

CHAPITRE 22 — Créer ses propres gabarits de page

Lorsque votre client clique sur « + Nouvelle page » dans le Studio, il choisit un gabarit de départ. Ce chapitre vous explique comment créer vos propres gabarits adaptés à la charte de votre site.

À quoi sert un gabarit

Un gabarit est un squelette de page qui comporte :

- Du HTML structuré (sections, classes CSS, balises sémantiques).
- Des slots pré-déclarés pour les zones éditables.
- Des valeurs par défaut, pour disposer d'une page complète dès la création.
- Des métadonnées (titre par défaut, slug, type de conteneur).

Lorsqu'un client crée une page depuis un gabarit, le module :

1. Crée une entrée dans la table `llx_website_page`.
2. Génère un fichier `page<N>.tpl.php` à partir du squelette.
3. Crée le wrapper Apache `<slug>.php`.
4. Lance un rescan des slots pour les détecter immédiatement.

Structure d'un gabarit

Un gabarit est un dossier dans `htdocs/custom/infrasstudio/templates/` :

```
templates/mon-gabarit/  
├─ meta.php          # Métadonnées du gabarit  
└─ skeleton.tpl.php # Le squelette HTML avec slots
```

Le fichier meta.php

Il retourne un tableau de configuration :

```
<?php  
return array(  
    'code'          => 'mon-gabarit',          // identifiant unique
```

```

'label'          => 'Mon Gabarit',                // affiché dans l'assistant
'category'      => 'page',                      // page | landing | blog
'description'   => "Description courte affichée sous le titre.",
'icon'          => 'fa-file-lines',            // icône FontAwesome
'type_container' => 'page',                    // type Dolibarr Website
'default_slug' => 'nouvelle-page',            // slug suggéré
);

```

| Champ | Description |
|-----------------------------|--|
| <code>code</code> | Identifiant unique. Doit correspondre au nom du dossier. |
| <code>label</code> | Texte affiché dans la grille de sélection de l'assistant. |
| <code>category</code> | Permet le regroupement visuel (page, landing, blog). |
| <code>icon</code> | Classe FontAwesome de l'icône de la tuile. |
| <code>type_container</code> | Valeur écrite dans <code>llx_website_page.type_container</code> . Valeurs standards : <code>page</code> <code>blogpost</code> <code>other</code> <code>menu</code> <code>.</code> |
| <code>default_slug</code> | Slug suggéré dans le formulaire. Le client peut le modifier. |

Le fichier skeleton.tpl.php

Il s'agit du squelette HTML, identique à un fichier tpl.php standard, à la différence qu'il contient des marqueurs qui seront remplacés au moment de la création.

Marqueurs disponibles

| Marqueur | Remplacé par |
|----------|--------------|
|----------|--------------|

| | |
|--------------------------|---|
| <code>@@PAGEID@@</code> | L'identifiant Dolibarr de la nouvelle page (par exemple 42). |
| <code>@@PAGEURL@@</code> | Le slug URL final (par exemple <code>about-keatic</code>). |
| <code>@@ISO2@@</code> | Le code ISO2 de la langue principale (<code>fr</code> , <code>en</code>). |

Exemple complet — gabarit page-libre

meta.php

```
<?php
return array(
    'code'          => 'page-libre',
    'label'         => 'Page libre',
    'category'      => 'page',
    'description'   => 'Une page simple avec un titre et un grand champ de texte riche.',
    'icon'          => 'fa-file-lines',
    'type_container' => 'page',
    'default_slug'  => 'nouvelle-page',
);
```

skeleton.tpl.php

```
<?php // BEGIN PHP
$websitekey = basename(__DIR__);
if (! defined('USEDOLIBARRSERVER') && ! defined('USEDOLIBARREDITOR')) {
    require_once __DIR__ . '/master.inc.php';
}
require_once DOL_DOCUMENT_ROOT . '/core/lib/website.lib.php';
require_once DOL_DOCUMENT_ROOT . '/core/website.inc.php';
ob_start();
try {
// END PHP ?>
<html lang="@@ISO2@">
<head>
```

```

<title>{{slot:page_title|type=text|default=Nouvelle page|label=Titre
SEO|group=seo}}</title>
<meta name="description"
content="{{slot:page_meta_description|type=text|default=|label=Meta description|group=seo}}"
/>
<link rel="canonical" href="<?php echo $website->virtualhost; ?>/@@PAGEURL@@.php" />
</head>
<body>
<?php includeContainer('header'); ?>
<main class="container">
<h1>{{slot:page_h1|type=text|default=Titre principal|label=H1|group=hero}}</h1>
<div class="content">
{{slot:page_body|type=richtext|default=<p>Contenu de la page.</p>|label=Contenu}}
</div>
</main>
<?php includeContainer('footer'); ?>
</body>
</html>
<?php // BEGIN PHP
} catch (Exception $e) { print $e->getMessage(); }
include dol_buildpath('/infrasstudio/core/tpl/website_output.tpl.php', 0);
// END PHP ?>

```

Le bloc final — N'oubliez pas la ligne `include`

`dol_buildpath('/infrasstudio/core/tpl/website_output.tpl.php', 0);`. C'est elle qui déclenche la résolution des slots et shortcodes au moment du rendu.

Localiser vos gabarits hors du module

Si vous souhaitez livrer des gabarits avec votre projet client sans modifier le module, définissez la constante :

```

// htdocs/conf/conf.php ou via dolibarr_set_const
$conf->global->INFRASTUDIO_TEMPLATE_EXTRA_DIR = '/var/www/monsite/templates';

```

Le module scanne ce répertoire en complément de `htdocs/custom/infrasstudio/templates/`.

Gabarits livrés par défaut

Code

Description

| | |
|------------------------------|--|
| <code>page-free</code> | Page libre avec titre et un grand champ texte riche. Pour les pages ponctuelles. |
| <code>blog-standard</code> | Article de blog générique avec hero, introduction, corps et appel à l'action. |
| <code>example-blog</code> | Article de blog au design moderne (hero CSS, accroche en italique, image secondaire, articles liés). Adaptable à votre charte. |
| <code>example-landing</code> | Page de destination produit complète (environ 70 slots). Hero, problème, solution, fonctionnalités, contact, FAQ. |

Conseil — Inspirez-vous de `example-landing` pour comprendre comment structurer un gabarit complexe avec environ 70 slots organisés en sections.

Bonnes pratiques pour vos gabarits

- Préfixez tous les slots du gabarit par un identifiant commun (par exemple `landing_`) pour éviter les collisions entre gabarits.
- Regroupez les slots avec `group=` par section logique (hero, fonctionnalités, contact, etc.).
- Donnez des valeurs par défaut représentatives. Le rédacteur dispose ainsi d'un exemple à modifier plutôt que d'une page vide intimidante.
- Incluez les slots SEO (`page_title`, `page_meta_description`) dans tout gabarit de type page.
- Incluez les balises Open Graph dans l'en-tête HTML pour le partage social.
- Incluez le helper hreflang si le site est multilingue.
- Testez le gabarit en créant une page réelle depuis l'assistant et vérifiez le rendu public.

Récapitulatif

Vous savez désormais :

- Comprendre l'utilité d'un gabarit (squelette, slots, valeurs par défaut).
- Créer un dossier `templates/<code>/` avec `meta.php` et `skeleton.tpl.php`.
- Renseigner les métadonnées (`code`, `label`, `category`, `icon`, `type_container`, `default_slug`).
- Utiliser les marqueurs `@@PAGEID@@`, `@@PAGEURL@@` et `@@IS02@@`.
- Localiser vos gabarits en dehors du module via `INFRASTUDIO_TEMPLATE_EXTRA_DIR`.
- Suivre les bonnes pratiques (préfixe, regroupement, valeurs par défaut, SEO, multilingue).

Le dernier chapitre de la Partie IV détaille le catalogue produit dynamique en profondeur.

Revision #2

Created 2026-05-07 11:37:08 UTC by Lucky Ranasolonirina

Updated 2026-05-08 07:44:27 UTC by Lucky Ranasolonirina