

# Документация исходного кода плагина «ServerHealthCheck» для GLPI

Плагин «ServerHealthcheck» написан для ITAM системы GLPI преимущественно на языке программирования PHP, присутствуют также вставки кода на HTML для генерации страниц плагина и SQL для обращений к базе данных. Плагин состоит из шести основных файлов. Далее будут приведены описания функций, расположенных в них. Дополнительно в исходном коде указаны комментарии.

## 1. Файл «setup.php»

При помощи кода, находящегося в данном файле, происходит инициализация плагина в системе GLPI.

1.1. `plugin_init_serverhealthcheck` – функция выполняет несколько действий:

- Регистрация основного класса плагина в GLPI
- Добавление необходимых вкладок на графический интерфейс системы, а также определение источников данных для них;
- Добавление нового типа отчета;
- Добавление нового типа класса виджетов.

Данная функция не принимает аргументов и ничего не возвращает.

1.2. `plugin_version_serverhealthcheck` – функция предоставляет GLPI данные о плагине: название, версия, имя автора, вариант лицензии распространения, адрес сайта плагина, минимальная и максимальная поддерживаемые версии самой системы GLPI. Функция не принимает аргументов и возвращает массив с значениями вышеописанных полей.

## 2. Файл «hook.php»

В данном файле определены функции для хуков плагина, требуемых GLPI по умолчанию, а именно функции вызываемые при обращении к хукам установки и удаления плагина.

- 2.1. `plugin_serverhealthcheck_install` – функция создает основную таблицу плагина, заносит в нее сервера из таблицы «`glpi_computers`», создает группу привилегированных пользователей «`ServerHealthCheck`», добавляет в созданную группу пользователя «`glpi`», создает папку «`reports`» в корневой директории плагина для будущего хранения отчетов. Функция не принимает аргументов и возвращает `true` в случае успешности выполнения всех операций установки плагина, в ином случае в логах GLPI появятся сообщения об ошибке с указанием некорректно выполненной операции.
- 2.2. `plugin_serverhealthcheck_uninstall` – функция удаляет основную таблицу плагина, удаляет группу «`ServerHealthCheck`» и все записи о принадлежности пользователей к ней в таблицу «`glpi_groups_users`», удаляет директорию «`reports`» и все файлы, находящиеся в ней. Функция не принимает аргументов и возвращает `true` в случае успешности выполнения всех операций удаления плагина, в ином случае в логах GLPI появятся сообщения об ошибке с указанием некорректно выполненной операции.

### 3. Файл «ServerHealthCheck.php»

Здесь определен основной класс плагина «ServerHealthCheck», который является дочерним для класса «CommonDBTM», а он в свою очередь наследует класс «CommonGLPI». Первый из них отвечает за работу с базами данных, а второй представляет собой центральный класс самого GLPI. В данном классе определены функции, которые GLPI использует для непосредственной работы плагина:

- getMenuName() – функция, которая возвращает строку, определяющую наименование вкладки плагина на пользовательском интерфейсе;
- canView() – определяет, может ли текущий авторизованный пользователь просматривать контент главной страницы плагина, возвращает значение типа bool.
- plugin\_serverhealthcheck\_getServerHealthData() – возвращает массив всех записей основной таблицы плагина;
- plugin\_serverhealthcheck\_updateServerHealthData (\$data) – обновляет данные о серверах на основе информации предоставленной с главной страницы плагина в переменной \$data через метод POST;
- plugin\_serverhealthcheck\_updateServerList() – обновляет список серверных систем в таблице плагина, основываясь на данных из таблицы GLPI с названием «glpi\_computers»;
- plugin\_serverhealthcheck\_getServers() – возвращает список серверов в таблице плагина;
- plugin\_serverhealthcheck\_updateServerState (\$id, \$state) – обновляет поле «state» для сервера с указанным id в таблице плагина;
- plugin\_serverhealthcheck\_gatherServerStates() – собирает информацию с использованием библиотеки ipmitool о состоянии серверов и обновляет их статус в таблице;
- plugin\_serverhealthcheck\_healthReport(\$access) – генерирует HTML страницу с публичной или приватной версией отчета в зависимости от предоставленного в переменной «access» уровня доступа;
- dashboardTypes() – определяет новый тип виджета для центральной панели;
- dashboardCards (\$cards) – добавляет новый тип контента виджета;
- cardWidget (\$params) – определяет шаблон содержимого виджета;
- cardDataProvider (\$params) – генерирует контент виджета плагина;

### 4. Файл «serverhealthcheck.php»

Данный файл представляет заготовку основной формы плагина на языке HTML, а также имеет внутри PHP скрипт, который отвечает за наполнение формы данными и их отправкой в случае нажатия одной из четырех кнопок.

### 5. Файл «posthandler.php»

Он принимает данные, отправленные из вышеописанной формы методом POST, обрабатывает их и выдает пользователю результат. Например, если была нажата кнопка сохранения изменений, то программа собирает данные из формы и обновляет их значения в таблице плагина, попутно проверяя поля ip и password на валидность. Для проверки первого поля используется регулярное выражение, для второго поля ключевым условием является длина строки, которая в целях безопасности не должна быть меньше восьми символов. Для кнопок обновления и сбора информации выполняются соответствующие действия и в новой

странице выдается результат об успешности или неуспешности их выполнения. В случае нажатия кнопки показа отчета генерируется приватная версия отчета для администратора ИТ-инфраструктуры организации. Для каждого из этих действий также реализована проверка привилегий пользователя на исполнение данных действий, которая предотвращает несанкционированный доступ к функциям плагина путем отправки POST запросов по адресу обработчика событий.

6. Файл «report.php»

Данный файл предоставляет заготовку HTML страницы, предоставляющей публичную версию отчета на вкладке Tools->Reports. За контент страницы отвечает вызываемая из основного класса функция `plugin_serverhealthcheck_healthReport` с строковым аргументом «public».