



# Développement d'extensions de console



<https://pandorafms.com/manual/!775/>

Permanent link:

[https://pandorafms.com/manual/!775/fr/documentation/pandorafms/technical\\_reference/07\\_anexo\\_console\\_extensions\\_development](https://pandorafms.com/manual/!775/fr/documentation/pandorafms/technical_reference/07_anexo_console_extensions_development)

2018/03/18 21:03



# Développement d'extensions de console

## Développement d'extensions de console

Les extensions sont un moyen de développer de nouvelles fonctionnalités pour votre console Pandora FMS en tant que *plugins*.

Dans ce chapitre, vous apprendrez à développer une extension.

### Types d'extensions

Une extension peut être un (ou plusieurs) des types suivants :

- Onglet Agent: Extensions qui apparaissent dans l'en-tête de tabulation de la vue d'opération et/ou d'édition de l'agent.
- Visibles: Extensions qui apparaissent dans le menu Pandora FMS.
- Invisibles: Extensions chargées et exécutées dans le `index.php` du menu Pandora FMS mais qui n'apparaissent pas dans le menu PFMS.

### Répertoire des extensions

Le répertoire des extensions est un sous-répertoire de l'installation locale de la Console Web Pandora FMS dont le nom est `extensions`.

```
euclides root ~ ls -la /var/www/html/pandora_console/extensions/
total 244
drwxr-xr-x. 11 apache apache 4096 Jun 18 15:43 .
drwxr-xr-x. 20 apache apache 4096 Jun 27 19:55 ..
-rwxr-xr-x. 1 apache apache 2668 May 24 15:33 agents_alerts.php
drwxr-xr-x. 2 apache apache 27 Jun 18 15:43 agents_modules
-rwxr-xr-x. 1 apache apache 37475 May 24 15:33 agents_modules.php
-rwxr-xr-x. 1 apache apache 7489 May 24 15:33 api_checker.php
drwxr-xr-x. 2 apache apache 43 Jun 18 15:43 dbmanager
-rwxr-xr-x. 1 apache apache 8985 May 24 15:33 dbmanager.php
-rwxr-xr-x. 1 apache apache 15898 May 24 15:33 db_status.php
drwxr-xr-x. 2 apache apache 31 Jun 18 15:43 disabled
-rwxr-xr-x. 1 apache apache 3180 May 24 15:33 extension_uploader.php
drwxr-xr-x. 3 apache apache 134 Jun 18 15:43 files_repo
-rwxr-xr-x. 1 apache apache 8240 May 24 15:33 files_repo.php
-rwxr-xr-x. 1 apache apache 9827 May 24 15:33 insert_data.php
drwxr-xr-x. 2 apache apache 22 Jun 18 15:43 matrix_events
drwxr-xr-x. 2 apache apache 27 Jun 18 15:43 module_groups
-rwxr-xr-x. 1 apache apache 17757 May 24 15:33 module_groups.php
drwxr-xr-x. 2 apache apache 25 Jun 18 15:43 net_tools
-rwxr-xr-x. 1 apache apache 3698 May 24 15:33 pandora_logs.php
-rwxr-xr-x. 1 apache apache 15928 May 24 15:33 quick_shell.php
drwxr-xr-x. 2 apache apache 75 Jun 18 15:43 realtime_graphs
-rwxr-xr-x. 1 apache apache 8708 May 24 15:33 realtime_graphs.php
-rwxr-xr-x. 1 apache apache 16332 May 24 15:33 resource_exportation.php
-rwxr-xr-x. 1 apache apache 47092 May 24 15:33 resource_registration.php
-rwxr-xr-x. 1 apache apache 3030 May 24 15:33 sample_agent.php
drwxr-xr-x. 2 apache apache 22 Jun 18 15:43 users
-rwxr-xr-x. 1 apache apache 7648 May 24 15:33 users_connected.php
euclides root ~
```

Ce répertoire contient les éléments suivants pour chaque extension :

Fichier principal de l'extension

Ce fichier contient le code à charger dans la Console Web Pandora FMS.

Sous-répertoire de l'extension

Il est facultatif et peut contenir le fichier image de l'icône (18 x 18 pixels) affiché à côté du nom de l'extension dans le menu, ainsi que d'autres fichiers tels que des traductions, des modules et des images.

## Structure d'extension

```
<?php
```

```

<Comments with license, author/s, etc...>

<php auxiliary code as functions, variables, classes that your extension use>

function <name of main function> () {
    <Main function Code>
}

/*-----*/

/* Adds the link in the operation menu */
extensions_add_operation_menu_option (<'Name Extension'>, '<father ID menu'>,
'<relative path Icon'>');

/* Adds the link in the godmode menu */
extensions_add_godmode_menu_option (<'Name Extension'>, '<ACL level'>', '<father
ID menu'>', '<relative path Icon'>)

/*-----*/

/* Sets the callback function to be called when the extension is selected in the
operation menu */
extensions_add_main_function (<'name of main function'>);

/* Sets the callback function to be called when the extension is selected in the
godmode menu */
extensions_add_godmode_function(<'name of godmode function'>);
?>

```

## API pour les extensions

L'API pour les extensions est encore en cours de développement et pourrait changer à l'avenir.

Les sections suivantes contiennent la description des fonctions de l'API pour les extensions.

### extensions\_add\_operation\_menu\_option

```
extensions_add_operation_menu_option ('', '', '')
```


Cette fonction ajoute le lien vers l'extension avec le nom donné dans le menu Opérations. Le troisième paramètre est facultatif et il s'agit du champ facultatif pour que l'image de l'icône (18 x 18 pixels) apparaisse à côté du lien.

Si ce dernier paramètre n'est pas défini, une icône d'écrou de bougie () est utilisée.

### **extensions\_add\_godmode\_menu\_option**

```
extensions_add_godmode_menu_option ('', '', '', '')
```

Cette fonction ajoute le lien vers l'extension avec le nom donné dans le menu Godmode si l'utilisateur a le niveau ACL requis comme indiqué dans le deuxième paramètre. Le quatrième paramètre est facultatif et il s'agit du champ facultatif pour que l'image de l'icône (18 x 18 pixels) apparaisse à côté du lien.

Si ce dernier paramètre n'est pas défini, une icône d'écrou de bougie () est utilisée.

### **extensions\_add\_main\_function**

```
extensions_add_main_function ('')
```

Il définit la fonction de *callback* qui sera renvoyée lorsque l'utilisateur cliquera sur le lien dans l'extension du menu des opérations.

### **extensions\_add\_godmode\_function**

```
extensions_add_godmode_function ('')
```

Il ajoute la fonction d'extension à appeler une fois que l'utilisateur accède à l'extension en godmode de la Console Web Pandora FMS au lieu de charger la fonction principale.

### **extensions\_add\_login\_function**

```
extensions_add_login_function ('')
```

Il ajoute la fonction d'extension à appeler une fois que l'utilisateur s'est connecté correctement à la Console Web Pandora FMS.

### **extensions\_add\_godmode\_tab\_agent**

```
extensions_add_godmode_tab_agent('', '', '', '')
```

Il ajoute un onglet supplémentaire à la vue de modification de l'agent afin que, lorsque cette option est sélectionnée, le code de la fonction du nom qui lui est transmis soit exécuté.

## extensions\_add\_opemode\_tab\_agent

```
extensions_add_opemode_tab_agent('', '', '{{false?nolink&}}', '')
```

Il ajoute un onglet supplémentaire à la vue d'opération de l'agent afin que, lorsque cette option est sélectionnée, le code de la fonction du nom qui lui est transmis soit exécuté.

## extensions\_add\_translation\_string\_function

```
extensions_add_translation_string_function('')
```

Il ajoute la fonction de *callback* qui sera appelée à partir de la fonction de traduction d'extension standard.

## Father IDs dans le menu

Liste des ID de chaînes disponibles pour son utilisation dans l'extension API. Si vous utilisez une valeur null ou n'incluez pas de paramètre dans la fonction `call`, l'extension apparaît uniquement dans le sous-menu de l'extension.

### Opération

- estado: Vue de surveillance.
- network: Vue réseau.
- reporting: Rapports et visualisation des données.
- gismaps: Vue SIG.
- événements: Vue des événements.
- workspace: Espace de travail utilisateur.

### Administration

- gagente: Gestion de la supervision.
- gmassive: Opérations massives.
- gmodules: Gestion des modules.
- galertas: Gestion des alertes.
- gusuarios: Gestion des utilisateurs.
- godgismaps: Gestion SIG.
- gserver: Gestion des serveurs.
- glog: Logs du système.
- gsetup: Configuration.
- gdbman: Maintenance de la base de données.

## Administration Enterprise

Ces éléments ne sont disponibles qu'avec la version Enterprise.

- **gpolicies**: Gestion des politiques

## Exemple

Monitoring / Views  
Combined table of agent group and module group

> Filters

Groups	General	Networking	Application	System	Miscellaneous	Performance	Database	Enviromental	Users	Nothing
Applications	0	0	0	0	0	0	0	0	0	0
Databases	0	0	0	0	0	0	0	0	0	0
Firewalls	0	0	0	0	0	0	0	0	0	0
Network	0	0	0	0	0	0	0	0	0	0
Servers	0	2	0	8	0	0	0	0	0	0
Unknown	0	0	0	32	0	32	0	0	0	7
Web	0	0	0	0	0	0	0	0	0	0
Workstations	0	0	0	0	0	0	0	0	0	0

> Legend

L'extension affiche un tableau où les colonnes sont des groupes de modules et les lignes sont des groupes d'agents. Chaque cellule a une couleur avec les significations suivantes :

- Vert : Lorsque tous les modules du groupe sont OK.
- Jaune : Lorsqu'il y a au moins un moniteur en alerte.
- Rouge : Lorsqu'au moins un moniteur tombe en panne.

Et cette extension se bloque du menu Opération sous Agents.

## Code source

```
<?php
/**
 * Pandora FMS- http://pandorafms.com
```



```
* =====
* Copyright (c) 2005-2009 Artica Soluciones Tecnologicas
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU General Public License
* as published by the Free Software Foundation for version 2.
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*/

/**
 * Extension specific translation function
 */
function extensions_translation() {
    $translates = array(
        'es' => array(
            "sentence" => "translated sentence"
        ),
        /*
        ... for other language's definitions.
        */
    );

    $args = func_get_args();
    $string = array_shift($args);
    $user_language = get_user_language ();
    if(isset($translates[$user_language][$string])){
        return vsprintf($translates[$user_language][$string], $args);
    }
    else{
        return false;
    }
}

/**
 * Translate the array texts using gettext
 */
function translate(&$item, $key) {
    $item = __($item);
}

/**
 * The main function of module groups and the enter point to
 * execute the code.
 */
function mainModuleGroups() {
    global $config; //the useful global var of Pandora Console, it has many data
    can you use

    //The big query
```

```

$sql = "select COUNT(id_agente) AS count, estado
FROM tagente_estado
WHERE utimestamp != 0 AND id_agente IN
      (SELECT id_agente FROM tagente WHERE id_grupo = %d AND disabled IS
FALSE)
      AND id_agente_modulo IN
      (SELECT id_agente_modulo
FROM tagente_modulo
WHERE id_module_group = %d AND disabled IS FALSE AND
delete_pending IS FALSE)
GROUP BY estado";

echo "<h1>" . ____("Combine table of agent group and module group") .
"</h1>";

echo "<p>" . ____("This table show in columns the modules group and for rows
agents group. The cell show all modules") . "</p>";

$agentGroups = get_user_groups ($config['id_user']);
$modelGroups = get_all_model_groups();
array_walk($modelGroups, 'translate'); //Translate all head titles to
language is set

$head = $modelGroups;
array_unshift($head, ' ');

//Metaobject use in print_table
$table = null;
$table->align[0] = 'right'; //Align to right the first column.
$table->style[0] = 'color: #ffffff; background-color: #778866; font-weight:
bolder;';
$table->head = $head;

//The content of table
$tableData = array();
//Create rows and celds
foreach ($agentGroups as $idAgentGroup => $name) {

    $row = array();

    array_push($row, $name);

    foreach ($modelGroups as $idModelGroup => $modelGroup) {
        $query = sprintf($sql,$idAgentGroup, $idModelGroup);
        $rowsDB = get_db_all_rows_sql ($query);

        $states = array();
        if ($rowsDB !== false) {
            foreach ($rowsDB as $rowDB) {
                $states[$rowDB['estado']] = $rowDB['count'];
            }
        }
    }
}

```

```

    }

    $count = 0;
    foreach ($states as $idState => $state) {
        $count = $state;
    }

    $color = 'transparent'; //Default color for cell
    if ($count == 0) {
        $color = '#babdb6'; //Grey when the cell for this model group
and agent group hasn't modules.
        $alinkStart = '';
        $alinkEnd = '';
    }
    else {
        $alinkStart = '<a
href="index.php?sec=estado&sec2=operation/agentes/status_monitor&status=-1&ag_group=' . $idAgentGroup .
        '&modulegroup=' . $idModelGroup . '">';
        $alinkEnd = '</a>';

        if (array_key_exists(0,$states) && (count($states) == 1))
            $color = '#8ae234'; //Green when the cell for this model
group and agent has OK state all modules.
        else {
            if (array_key_exists(1,$states))
                $color = '#cc0000'; //Red when the cell for this model
group and agent has at least one module in critical state and the rest in any
state.

            else
                $color = '#fce94f'; //Yellow when the cell for this
model group and agent has at least one in warning state and the rest in green
state.

        }
    }

    array_push($row,
        '<div
            style="background: ' . $color . ' ;
            height: 15px;
            margin-left: auto; margin-right: auto;
            text-align: center; padding-top: 5px;">
            ' . $alinkStart . $count . ' modules' . $alinkEnd .
        '</div>');
    }
    array_push($tableData,$row);
}
$table->data = $tableData;

print_table($table);

echo "<p>" . ___("The colours meaning:") .

```

```

        "<ul>" .
        '<li style="clear: both;">
            <div style="float: left; background: #babdb6; height: 20px; width:
80px;margin-right: 5px; margin-bottom: 5px;"> </div>' .
            ___("Grey when the cell for this model group and agent group hasn't
modules.") . "</li>" .
        '<li style="clear: both;">
            <div style="float: left; background: #8ae234; height: 20px; width:
80px;margin-right: 5px; margin-bottom: 5px;"> </div>' .
            ___("Green when the cell for this model group and agent has OK state
all modules.") . "</li>" .
        '<li style="clear: both;"><div style="float: left; background: #cc0000;
height: 20px; width: 80px;margin-right: 5px; margin-bottom: 5px;"> </div>' .
            ___("Red when the cell for this model group and agent has at least
one module in critical state and the rest in any state.") . "</li>" .
        '<li style="clear: both;"><div style="float: left; background: #fce94f;
height: 20px; width: 80px;margin-right: 5px; margin-bottom: 5px;"> </div>' .
            ___("Yellow when the cell for this model group and agent has at
least one in warning state and the rest in green state.") . "</li>" .
        "</ul>" .
        "</p>";
}

extensions_add_operation_menu_option("Modules groups", 'estado',
'module_groups/icon_menu.png');
extensions_add_main_function('mainModuleGroups');
extensions_add_translation_string_function('extensions_translation');
?>

```

## Explication

Dans le code source, il y a deux parties :

- Le code source de l'extension.
- Les fonctions d'appel API.

L'ordre des parties est indifférent, mais il est préférable de placer les “ fonctions d'appel API ” sous votre fichier d'extension principal, car les guides de style avertissent que cette partie doit être placée en dessous pour que toutes les extensions aient à peu près le même style.

## Code source d'extension

Dans le cas de cet exemple, il y a deux fonctions dans le même fichier, mais si vous avez un code plus complexe, il est préférable de le diviser en plusieurs fichiers (et de l'enregistrer dans une extension de sous-répertoire). Il s'agit des fonctions suivantes :

- Fonction `extensions_translation()`  
Fonction spécifique pour l'extension des traductions. Elle est appelée à partir de la fonction principale

de traduction.

- Fonction `translate(&$item, $key)`

L'utilisation de cette fonction pour callback dans `array_walk function`. Car la fonction principale conserve les titres des colonnes et les titres des lignes sélectionnées sans traductions

- Fonction `mainModuleGroups()`

C'est le noyau de l'extension et a un grand nombre de lignes. Tout le code n'est pas affiché, mais quelques parties importantes :

- La première est l'accès au `global var config`. Dans cette var, vous avez de nombreuses configurations et valeurs par défaut pour de nombreux éléments de la console Pandora FMS.
- Le second var est la requête dans `MySQL` dans une *chaîne*. Et le `%d` est le `format placeholders` est pour `Id Group` et `Id Module Group` ; ce sont des substituts pour la valeur dans `sprintf function`.
- Quelques echos pour imprimer le texte devant le tableau
- Extrait de deux sélections de la base de données avec une dimension avec l'index en tant qu' `id` et le contenu comme titre pour les colonnes (`Module groups`) et les lignes (`Agent group`).
- Il traduit les titres de la sélection du `Model Group`.
- Il stabilise le *meta-objet* rempli de lignes et imprime.
  - Devant les `foreach ~ loops`, il définit comme `$table` l'en-tête et les styles de la table.
  - La première *loop* est pour les lignes (chaque groupe d'agents).
    - La deuxième *loop* est pour les colonnes en rangée normale (chaque groupe de modules).
      - Ensuite, pour chaque cellule, il y a deux nombres, `id model group` et `id agent group`. Avec ces deux nombres, il interroge la base de données et obtient les fichiers.
      - Il traite la sélection résultante pour obtenir une autre sélection et l'index est une intégration de différents types d'état du moniteur et le contenu est un calcul du moniteur dans cet état.
      - Tout ce qui reste à faire est de remplir le contenu de la cellule en HTML. La façon de procéder est simple : si le calcul de tous les états est nul, le *background* pour `div` en CSS est gris. Si `$states[1] != 0` ou si au moins un moniteur est dans un état critique, alors le `div` aura une couleur rouge. Si la sélection n'a qu'une seule cellule et qu'elle est en mode normal, la couleur verte sera sur cette `div`. Dans les autres cas, le jaune sera la couleur pour `div`.
      - Ajouter un *lien* dans la cellule si le calcul est supérieur à 0.
      - Enregistrer la série sur `$table`, et commencer une autre itération de `foreach`.
- Imprimer le tableau.
- Imprimer la légende et d'autres notes au bas de la page.

## API call functions

Il y a peu de lignes de code. Les opérations sur ces lignes sont :

- Insérer l'extension dans le menu Pandora FMS.

Il apparaît comme :

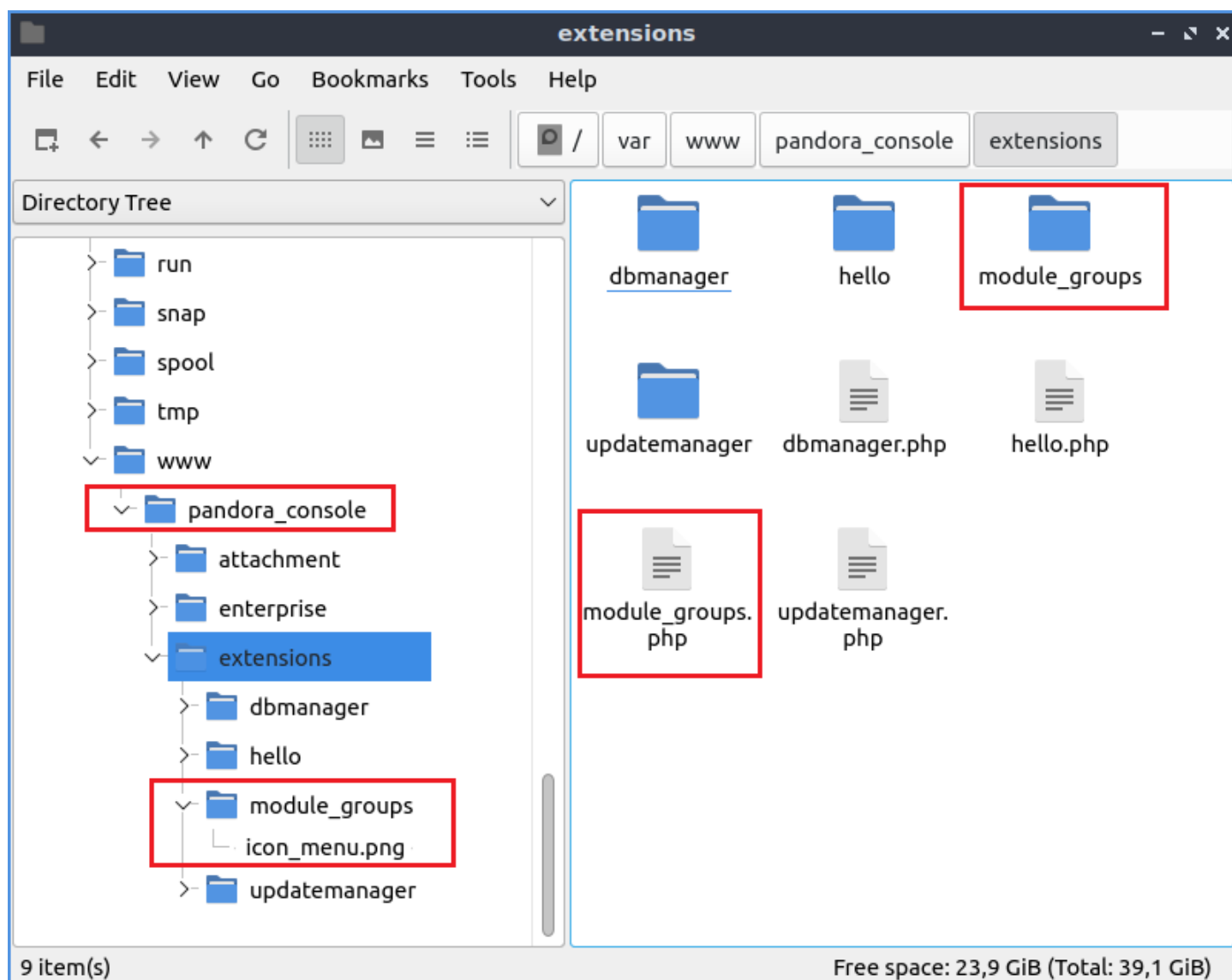
```
extensions_add_operation_menu_option("Modules groups", 'estado',
'module_groups/icon_menu.png');
```

Où :

- 'Modules groups' est le nom qui apparaît dans le sous-menu des agents.
- ' état' est l'élément qui pend de l'extension.
- 'module\_groups/icon\_menu.png' est l'icône de l'image qui apparaît dans le sous-menu, le combo est lié au répertoire d'extension que vous avez.
- Il définit la fonction principale de cette extension.  
Et il apparaît comme `extensions_add_main_function('mainModuleGroups')` ; où :
  - 'mainModuleGroups' est le nom de la fonction principale de l'extension.

L'ordre de désignation des fonctions est indifférent.

## Organisation du répertoire



L'installation de l'extension est très facile, car la console Pandora FMS recherche de nouvelles extensions et les ajoute au système lorsque vous les trouvez.

Il vous suffit de copier tous les fichiers d'extension dans l'extension de répertoire [de votre installation Pandora FMS Console](#). Mais vous devez installer les autorisations pour que la console Pandora FMS puisse lire les fichiers et les sous-répertoires d'extension.

Dans l'exemple d'image, l'extension a cette structure de répertoire :

- `module_groups`
  - `icon_menu.png`
- `module_groups.php`

Et l'extension du répertoire est par exemple dans `/var/www/pandora_console`.

### **Sous-répertoire**

Dans ce cas, l'exemple a un sous-répertoire et généralement toute extension doit avoir un sous-répertoire. Le sous-répertoire a le même nom que le nom de l'extension et le fichier principal. Le sous-répertoire de l'exemple n'a qu'un seul fichier d'image d'icône (`icon_menu.png`). Cette icône apparaît dans le menu Pandora FMS.

[Retour à l'index de la documentation du SGF Pandora](#)