



Développement d'extensions de console



<https://pandorafms.com/manual/!778/>

Permanent link:

https://pandorafms.com/manual/!778/fr/documentation/pandorafms/technical_reference/07_anexo_console_extensions_development

12/03 19:30





Développement d'extensions de console

Introduction

Les extensions permettent de développer de nouvelles fonctionnalités pour la console Web de Pandora FMS sous la forme de *plugins*.

Types d'extensions

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

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

Répertoire des extensions

Le répertoire des extensions est un sous-répertoire de l'installation locale de la console Web Pandora FMS qui porte le nom `extensions`. Ce répertoire contient les éléments suivants pour chaque extension :

Fichier principal de l'extension

Ce fichier contient le code qui doit être chargé dans la console Web de Pandora FMS.

Sous-répertoire de l'extension

Il est facultatif et peut contenir le fichier image de l'icône (18 par 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 de l'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 peut changer à l'avenir.

extensions_add_operation_menu_option

```
extensions_add_operation_menu_option (<string name>, <father ID menu>,
<relative path Icon>)
```

Cette fonction ajoute le lien à l'extension dont le nom est indiqué dans le menu Opérations. Le troisième paramètre est facultatif et il s'agit du champ facultatif pour l'image de l'icône (18 par 18 pixels) qui apparaîtra à côté du lien.

Si ce dernier paramètre n'est pas défini, l'icône d'une fiche sera utilisée (📄).

extensions_add_godmode_menu_option

```
extensions_add_godmode_menu_option ('<Name Extension>', '<ACL level>' , '<father ID menu>', '<relative path Icon>')
```

Cette fonction ajoute le lien vers l'extension dont le nom est indiqué dans le menu Godmode si l'utilisateur a le niveau ACL requis, comme l'indique le deuxième paramètre. Le quatrième paramètre est facultatif et correspond au champ relatif de l'image de l'icône (18 x 18 pixels) qui doit apparaître à côté du lien.

Si ce dernier paramètre n'est pas défini, l'icône d'une fiche sera utilisée (📄).

extensions_add_main_function

```
extensions_add_main_function ('<name of main function>')
```

Il définit la fonction de callback à renvoyer lorsque l'utilisateur clique sur le lien d'extension du menu d'exploitation.

extensions_add_godmode_function

```
extensions_add_godmode_function ('<name of godmode function>')
```

Ajouter la fonction d'extension à appeler une fois que l'utilisateur va à l'extension dans le godmode de la console web Pandora FMS au lieu de charger la fonction principale.

extensions_add_login_function

```
extensions_add_login_function ('<name of 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('<ID of extension tab>', '<Name of extension tab>', '<Image file with relative dir>', '<Name of function to show content of godmode tab agent>')
```

Il ajoute un onglet supplémentaire à la vue d'édition de l'agent de sorte que lorsqu'il est sélectionné, le code de la fonction du nom qui lui est passé est exécuté.

extensions_add_opemode_tab_agent

```
extensions_add_opemode_tab_agent('<ID of extension tab>', '<Name of extension tab>', '<Image file with relative dir>', '<Name of function to show content of operation tab agent>')
```

Il ajoute un onglet supplémentaire à la vue de l'opération de l'agent qui provoque l'exécution du code de la fonction du nom passé lorsqu'il est sélectionné.

extensions_add_translation_string_function

```
extensions_add_translation_string_function('<Name of translation function>')
```

Il ajoute la fonction de callback à appeler à partir de la fonction de traduction de l'extension standard.

Menu Father IDs

Liste des IP strings pouvant être utilisées dans l'API d'extension. Si vous utilisez une valeur nulle ou n'incluez pas de paramètre dans la fonction `call`, l'extension n'apparaît que dans le sous-menu d'extension.

Operation

- `estado`: Vue de supervision.
- `network`: Vue du réseau.
- `reporting`: Rapports et visualisation des données.
- `gismaps`: Vue SIG.
- `eventos`: Vue de l'événement.
- `workspace`: Espace de travail de l'utilisateur.

Administration

- gagente : Suivi de la gestion.
- gmassive : Opérations de masse.
- gmodules : Gestion des modules.
- galertas : Gestion des alertes.
- gusuarios : Gestion des utilisateurs.
- godgismaps : Gestion des SIG.
- gserver : Gestion du serveur.
- glog : Logs du système.
- gsetup : Configuration.
- gdbman : Maintenance de la base de données.
- 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 dont les colonnes sont des groupes de modules et les lignes des groupes d'agents. Chaque cellule a une couleur avec les significations suivantes :

- Vert : Lorsque tous les modules du groupe sont OK.

- Jaune : Lorsqu'au moins un moniteur est en alerte.
- Rouge : Lorsqu'au moins un moniteur est défaillant.

Cette extension se trouve dans le menu Opération sous Agents.

Code source

```
<?php
/**
 * Pandora FMS - http://pandorafms.com
 * =====
 * Copyright (c) 2005-2024 PFMS
 *
 * 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

Le code source se compose de deux parties :

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

L'ordre des parties est indifférent, il est juste préférable de placer le « calling API functions » en

dessous de votre fichier d'extension principal car les guides de style conseillent de placer cette partie en dessous afin que toutes les extensions aient approximativement le même style.

Code source de l'extension

Dans le cas de cet exemple, il y a deux fonctions dans le même fichier. Si vous aviez un code plus complexe, il serait préférable de le diviser en plusieurs fichiers (et de l'enregistrer dans une extension de sous-répertoire). Les fonctions sont les 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, $key)` L'utilisation de cette fonction pour callback en [array_walk function](#), car la fonction principale conserve les en-têtes de colonne et de ligne sélectionnés non traduits.
- Fonction `mainModuleGroups()` C'est le cœur de l'extension et il comporte un grand nombre de lignes. Quelques éléments importants :
 - La première est l'accès à la `var config` globale. Ce `var` contient de nombreuses configurations et valeurs par défaut pour de nombreux éléments de la console Web de Pandora FMS.
 - Le second `var` est la requête MySQL dans un fichier `string`. Le `%d` est le format placeholder pour `Id Group` et `Id Module Group`. Ce sont les substituts de la valeur en [sprintf function](#).
 - Une commande `echo` pour imprimer le texte devant le tableau.
 - Extraire deux sélections de la base de données à une dimension avec l'index comme `id` et le contenu comme le titre de la colonne (`Module groups`) et des lignes (`Agent group`).
 - Traduire les titres de la sélection `Module Group`.
 - Rendre le meta-object stable et le remplit de lignes et d'impressions.
 - Devant les boucles `for each ~ loop`, définir comme `$table` l'en-tête et les styles du tableau.
 - La première cycle (loop) concerne les lignes (chaque groupe d'agents).
 - Le deuxième cycle concerne les colonnes de lignes normales (chaque groupe de modules).
 - Ensuite, pour chaque cellule, il y a deux nombres, `id module group` et `id agent group`. Ces deux numéros lui permettent d'interroger la base de données et d'obtenir 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.
 - La dernière chose à faire est de remplir le contenu de la cellule en HTML : Si le nombre de tous les états est égal à zéro, la couleur d'arrière-plan de `div` dans le CSS est le gris. Si `$states[1] != 0` ou s'il y a au moins un moniteur en état critique, la `div` aura une couleur rouge. Si la sélection ne comporte qu'une seule cellule et qu'elle est en mode normal, la couleur de cette `div` sera le vert. Dans tous les autres cas, le jaune sera la couleur de la `div`.
 - Ajouter un lien dans la cellule si le compte est supérieur à 0.
 - Enregistrer la série dans `$table`, et commencez une autre itération de `foreach`.
 - Imprimer le tableau.
 - Imprimer la légende et les autres notes en bas de page.

API calls functions

- Insertion de l'extension dans le menu de Pandora FMS :

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

- `Modules groups` est le nom qui apparaît dans le sous-menu de l'agent.
- `estado` est l'élément suspendu à l'extension.
- `module_groups/icon_menu.png` est l'icône de l'image qui apparaît dans le sous-menu, le champ est lié au répertoire d'extensions dont vous disposez.
- Définir la fonction principale de cette extension. Elle apparaît comme `extensions_add_main_function('mainModuleGroups');` où `mainModuleGroups` est le nom de la fonction principale de l'extension.

L'ordre dans lequel les fonctions sont nommées n'est pas pertinent.

Organisation du conseil d'administration

L'extension a la structure de répertoire suivante :

```
+module_groups  
|  
+- - icon_menu.png  
|  
+module_groups.php
```

L'extension du répertoire est dans `/var/www/pandora_console`.

Vous ne devez copier tous les fichiers d'extension que dans le répertoire d'extension d'[installation de la console Pandora FMS](#). Vous devez installer les permissions pour que la console Pandora FMS puisse lire les fichiers et les sous-répertoires de l'extension. La console Pandora FMS recherche de nouvelles extensions et les ajoute au système lorsqu'elle les trouve.

Sous-répertoire

L'exemple n'a qu'un seul sous-répertoire (et en général, toute extension devrait avoir un sous-répertoire). Le sous-répertoire porte le même nom que le nom de l'extension et le fichier principal. Le sous-répertoire de l'exemple ne contient qu'un seul fichier d'image d'icône (`icon_menu.png`). Cette icône apparaîtra dans le menu Pandora FMS.

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