

Plugins

Voici à quoi ressemble la structure minimale d'un plugin:

```
monplugin/  
├─ compositions/  
│ ── article-pleine-page.html  
│ ── article-pleine-page.xml  
│ ── rubrique-actualites.xml  
│ ── rubrique-actualites.html  
├─ css/ # (surcharges css de ibootstrap)  
│ ── plugins/  
│ │ ── _owl_carousel.scss  
│ ── config_variables.scss.html # (fichier pour récupérer des variables SPIP avec la syntaxe SPIP)  
│ ── fonts.css  
│ ── main.scss  
│ ── *.scss # (fichiers SCSS du thème à intégrer avec @import dans main.scss)  
├─ fonts/  
├─ formulaires/  
│ ── configurer_monplugin.html  
├─ images/  
├─ inclure/  
│ ── menu.html  
│ ── share_social_links.html  
│ ── social_links.html  
├─ js/  
│ ── widgets/  
│ │ ── owl_carousel.js  
│ ── main.js # (indispensable)  
├─ lang/ # (indispensable)  
│ ── monplugin_fr.php  
│ ── paquet-monplugin_fr.php.bak  
├─ layouts/  
│ ── inc/ # (surcharges de ibootstrap)  
│ │ ── footer-defaut.html  
│ │ ── header.html  
│ │ ── navigation.html
```

```
| | └─ page-header.html
| | └─ page.html
└─ pages/ # (surcharges de pages existantes de ibootstrap)
| └─ sommaire/
| | └─ inclure/
| | | └─ about.html
| | | └─ contact.html
| | | └─ header.html
| | | └─ news.html
| | | └─ services.html
| | └─ content.html
| | └─ aside.html
| | └─ pre-content.html
└─ prive/
| └─ squelettes/
| | └─ contenu/
| | | └─ configurer_ibootstrap.html # (surcharge de ibootstrap)
| └─ themes/
| | └─ spip/
| | | └─ images/
| | | | └─ monplugin-16.png
| | | | └─ monplugin-24.png
| | | | └─ monplugin-32.png
| | | | └─ monplugin.png
└─ vendor/
| └─ owl.carousel/
| | └─ css/
| | | └─ owl.carousel.min.css
| | | └─ owl.theme.default.min.css
| | └─ js/
| | | └─ owl.carousel.min.js
└─ paquet.xml # (indispensable)
└─ monplugin_fonctions.php
└─ monplugin_pipelines.php
└─ favicon.ico
└─ imentionslegales.js.html # (pour les cookies)
```

Mais comme nous l'avons vu précédemment, il n'y a que le fichier `paquet.xml` et le dossier `lang` qui sont toujours indispensables.

Exemple de `paquet.xml`

```
<paquet
  prefix="monplugin"
  categorie="squelette"
  version="0.0.1"
  etat="dev"
  compatibilite="[4.0.0;["
>
  <nom>Mon plugin</nom>
  <auteur>Nom Prénom</auteur>
  <licence lien="http://www.gnu.org/licenses/gpl-3.0.html">GPL 3</licence>

  <!-- Pipelines utilisés -->
  <!-- Exemple insert_head implémenté dans le fichier monplugin_pipelines.php -->
  <pipeline nom="insert_head" inclure="monplugin_pipelines.php" />
  <pipeline nom="insert_head_css" inclure="monplugin_pipelines.php" />

  <!-- Dépendances du plugin -->
  <!-- Exemple plugin bootstrap version 6.0.0 ou plus -->
  <necessite nom="bootstrap" compatibilite="[6.0.0;[" />

</paquet>
```

Le `pipeline insert_head` permet d'ajouter des scripts dans le head des pages publiques du site.

Les compositions

Les **compositions** permettent de proposer plusieurs mises en page différentes pour un même type d'objet (article, rubrique, etc.).

Pour chaque composition, vous devez créer **deux fichiers** dans le répertoire `compositions/` :

1. **Un fichier HTML** : le squelette de la composition
2. **Un fichier XML** : le descriptif de la composition

Exemple : composition "Actualités" pour les rubriques

Fichier `compositions/rubrique-actualites.html` :

```
<BOUCLE_principale(RUBRIQUES) {id_rubrique}>
  <INCLUDE{fond=layouts/#CONFIG{ibootstrap/layout,layout},env}{bs_page=rubrique-actualites}{composition=#COMPOSITION} />
</BOUCLE_principale>
```

Fichier `compositions/rubrique-actualites.xml` :

```
<composition>
  <nom><:monplugin:rubrique_actualites:></nom>
  <description><:monplugin:rubrique_actualites_desc:></description>
  <icon>images/rubrique-actualites.png</icon>
</composition>
```

Une fois les fichiers créés, la composition apparaît dans l'interface d'édition de l'objet concerné (article, rubrique...).

Les rédacteurs peuvent alors choisir la composition souhaitée dans le menu déroulant "Composition".

Surcharge des squelettes de pages

SPIP utilise un système de **hiérarchie de fichiers** pour déterminer quel squelette afficher.

Une surcharge sert à personnaliser une page existante (d'un plugin parent comme `ibootstrap`), créez un fichier avec le **même nom** dans votre plugin. SPIP utilisera automatiquement votre version.

Hiérarchie de recherche des squelettes

SPIP cherche les squelettes dans cet ordre :

1. **Répertoire** `squelettes/` (à la racine de SPIP)
2. **Plugins activés** (ordre de priorité des plugins)
3. **Répertoire** `dist/` (squelettes par défaut de SPIP)

Exemples de surcharges courantes

Surcharge de la page d'accueil:

Créez `sommaire.html` à la racine de votre plugin ou dans `squelettes/`, pour ajouter une variable d'env `pleine_page` :

```
<INCLUDE{fond=layouts/#CONFIG{ibootstrap/layout,layout},env}{bs_page=sommaire}{breadcrumb=non}{pl  
eine_page=oui}  
/>
```

Surcharge de la page auteur.html pour la désactiver:

Créez `auteur.html` et laissez-le vide.

Surcharge partielle avec `layouts/`:

Pour surcharger uniquement des **fragments** de mise en page (header, footer, navigation...), utilisez le répertoire `layouts/inc/`. Par exemple dans le fichier `layouts/inc/header.html` (créez le si n'existe pas).

Collez le code suivant:

```
<INCLUDE{fond=layouts/inc/navigation,env} />  
[(#ENV{bs_page}|=={sommaire}|non)<INCLUDE{fond=layouts/header-objet, env} />]
```

Cette structure permet de surcharger seulement le header tout en conservant la structure principale du plugin parent.

Ordre de priorité avec iBootstrap

Si vous utilisez le plugin `ibootstrap`, voici comment SPIP recherche les fichiers :

1. `monplugin/layouts/inc/header.html` (votre surcharge)
2. `ibootstrap/layouts/inc/header.html` (version du plugin)
3. `dist/header.html` (version par défaut)

Pipelines

Les pipelines sont des points d'entrée prévus par SPIP qui permettent aux plugins d'intervenir à différents moments du fonctionnement du site. Ils offrent la possibilité d'ajouter, modifier ou filtrer du code et du contenu, que ce soit lors de l'affichage des pages, du traitement des formulaires ou d'autres étapes du cycle de vie de SPIP.

Exemple de fichier `monplugin_pipelines.php`

```
<?php  
  
if (!defined('_Ecrire_inc_version')) {  
    return;
```

```
}  
  
function monplugin_insert_head($flux) {  
    $flux .= "<script src=\"" . find_in_path('/js/widgets/owl-carousel.js') . "\" type='text/javascript'  
defer></script>\n";  
    return $flux;  
}
```

Revision #6

Created 24 November 2025 18:20:12

Updated 8 December 2025 12:44:18