Kiubi™ Un outil, c'est tout.

Guide du Designer

1ère partie, Généralités

23 mars 2009
Version 1.1
Par Marc Beyer

Le guide du Designer est une documentation technique des possibilités de personnalisation offertes par Kiubi.

Cette personnalisation peut se faire à deux niveaux :

1. en utilisant un thème graphique personnalisé, qui vous garanti un accès complet aux fichiers HTML, CSS et XML qui composent le site internet. La 1ère partie du guide du Designer abordera ce point en détail.

2. en modifiant les paramètres, les configurations et les modèles graphiques des différents Widgets proposés par Kiubi. Les 2ème et 3ème parties du guide du Designer aborderont ce point en détail.

Guide du Designer **1**
1ère partie, Généralités **1**
1 Utiliser un thème graphique personnalisé **4**
1.1 Créer un thème graphique personnalisé 4
1.1.1 Première méthode (recommandée) 4
1.1.2 Seconde méthode 4
1.2 L'arborescence des fichiers 4
1.2.1 Prérequis 5
1.2.2 Le fichier « theme.xml » 5
1.2.3 L'arborescence des fichiers d'un thème graphique personnalisé 5
2 Pré-requis **7**
2.1 Les balises 7
2.1.1 Définition 7
2.1.2 Exemple d'utilisation d'une balise 8
2.2 Les balises globales 8
2.2.1 Liste des balises globales 8
2.3 Les filtres 12
2.4 Les blocs 12
3 Les types de billets **13**
3.1 Définition 13
3.2 Principe 14
3.3 Le fichier « config.xml » 14
3.3.1 Exemple de code source 14
3.3.2 Description 15
3.3.3 Le champ "liste" 15
3.4 Le fichier « index.html » 15
3.4.1 Exemple de code source 16
3.4.2 Description 16
3.4.3 Balises disponibles 16
3.5 Ajouter un nouveau type de billet 17
4 Les types des produits **17**
4.1 Définition 17
4.2 Principe 18
4.3 Le fichier « config.xml » 18
4.3.1 Exemple de code source 19
4.3.2 Description 19
4.3.3 Le champ "liste" 19
4.4 Le fichier « index.html » 20
4.4.1 Exemple de code source 20
4.4.2 Balises disponibles 22
4.5 Ajouter un nouveau type de produit 29
5 La Médiathèque **30**
5.1 Définition 30
5.2 Accéder à un fichier de la Médiathèque 30
5.3 Le cas particulier des images 30
6 Les includes **30**
6.1 Définition 30
6.2 Principe 31
7 Les modules **31**
7.1 Définition 31

7.2 La page d'erreur 404 ... 31
7.2.1 Répertoire ... 31
7.2.2 Exemple du code source ... 31
7.3 La page de fermeture du site ... 32
7.3.1 Répertoire ... 32
7.3.2 Exemple du code source ... 32
7.4 Le chemin de navigation ... 32
7.4.1 Répertoire ... 33
7.4.2 Exemple du code source ... 33
7.4.3 Balises disponibles ... 33
8 Les modèles de mise en page (templates) ... 34
8.1 Définition ... 34
8.2 Principe ... 34
8.3 Le fichier « desc.xml » ... 35
8.3.1 Exemple de code source ... 35
8.3.2 Description ... 35
8.4 Le fichier « structure.xhml » ... 35
8.4.1 Exemple de code source ... 35
8.4.2 Description ... 36
8.5 Le fichier « index.html » ... 36
8.5.1 Exemple de code source ... 36
8.5.2 Description ... 38
8.5.3 Balises disponibles ... 38
8.5.4 Le moteur de recherche ... 39
8.6 Ajouter un nouveau modèle de mise en page ... 40
9 Les Widgets ... 40
9.1 Définition ... 40
9.2 Classement ... 41
9.3 Personnalisation ... 41
9.3.1 Modifier les fichiers sources d'un widget ... 41
9.3.2 Ajouter des modèles graphiques supplémentaires à un widget ... 42
10 Les fonctionnalités et les balises communes des widgets ... 42
10.1 {alterne} ... 42
10.2 {compteur} ... 43
10.3 {intitule} ... 43
10.4 {ctl} ... 44
10.5 La navigation ... 44
10.5.1 Balises disponibles ... 46

1 Utiliser un thème graphique personnalisé

A partir de la formule « Entreprise » de Kiubi il est possible d'utiliser un thème graphique personnalisé qui vous permet d'intégrer une ligne graphique réalisée sur mesure pour un site internet.

Kiubi permet de personnaliser à 100% un site, grâce à son accès aux gabarits de mise en page. **Vous pourrez donc directement modifier le code source de votre site internet. Aucune limitation graphique particulière n'est imposée !**

Cependant, comme tout système destiné à créer des sites professionnels et respectueux des standards du web, concevoir un thème graphique sur mesure requiert la maîtrise du XHTML et des CSS, **un thème graphique personnalisé est donc réservé aux utilisateurs expérimentés !**

1.1 Créer un thème graphique personnalisé

1.1.1 Première méthode (recommandée)

Pour créer un thème graphique personnalisé, rendez-vous dans la Console d'administration dans « *Apparence / Gestion des thèmes* ». Par défaut, un des thèmes graphiques fournis en standard par Kiubi (appelé thème graphique par défaut) est utilisé. Il suffit de cliquer sur le bouton « *Utiliser un thème personnalisé* » et de compléter et valider le formulaire. Dès cet instant, le thème graphique personnalisé à été créé et initialisé à partir du thème par défaut choisi.

Vous pouvez vous connecter à votre espace FTP avec les informations fournis dans la rubrique « *Apparence / Gestion des thèmes* » ou la rubrique « *Mon compte / Préférences / FTP* » pour accéder à l'intégralité des fichiers du thème graphique.

1.1.2 Seconde méthode

Pour utiliser un thème graphique personnalisé vous devez au préalable vous connecter à votre espace FTP (les informations de connexion sont disponibles dans la rubrique « *Mon compte / Préférences / FTP* »). Une fois connecté à votre espace FTP, vous devez publier les fichiers utiles à la création d'un thème graphique personnalisé dans le répertoire « *theme* ».

Par défaut le répertoire « *theme* » de votre compte FTP est vide, vous pouvez télécharger les fichiers sources de tous les thèmes graphiques standards de Kiubi dans la rubrique « *Apparence / Gestion des thèmes* ».

Décompressez le fichier téléchargé avec un logiciel de décompression (7zip, Winzip, Winrar, ...) puis publiez les fichiers directement dans le répertoire « *theme* » de votre compte FTP.

1.2 L'arborescence des fichiers

Une fois connecté à votre espace FTP, deux répertoires sont disponibles :

- le répertoire « *mediatheque* » : nécessaire à la Médiathèque pour y publier des fichiers de plus de 7Mo.
- le répertoire « *theme* » : contenant les fichiers de votre thème graphique personnalisé.

1.2.1 Prérequis

Afin de déclarer un thème graphique personnalisé, il est indispensable de bien configurer les fichiers suivants :

- *illustration.jpg :* Image d'illustration du thème graphique personnalisé destinée à être affichée dans la gestion des thèmes afin d'identifier plus facilement le thème en question. Il est recommandé d'utiliser une image de 300px par 225px.

- *theme.xml :* Ce fichier contient des informations descriptives du thème graphique. **Il est indispensable pour déclarer le thème graphique personnalisé.**

1.2.2 Le fichier « theme.xml »

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

<theme img="illustration.jpg" intitule="Black Lagoon" style="Moderne" version="1.2"
date="2006/08/04" >

    <auteur nom="Troll d'idées" contact="contact@troll-idees.com" site="www.troll-idees.com"
/>

</theme>
```

Ce fichier contient une seule balise principale `<theme>` qui possède 5 attributs :

- *img :* nom du fichier image qui est utilisé dans la console d'administration pour représenter ce modèle (correspond au fichier image nommé ci-dessus),
- *intitule :* intitule du thème qui est affiché dans la console d'administration,
- *style :* le style du thème (moderne, classique, etc...),
- *version :* la version du thème qui est affiché dans la console d'administration,
- *date :* la date de la dernière mise à jour du thème.

A l'intérieur, nous avons une seule balise `<auteur>` qui permet d'ajouter des informations sur l'auteur du thème. Cette balise contient les attributs suivants :

- *nom :* le nom de l'auteur qui est affiché dans la console d'administration,
- *contact :* l'email de l'auteur,
- *site :* l'adresse du site internet de l'auteur.

1.2.3 L'arborescence des fichiers d'un thème graphique personnalisé

L'arborescence d'un thème graphique personnalisé contenu dans le répertoire « *theme* » de votre espace FTP est organisée de la sorte :

fr	**Répertoire de langue** (seulement le français pour l'instant)
billets post_2_col post_img_left post_img_right post_simple config.xml index.html	**Répertoire des types de billets** Chaque répertoire (ici : *post_2_col*, *post_img_left*, *post_img_right* et *post_simple*) correspond à un type de billet et contient ses propres fichiers
images	**Répertoire des images** (hors Médiathèque)

includes	**Répertoire des fichiers includes**
modules 404 chemin fermerture	**Répertoire des modules** Chaque répertoire correspond à un module et contient ses propres fichiers.
produits simple config.xml index.html styles.css	**Répertoire des types de produits** Chaque répertoire (ici : *simple*) correspond à un type de produit et contient ses propres fichiers.
templates 2_colonnes_droite 2_colonnes_gauche desc.xml illustration_2col.gif index.html layout.css structure.xhtml styles.css print.css styles.css styles_70.css styles_90.css	**Répertoire des templates principaux** Ce répertoire contient tous les templates principaux du thème. Chaque répertoire (ici : *2_colonnes_droite* et *2_colonnes_gauche*) correspond à un template et contient ses propres fichiers.
widgets	**Répertoire de tous les widgets**
blog archives categories derniers_billets_postes detail_billet les_plus_commentes liens liste_billets liste_commentaires poster_commentaire	**Répertoire des widgets du service Blog** Chaque widget à son propre répertoire principal contenant ses propres fichiers.
catalogue categories liste_categories liste_commentaires liste_produits poster_commentaire produits_vedettes tags	**Répertoire des widgets du service Catalogue** Chaque widget à son propre répertoire principal contenant ses propres fichiers.
commandes livraison paiement panier panier_detail validation	**Répertoire des widgets du service Commandes** Chaque widget à son propre répertoire principal contenant ses propres fichiers.

communication contact newsletter syndication_blog	**Répertoire des widgets du service Communication** Chaque widget à son propre répertoire principal contenant ses propres fichiers.
compte identification identification_rapide inscription tableau_bord	**Répertoire des widgets du service Compte** Chaque widget à son propre répertoire principal contenant ses propres fichiers.
recherche simple	**Répertoire des widgets du service Recherche** Chaque widget à son propre répertoire principal contenant ses propres fichiers.
site_web bloc_extrait avec_lien_vers_page sans_image index.html styles.css contenu_page_libre menu_h menu_v	**Répertoire des widgets du service Site web** Chaque widget à son propre répertoire principal contenant ses propres fichiers. Chaque widget peut avoir plusieurs templates secondaires différents (ici : *avec_lien_vers_page* et *sans_image)* contenant leurs propres fichiers correspondant chacun à un sous-répertoire du répertoire principal du widget.
illustration.jpg theme.xml	**Fichier de description et de déclaration du thème** Ces fichiers définissent les informations affichées dans la gestion des thèmes de la console d'administration.

2 Pré-requis

2.1 Les balises

2.1.1 Définition

Les balises de Kiubi sont de petits codes que vous pouvez inclure un peu partout et qui servent à afficher toutes les informations que vous avez saisies à l'aide de votre console d'administration dans les pages de votre site. On reconnaît facilement ces balises car elles sont composées d'un mot entouré d'accolades. Exemple : `{titre}`

Kiubi utilise des ensembles de fichiers HTML (des modèles graphiques) pour mettre en forme votre site. Ils composent le squelette du site. Les balises servent à donner vie au squelette en indiquant à Kiubi où injecter exactement le titre de la page, le contenu des billets, la date de création du dernier billet de votre blog, … Elles sont pour ainsi dire omniprésentes. Tous les modèles graphiques ont besoin de balises pour fonctionner et vous pouvez également saisir certaines balises dans les éditeurs de texte visuels (éditeurs « Wysiwyg »).

Il existe trois grandes familles de balises :

- les balises globales sont utilisables partout (modèles graphiques, éditeurs de texte visuels, champs d'information, …),
- les balises des modèles de mise en pages qui ne fonctionnent que dans les templates principaux,
- les balises des widgets qui ne fonctionnent que dans leur widget.

2.1.2 Exemple d'utilisation d'une balise

Prenons un exemple de modèle de mise en page tel qu'on pourrait en trouver dans les sous-répertoires de « */fr/template/* ».

```
<head>
<meta name="keywords" content="{metaKeywords}" />
<meta name="description" content="{metaDescription}" />
<title>{title}</title>
</head>
```

On voit ici que le titre de la page, la description et les mots clés de l'entête du fichier HTML sont définis à l'aide de balises et seront remplis automatiquement avec les valeurs configurées dans la console d'administration.

Avertissement : une balise utilisée dans un widget, n'est pas forcément utilisable dans un autre ou dans tous les widgets. A chaque widget ses balises !

2.2 Les balises globales

Les balises globales sont disponibles et utilisables partout dans Kiubi (modèles graphiques, éditeurs de texte visuels, champs d'information, …).

2.2.1 Liste des balises globales

`{racine}`

Indique le chemin relatif vers la racine du site où que l'on soit situé dans l'arborescence.

`{lg}`

Indique le code langue courant (toujours `fr` pour l'instant).

`{baseLangue}`

Indique le chemin relatif vers la racine du site dans la langue courante.

`{lien_pagecourante}`

URL de la page courante.

`{intitule_pagecourante}`

Intitulé de la page courante.

`{lien_pageparente}`

URL de la page parente (niveau hiérarchique supérieur dans l'arborescence).

`{intitule_pageparente}`

Indique l'intitulé de la page parente.

`{SOCIETE.contact_nom}`

Affiche le nom du contact principal du site internet figurant dans "*Mon compte / Préférences*".

`{SOCIETE.contact_prenom}`

Affiche le prénom du contact principal du site internet figurant dans "*Mon compte / Préférences*".

`{SOCIETE.contact_email}`

Affiche l'e-mail du contact principal du site internet figurant dans "*Mon compte / Préférences*".

`{SOCIETE.societe_nom}`

Affiche le nom de la société responsable du site internet figurant dans "*Mon compte / Préférences*".

`{SOCIETE.societe_adresse}`

Affiche l'adresse de la société responsable du site internet figurant dans "*Mon compte / Préférences*".

`{SOCIETE.societe_cp}`

Affiche le code postal de la société responsable du site internet figurant dans "*Mon compte / Préférences*".

`{SOCIETE.societe_ville}`

Affiche la ville de la société responsable du site internet figurant dans "*Mon compte / Préférences*".

`{SOCIETE.societe_pays}`

Affiche le pays de la société responsable du site internet figurant dans "*Mon compte / Préférences*".

`{SOCIETE.societe_tel}`

Affiche le numéro de téléphone de la société responsable du site internet figurant dans "*Mon compte / Préférences*".

`{SOCIETE.societe_mobile}`

Affiche le numéro de mobile de la société responsable du site internet figurant dans "*Mon compte / Préférences*".

`{SOCIETE.societe_fax}`

Affiche le numéro de fax de la société responsable du site internet figurant dans "*Mon compte / Préférences*".

`{SOCIETE.societe_cnil}`

Affiche le numéro de déclaration CNIL du site internet figurant dans "*Mon compte / Préférences*".

`{SOCIETE.societe_capital}`

Affiche le capital de la société responsable du site internet figurant dans "*Mon compte / Préférences*".

`{SOCIETE.societe_siret}`

Affiche le numéro siret de la société responsable du site internet figurant dans "*Mon compte / Préférences*".

`{SOCIETE.societe_rcs}`

Affiche le RCS de la société responsable du site internet figurant dans "*Mon compte / Préférences*".

`{SOCIETE.societe_ape}`

Affiche le code APE (ou NAF) de la société responsable du site internet figurant dans "*Mon compte / Préférences*".

`{SOCIETE.societe_tva}`

Affiche le numéro de TVA intracommunautaire de la société responsable du site internet figurant dans "*Mon compte / Préférences*".

`{SOCIETE.societe_forme}`

Affiche la forme juridique de la société responsable du site internet figurant dans "*Mon compte / Préférences*".

`{accroche_site}`

Affiche l'accroche du site figurant dans "*Mon site / Apparence / Bloc-marque*".

`{desc_site}`

Affiche la description du site figurant dans "*Mon site / Apparence / Bloc-marque*".

`{logo_site}`

Affiche le logo du site figurant dans "*Mon site / Apparence / Bloc-marque*".

`{domaine}`

Affiche le nom du domaine courant du site.

`{liste_domaines}`

Liste des noms de domaines du site.

`{site_nom}`

Affiche le nom du site figurant dans "*Mon compte / Préférences*".

`{num_jour_maintenant}`

Affiche le numéro du jour courant (01 - 31).

`{jour_semaine_maintenant}`

Affiche le jour de la semaine courant (lundi - dimanche).

`{num_mois_maintenant}`

Affiche le numéro du mois courant (01 - 12).

`{mois_maintenant}`

Affiche le mois courant (janvier - décembre).

`{mois_abrev_maintenant}`

Affiche le mois courant abrégé (jan - déc). Il vaut mieux utiliser `{mois_abrev_maintenant}` que `{mois_maintenant|left|3}` car la découpe se fait plus intelligemment pour les mois en quatre lettres.

`{annee_maintenant}`

Affiche l'année courante (2007).

`{heure_maintenant}`

Affiche l'heure courante (00-23).

`{minute_maintenant}`

Affiche les minutes courantes (00-59).

`{seconde_maintenant}`

Affiche les secondes courantes (00-59).

`{timestamp_maintenant}`

Affiche le timestamp courant

`{FILE "fichier.html"}`

Cette balise est très spéciale car elle permet d'inclure un modèle graphique dans le modèle graphique courant. Il suffit pour cela d'indiquer le chemin du modèle graphique que vous voulez inclure. Ce fichier doit se trouver dans le thème courant. Cette balise est souvent utilisée pour inclure à différents endroits des bouts de page qui reviennent de manière récurrente.
Par exemple, dans les thèmes Kiubi, le copyright en bas de page est inclus dans toutes les pages grâce à la balise : `{FILE "/fr/includes/copyright.html"}`

`{g_vignette_l}`

Largeur en pixels des grandes vignettes de la médiathèque.

`{vignette_l}`

Largeur en pixels des vignettes de la médiathèque.

`{g_vignette_h}`

Hauteur en pixels des grandes vignettes de la médiathèque.

`{vignette_h}`

Hauteur en pixels des vignettes de la médiathèque.

`{g_miniature_l}`

Largeur en pixels des grandes miniatures de la médiathèque.

`{miniature_l}`

Largeur en pixels des miniatures de la médiathèque.

`{g_miniature_h}`

Hauteur en pixels des grandes miniatures de la médiathèque.

`{miniature_h}`

Hauteur en pixels des miniatures de la médiathèque.

2.3 Les filtres

Kiubi supporte un ensemble de filtres qui peuvent être appliqués aux balises. Ces filtres ont pour but de changer la valeur de la balise pour mieux répondre à un besoin précis.

On utilise ces filtres en ajoutant une barre verticale `|` dans la balise. Exemple : `{balise|filtre}`.
Certains filtres peuvent prendre des arguments, ils seront alors séparés par une autre barre. Exemple : `{balise|filtre|argument}`.

Les filtres disponibles sont :

- *ucfirst :* met le premier caractère en majuscule
- *strtoupper :* met tout en majuscule
- *strtolower :* met tout en minuscule
- *rawurlencode :* encode le contenu de la balise pour qu'il puisse être transmis par url
- *htmlentities :* encode en html les caractères spéciaux
- *left|x :* affiche les x premiers caractères à partir de la gauche
- *stripText|x :* coupe le texte à x caractères, sans couper un mot
- *mot|x :* affiche le xième mot

Ainsi `{intitule_pagecourante|strtoupper}` affichera le nom de la page courante en majuscule. Tandis que `{SOCIETE.societe_cp|left|2}` affichera le numéro de département extrait du code postal du contact principal du site.

2.4 Les blocs

Les blocs sont des portions de HTML encadrés par des codes spécifiques qui vont pouvoir être affichés ou masqués selon le besoin, et même dupliqués. Ils se présentent toujours de cette façon :

```
<!-- BEGIN:bloc --> <!-- END:bloc -->
```

Les blocs sont pour la plupart du temps imbriqués dans d'autres blocs. La documentation ci-après précise toujours dans quel bloc placer tel autre bloc (ou telle balise). Si rien n'est précisé c'est qu'il faut au moins placer le bloc dans le bloc `<!-- BEGIN:main --> <!-- END:main -->`.

Voici un exemple de blocs correctement imbriqués :

```
<!-- BEGIN:main -->
     <!-- BEGIN:bloc -->
         <!-- BEGIN:bloc2 -->
         <!-- END:bloc2 -->

         <!-- BEGIN:bloc3 -->
         <!-- END:bloc3 -->
     <!-- END:bloc -->
 <!-- END:main -->
```

L'exemple suivant est faux et amènera à des comportements imprévisibles. **Les blocs "bloc2" et "bloc3" sont mélangés l'un dans l'autre, ce qui est incorrect :**

```
<!-- BEGIN:main -->
         <!-- BEGIN:bloc2 -->
             <!-- BEGIN:bloc3 -->
         <!-- END:bloc2 -->
             <!-- END:bloc3 --> // cette fermeture de bloc ne devrait pas se trouver ici !!!
 <!-- END:main -->
```

Certaines balises ne sont accessibles qu'à l'intérieur de blocs spécifiques. Si c'est le cas, cette documentation le précisera.

Avertissement : un même bloc ne peut pas être répété deux fois dans le même fichier !

Avertissement : un bloc qui est utilisé dans un widget, n'est pas forcément utilisable dans un autre ou dans tous les widgets. A chaque widget ses blocs !

Avertissement : tous les modèles graphiques doivent commencer par `<!-- BEGIN:main -->` et se terminer par `<!-- END:main -->`. Oublier ce bloc est une erreur très courante. Si vous créez ou modifiez un modèle graphique et que rien ne s'affiche dans votre site, pensez à vérifier ce bloc. De même, tout ce qui est en dehors du bloc `<!-- BEGIN:main -->` `<!-- END:main -->` est ignoré par Kiubi !

3 Les types de billets

3.1 Définition

Un billet permet d'ajouter du contenu à une page libre du Site web. Chaque billet peut être considéré comme un paragraphe, un bloc de contenu, pouvant contenir du texte, des images, des liens, des vidéos, du code HTML, ...
Lors de l'ouverture d'un site, Kiubi vous propose par défaut 4 types de billets différents dans le service Site web :

- Titres et texte.
- Titres, texte et image à gauche.

- Titres, texte et image à droite.
- Titre et texte sur 2 colonnes.

Chaque type de billet peut être structuré de manière différente et permet d'intégrer et de présenter du contenu suivant ses besoins.

Si vous utilisez un thème graphique personnalisé, vous pouvez modifier les types de billets par défaut et créer autant de types de billet que souhaité.

3.2 Principe

Le principe de fonctionnement d'un type billet du Site web est très simple. Tous les types de billets de votre site sont regroupés dans le répertoire « *theme/fr/billets/* » de votre espace FTP. Chaque sous-répertoire correspond à un type de billet.

Un billet est composé, en plus d'une structure HTML de votre choix, d'un maximum de 17 champs d'information :

- titre
- sstitre
- texte1
- …
- texte15

Chaque champ peut contenir n'importe quel type de donnée. Par convention (mais ce n'est pas obligatoire), les champs texte10 à texte15 sont utilisés pour stocker des éléments de la médiathèque (image, fichier, …), plus précisément l'identifiant faisant référence à l'élément de la médiathèque.

Ces champs sont à définir dans le fichier « *config.xml* » de chaque type de billet, ce qui permet de générer automatiquement le formulaire de saisie utilisé dans la console d'administration. **Ce fichier est obligatoire.**

Chaque champ pourrait être par la suite affiché dans le fichier « *index.html* » de chaque type de billet ou dans chaque widget affichant des billets, en fonction de ses besoins. **Ce fichier est obligatoire.**

Comme tous les éléments de Kiubi, un billet (ou plutôt une liste de billet) est affiché grâce à un widget. C'est principalement le widget "*Contenu de la page libre*" qui est utilisé pour afficher les billets dans votre site internet.

3.3 Le fichier « config.xml »

Ce fichier contient des informations descriptives du type de billet qui seront utilisée à la fois dans la console d'administration et le site internet. Il indique à Kiubi de quels champs est composé le type de billet.

3.3.1 Exemple de code source

```
<?xml version="1.0" encoding="iso-8859-1"?>
<type tri="2">
    <desc>Titres, texte et image (à gauche)</desc>
    <listechamps>
        <champ champ="titre" type="texte" intitule="Titre"/>
        <champ champ="sstitre" type="text" intitule="Sous-titre" />
```

```
        <champ champ="texte10" type="image" intitule="Image" />
        <champ champ="texte1" type="wysiwyg" intitule="Contenu" />
    </listechamps>
</type>
```

3.3.2 Description

Ce fichier contient une seule balise principale `<type>` qui possède un seul attribut :

- *tri :* nombre qui sert à définir l'ordre dans le quel vont apparaître les types de billet dans les listes déroulantes de la console d'administration.

A l'intérieur, une seule balise `<desc>` qui contient la description du type de billet qui sera repris dans la console d'administration.

La balise `<listechamps>` sert à déclarer tous les champs qui seront utilisés dans ce type de billet.
Il existe alors une balise `<champ>` par champ qui contient les attributs suivants :

- *champ :* code du champ qui sera aussi utilisé dans les templates : titre, sstitre, texte1 à texte15.
- *type :* le type du champ indique quelle sorte d'information est stockée dans le champ.
  Il existe quatre types de champ :
  - *texte :* champ de saisie simple
  - *textarea :* champ de saisie multiligne
  - *liste :* champ de liste déroulante
  - *wysiwyg :* éditeur de texte visuel
  - *image :* fichier image de la médiathèque
  - *fichier :* fichier quelconque de la médiathèque
- *intitule :* intitulé du champ qui va être repris dans la console d'administration dans le formulaire de saisie d'un billet.

3.3.3 Le champ "liste"

Le champ de type "liste" se comporte comme une liste déroulante classique. Il est construit de la manière suivante :

```
<champ champ="texte2" type="select" intitule="Liste">
        <option value="Valeur1">Label1</option>
        <option value="Valeur2">Label2</option>
        <option value="Valeur3">Label3</option>
</champ>
```

Le contenu de l'attribut `value` est stocké dans le champ `texte2`. Le label n'est affiché que dans le détail du billet dans la console d'administration.

3.4 Le fichier « index.html »

Le fichier « *index.html* » contient le code HTML du type de billet.

3.4.1 Exemple de code source

```
<!-- BEGIN: main -->
    <!-- BEGIN: titre -->
        <h2>{titre}</h2>
    <!-- END: titre -->
    <!-- BEGIN: sstitre -->
        <h3>{sstitre}</h3>
    <!-- END: sstitre -->
    <!-- BEGIN:texte10 -->
        <img src="/media/vignette/{texte10}" alt="{titre}"/>
    <!-- END:texte10 -->
    <!-- BEGIN: texte1 -->
        <p>{texte1}</p>
    <!-- END: texte1 -->
<!-- END: main -->
```

3.4.2 Description

Les balises `{titre}`, `{sstitre}`, … servent à indiquer où afficher le contenu des champs du billet.

3.4.3 Balises disponibles

`{auteur}`

Nom de l'auteur du billet.

`{auteur_modif}`

Nom de l'utilisateur qui a modifié le billet en dernier.

`{billet_id}`

Id du billet.

`{groupe_billet}`

Groupe du billet.

`{intitule_page}`

Intitulé de la page libre dans laquelle se trouve le billet.

`{lien_billet}`

URL un peu spéciale vers la page libre du billet. Les widgets "*Contenu de la page libre*" de la page libre du billet qui peuvent afficher ce billet n'afficheront que ce billet là, et tous les autres widgets fonctionneront normalement. C'est très utile pour faire des pages de listing et de détail.

`{lien_page}`

URL vers la page libre dans laquelle se trouve le billet.

`{lg}`

Langue du billet.
Affiche toujours `fr` pour l'instant.

`{nom_optim}`

Nom optimisé de la page libre du billet (c'est à dire la dernière partie de l'URL de la page), tel qu'il est défini dans les options avancées de la page d'édition de la page libre.

`{sstitre}`

Sous-titre du billet.

`{texte1}` à `{texte15}`
Affiche les champs texte1 à texte15.

`{titre}`

Titre du billet.

`{type_billet}`

Type du billet : affiche le nom du répertoire du billet sur l'espace FTP.

`<!-- BEGIN:titre--> <!-- END:titre-->`

Le contenu du bloc ne s'affiche que si le titre du billet n'est pas vide.

`<!-- BEGIN:sstitre--> <!-- END:sstitre-->`

Le contenu du bloc ne s'affiche que si le sous-titre du billet n'est pas vide.

`<!-- BEGIN:texte1--> <!-- END:texte1-->` à `<!-- BEGIN:texte15--> <!-- END:texte15-->`
Les contenus des blocs ne s'affichent que si les contenus des champs textes correspondants ne sont pas vides.

3.5 Ajouter un nouveau type de billet

Pour ajouter un type de billet, il suffit de créer un nouveau sous-répertoire au répertoire *theme/fr/billets/* et d'y copier les fichiers « *config.xml* » et « *index.html* » correspondant à votre nouveau type de billet. Le nouveau type de billet sera alors automatiquement proposé dans le service Site web de la console d'administration.

4 Les types des produits

4.1 Définition

Un type de produit fonctionne comme un type de billet mais est adapté à votre catalogue produit. Lors de l'ouverture d'un site, Kiubi vous propose par défaut un seul type de produit dans le service Catalogue :

- Produit simple.

Chaque type de produit peut être structuré de manière différente et permet d'intégrer et de présenter du contenu suivant ses besoins. Contrairement aux différents types de billet nécessaires au Site web, il est plus rarement utile de multiplier les types de produit, un seul type (proposé par défaut) étant bien souvent suffisant.

Si vous utilisez un thème graphique personnalisé, vous pouvez modifier le type de produit par défaut et créer des nouveaux types de produit.

4.2 Principe

Le principe de fonctionnement d'un type de produit du Catalogue est très simple. Tous les types de produits de votre site sont regroupés dans le répertoire « *theme/fr/produits/* » de votre espace FTP. Chaque sous-répertoire correspond à un type de produit.

Un produit peut être composé, en plus d'une structure HTML de votre choix et des informations spécifiques à un produit (intitulé, chapô, description, variantes, illustration, …), d'un maximum de 15 champs d'information supplémentaires (qui ne sont pas utilisés dans le type de produit par défaut proposé par Kiubi) :

- texte1
- …
- texte15

Chaque champ peut contenir n'importe quel type de donnée. Par convention (mais ce n'est pas obligatoire), les champs texte10 à texte15 sont utilisés pour stocker des éléments de la médiathèque (image, fichier, …), plus précisément l'identifiant faisant référence à l'élément de la médiathèque.

Ces champs sont à définir, au besoin, dans le fichier « *config.xml* » de chaque type de produit, ce qui permet de générer et d'ajouter automatiquement le formulaire de saisie correspondant lors de l'édition d'un produit dans la console d'administration. **Ce fichier est obligatoire.**

Chaque champ pourrait être par la suite affiché dans le fichier « *index.html* » de chaque type produit ou dans chaque widget affichant des produits, en fonction de ses besoins. **Ce fichier est obligatoire.**

Un fichier *styles.css* peut être ajouté afin de contrôler la mise en forme du widget « *Détail d'un produit* ». Ce fichier est automatiquement chargé dans la balise `<head>` si le widget est présent dans la page. **Ce fichier n'est pas obligatoire.**

Comme tous les éléments de Kiubi, un produit (ou une liste de produit) est affiché grâce à un widget. Ce sont principalement certains widgets du service « *Catalogue* » qui sont utilisés pour afficher des produits.

Avertissement : les fichiers « *index.html* » et « *styles.css* » sont exclusivement utilisés par le widget « *Détail d'un produit* ».

4.3 Le fichier « config.xml »

Ce fichier contient des informations descriptives du type de produit qui seront utilisée à la fois dans la console d'administration et le site internet. Il indique à Kiubi de quels champs d'information supplémentaires est composé le type de produit. **Ces champs d'information supplémentaires sont facultatifs.**

4.3.1 Exemple de code source

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE type SYSTEM "http://www.kiubi-admin.com/DTD/typesproduits.dtd">
<type tri="1">
	<desc>Produit simple</desc>
	<listechamps>
		<champ champ="texte1" type="wysiwyg" intitule="Fiche technique" />
	</listechamps>
</type>
```

4.3.2 Description

Ce fichier contient une seule balise principale `<type>` qui possède un seul attribut :

- *tri :* nombre qui sert à définir l'ordre dans le quel vont apparaître les types de produit dans les listes déroulantes de la console d'administration.

A l'intérieur, une seule balise `<desc>` qui contient la description du type de produit qui sera repris dans la console d'administration.

La balise `<listechamps>` sert à déclarer tous les champs d'information supplémentaires qui seront utilisés dans ce type de produit.
Il existe alors une balise `<champ>` par champ qui contient les attributs suivants :

- *champ :* code du champ qui sera aussi utilisé dans les templates ; texte1 à texte15.
- *type :* le type du champ indique quelle sorte d'information est stockée dans le champ.
  Il existe quatre types de champ :
  - *texte :* champ de saisie simple
  - *textarea :* champ de saisie multiligne
  - *liste :* champ de liste déroulante
  - *wysiwyg :* éditeur de texte visuel
  - *image :* fichier image de la médiathèque
  - *fichier :* fichier quelconque de la médiathèque
- *intitule :* intitulé du champ qui va être repris dans la console d'administration dans le formulaire de saisie d'un billet.

4.3.3 Le champ "liste"

Le champ de type "liste" se comporte comme une liste déroulante classique. Il est construit de la manière suivante :

```
<champ champ="texte2" type="select" intitule="Liste">
	<option value="Valeur1">Label1</option>
	<option value="Valeur2">Label2</option>
	<option value="Valeur3">Label3</option>
</champ>
```

Le contenu de l'attribut `value` est stocké dans le champ `texte2`. Le label n'est affiché que dans le détail du produit dans la console d'administration.

4.4 Le fichier « index.html »

Le fichier « *index.html* » contient le code HTML du type de produit, **c'est-à-dire du détail du produit affiché grâce au widget « *Détail d'un produit* ».**

4.4.1 Exemple de code source

```
<div class="produit_detail">

  <!-- BEGIN: illustrations -->

  <div class="produit_images" style="width: {g_vignette_l}px;"><a
href="{racine}/media/{illustration_principale}" rel="lightbox[zoom{produit_id}]"
title="{intitule_produit}" class="illustration"><img
src="{racine}/media/g_vignette/{illustration_principale}"
alt="{intitule_produit|htmlentities}" /></a>

    <div class="liste_images">

      <!-- BEGIN: illustration -->

      <a href="{racine}/media/{illustration}" rel="lightbox[zoom{produit_id}]"
title="{intitule_produit}"><img src="{racine}/media/miniature/{illustration}"
alt="{intitule_produit|htmlentities}" /></a>

      <!-- END: illustration -->

    </div>

  </div>

  <!-- END: illustrations -->

  <!-- BEGIN: noillustrations -->

  <div class="produit_images" style="width: {g_vignette_l}px;"><img
src="{racine}/{theme}/fr/images/produit.gif" alt="{intitule_produit|htmlentities}"
style="width: {g_vignette_l}px; height: {g_vignette_h}px;" /></div>

  <!-- END: noillustrations -->

  <div class="post_produit" style="margin-right: {g_vignette_l}px;">

    <h2>{intitule_produit}</h2>

    <!-- BEGIN:tags -->

    <p class="tags">Tags : {tags}</p>

    <!-- END:tags -->

    <!-- BEGIN:non_disponible -->

    <p class="alerte">Disponible à partir du
{num_jour_disponibilite}/{num_mois_disponibilite}/{annee_disponibilite}</p>

    <!-- END:non_disponible -->

    <!-- BEGIN:rupture_partielle -->

    <p class="alerte">Toutes les variantes ne sont plus disponibles</p>

    <!-- END:rupture_partielle -->

    <!-- BEGIN:rupture -->

    <p class="alerte">Produit en rupture de stock</p>

    <!-- END:rupture -->

    <p class="prix"><span class="<!-- BEGIN:prix_promo -->promo<!-- END:prix_promo --
>">{prix}</span>

      <!-- BEGIN:prix_barre -->

      <s>{prix_barre}</s>

      <!-- END:prix_barre -->
```

```
    </p>
    <div class="produit_infos">
      <form id="fp{serial}_{produit_id}" method="post"
action="{baseLangue}/ecommerce/panier.html">
        <!-- BEGIN:variantes -->
        <ul class="variantes">
          <!-- BEGIN:variante -->
          <li>
            <!-- BEGIN:en_stock -->
            <input name="vid" type="radio" value="{variante_id}"
id="variante{produit_id}_{variante_id}" {checked}/>
            <label for="variante{produit_id}_{variante_id}">
            {intitule_variante} - {prix_variante}
            <!-- BEGIN:prix_barre -->
             <s>{prix_variante_barre}</s>
            <!-- END:prix_barre -->
            <!-- END:en_stock -->
            <!-- BEGIN:rupture -->
            <input name="vid" type="radio" value="{variante_id}"
id="variante{produit_id}_{variante_id}" disabled="disabled"/>
            <label for="variante{produit_id}_{variante_id}"
class="rupture">{intitule_variante} - {prix_variante}
            <!-- BEGIN:prix_barre -->
             <s>{prix_variante_barre}</s>
            <!-- END:prix_barre -->
            <!-- END:rupture -->
            <!-- BEGIN: eco_taxe -->
            <span class="eco_taxe">dont {eco_taxe} d'&eacute;co-part</span>
            <!-- END: eco_taxe -->
            </label>
          </li>
          <!-- END:variante -->
        </ul>
        <!-- END:variantes -->
        <!-- BEGIN:panier-->
        <!-- <a href="javascript:void(0);" onclick="Ssubmit('fp{serial}_{produit_id}')"
class="bt_panier" title="Ajouter au panier">Ajouter au panier</a>-->
        <input type="submit" class="bt_panier" value="Ajouter au panier" />
        <!-- END:panier-->
        <input type="hidden" name="pid" value="{produit_id}" />
        <input type="hidden" name="act" value="addP" />
        <input type="hidden" name="ctl" value="{ctl}" />
      </form>
    </div>
  </div>
```

```
  <!-- BEGIN: description -->
  <p class="desc">{description}</p>
  <!-- END: description -->
  <div class="post_footer"></div>
</div>
```

4.4.2 Balises disponibles

`{type_prix}`

Type du prix ; `HT` ou `TTC`.

`{produit_id}`

Id du produit.

`{type_produit}`

Type du produit
Affiche le nom du répertoire du produit sur l'espace FTP.

`{en_avant}`

Affiche `1` si le produit est un produit vedette, sinon affiche `0`.

`{nb_commentaires}`

Nombre de commentaires.

`{note_moyenne}`

Note moyenne ; `0` à `10`.

`{num_jour_disponibilite}`

Affiche le numéro du jour courant (01 - 31).

`{jour_semaine_disponibilite}`

Affiche le jour de la semaine courant (lundi - dimanche).

`{num_mois_disponibilite}`

Affiche le numéro du mois courant (01 - 12).

`{mois_disponibilite}`

Affiche le mois courant (janvier - décembre).

`{mois_abrev_disponibilite}`

Affiche le mois courant abrégé (jan - déc). Il vaut mieux utiliser `{mois_abrev_maintenant}` que `{mois_maintenant|left|3}` car la découpe se fait plus intelligemment pour les mois en quatre lettres.

`{annee_disponibilite}`

Affiche l'année courante (2007).

`{heure_disponibilite}`

Affiche l'heure courante (00-23).

`{minute_disponibilite}`

Affiche les minutes courantes (00-59).

`{seconde_disponibilite}`

Affiche les secondes courantes (00-59).

`{timestamp_disponibilite}`

Affiche le timestamp de la date de disponibilité.

`{date_crea}`

Date de création du produit au format anglo-saxon.

`{date_modif}`

Date de modification du produit au format anglo-saxon.

`{auteur}`

Auteur du produit.

`{intitule_produit}`

Intitulé du produit.

`{categorie_id}`

Id de la catégorie en cours du produit.

`{intitule_categorie}`

Intitulé de la catégorie en cours du produit.

`{categ_optim}`

Nom optimisé de la catégorie en cours du produit, tel qu'il est défini dans les options avancées de la page d'édition de la catégorie. Représente le « dossier virtuel » utilisé dans la réécriture d'URL pour arriver sur la page de la catégorie.

`{illustration}`

Numéro de fichier dans la médiathèque de l'illustration principale du produit.

`{lien_produit}`

Lien vers la page de détail du produit.

`{accroche}`

Accroche du produit. Affiche le chapô du produit ou son contenu si son chapô est vide.

`{chapo}`

Chapô du produit.

`{description}`

Description du produit.

`{tags}`

Affiche la liste des tags du produit. Le lien renvoi vers la catégorie en cours du produit restreint au tag cliqué.

`{tags_recherche}`

Affiche la liste des tags du produit. Le lien renvoi vers le moteur de recherche restreint au tag cliqué.

`{prix_min}`

Affiche le prix minimum toutes variantes confondues.

`{prix_max}`

Affiche le prix maximum toutes variantes confondues.

`{prix_barre}`

Affiche le prix normal si toutes les variantes ont un prix promo renseigné identique, sinon n'affiche rien.

`{prix}`

Comportement standard du prix.

- *Si une variante à un prix :* affiche le prix
- *Si une variante à un prix promo :* affiche le prix promo
- *Si plusieurs variantes ont un prix identique :* affiche le prix
- *Si plusieurs variantes ont un prix promo identique :* affiche le prix promo
- *Si plusieurs variantes ont un prix (promo) différent :* affiche « prix minium (promo comprise) - prix maximum (promo comprise) »

`{prix_HT_min}`

Affiche le prix minimum (promo comprise) HT toutes variantes confondues.

`{prix_TTC_min}`

Affiche le prix minimum (promo comprise) TTC toutes variantes confondues.

`{prix_HT_max}`

Affiche le prix maximum (promo comprise) HT toutes variantes confondues.

`{prix_TTC_max}`

Affiche le prix maximum (promo comprise) TTC toutes variantes confondues.

`{prix_HT}`

Comportement standard du prix, mais toujours en HT.

`{prix_TTC}`

Comportement standard du prix, mais toujours en TTC.

`{texte1}` à `{texte15}`

Affiche les champs texte1 à texte15.

`{nb_achats}`

Quantité de produits vendus, toutes variantes confondues.

`{nb_vus}`

Nombre de consultations de la page de détail du produit.

`<!-- BEGIN: illustrations -->  <!-- END: illustrations -->`

Le contenu du bloc s'affiche si le produit a des illustrations.

`<!-- BEGIN: noillustrations -->  <!-- END: noillustrations -->`

Le contenu du bloc s'affiche si le produit n'a pas d'illustration.

`<!-- BEGIN: illustration -->  <!-- END: illustration -->`

Le contenu du bloc s'affiche pour chaque illustration.
Doit être contenu entre `<!-- BEGIN: illustrations -->` et `<!-- END: illustrations -->`

`{illustration}`

Numéro de fichier dans la médiathèque.
Doit être contenu entre `<!-- BEGIN: illustration -->` et `<!-- END: illustration -->`

`{favoris}`

Affiche `1` si l'illustration est définie en tant qu'illustration principale, sinon affiche `0`.
Doit être contenu entre `<!-- BEGIN: illustration -->` et `<!-- END: illustration -->`

`{intitule_fichier}`

Intitulé du fichier dans la médiathèque.
Doit être contenu entre `<!-- BEGIN: illustration -->` et `<!-- END: illustration -->`

`{ext}`

Extention du fichier dans la médiathèque.

Doit être contenu entre `<!-- BEGIN: illustration -->` et `<!-- END: illustration -->`

`{nom_origine}`

Nom d'origine du fichier à l'upload dans la médiathèque.
Doit être contenu entre `<!-- BEGIN: illustration -->` et `<!-- END: illustration -->`

`<!-- BEGIN: vedette -->  <!-- END: vedette -->`

Le contenu du bloc s'affiche si le produit est un produit vedette.

`<!-- BEGIN: accroche -->  <!-- END: accroche -->`

Le contenu du bloc s'affiche si l'accroche du produit a une valeur.

`<!-- BEGIN: chapo -->  <!-- END: chapo -->`

Le contenu du bloc s'affiche si le chapô du produit a une valeur.

`<!-- BEGIN: description -->  <!-- END: description -->`

Le contenu du bloc s'affiche si la description du produit a une valeur.

`<!-- BEGIN: tags -->  <!-- END: tags -->`

Le contenu du bloc s'affiche si le produit a un ou plusieurs tags.

`<!-- BEGIN: prix_promo -->  <!-- END: prix_promo -->`

Le contenu du bloc s'affiche si le produit a au moins une variante en promo.

`<!-- BEGIN: prix_barre -->  <!-- END: prix_barre -->`

Le contenu du bloc s'affiche si le produit a toutes ces variantes avec le même prix et le même prix promo.

`<!-- BEGIN:texte1--> <!-- END:texte1-->` à `<!-- BEGIN:texte15--> <!-- END:texte15-->`
Les contenus des blocs ne s'affichent que si les contenus des champs textes correspondants ne sont pas vides.

`<!-- BEGIN: non_disponible -->  <!-- END: non_disponible -->`

Le contenu du bloc s'affiche si le produit n'est pas encore disponible (date de disponibilité suppérieure à la date courante) .

`<!-- BEGIN: rupture -->  <!-- END: rupture -->`

Le contenu du bloc s'affiche si toutes les variantes du produit sont en rupture de stock.

`<!-- BEGIN: rupture_partielle -->  <!-- END: rupture_partielle -->`

Le contenu du bloc s'affiche si une variante du produit est en rupture de stock.

`<!-- BEGIN: achetable -->  <!-- END: achetable -->`

Le contenu du bloc s'affiche si le produit peut être ajouté au panier (en raison du stock, de la date de disponibilité, de la présence des fonctionnalités e-commerce).

`<!-- BEGIN: noachetable -->  <!-- END: noachetable -->`

Le contenu du bloc s'affiche si le produit ne peut pas être ajouté au panier (en raison du stock, de la date de disponibilité, de l'absence des fonctionnalités e-commerce).

`<!-- BEGIN: panier -->  <!-- END: panier -->`

Le contenu du bloc s'affiche si le produit peut être ajouté au panier (en raison du stock, de la date de disponibilité, de la présence des fonctionnalités e-commerce).

`<!-- BEGIN: nopanier -->  <!-- END: nopanier -->`

Le contenu du bloc s'affiche si le produit ne peut pas être ajouté au panier (en raison du stock, de la date de disponibilité, de l'absence des fonctionnalités e-commerce).

`<!-- BEGIN: variantes -->  <!-- END: variantes -->`

Le contenu du bloc s'affiche si le produit a au moins une variante.

`<!-- BEGIN: novariantes -->  <!-- END: novariantes -->`

Le contenu du bloc s'affiche si le produit n'a pas de variante.

`<!-- BEGIN: variantes_optim -->  <!-- END: variantes_optim -->`

Le contenu de bloc s'affiche si le produit à plus d'une variante.

`<!-- BEGIN: novariantes_optim -->  <!-- END: novariantes_optim -->`

Le contenu du bloc s'affiche si le produit n'a pas de variante.

`<!-- BEGIN: variante -->  <!-- END: variante -->`

Le contenu du bloc s'affiche pour chaque variante du produit.
Doit être contenu entre `<!-- BEGIN: variantes -->` et `<!-- END: variantes -->` ou entre `<!-- BEGIN: variantes_optim -->` et `<!-- END: variantes_optim -->`

`{variante_id}`

Id de la variante.
Doit être contenu entre `<!-- BEGIN: variante -->` et `<!-- END: variante -->`

`{reference}`

Référence de la variante.
Doit être contenu entre `<!-- BEGIN: variante -->` et `<!-- END: variante -->`

`{intitule_variante}`

Intitulé de la variante.
Doit être contenu entre `<!-- BEGIN: variante -->` et `<!-- END: variante -->`

`{prix_variante}`

Comportement standard du prix de la variante.

- Affiche le prix normal si le prix n'a pas de promo.
- Affiche le prix promo si le prix a une promo renseignée.

Doit être contenu entre `<!-- BEGIN: variante -->` et `<!-- END: variante -->`

`{prix_variante_HT}`

Comportement standard du prix de la variante, mais toujours en HT.
Doit être contenu entre `<!-- BEGIN: variante -->` et `<!-- END: variante -->`

`{prix_variante_TTC}`

Comportement standard du prix de la variante, mais toujours en TTC.
Doit être contenu entre `<!-- BEGIN: variante -->` et `<!-- END: variante -->`

`{promo_variante_HT}`

Affiche le prix promo HT si renseigné, sinon n'affiche rien.
Doit être contenu entre `<!-- BEGIN: variante -->` et `<!-- END: variante -->`

`{promo_variante_TTC}`

Affiche le prix promo TTC si renseigné, sinon n'affiche rien.
Doit être contenu entre `<!-- BEGIN: variante -->` et `<!-- END: variante -->`

`{prix_variante_barre}`

Affiche le prix normal si le prix promo est renseigné, sinon n'affiche rien.
Doit être contenu entre `<!-- BEGIN: variante -->` et `<!-- END: variante -->`

`{prix_variante_barre_HT}`

Affiche le prix normal HT si le prix promo est renseigné, sinon n'affiche rien.
Doit être contenu entre `<!-- BEGIN: variante -->` et `<!-- END: variante -->`

`{prix_variante_barre_TTC}`

Affiche le prix normal TTC si le prix promo est renseigné, sinon n'affiche rien.
Doit être contenu entre `<!-- BEGIN: variante -->` et `<!-- END: variante -->`

`{poids}`

Poids de la variante.
Doit être contenu entre `<!-- BEGIN: variante -->` et `<!-- END: variante -->`

`{stock_illimite}`

Affiche `1` si le stock est illimité, sinon affiche `0`.
Doit être contenu entre `<!-- BEGIN: variante -->` et `<!-- END: variante -->`

`{stock}`

Stock de la variante.
Doit être contenu entre `<!-- BEGIN: variante -->` et `<!-- END: variante -->`

`{checked}`

Affiche `checked="checked"` pour la première variante, sinon n'affiche rien. Utilisé pour les boutons radios.
Doit être contenu entre `<!-- BEGIN: variante -->` et `<!-- END: variante -->`

`{selected}`

Affiche `selected="selected"` pour la première variante, sinon n'affiche rien. Utilisé pour les listes déroulantes.
Doit être contenu entre `<!-- BEGIN: variante -->` et `<!-- END: variante -->`

`<!-- BEGIN: rupture -->  <!-- END: rupture -->`

Le contenu du bloc s'affiche si une variante est en rupture de stock.
Doit être contenu entre `<!-- BEGIN: variante -->` et `<!-- END: variante -->`

`<!-- BEGIN: prix_barre -->  <!-- END: prix_barre -->`

Le contenu du bloc s'affiche si une variante est en rupture de stock et si la variante a un prix promo renseigné.
Doit être contenu entre `<!-- BEGIN: en_stock -->` et `<!-- END: en_stock -->` ou entre `<!-- BEGIN: rupture -->` et `<!-- END: rupture -->`

`<!-- BEGIN: en_stock -->  <!-- END: en_stock -->`

Le contenu du bloc s'affiche si une variante est en stock.
Doit être contenu entre `<!-- BEGIN: variante -->` et `<!-- END: variante -->`

`<!-- BEGIN: eco_taxe -->  <!-- END: eco_taxe -->`

Le contenu du bloc s'affiche si une variante à une éco-participation de définie.
Doit être contenu entre `<!-- BEGIN: variante -->` et `<!-- END: variante -->`

`{eco_taxe}`

Montant de l'éco-participation.
Doit être contenu entre `<!-- BEGIN: eco_taxe -->` et `<!-- END: eco_taxe -->`

`{serial}`

Identifiant du widget dans la page. Sert surtout pour du DHTML afin d'identifier de manière unique des éléments HTML dans la page. Exemple : `<div id="element{serial}">`

4.5 Ajouter un nouveau type de produit

Pour ajouter un type de produit, il suffit de créer un nouveau sous-répertoire au répertoire « *theme/fr/produits/* » et d'y copier les fichiers « *config.xml* » et « *index.html* » (et « *styles.css* ») correspondant à votre nouveau type de produit. Le nouveau type de produit sera alors automatiquement proposé dans le service Catalogue de la console d'administration.

5 La Médiathèque

5.1 Définition

La Médiathèque de Kiubi permet de gérer les fichiers (images, documents textes, pdf, sons, vidéos...) que vous allez utiliser pour enrichir le contenu de votre site internet. Vous pouvez publier des fichiers **de moins de** 7Mo directement dans la Médiathèque en passant par la console d'administration (requiert le plugin Adobe Flash Player 9 ou supérieur). Pour des fichiers **de plus de** 7Mo, vous devez les publier dans votre espace FTP, dans le répertoire « *mediatheque* ».

5.2 Accéder à un fichier de la Médiathèque

Tous les fichiers publiés dans la Médiathèque sont stockés dans le répertoire virtuel « *media* » et sont identifiés par un numéro identifiant qui leur est propre. Un fichier de la Médiathèque sera donc accessible à l'adresse :

`http://nom_de_domaine/media/numero_identifiant`

Généralement le numéro identifiant est ajouté automatiquement si le fichier est inséré directement via la Médiathèque dans une zone de mise en forme de texte ou dans un champ d'information de type image ou fichier.

5.3 Le cas particulier des images

Kiubi permet de générer automatiquement 4 tailles différentes en plus de la taille d'origine d'une image publiée dans la Médiathèque. Ces tailles sont configurables dans la console d'administration rubrique « *Modules / Médiathèque* » :

- *grande vignette*, par défaut configurée en 350px par 350px
- *vignette*, par défaut configurée en 200px par 200px
- *grande miniature*, par défaut configurée en 140px par 140px
- *miniature*, par défaut configurée en 80px par 80px

Ces différentes tailles d'images sont accessibles aux adresses :

`http://nom_de_domaine/media/g_vignette/numero_identifiant`

`http://nom_de_domaine/media/vignette/numero_identifiant`

`http://nom_de_domaine/media/g_miniature/numero_identifiant`

`http://nom_de_domaine/media/miniature/numero_identifiant`

La taille d'origine de l'image restant accessible à l'adresse :

`http://nom_de_domaine/media/numero_identifiant`

6 Les includes

6.1 Définition

Un include permet d'inclure le contenu d'un fichier HTML dans n'importe quel autre modèle graphique.

6.2 Principe

Pour utiliser un include, il suffit d'indiquer le chemin du fichier que vous voulez inclure sous la forme :

```
{FILE "/fr/includes/date.html"}
```

Tous les includes de Kiubi sont regroupés dans le répertoire « */theme/fr/includes/* » qui contient par défaut 4 fichiers :

- *chemin.html*
- *copyright.html*
- *date.html*
- *taille_texte.html*

Un include est généralement utilisé pour inclure à différents endroits des bouts de code qui reviennent de manière récurrente.
Par exemple, dans les modèles de mise en pages de Kiubi, le copyright en bas de page est inclus grâce à la balise : `{FILE "/fr/includes/copyright.html"}`

7 Les modules

7.1 Définition

Les modules regroupent 3 éléments particuliers de Kiubi :

- La page d'erreur 404,
- La page de fermeture du site,
- Le chemin de navigation.

Tous les modules sont regroupés dans le répertoire « */theme/fr/modules/* ».

7.2 La page d'erreur 404

La page d'erreur 404 est la page qui est affichée si une adresse erronée de votre site est demandée.

7.2.1 Répertoire

/theme/fr/modules/404

7.2.2 Exemple du code source

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

<meta name="keywords" content="{metaKeywords}">

<meta name="description" content="{metaDescription}">
```

```
<title>{site_nom} | Erreur 404, cette page n'existe pas !</title>
</head>
<body>
<h1>Erreur 404, cette page n'existe pas !
<br />
<a href="{racine}/" title="Retour au site">Retour au site</a></h1>
</body>
</html>
```

7.3 La page de fermeture du site

La page de fermeture est la page qui est affichée si le site est fermé (soit par manque de crédit, soit volontairement via la rubrique « *Mon compte / Préférences* »).

7.3.1 Répertoire

/theme/fr/modules/fermerture

7.3.2 Exemple du code source

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="keywords" content="{metaKeywords}">
<meta name="description" content="{metaDescription}">
<title>{site_nom} est actuellement ferm&eacute; !</title>
</head>
<body>
<h1>{site_nom} est actuellement fermé !</h1>
</body>
</html>
```

7.4 Le chemin de navigation

Le chemin de navigation, ou fil d'Ariane, est une aide à la navigation sous forme de signalisation de la localisation de l'internaute dans le site internet. Il se présente généralement sous la forme :

```
Accueil / Catalogue / Nom de la catégorie / Nom du produit
```

Le chemin de navigation est construit automatiquement par Kiubi, les intitulés des éléments qu'il affiche ne sont donc pas tous personnalisables. Chaque service aura son propre intitulé :

- Site web : `Accueil / Nom de la page`
- Blog : `Accueil / Blog / Nom de la catégorie`
- Catalogue : `Accueil / Catalogue / Nom de la catégorie / Nom du produit`

- E-commerce : `Accueil / Etape de la commande`
- Gestion de comptes : `Accueil / Gestion de comptes`
- Contact : `Accueil / Contact`

Par défaut, ce module est affiché via l'include `{FILE "/fr/includes/chemin.html"}` dans les modèles de mise en page**. Cet include contient la balise** `{MODULE.cms/chemin}` **utilisée pour appeler le module**.

7.4.1 Répertoire

/theme/fr/modules/chemin

7.4.2 Exemple du code source

```
<a href="{baseLangue}/" title="Accueil">Accueil</a> /
<!-- BEGIN:page -->
<a href="{lien}" title="{page}" target="{cible}">{page}</a>
<!-- BEGIN:separateur -->
/
<!-- END:separateur -->
<!-- END:page-->
```

7.4.3 Balises disponibles

`<!-- BEGIN:page--> <!-- END:page-->`

Le contenu du bloc s'affiche pour chaque page.

`{cible}`

Cible du lien (attribut `target`).
Doit être contenu entre `<!-- BEGIN: page -->` et `<!-- END: page -->`

`{lien}`

URL du lien vers la page.
Doit être contenu entre `<!-- BEGIN: page -->` et `<!-- END: page -->`

`{page}`

Nom de la page.
Doit être contenu entre `<!-- BEGIN: page -->` et `<!-- END: page -->`

`{type}`

Type de la page.
Affiche `page` pour une page libre, `lien_ext` pour un lien externe, `lien_int` pour un lien interne et n'affiche rien pour un séparateur.
Doit être contenu entre `<!-- BEGIN: page -->` et `<!-- END: page -->`

`<!-- BEGIN:separateur--> <!-- END:separateur-->`

Le contenu du bloc va s'afficher entre chaque page du chemin.
Doit être contenu entre `<!-- BEGIN: page -->` et `<!-- END: page -->`

8 Les modèles de mise en page (templates)

8.1 Définition

Les modèles de mise en page, ou templates, constituent les squelettes du site dans lesquels vont s'intégrer les Widgets. Chaque modèle de mise en page contient une structure HTML complète qui lui est propre, des feuilles de styles, des codes javascripts, … mais également la position de chaque zone où vont se placer les Widgets.

Chaque modèle peut être utilisé pour configurer une page type dans l'édition d'un ensemble de mise en page dans la rubrique « *Apparence / Mise en page* » de la console d'administration. Il est donc possible d'utiliser plusieurs modèles de mise en pages différents pour configurer les pages types.

Si vous utilisez un thème graphique personnalisé, vous pouvez modifier les modèles de mise en page et créer des nouveaux modèles.

Avertissement : il est très fortement conseillé de consulter le guide de l'utilisateur pour plus d'information sur les ensembles de mise en pages et les pages types (http://aide.kiubi.com/La-gestion-de-l_Apparence.html**), au risque de ne rien comprendre à ce qui suit !**

8.2 Principe

Une fois connecté à votre espace FTP, tous les modèles de mise en page sont regroupés dans le répertoire « *theme/fr/templates/* ». Chacun de ces sous-répertoires est un modèle de mise en page. Un modèle contient les fichiers suivants :

- *desc.xml :* permet de décrire le modèle de mise en page et déclarer les zones. **Ce fichier est obligatoire.**
- *illustration.gif :* image qui illustre le modèle de mise en page dans la console d'administration. **Ce fichier est obligatoire mais peut être renommé.**
- *index.html :* contient le code HTML du modèle de mise en page. **Ce fichier est obligatoire.**
- *layout.css :* feuille de styles qui définie l'agencement des blocs principaux de mise en page du site. **Ce fichier est facultatif mais il est recommandé de le conserver.**
- *structure.xhtml :* permet de définir la structure du modèle de mise en page utilisée dans l'édition d'une page type. Ce fichier n'a aucun lien avec le rendu du site internet et n'est utilisé que pour la console d'administration. **Ce fichier est obligatoire.**
- *styles.css : f*euille de styles qui définie l'apparence des éléments propres au modèle de mise en page du site à lequel elle est rattachée. **Ce fichier est facultatif mais il est recommandé de le conserver.**

Par défaut, 3 autres fichiers sont présents dans le répertoire « *theme/fr/templates/* ». **Ces fichiers sont facultatifs mais il est recommandé de les conserver** :

- *print.css :* feuille de styles d'impression.
- *styles.css :* feuille de styles principale qui définie l'apparence des éléments communs à tout le site.
- *styles_70.css :* feuille de styles qui est chargée lorsqu'un internaute clique sur le bouton « *Diminuer la taille du texte* ».
- *styles_90.css :* feuille de styles qui est chargée lorsqu'un internaute clique sur le bouton « *Augmenter la taille du texte* ».

8.3 Le fichier « desc.xml »

Ce fichier contient des informations descriptives du modèle de mise en page. Il permet principalement de déclarer les zones du modèle dans lesquelles pourront être insérés des Widgets.

8.3.1 Exemple de code source

```
<?xml version="1.0" encoding="iso-8859-1"?>
<modele img="illustration.gif" intitule="Modèle par défaut">
    <zones>
        <zone id="entete" colmax="1" intitule="Entete" defaut="1"/>
        <zone id="menu" colmax="1" intitule="Menu" defaut="1"/>
        <zone id="sidebar" colmax="1" intitule="Barre latérale" defaut="1"/>
        <zone id="contenu" colmax="3" intitule="Contenu" defaut="1" />
        <zone id="piedepage" colmax="3" intitule="Pied de page" defaut="1"/>
    </zones>
</modele>
```

8.3.2 Description

Ce fichier contient une seule balise principale `<modele>` qui possède 2 attributs :

- *img :* nom du fichier image qui est utilisé dans la console d'administration pour représenter ce modèle.
- *intitule :* intitule du modèle de mise en page qui est utilisé dans la console d'administration

A l'intérieur, nous avons une seule balise `<zones>` qui contient tout un ensemble de balises `<zone>`. La balise `<zone>` sert à déclarer une zone où vont pouvoir être inséré les Widgets. **L'id des zones devra être repris dans les fichiers "*structure.xhtml*" et "*index.html*".** Elles contiennent les attributs suivants :

- *id :* code alphanumérique qui sera réutilisé pour identifié la zone dans les fichiers « *index.html* » et « *structure.xhtml* ».
- *colmax :* nombre maximum de colonnes des blocs de présentation qui peuvent être insérés dans cette zone (de 1 à 3).
- *intitule :* intitulé de la zone qui sera repris dans la console d'administration dans l'édition d'une page type.
- *defaut :* nombre de colonnes des blocs de présentation insérées par défaut dans la zone.

8.4 Le fichier « structure.xhml »

Ce fichier est une représentation schématique de la disposition des zones dans le modèle de mise en page (les zones déclarées dans le fichier « *desc.xml* »). Il est utilisé pour générer la page d'édition d'une page type dans la console d'administration. Il n'est en aucun cas utilisé pour effectuer un rendu du site internet.

8.4.1 Exemple de code source

```
<table>
    <tr>
       <td colspan="2">{entete}</td>
    </tr>
    <tr>
```

```
        <td colspan="2">{menu}</td>
    </tr>
    <tr>
        <td width="30%">{sidebar}</td>
        <td>{contenu}</td>
    </tr>
    <tr>
        <td colspan="2">{piedepage}</td>
    </tr>
</table>
```

8.4.2 Description

Les zones sont appelées directement en mettant leurs ID entre accolade.

Il est obligatoire d'utiliser un code HTML simple, sans scripts ni styles. Seules les balises TABLE, TD et TR sont autorisées.

8.5 Le fichier « index.html »

Alors que le fichier « *structure.xhtml* » est une vue abstraite de la page utilisée seulement dans la console d'administration, le fichier « *index.html* » contient le code HTML concret du modèle de mise en page qui sera affiché dans le site internet. Il est souvent accompagné dans les thèmes de Kiubi par une ou plusieurs feuilles de styles.

8.5.1 Exemple de code source

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

<meta name="keywords" content="{metaKeywords}" />

<meta name="description" content="{metaDescription}" />

<title>{title}</title>

<link href="{racine}/{theme}/{lg}/templates/{modele}/styles.css" rel="stylesheet"
type="text/css" media="screen" />

<link href="{racine}/{theme}/{lg}/templates/styles_70.css" rel="alternate stylesheet"
type="text/css" title="taille_70" media="screen" />

<link href="{racine}/{theme}/{lg}/templates/styles_90.css" rel="alternate stylesheet"
type="text/css" title="taille_90" media="screen" />

<link href="{racine}/{theme}/{lg}/templates/print.css" rel="stylesheet" type="text/css"
media="print" />

<!-- Charge les feuilles de styles des Widgets et des modules -->

{Stylesheet}

{StylesheetCache}

<!-- Charge les js des Widgets et des modules -->

{Javascripts}
```

```
<!-- js nécessaire au module de changement de taille du texte -->
<script language="javascript" type="text/javascript">
        var defaultStyle = 'taille_70';
        loadStyleSheet();
</script>
<!-- Eléments nécessaires au module lightbox -->
<link href="/js/lightbox/lightbox.css" rel="stylesheet" type="text/css" media="screen" />
<!-- Contrôle des éléments qui doivent être initialisés au chargement de la page -->
<script language="javascript" type="text/javascript">
        poolOnLoad.push( function(fct){
                {JavascriptsOnLoad}
        });
</script>
<!-- Charges les scripts des Options avancées du Back-office -->
{js_head}
</head>
<body>
<div id="page">
  <div id="header">{ZONE.entete}</div>
  <div id="tools">{FILE "/fr/includes/date.html"}</div>
  <div id="top">
    <div id="search">
      <form action="{baseLangue}/recherche/{service}/" method="get">
        <table border="0" cellspacing="0" cellpadding="0">
          <tr>
            <td>
              <input type="text" name="r" class="textfield" />
              </td>
            <td><input type="submit" value="" class="bt_search" title="Rechercher" /></td>
          </tr>
        </table>
      </form>
    </div>
    <!-- BEGIN:logo_site -->
    <div id="logo"><a href="{racine}/" title="{accroche_site}"><img
src="{racine}/media/{logo_site}" alt="{accroche_site}" /></a></div>
    <!-- END:logo_site -->
    <!-- BEGIN:accroche_site-->
    <h1><a href="{racine}/" title="{accroche_site}">{accroche_site}</a></h1>
    <!-- END:accroche_site-->
    <!-- BEGIN:desc_site-->
    <h2>{desc_site}</h2>
    <!-- END:desc_site-->
```

```
  </div>
  <div id="menu">{FILE "/fr/includes/taille_texte.html"}{ZONE.menu}</div>
  {FILE "/fr/includes/chemin.html"}
  <div id="container">
    <div id="sidebar">{ZONE.sidebar}</div>
    <div id="main">{ZONE.contenu}</div>
    <div style="clear: both;"></div>
    <div class="bt_top"><a href="#top" title="Haut de la page">Haut de la page</a></div>
  </div>
  <div id="footer"> {ZONE.piedepage}
    &copy; Copyright {MODULE.cms/date|Y} {SOCIETE.societe_nom} - Tous droits reservés.
    {FILE "/fr/includes/copyright.html"} </div>
</div>
<!-- Charges les scripts des Options avancées du Back-office -->
<script type="text/javascript" src="/js/scriptaculous-
1.6.4/scriptaculous.js?load=effects"></script>
<script type="text/javascript" src="/js/lightbox/lightbox.js"></script>
{js_body}
</body>
</html>
```

8.5.2 Description

Les balises `{ZONE.}` servent à indiquer où insérer les Widgets. Les noms des zones sont déclarés dans le fichier « *desc.xml* ».

8.5.3 Balises disponibles

`{Stylesheet}`

Insère les feuilles de styles des Widgets et des modules, si besoin.

`{Javascripts}`

Insère les javascripts nécessaires au bon fonctionnement de Kiubi.

`{JavascriptsOnLoad}`

Insère les javascripts nécessaires au bon fonctionnement des Widgets et des modules, si besoin.

`{js_body}`

Insère du javascript destiné à être exécuté juste avant la fermeture de la balise `<body>` de la page. A configurer dans "*Mon compte / Préférences / Metas par défaut*" ou dans les options avancées de la page.

`{js_head}`

Insère du javascript destiné à être exécuté dans la balise `<head>` de la page. A configurer dans "*Mon compte / Préférences / Metas par défaut*" ou dans les options avancées de la page.

`{metaKeywords}`

Insère les mots clés de la page `<meta keywords>`. A configurer dans "*Mon compte / Préférences / Metas par défaut*" ou dans les options avancées de la page.

`{metaDescription}`

Insère la description de la page `<meta description>`. A configurer dans "*Mon compte / Préférences / Metas par défaut*" ou dans les options avancées de la page.

`{modele}`

Affiche le nom du répertoire du modèle de mise en page utilisé.

`{service}`

Affiche le code du service en cours. Cette balise sert, par exemple, au moteur de recherche pour qu'il fasse des recherches dans le service en cours.
Affiche `cms` si le service en cours est le Site web, Contact et Gestion de compte.
Affiche `blog` si le service en cours est le Blog.
Affiche `catalogue` si le service en cours est le Catalogue et E-commerce (Commandes).

`{title}`

Insère le titre de la page `<title>`. A configurer dans "*Mon compte / Préférences / Metas par défaut*" ou dans les options avancées de la page.

`{ZONE.xxx}`

Insère les Widgets de la zone "*xxx*" telle qu'ils ont été configurés dans l'édition d'une page type. Les noms (identifiants) des zones sont définis dans le fichier "*structure.xhtml*" propre au modèle de mise en page.

`<!-- BEGIN:accroche_site--> <!-- END:accroche_site-->`

Le bloc s'affiche si l'option "*Afficher l'accroche*" est activée dans "*Apparence / Bloc-marque*".

`<!-- BEGIN:desc_site--> <!-- END:desc_site-->`

Le bloc s'affiche si l'option "*Afficher la description*" est activée dans "*Apparence / Bloc-marque*".

`<!-- BEGIN:logo_site--> <!-- END:logo_site-->`

Le bloc s'affiche si l'option "*Afficher le logo*" est activée dans "*Apparence / Bloc-marque*".

8.5.4 Le moteur de recherche

Il est possible d'intégrer un moteur de recherche dans un modèle de mise en page.

```
<div id="search">
  <form action="{baseLangue}/recherche/{service}/" method="get">
    <table border="0" cellspacing="0" cellpadding="0">
      <tr>
        <td><input type="text" name="r" class="textfield" />
```

```
        </td>
        <td><input type="submit" value="" class="bt_search" title="Rechercher" /></td>
      </tr>
    </table>
  </form>
</div>
```

Le moteur de recherche, à ne pas confondre avec les résultats d'une recherche (le Widget « *Recherche simple* »), permet d'effectuer une recherche dans le service en cours dans le site internet. En effet, Kiubi étant multiservices, le moteur de recherche ne peut pas faire de recherche dans l'intégralité du site internet (contenu rédactionnel, blog et catalogue simultanément). La recherche ne s'effectuera que dans le service actuellement visité par l'internaute.

Exemple : si l'internaute visite le blog et qu'il effectue une recherche, la recherche ne sera effectuée que dans le contenu du blog. Il en va de même pour le catalogue de produit et pour le contenu rédactionnel.

La balise `{service}` permet de définir automatiquement le service en cours. Il est cependant possible de forcer une recherche dans un service particulier en remplaçant la balise `{service}` par le service concerné :

- `cms` pour le Site web.
- `blog` pour le Blog.
- `catalogue` pour le Catalogue.

Le paramètre « `name` » du champ de recherche doit impérativement avoir « `r` » pour valeur.

8.6 Ajouter un nouveau modèle de mise en page

Pour ajouter un nouveau modèle de mise en page, il suffit de créer un nouveau sous-répertoire au répertoire « *theme/fr/templates/* » et d'y copier tous les fichiers correspondant à votre nouveau modèle. Le nouveau modèle sera alors automatiquement proposé dans l'édition d'un ensemble de mise en page de la console d'administration.

9 Les Widgets

9.1 Définition

Les Widgets de Kiubi forment la base des différents modules fonctionnels qui composent le site internet. Dans la gestion de l'Apparence, chaque Widget peut être glissé/déposé dans une zone d'une page type et configuré selon un certain nombre de paramètres qui lui sont propres. Afficher un menu, une liste de produits ou même un formulaire de contact, tout ceci est possible grâce aux widgets.

Les 2ème et 3ème parties du guide du Designer décrivent et expliquent en détail les différents paramètres et configurations possibles pour tous les Widgets. De plus, vous y trouverez toutes les balises disponibles par Widgets et dans quelles conditions les utiliser. La compréhension de ces différentes informations (paramètres + configurations + balises) vous permettra de personnaliser l'intégralité des Widgets de Kiubi.

9.2 Classement

Les widgets sont classés selon leur service et leur type :

Services	Widgets	Services	Widgets
Site web	Menu horizontal	**Catalogue**	Catégories
	Menu vertical		Tags
	Contenu d'une page libre		Liste des catégories
	Bloc d'extraits		Liste des produits
Blog	Catégories		Détail d'un produit
	Archives		Ajouter un commentaire
	Liste des billets		Liste des commentaires
	détail d'un billet		Produits vedettes
	liste des commentaires		Produits les plus…
	Ajout de commentaires		Produits également achetés
	Derniers billets postés	**E-commerce**	Panier
	Billets les plus commentés		Détail du panier
	Liens		Livraison
Recherche	Recherche simple		Paiement
Gestion de compte	Identification rapide		Validation
	Identification	**Communication**	Formulaire de contact
	Formulaire d'inscription		Syndication du blog
	Tableau de bord		Inscription à la newsletter
		Boite à outils	Fragment de code

9.3 Personnalisation

Chaque widget de Kiubi peut être personnalisé de deux manières différentes :

- Dans tous les cas, en modifiant ses paramètres et sa configuration, directement lors de l'édition du widget dans la gestion de l'Apparence de la Console d'administration de Kiubi.
- Dans le cas de l'utilisation d'un thème graphique personnalisé, en modifiant les fichiers sources (HTML et CSS) qui le compose.

9.3.1 Modifier les fichiers sources d'un widget

Si vous utilisez un thème graphique personnalisé, vous pouvez éditer et modifier les fichiers sources, c'est-à-dire les fichiers HTML et CSS qui composent et définissent l'apparence d'un widget. Le paramétrage et la configuration d'un widget seront cependant toujours à faire lors de l'édition d'un widget dans la gestion de l'Apparence de la Console d'administration de Kiubi.

Dans votre espace FTP, tous les widgets sont regroupés dans le répertoire « *theme/fr/widgets/* » et classés dans leur propre répertoire, selon leur service (un répertoire par widget classé dans un répertoire par service).

Chaque répertoire principal d'un widget contient :

- un fichier *index.html* structuré de manière différente en fonction du widget. **Ce fichier *index.html* est obligatoire.**

- un fichier *styles.css* peut être ajouté au widget afin de contrôler sa mise en forme. Ce fichier est automatiquement chargé dans la balise `<head>` si le widget est présent dans la page. **Ce fichier *styles.css* n'est pas obligatoire.**

9.3.2 Ajouter des modèles graphiques supplémentaires à un widget

Un même widget peut avoir plusieurs modèles graphiques, c'est-à-dire qu'il peut avoir plusieurs apparences différentes.

Pour ajouter un modèle graphique à un widget, il suffit de créer un sous-répertoire au répertoire principal du widget et d'y copier/coller le fichier *index.html* et/ou *styles.css* correspondant au widget.
Si les deux fichiers sont présents, ce nouveau modèle graphique utilisera les deux fichiers.
Si un seul des deux fichiers est présent, ce nouveau modèle graphique utilisera le fichier présent et complètera par celui présent à la racine du répertoire principal du widget.

Il suffit de choisir ensuite le modèle graphique correspondant lors de l'édition d'un widget dans la gestion de l'Apparence de la Console d'administration de Kiubi.

Il est donc possible, avec le même widget, d'avoir non seulement plusieurs paramétrages et configurations possibles mais en plus, chaque widget peut avoir une infinité d'apparences différentes !

10 Les fonctionnalités et les balises communes des widgets

Un certain nombre de widgets possèdent des fonctionnalités et des balises communes.

10.1 {alterne}

La balise `{alterne}` est utilisée dans les widgets qui génèrent des listings comme par exemple le widget du "*Blog / Liste des Billets*". La balise affichera alternativement `1` ou `2` si on la place dans le bloc qui est répété dans le widget.

Exemple : le widget "*Blog / Liste des Billets*" :

```
<!-- BEGIN:main -->
<!-- BEGIN:billet -->
<ul>
    <li class="color_{alterne}">{alterne}, {titre}</li>
</ul>
<!-- END:billet -->
<!-- END:main -->
```

Si on a 5 billets dans son blog, le résultat sera :

```
1, Titre du premier billet
2, Titre du deuxième billet
```

```
1, Titre du troisième billet
2, Titre du quatrième billet
1, Titre du cinquième billet
```

Cela vous permet, par exemple, d'appeler alternativement l'un ou l'autre style pour changer la couleur de fond et rendre ainsi vos listings plus lisibles.

Les balises `{alterne_3}`, `{alterne_4}` et `{alterne_5}` sont également disponibles et renvoient respectivement les valeurs :

```
1, 2, 3
1, 2, 3, 4
1, 2, 3, 4, 5
```

10.2 {compteur}

La balise `{compteur}` est utilisée dans les widgets qui génèrent des listings comme par exemple le widget du "*Blog / Liste des commentaires*". La balise affichera un compteur commençant à `1` et qui sera incrémenté de 1 à chaque fois.

Exemple : le widget "*Blog / Liste des Billets*" :

```
<!-- BEGIN:main -->
<!-- BEGIN:commentaire-->
<ul>
    <li>{compteur}. {commentaire}</li>
</ul>
<!-- END:commentaire-->
<!-- END:main -->
```

Si on a 3 commentaires pour un billet de son blog, le résultat sera :

```
- 1. Premier commentaire
- 2. Deuxième commentaire
- 3. Troisième commentaire
```

Chaque ligne est numérotée automatiquement.

10.3 {intitule}

Cette fonctionnalité est une association entre un bloc et une balise. Les widgets pouvant utiliser la fonctionnalité "*Intitulé*" ont un bloc supplémentaire à placer dans le bloc "*main*" :

```
<!-- BEGIN:main -->
```

```
    <!-- BEGIN:intitule -->
        {intitule}
    <!-- END:intitule -->
<!-- END:main -->
```

Le bloc `<!-- BEGIN:intitule --> <!-- END:intitule -->` ne s'affiche que si l'intitulé du widget est rempli. Cet intitulé est paramétrable lors de l'édition d'un widget dans la gestion de l'Apparence de la Console d'administration de Kiubi.

10.4 {ctl}

Cette balise est présente dans certain widgets qui utilisent un formulaire. Elle n'a qu'une utilité technique et est présente sous la forme :

```
<input type="hidden" name="ctl" value="{ctl}"/>
```

Il ne faut pas la supprimer, sinon le formulaire ne fonctionnera plus !

10.5 La navigation

La navigation est une fonctionnalité que l'on retrouve dans les widgets qui génèrent des listings et dont la configuration permet l'affichage. C'est un ensemble de blocs qui permet d'afficher des barres de navigations complètes. Les résultats sont affichés sur plusieurs pages et la navigation permet de passer d'une page à l'autre.

Les options de navigation (nombre d'élément par page, afficher ou cacher la navigation) sont toujours configurables dans le widget correspondant.

La barre de navigation comprend :

- des liens vers les pages suivantes et précédentes,
- des liens vers la première et dernière page,
- une liste de numéro de page comprenant un lien pour arriver directement sur la page en question.

Pour plus de souplesse, il est possible de mettre deux barres de navigation dans un même widget (une barre en haut et une en bas du listing, par exemple). Les balises et blocs ci-dessous fonctionnent dans les blocs `<!-- BEGIN: nav1 --> <!-- END: nav1 -->` et `<!-- BEGIN: nav2 --> <!-- END: nav2 -->`.

Voici un exemple de ce que cela peux donner en HTML :

```
<!-- BEGIN: nav1 -->
   <!-- BEGIN: premier -->
   <a href="{lien_premier}">premiere page</a> |
   <!-- END: premier -->
   <!-- BEGIN: precedent -->
   <a href="{lien_precedent}">page precedente</a> |
   <!-- END: precedent -->
   <!-- BEGIN: pages -->
```

```
   <a href="{lien_page}" class="{selected}">{page}</a>
   <!-- END: pages -->
   <!-- BEGIN: suivant -->
   | <a href="{lien_suivant}">page suivante</a>
   <!-- END: suivant -->
   <!-- BEGIN: dernier -->
   | <a href="{lien_dernier}">derniere page</a>
   <!-- END: dernier -->
 <!-- END: nav1 -->
<br />
<br />
 page {num_page} sur {nb_pages} ({nb_resultats} billets trouvés)<br />
<br />
[listing ....]<br />
<br />
<!-- BEGIN: nav2 -->
  <!-- BEGIN: premier -->
  <a href="{lien_premier}">premiere page</a> |
  <!-- END: premier -->
  <!-- BEGIN: precedent -->
  <a href="{lien_precedent}">page precedente</a> |
  <!-- END: precedent -->
  <!-- BEGIN: pages -->
  <a href="{lien_page}" class="{selected}">{page}</a>
  <!-- END: pages -->
  <!-- BEGIN: suivant -->
  | <a href="{lien_suivant}">page suivante</a>
  <!-- END: suivant -->
  <!-- BEGIN: dernier -->
  | <a href="{lien_dernier}">derniere page</a>
  <!-- END: dernier -->
<!-- END: nav2 -->
```

Ce qui donnera cela :

```
premiere page | page precedente | 1 2 3 4 5 | page suivante |  derniere page

 page 2 sur 5 (53 billets trouvés)

[listing ...]

 premiere page | page precedente | 1 2 3 4 5 | page suivante |  derniere page
```

10.5.1 Balises disponibles

`{lien_dernier}`

URL vers la dernière page.
Doit être contenu entre `<!-- BEGIN: nav1 -->` et `<!-- END: nav1 -->` ou `<!-- BEGIN: nav2 -->` et `<!-- END: nav2 -->`

`{lien_precedent}`

URL vers la page précédente.
Doit être contenu entre `<!-- BEGIN: nav1 -->` et `<!-- END: nav1 -->` ou `<!-- BEGIN: nav2 -->` et `<!-- END: nav2 -->`

`{lien_premier}`

URL vers la première page.
Doit être contenu entre `<!-- BEGIN: nav1 -->` et `<!-- END: nav1 -->` ou `<!-- BEGIN: nav2 -->` et `<!-- END: nav2 -->`

`{lien_suivant}`

URL vers la page suivante.
Doit être contenu entre `<!-- BEGIN: nav1 -->` et `<!-- END: nav1 -->` ou `<!-- BEGIN: nav2 -->` et `<!-- END: nav2 -->`

`{nb_pages}`

Nombre de page du listing.
Doit être contenu entre `<!-- BEGIN: nav1 -->` et `<!-- END: nav1 -->` ou `<!-- BEGIN: nav2 -->` et `<!-- END: nav2 -->`

`{nb_resultats}`

Nombre d'élément dans le listing.
Doit être contenu entre `<!-- BEGIN: nav1 -->` et `<!-- END: nav1 -->` ou `<!-- BEGIN: nav2 -->` et `<!-- END: nav2 -->`

`{num_page}`

Numéro de la page courante.
Doit être contenu entre `<!-- BEGIN: nav1 -->` et `<!-- END: nav1 -->` ou `<!-- BEGIN: nav2 -->` et `<!-- END: nav2 -->`

`<!-- BEGIN:dernier--> <!-- END:dernier-->`

Le bloc s'affiche si on n'est pas sur la dernière page du listing.
Doit être contenu entre `<!-- BEGIN: nav1 -->` et `<!-- END: nav1 -->` ou `<!-- BEGIN: nav2 -->` et `<!-- END: nav2 -->`

`<!-- BEGIN:pages--> <!-- END:pages-->`

Le contenu du bloc s'affiche pour chaque page.
Doit être contenu entre `<!-- BEGIN: nav1 -->` et `<!-- END: nav1 -->` ou `<!-- BEGIN: nav2 -->` et `<!-- END: nav2 -->`

`{lien_page}`

URL vers la page.
Doit être contenu entre `<!-- BEGIN: pages -->` et `<!-- END: pages -->`

`{page}`

Numéro de la page.
Doit être contenu entre `<!-- BEGIN: pages -->` et `<!-- END: pages -->`

`{selected}`

Affiche "selected" si la page est la page courante. Sinon n'affiche rien.
Doit être contenu entre `<!-- BEGIN: pages -->` et `<!-- END: pages -->`

`<!-- BEGIN:precedent--> <!-- END:precedent-->`

Le contenu du bloc s'affiche si on n'est pas sur la première page du listing.
Doit être contenu entre `<!-- BEGIN: nav1 -->` et `<!-- END: nav1 -->` ou `<!-- BEGIN: nav2 -->` et `<!-- END: nav2 -->`

`<!-- BEGIN:premier--> <!-- END:premier-->`

Le contenu du bloc s'affiche si on n'est pas sur la première page du listing.
Doit être contenu entre `<!-- BEGIN: nav1 -->` et `<!-- END: nav1 -->` ou `<!-- BEGIN: nav2 -->` et `<!-- END: nav2 -->`

`<!-- BEGIN:suivant--> <!-- END:suivant-->`

Le contenu du bloc s'affiche si on n'est pas sur la dernière page du listing.
Doit être contenu entre `<!-- BEGIN: nav1 -->` et `<!-- END: nav1 -->` ou `<!-- BEGIN: nav2 -->` et `<!-- END: nav2 -->`