**CMS Miki**

**Guide spécifique**

**V 1.0**

**30.11.2010 - Hervé Torche**

Table des matières

[2 Préambule 3](#_Toc280621830)

[3 Mise en place d’un gabarit 4](#_Toc280621831)

[4 Feuilles de style CSS 5](#_Toc280621832)

[5 Menus 5](#_Toc280621833)

[6 Liens sur les pages 6](#_Toc280621834)

[6.1 Lien depuis une page intégrée dans le Miki sur une page du CMS 6](#_Toc280621835)

[6.2 Lien depuis une page non-intégrée dans le Miki sur une page du CMS 6](#_Toc280621836)

[7 Redirection d’une page en php 8](#_Toc280621837)

[8 Donner un titre spécifique (balise meta « title ») à une page 8](#_Toc280621838)

[9 Images, scripts javascript externes et feuilles de style CSS externes 9](#_Toc280621839)

[10 Internationalisation d’un site web 9](#_Toc280621840)

[10.1 Gestion des langues dans le Miki 10](#_Toc280621841)

[10.2 Changement de langue par le visiteur du site Internet 10](#_Toc280621842)

[11 Navigation – dernière page visitée 10](#_Toc280621843)

[12 Pages sécurisées – connexion des membres 11](#_Toc280621844)

[13 Récupération des variables de configuration 12](#_Toc280621845)

[14 Configuration d’un nouveau site Internet 12](#_Toc280621846)

# Préambule

Ce document est un guide permettant au développeur débutant avec le CMS Miki de prendre cet outil en main et de pouvoir utiliser certaines fonctions spécifiques.

Le CMS Miki est très simple d’utilisation et est basé sur une architecture orientée objets. Tous les éléments du CMS sont donc représentés par des objets. Ceux-ci évoluent au fil des projets et des besoins clients. Leur documentation complète (et mise à jour régulièrement) est disponible à l’adresse suivante : <http://www.fbw-one.com/miki_help>.

Pour toute question relative au CMS ou à ce document, veuillez contacter Hervé Torche par e-mail à l’adresse suivante : [herve@fbw-one.com](mailto:herve@fbw-one.com).

# Mise en place d’un gabarit

Un gabarit permet de mettre en place le visuel du site Internet une seule fois. Il est ainsi plus facile et plus rapide de maintenir à jour ou de modifier ce visuel de par le fait que les modifications sont apportées à un seul endroit et sont répercutées sur toutes les pages utilisant ce gabarit.

Un gabarit est lié à une feuille de style CSS (voir chapitre « Feuilles de style CSS »).

Un gabarit est intégré dans le Miki via le menu « Disposition » 🡪 « Gabarits ». Il suffit de donner un nom au gabarit, de lui affecter une feuille de style CSS (optionnel mais très conseillé) et de lui donner un contenu. Le contenu peut, au choix, être directement écrit dans le Miki (dans ce cas, les modifications devront être apportées également dans le Miki) ou alors dans un fichier PHP (à enregistrer dans le répertoire « template » du projet) que l’on insèrera dans le champ « Contenu » du gabarit via l’instruction PHP suivante :

<?php require\_once("template/mon\_template.php"); ?>

A l’intérieur du gabarit, le code [miki\_content] est obligatoire. Lors de l’affichage d’une page utilisant ce gabarit, le contenu de la page sera intégré dans le gabarit à la place de ce code.

Il est également possible d’intégrer un menu à l’intérieur d’un gabarit en ajoutant le code suivant [miki\_menu='mon\_menu'] (Pour plus d’information sur les menus, voir le chapitre « Menus »).

A VENIR : intégration des sections de gabarit

# Feuilles de style CSS

Comme vu dans le chapitre précédent, un gabarit peut-être (c’est très conseillé) lié à une feuille de style CSS.

Une feuille de style CSS peut être insérée dans le Miki via le menu « Disposition » 🡪 « Feuille de style ».

Comme pour les gabarits, il suffit de lui donner un nom et un contenu. Pour le contenu, toujours comme pour les gabarits, celui-ci peut, au choix, être directement écrit dans le Miki (dans ce cas, les modifications devront être apportées également dans le Miki) ou alors dans un fichier CSS (à enregistrer dans le répertoire « css » du projet) que l’on insèrera dans le champ « Contenu » de la feuille de style via l’instruction PHP suivante :

<?php require\_once("css/ma\_feuille\_de\_style.css"); ?>

A la différence d’une feuille de style classique, les fichiers CSS intégrés dans le Miki peuvent contenir, si besoin, du code PHP. En effet, le CMS utilise des fonctions PHP pour inclure les feuilles de style dans les gabarits. Le code PHP à l’intérieur d’un fichier CSS est donc analysé et interprété.

# Menus

Le CMS Miki gère également les menus du site Internet. Un menu peut être inséré dans le Miki via le menu « Disposition » 🡪 « Menus ».

Comme pour les gabarits, il suffit de lui donner un nom, de lui affecter une feuille de style (optionnelle) et un contenu.   
Pour la feuille de style (voir chapitre « Feuilles de style CSS ») on peut lui en affecter une dans le but de séparer la feuille de style du menu de celle du gabarit, mais ce n’est pas obligatoire. On peut directement intégrer le code CSS dédié au menu dans la feuille de style liée gabarit utilisant le menu   
Pour le contenu, toujours comme pour les gabarits, celui-ci peut, au choix, être directement écrit dans le Miki (dans ce cas, les modifications devront être apportées également dans le Miki) ou alors dans un fichier PHP (à enregistrer dans le répertoire « menus » du projet) que l’on insèrera dans le champ « Contenu » du menu via l’instruction PHP suivante :

<?php require\_once("menus/mon\_menu.php"); ?>

Pour inséré le menu dans un gabarit, il suffit d’ajouter le code suivant à l’emplacement prévu pour le menu : [miki\_menu='mon\_menu'] (où « mon\_menu » correspond au nom du menu configuré dans le Miki).

Par défaut, le CMS Miki contient un menu sur plusieurs niveaux du nom de « menu\_default ». Le code de ce menu se trouve dans le fichier « menu\_default.php » du répertoire « menus ». Ce menu peut être utilisé tel quel et mise en page via CSS ou alors modifié pour répondre au besoin du projet.

# Liens sur les pages

## Lien depuis une page intégrée dans le Miki sur une page du CMS

Il faut mettre le code suivant : [miki\_page=’nom\_de\_la\_page’]

« nom\_de\_la\_page » correspond au nom de la page donné dans le Miki.

Il est possible de passer des paramètres en GET à cette page en mettant le code suivant : [miki\_page=’nom\_de\_la\_page’ params=’mon\_param1=ma\_val1&&mon\_param2=ma\_val2’]

Le différents paramètres doivent être séparés par « && »

## Lien depuis une page non-intégrée dans le Miki sur une page du CMS

La page source n’étant pas intégrée dans le Miki, il faut ici insérer les fonctions utilisées par le Miki manuellement.

Voici un exemple :

<?php  
 require\_once ("include/headers.php");  
 require\_once("set\_links.php");  
  
 $code = "<a href='[miki\_page='nom\_de\_la\_page\_destination' params='mon\_param1=ma\_val1']'>mon lien</a>";  
 set\_links($code, $\_SESSION['lang']);  
 echo $code;  
?>

La page incluse « headers.php » contient toutes les informations nécessaires au CMS.

La page incluse « set\_links.php » contient les fonctions faisant la correspondance entre le code *[miki\_page=’…’]* et le lien réel de la page.

Pour « headers.php » il est nécessaire de spécifier le chemin complet en partant depuis le répertoire où est située la page (en générale la racine du site), soit « include/headers.php ». Ensuite, pour les pages des répertoires « include », « scripts » et « class », il n’est plus nécessaire de spécifier le chemin complet. Ceux-ci sont en effet déjà enregistrés dans la configuration « include\_path » de PHP.

La fonction « set\_links() », contenue dans la page « set\_links.php » doit être utilisée afin de faire correspondre le code [miki\_page=’…’] et le lien réel de la page.

Il faut utiliser ce code dans les pages n’étant pas intégrées dans le Miki telles que des pages de validation de formulaire p.ex. Ce sont des pages qui n’affichent rien à l’écran. Elles peuvent effectuer différentes opérations (création/modification/suppression d’un objet, envoi d’un e-mail, etc.) puis rediriger vers une autre page.

Voici un exemple concret récupérant les informations d’un formulaire de demande de contact.

On peut voir ici différentes fonction du Miki qui sont expliquées plus loin dans ce document.

<?php  
 require\_once("include/headers.php");  
 require\_once("set\_links.php");

// récupert les information dans une personne (ne sera pas sauvée dans la BDD)  
 $person = new Miki\_person();  
 $person->firstname = stripslashes($\_POST['firstname']);  
 $person->lastname = stripslashes($\_POST['lastname']);  
 $person->address = stripslashes($\_POST['address']);  
 $person->npa = $\_POST['npa'];  
 $person->city = stripslashes($\_POST['city']);  
 $person->country = stripslashes($\_POST['country']);  
 $person->email1 = $\_POST['email'];  
 $person->tel1 = $\_POST['tel'];

$message = nl2br(stripslashes($\_POST['message']));

$\_SESSION['person'] = $person;  
 $\_SESSION['message'] = $message;

$sitename = Miki\_configuration::get('sitename');  
 $email\_answer = Miki\_configuration::get('email\_answer');

// création du mail  
 $mail = new miki\_email('contact', 'fr');

$mail->From = $person->email1;  
 $mail->FromName = $person->lastname ." " .$person->firstname;

// prépare les variables nécessaires à la création de l'e-mail  
 $vars\_array['person\_to'] = $person;  
 $vars\_array['message'] = $message;  
 $vars\_array['sitename'] = $sitename;

// initialise le contenu de l'e-mail  
 $mail->init($vars\_array);

$mail->AddAddress($email\_answer);

if(!$mail->Send())  
 $\_SESSION['message\_error'] = "Une erreur est survenue lors de l'envoi du message : <br />" .$mail->ErrorInfo;  
 else{  
 $\_SESSION['message\_success'] = "Votre message nous est parvenu avec succès.<br />Nous allons traiter votre message dans les plus brefs délais.";  
 unset($\_SESSION['person']);  
 unset($\_SESSION['message']);  
 }

$mail->ClearAddresses();

$code = "[miki\_page='contact']";  
 set\_links($code, $\_SESSION['lang']);  
 miki\_redirect($code);  
?>

# Redirection d’une page en php

Dans un code PHP, pour rediriger l’internaute vers une autre page, il faut utiliser la fonction suivante :

Miki\_redirect(‘mon\_url’);

* « mon\_url » peut correspondre à un lien en dur pour une page externe au site (p.ex <http://www.google.com>) ou à un lien interne en utilisant le code *[miki\_page=’…’]*

# Donner un titre spécifique (balise meta « title ») à une page

D’ordinaire, le titre (balise meta « title ») d’une page intégrée dans le Miki est donné dans le Miki via le champ « Titre » du formulaire d’ajout/édition d’une page.

Si une page doit avoir un titre différent selon son contenu, il est possible de surcharger ce titre en insérant dans la page la balise suivante : <title2>Mon titre différent</title2>

Ceci est utile p.ex. dans une page affichant les détails d’une personne, d’un événement, d’une actualité. Dans le cadre d’une page affichant les détails d’un événement, nous aurons p.ex. le code suivant :

<?php   
 require\_once ("config/event\_category.php");

// si pas d'id d'événement spécifié, on retourne à la page précédente  
 if (!isset($\_GET['eid']) || !is\_numeric($\_GET['eid'])){  
 miki\_redirect($\_SESSION['url\_back']);  
 }

try{  
 $event = new Miki\_event($\_GET['eid']);  
 $title = $event->title[$\_SESSION['lang']];  
 }  
 catch(Exception $e){  
 miki\_redirect($\_SESSION['url\_back']);  
 }  
?>

<title2><?php echo $title; ?></title2>

# Images, scripts javascript externes et feuilles de style CSS externes

Pour l’intégration d’une image, d’un script Javascript externe ou d’une feuille de style CSS externe dans une page du CMS, il est important d’ajouter le code suivant dans l’attribut « src » de la balise : < ?php echo URL\_BASE ; ?>

La constante « URL\_BASE » pointe toujours à la racine du site Internet. Il suffit donc d’ajouter le chemin de l’objet à insérer à partir de la racine du site.

P.ex., si on veut insérer l’image « mon\_image.jpg » se trouvant dans le répertoire « pictures » du site, il faut mettre la balise suivante :

<img src="< ?php echo URL\_BASE; ?>pictures/mon-image.jpg" />

Pour le script Javascript « mon\_script.js » se trouvant dans le répertoire « scripts » :

<script type="text/javascript" src="<?php echo URL\_BASE; ?>scripts/mon\_script.js"></script>

Pour la feuille de style CSS « mon\_style.css » se trouvant dans le répertoire « css » :

<link rel="stylesheet" type="text/css" href="<?php echo URL\_BASE; ?>css/mon\_style.css" />

# Internationalisation d’un site web

Beaucoup de nos clients demandent un site en plusieurs langues. Dans tous les cas (même sans demande explicite) nous préparons les sites pour l’internationalisation.

Le Miki gère bien le multilingue pour ce qui est du contenu entré directement à l’intérieur du CMS mais pour toutes les pages, gabarits, menus ou autres fichiers n’étant pas « entrés » directement dans le CMS (tous les fichiers qui sont donc « codés à la main »), il est nécessaire d’utiliser la fonction « Gettext » de PHP (pour plus d’informations sur « Gettext », voir ici : <http://php.net/manual/fr/book.gettext.php>).

Pour chaque mot ou phrase écrit en dur dans le code (que ce soit en Français ou dans une autre langue) il faut donc utiliser Gettext.

La fonction utilisée est la suivante : < ?php echo \_("mon texte qui pourra être traduit"); ?>

Si besoin, le client pourra donc traduire lui-même tous les textes insérés via cette fonction et permettre une internationalisation du site web.

Si une variable PHP doit intervenir dans une phrase, il faut ajouter la fonction « printf » (pour plus d’infos sur printf, voir ici : <http://ch2.php.net/manual/fr/function.printf.php>)

Exemple :

<?php   
 $prenom = "Jean" ;  
 printf(\_("Hello %s!"), $prenom);  
 ?>

## Gestion des langues dans le Miki

Pour permettre à un site Internet d’être disponible en plusieurs langues, celles-ci doivent être configurées dans le Miki via le menu « Administration » 🡪 « Langues ».

Une langue est formée d’un nom (p.ex. Français), d’un code de 2 caractères (p.ex FR) ainsi que d’une image représentant la langue (optionnel). Une langue (une seule) peut être définie comme langue par défaut. Celle-ci sera donc la langue par défaut lors de l’affichage des pages du site Internet si aucune langue n’est spécifiée par le visiteur.

Il est possible de récupérer (en PHP) la langue dans laquelle une page est affichée en lisant la valeur de la variable de session $\_SESSION["lang"]   
Cette variable vous donne le code (2 caractères) de la langue. Vous pouvez ensuite récupérer l’objet représentant la langue via la fonction « load\_from\_code » de la classe « Miki\_language ».

Exemple :   
$langue = new Miki\_language() ;  
$langue->load\_from\_code($\_SESSION["lang"]);

Pour plus d’informations, veuillez vous référer à l’aide API disponible à l’adresse suivante :   
<http://www.fbw-one.com/miki_help>

## Changement de langue par le visiteur du site Internet

Pour permettre au visiteur de changer de langue d’affichage, le script PHP « langues.php » est présent dans le répertoire « include » du projet. Il suffit de faire un « include » de ce fichier dans un gabarit pour afficher les langues disponibles à l’endroit désiré. Ce script peut bien entendu être modifié à volonté. Il faut toutefois faire attention à conserver toutes les variables passée à la page (voir dans le script).

# Navigation – dernière page visitée

Le CMS Miki conserve dans la variable de session $\_SESSION["url\_back"] la dernière page (intégrée au Miki) visitée. Il est donc possible dans toute page PHP du projet de rediriger l’utilisateur sur la dernière page visitée en utilisant cette variable.

Exemple :

miki\_redirect($\_SESSION['url\_back']);

# Pages sécurisées – connexion des membres

Le Miki permet de mettre en place très simplement et de façon très efficace une ou plusieurs parties du site sécurisées.

L’accès aux pages d’une partie sécurisée se fait via un login et un mot de passe. Les membres peuvent être gérés entièrement (ajout/modification/suppression) via le Miki mais on met généralement en place un système de création et de modification de compte dans le site Internet pour permettre au non-membre de le devenir.

Une inscription peut devoir faire l’objet d’une validation de la part de l’administrateur du site (voir fonction « send\_inscription » de la classe « Miki\_person »).

Pour la connexion d’un membre sur le site Internet, il suffit de poster les variables « username » et « password » au script « test\_connexion.php » qui se trouve dans le répertoire « scripts ». Pour permettre au membre de mémoriser son accès depuis le même ordinateur, il suffit de poster également la variable « memorize » (si memorize = 1, l’accès est mémorisé).

Pour définir une page comme étant « sécurisée », il faut définir le type de login « dans le Miki ». Si le type de login = 0, la page n’est pas sécurisée. Si > 0, la page est sécurisée. Il est ainsi possible de définir plusieurs zones de sécurité différents en donnant un type de login = 1 à une certaines zone, un type de login = 2 à une autre zone et ainsi de suite.

Lors de l’affichage d’une page intégrée dans le Miki possédant un type de login > 0, une vérification automatique est effectuée pour savoir si le visiteur est authentifié ou non. S’il n’est pas authentifié, la page ne s’affiche pas et le visiteur est automatiquement redirigé sur la page d’accueil (la page de redirection est à configurer dans le fichier « index.php » à la ligne 239 « $page->load\_from\_name('accueil'); ». Il suffit de modifier « accueil » par le nom de la page sur laquelle le visiteur non authentifié sera redirigé.

Pour les pages non intégrée dans le Miki, il est possible de vérifier si le visiteur est authentifié, en appelant la fonction « is\_connected() » (il faut au préalable avoir inclus la page « headers.php » du répertoire « include »).

Cette fonction « is\_connected() » retourne un objet de type « Miki\_person » représentant le membre connecté si le visiteur est authentifié ou **false** sinon. Pour vérifier si un utilisateur est connecté, il faut tester la valeur de retour avec un triple égal de PHP (===).

Exemple :

< ?php  
 $person\_connected = is\_connected();

// si l'événement est privé et que l'utilisateur n'est pas authentifié, on le redirige vers la page de connexion  
 if ($person\_connected === false){  
 /\*   
 instruction si le membre n’est pas connecté. Exemple :   
 $ma\_page->load\_from\_name('login');  
 miki\_redirect($ma\_page->get\_url\_simple($lang));  
 \*/  
 }  
 else{  
 // instruction si le membre est connecté.  
 }  
?>

Cette fonctione peut également être utilisée dans les pages intégrées au Miki afin de récupérer l’objet de type « Miki\_person » représentant le membre connecté.

# Récupération des variables de configuration

Dans le menu « Administration » 🡪 « Configuration du site Internet » du Miki, on peut entrer plusieurs configurations.

Les plus importantes sont les 3 premières de l’onglet « Général ». Voici leur explication :

* Nom du site Internet : Le nom du site Internet est utilisé principalement dans les signatures des e-mails ainsi qu’à chaque fois qu’on a besoin d’afficher le nom du site Internet à l’écran.
* URL du site Internet : De la même façon que le nom du site Internet, l’url du site Internet est utilisée principalement dans les e-mails lorsque l’on a besoin d’y insérer un lien vers le site. L’url est également utilisée lorsque l’on veut rediriger l’utilisateur vers la page d’accueil (via la fonction miki\_redirect()).
* Adresse e-mail de l'expéditeur des e-mails : Cette adresse est utilisée comme adresse d’expéditeur pour tous les e-mails destinés aux membres (ou visiteurs) du site Internet. Elle est également utilisée comme destinataire pour tous les mails destinés à l’administrateur du site web. Cette adresse fait aussi office d’expéditeur lors de l’envoi d’une newsletter grâce au module « Miki Newsletter ».
* Identifiant Google Analytics : L’identifiant du site Internet sur Google Analytics. Ceci permet d’activer les statistiques du site.

Pour récupérer les valeurs de ces configurations, la classe Miki\_configuration propose une fonction statique « get ». Son utilisation est la suivante :

Miki\_configuration::get('nom\_de\_la\_variable');

Voici les différents noms de variable à utiliser pour récupérer les informations entrées dans le Miki :

* **'sitename'** pour le nom du site Internet.
* **'site\_url'** pour l’url.
* **'email\_answer'** pour l’adresse e-mail de l'expéditeur des e-mails

# Configuration d’un nouveau site Internet

Lors du démarrage d’un nouveau projet, quelques opérations sont nécessaires afin de configurer le Miki :

* Installer la base de données.
* Configurer la connexion à la base de données (fichier « config.php » du répertoire « scripts »).
* Configurer le nom du site Internet (menu « Administration » 🡪 « Configuration du site Internet » dans le Miki).
* Configurer l’URL du site Internet (menu « Administration » 🡪 « Configuration du site Internet » dans le Miki).  
  L’URL du site Internet correspond à l’adresse grâce à laquelle vous accéder au site. Ex. pour un site en développement : <http://mon-site.com/beta>
* Configurer l’adresse e-mail (menu « Administration » 🡪 « Configuration du site Internet » dans le Miki).   
  Cette adresse e-mail sera utilisée en tant qu’expéditeur des mails adressés aux membres su site ou aux abonnés de la newsletter. C’est également à cette adresse que les mails à destination de l’administrateur du site Internet seront adressés par défaut.