[idDeLaBalise]}}
```

Soit :

```
{{inc :style [add_01]}}
```

On rajoute un appel au style dans le template :



Et on remarque que la mise en page s'est adaptée :

