Documentation of the source code of the "ServerHealthCheck" plugin for GLPI

Plugin «ServerHealthcheck» is written for ITAM system GLPI mainly in PHP programming language, there are also inserts of HTML code for generating plugin pages and SQL for database calls. The plugin consists of six main files. Next, descriptions of the functions located in them will be given. Additionally, there are comments in the source code.

1. File «setup.php»

Using the code in this file, the plugin is initialized in the GLPI system.

1.1. plugin_init_serverhealthcheck – function performs several actions:

- Registering the main plugin class in GLPI
- • Adding the necessary tabs to the graphical interface of the system, as well as defining data sources for them;
- • Adding a new report type;
- • Adding a new widget class type.

This function takes no arguments and returns nothing.

1.2. plugin_version_serverhealthcheck – the function provides GLPI with information about the plugin: name, version, author name, distribution license option, plugin site address, minimum and maximum supported versions of the GLPI system itself. The function takes no arguments and returns an array with the values of the above fields.

2. File «hook.php»

This file defines the functions for plugin hooks required by GLPI by default, namely the functions called when accessing plugin installation and removal hooks..

2.1. plugin_serverhealthcheck_install – the function creates the main plugin table, enters servers from the glpi_computers table into it, creates the ServerHealthCheck privileged users group, adds the glpi user to the created group, creates the reports folder in the plugin root directory for future storage of reports. The function takes no arguments and returns true if all plugin installation operations were completed successfully, otherwise error messages will appear in the GLPI logs indicating an incorrectly performed operation.

2.2. plugin_serverhealthcheck_uninstall – the function deletes the main plugin table, deletes the "ServerHealthCheck" group and all records about users belonging to it in the "glpi_groups_users" table, deletes the "reports" directory and all files located in it. The function takes no arguments and returns true if all plugin removal operations were successful, otherwise GLPI logs will contain error messages indicating an incorrectly performed operation.

3. File«ServerHealthCheck.php»

Here, the main plugin class "ServerHealthCheck" is defined, which is a child of the "CommonDBTM" class, and it in turn inherits the "CommonGLPI" class. The first of them is responsible for working with databases, and the second is the central class of the GLPI itself. This class defines the functions that GLPI uses for the direct operation of the plugin.:

- getMenuName() – a function that returns a string specifying the name of the plugin tab on the user interface;
- canView() – determines whether the current authorized user can view the content of the main page of the plugin, returns a bool value.
- plugin_serverhealthcheck_getServerHealthData() – returns an array of all records in the plugin's main table;
- plugin_serverhealtthcheck_updateServerHealthData ($data) – updates the server data based on the information provided from the main page of the plugin in the $data variable via the POST method;
- plugin_serverhealthcheck_updateServerList() – updates the list of server systems in the plugin table based on data from the GLPI table named "glpi_computers";
- plugin_serverhealthcheck_getServers() – returns a list of servers in the plugin table;
- plugin_serverhealthcheck_updateServerState ($id, $state) – updates the "state" field for the server with the specified id in the plugin table;
- plugin_serverhealthcheck_gatherServerStates() – collects information about the status of servers using the ipmitool library and updates their status in the table;
- plugin_serverhealthcheck_healthReport($access) – generates an HTML page with a public or private version of the report, depending on the access level provided in the "access" variable;
- dashboardTypes() – defines a new widget type for the center dashboard;
- dashboardCards ($cards) – adds a new widget content type;
- cardWidget ($params) – defines the widget's content template;
- cardDataProvider ($params) – generates plugin widget content;

4. File «serverhealthcheck.php»

This file represents a blank of the main form of the plugin in HTML, and also has a PHP script inside that is responsible for filling the form with data and submitting it if one of the four buttons is pressed.

5. File «posthandler.php»

It accepts the data sent from the above form using the POST method, processes it and gives the result to the user. For example, if the save changes button was pressed, the program collects data from the form and updates their values in the plugin table, simultaneously checking the ip and password fields for validity. To check the first field, a regular expression is used, for the second field, the key condition is the length of the string, which, for security reasons, should not be less than eight characters. For the update and information collection buttons, the corresponding actions are performed and the result on the success or failure of their execution is displayed on the new page. If you click the report display button, a private version of the report is generated for the administrator of the organization's IT infrastructure. For each of these actions, a check of the user's privileges to perform these actions is also implemented, which prevents unauthorized access to the plugin functions by sending POST requests to the address of the event handler.

6. File «report.php»

This file provides blank of HTML page that provides a public version of the report on the Tools->Reports tab. The plugin_serverhealthcheck_healthReport function called from the main class with the "public" string argument is responsible for the page content generation.