



Desarrollo de extensiones de consola



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

parent link:

https://pandorafms.com/manual/!current/es/documentation/pandorafms/technical_reference/07_anexo_console_extensions_development

2020/06/10 14:36



Desarrollo de extensiones de consola

Desarrollo de extensiones de consola

Las extensiones son una manera de desarrollar una nueva funcionalidad para su Pandora FMS Console como *plugins*. En este capítulo se explica cómo desarrollar una extensión.

Tipos de Extensiones

Una extensión puede ser alguno (o varios) de los siguientes tipos:

- Pestaña agente: Extensiones que aparecen en la cabecera de pestañas de la vista de operación y/o edición del agente.
- Visibles: Extensiones que aparecen en el Menú de Pandora FMS.
- Invisibles: Extensiones que se cargan y ejecutan en el `index.php` del Menú de Pandora FMS pero que no aparecen en el Menú de PFMS.

Directorio de Extensiones

El directorio de extensiones es un subdirectorio dentro de la instalación local de la Consola web de Pandora FMS que tiene el nombre de `extensions`. Este directorio contiene lo siguiente para cada extensión:

Fichero principal de la extensión

Este fichero tiene el código que se ha de cargar en la Consola web de Pandora FMS.

Subdirectorio de extensión

Es opcional y puede contener el fichero de imagen del icono (18 por 18 pixeles) que se muestra al lado del nombre de la extensión en el menú, así como otros ficheros como traducciones, módulos e imágenes.

Estructura de la extensión

```
<?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 para Extensiones

El API para extensiones está todavía en desarrollo y puede cambiar en el futuro.

Las siguientes secciones contienen la descripción de las funciones en el API para extensiones.


extensions_add_operation_menu_option

```

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

```


Esta función añade el enlace a la extensión con el nombre dado en el menú Operaciones. El tercer parámetro es opcional y este es el campo opcional para que la imagen de icono (18 por 18 pixeles) aparezca al lado del enlace.

Si este último parámetro no está definido, se utilizará el icono de un enchufe ()

extensions_add_godmode_menu_option

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

Esta función añade el enlace a la extensión con el nombre dado en el menú Godmode si el usuario tiene el nivel de ACL requerido tal y como indica el segundo parámetro. El cuarto parámetro es opcional y es el campo relativo para que a la imagen del icono (18 por 18 pixelse) aparezca al lado del enlace.

Si este último parámetro no está definido, se utilizará el icono de un enchufe ().

extensions_add_main_function

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

Fija la función *callback* que devolverá cuando el usuario haga clic en el enlace de la extensión del menú operación.

extensions_add_godmode_function

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

Añade la función de extensión para llamar una vez que el usuario vaya a la extensión en el godmode de la Consola web de Pandora FMS en lugar de cargar la función principal.

extensions_add_login_function

```
extensions_add_login_function ('<name of login function>')
```

Añade la función de extensión para llamar una vez que el usuario inicie sesión correctamente en la Consola web de 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>')
```

Añade una pestaña más a la vista de edición del agente para que cuando se seleccione se ejecute el código de la función del nombre que se le pasa.

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>')
```

Añade una pestaña más a la vista de operación del agente que hace que cuando se selecciona se ejecute el código de la función del nombre que se le pasa.

extensions_add_translation_string_function

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

Añade la función de *callback* que será llamada desde la función estándar de traducción de extensiones.

Father IDs en el menú

Lista de strings IDs disponibles para usar en extensión API. Si utiliza un valor nulo o no incluye parámetro en la función `call`, entonces la extensión aparece únicamente en el submenú de extensión

Operación

- estado: Vista de monitorización.
- network: Vista de redes.
- reporting: Informes y visualización de datos.
- gismaps: Vista GIS.
- eventos: Vista de eventos.
- workspace: Espacio de trabajo del usuario.

Administration

- gagent: Gestión de la monitorización.
- gmassive: Operaciones masivas.
- gmodules: Gestión de módulos.
- galertas: Gestión de alertas.
- gusuarios: Gestión de usuarios.
- godgismaps: Gestión GIS.
- gserver: Gestión de servidores.
- glog: *Logs* del sistema.
- gsetup: Configuración.
- gdbman: Mantenimiento BBDD.
- gpolicies: Gestión de políticas

Ejemplo

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

La extensión muestra una tabla donde las columnas son grupos de módulos y las filas grupos de agentes. Cada celda tiene un color con los siguientes significados:

- Verde: cuando todos los módulos del grupo están OK.
- Amarillo: cuando existe al menos un monitor en alerta.
- Rojo: cuando al menos un monitor falla.

Y esta extensión cuelga del menú Operación en Agentes.

Código fuente

```
<?php
```

```
/**
```

```
* Pandora FMS - http://pandorafms.com
```

```
* =====
```

```
* Copyright (c) 2005-2023 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');
?>

```

Explicación

En el código fuente existen dos partes:

- El código fuente de la extensión.
- Las funciones de llamada API.

El orden de las partes es indiferente, pero es mejor colocar las “funciones API de llamada” debajo de su fichero principal de extensión ya que las guías de estilo advierten que se debe situar esta parte debajo para que todas las extensiones tengan aproximadamente el mismo estilo.

Código fuente de extensión

En el caso de este ejemplo hay dos funciones en el mismo fichero, pero si tiene un código más complejo, entonces es mejor dividirlo en muchos ficheros (y guardarlo en extensión de subdirectorio). Las funciones son:

- Función `extensions_translation()`
Función específica para la extensión de traducciones. Se le llama desde la función principal de traducciones.
- Función `translate(&$item, $key)`
El uso de esta función para callback en [array_walk function](#). Debido a que la función principal mantiene los títulos de las columnas y los títulos de las filas seleccionados sin traducciones
- Función `mainModuleGroups()`

Es el núcleo de la extensión y tiene una enorme cantidad de líneas. No se visualiza todo el código, pero sí algunas partes importantes:

- La primera es el acceso al `global var config`. En este `var` tiene muchas configuraciones y valores por defecto para muchos elementos de la consola de Pandora FMS.
- El segundo `var` es el query en MySQL en un *string*. Y el `%d` es el format placeholders es para `Id Group` y `Id Module Group`; estos son sustitutos para el valor en *sprintf function*.
- Algunos echos para imprimir el texto delante de la tabla
- Extracto de dos selecciones de la BBDD con una dimensión con el índice como `id` y el contenido como título para columnas (`Module groups`) y filas (`Agent group`).
- Traduce los títulos de la selección del `Module Group`.
- Hace estable el *meta-object* rellena con filas e imprime.
 - Delante de los `foreach ~ loops`, define como `$table` la cabecera y estilos de la tabla.
 - El primer *loop* es para las filas (cada grupo de agentes).
 - El segundo *loop* es para las columnas en hilera normal (cada grupo de módulos).
 - Después, para cada celda, hay dos números, `id module group` e `id agent group`. Con estos dos números hace una *query* a la base de datos y obtiene los ficheros.
 - Procesa la selección resultante para obtener otra selección y el índice es una integración de diferentes tipos de estado de monitor y el contenido es un cómputo del monitor en ese estado.
 - Lo único que queda por hacer es rellenar el contenido de la celda en HTML. El modo de hacerlo es sencillo: si el cómputo de todos los estados es cero, el *background* para `div` en CSS es gris. Si `$states[1] != 0` o existe al menos un monitor en estado crítico, entonces el `div` tendrá un color rojo. Si la selección tiene sólo una celda y está en modo normal entonces el color verde estará en esta `div`. En los otros casos el amarillo será el color para `div`.
 - Añadir un *link* en la celda si el cómputo es mayor que 0.
 - Guardar la serie en `$table`, y empezar otra iteración de `foreach`.
- Imprimir la tabla.
- Imprimir la leyenda y otras notas en la parte inferior de la página.

API calls functions

Hay pocas líneas de código. Las operaciones en estas líneas son:

- Insertar la extensión en el menú de Pandora FMS.
Aparece como:

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

Donde:

- 'Modules groups' es el nombre que aparece en el submenú de agentes.
- 'estado' es el elemento que cuelga de la extensión.
- 'module_groups/icon_menu.png' es el icono de la imagen que aparece en el submenú, el campo está relacionado con el directorio de extensión que tenga.
- Define la función principal de esta extensión.
Y aparece como `extensions_add_main_function('mainModuleGroups');` donde:
 - 'mainModuleGroups' es el nombre de la función principal de la extensión.

El orden para nombrar las funciones es indiferente.

Organización del Directorio

La extensión tiene esta estructura de directorio:

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

Y la extensión del directorio está en `/var/www/pandora_console`.

Únicamente deberá copiar todos los ficheros de extensiones a la extensión del directorio en su [instalación de Pandora FMS Console](#). Debe instalar los permisos para que la consola de Pandora FMS pueda leer los ficheros y los subdirectorios de extensión. La consola de Pandora FMS busca nuevas extensiones y las añade al sistema cuando las encuentra.

Subdirectorio

En este caso, el ejemplo tiene un subdirectorio y generalmente cualquier extensión debe tener un subdirectorio. El subdirectorio tiene el mismo nombre que el nombre de la extensión y el fichero principal. El subdirectorio del ejemplo tiene sólo un fichero de imagen de icono (`icon_menu.png`). Este icono aparece en el Menú de Pandora FMS.

[Volver al Índice de Documentación Pandora FMS](#)