

Оглавление

Установка	1
Использование	1
Написание своих плагинов.....	4

Установка

1. Скопировать все из папки Modules в папку Modules проекта
2. Скопировать все из папки Templates в папку Templates проекта
3. Скопировать скрипт из папки scripts в папку scripts проекта (данный скрипт будет запускаться вместе с главным циклом)
4. Выполнить Sql из файла sql_add.sql для создания таблиц в БД, после чего в таблице admin_users дать права на использование данного модуля (сводится все к тому что добавляется строка scheduled_job в поле ACCESS)

После этого зайти в панель управления, новый модуль будет доступен в Системе (имя scheduled_job)

Использование

- Заходим в Панель Управления/scheduled_job
- Изначально список заданий пуст
- Жмем Add new job



- Появляется новое задание (По-умолчанию оно неактивно)

+Add new job

Запланированные задания

Name: NEW Last run date: Next run date:	Status: 0	Crontab: *****	Is Periodical: 0	Is Active: 0	Редактировать / Удалить
---	-----------	----------------	------------------	--------------	---

- Жмем редактировать
- Поля:
 - Название – имя нашего задания
 - Crontab – расписание запуска в формате кронтаб (<http://ru.wikipedia.org/wiki/Cron>)
 - Is_active – флаг активности (если активно – задание будет выполняться)
 - Is_periodical – флаг периодичности (если сброшен – задание выполнится только один раз, далее автоматически деактивируется; если активен – задание будет выполняться на регулярной основе)
 - Задания – список действий, которые будут выполняться

Сейчас в системе существует только одно действие – ScheduledJobActionSetObjectProp, Единственное, что оно умеет это устанавливать значение для определенного объекта.

Выбираем действие, жмем кнопку Add, далее необходимо ввести значение параметра, Для данного действия значение должно иметь вид [имя объекта].[имя свойства]=[значение], если формат будет неправильный или данный объект/свойство не существует, то система не даст создать такое действие

(Внимание! Проверка существования объекта производится с помощью метода getGlobal, в том случае если объект не существует, этот метод возвращает 0, т.е. возможна ситуация когда объект существует, но его текущее значение = 0, тогда задание не будет создано, данную проблему можно решить по-разному, пока просто надо иметь ввиду, что такое возможно)

- Выбрали задание

[<<< Назад](#)

Новая запись

Название:

Crontab:

Is_active: ☒

Is_periodical: ☒

Действия:

typeName	params	Set	GetInfo	Delete
Добавить	<input type="text" value="ScheduledJobActionSetObjectPr"/>	Add		

[Отмена](#)

- Получаем новое задание

[<<< назад](#)

Новая запись

Название:

Crontab:

Is_active: ☒

Is_periodical: ☒

Действия:

typeName	params	Set	GetInfo	Delete
ScheduledJobActionSetObjectProp	ThisComputer, 1w_temp=2	установить	Get info	Delete
Добавить	<input type="text" value=""/>	Add		

[Отмена](#)

- Возвращаемся назад

Такой скрин будет у пользователей версии 0.6.0

В версии 0.5.0 нулей в датах не будет

Запланированные задания

Name: NEW Last run date: 0000-00-00 00:00:00 Next run date: 0000-00-00 00:00:00	Status: 0	Crontab: 0 10 * * *	Is Periodical: 1	Is Active: 1	Редактировать / Удалить
---	-----------	---------------------	------------------	--------------	---

Написание своих плагинов

Изначально была задумка, чтобы любой желающий, как можно проще мог добавить свои специфические действия, нужные только ему

В итоге, для того чтобы добавить новое действие и система могла его выполнить, необходимо сделать следующее:

В папке `modules/scheduled_job/Actions` создать новый файл, в файле должен быть класс с 3-мя методами (имя класса должно совпадать с именем файла)

Пример:

Actions/MyNewAction.php

```
class MyNewAction extends ScheduledJobActionAbstract
```

```
{
```

public function run(\$params) – данный метод вызывается, когда выполняется действие, \$params – параметры, которые мы заполняем в интерфейсе

public function getInfo() - данный метод вызывается, когда мы жмем на кнопку GetInfo в интерфейсе (см. скрины выше), он возвращает краткую инфу о назначении действия

public function check(\$params) - данный метод вызывается, когда мы ставим новое значение параметра или создаем действие, он проверяет параметры на корректность (как раз данный метод проверяет существование объекта в действии ScheduledJobActionSetObjectProp)

```
}
```

Далее, необходимо зайти в файл конфига в папке `modules/scheduled_job/conf.php`

И прописать новое действие (цифра слева – идентификатор, после создания нового действия и его привязки к заданиям, ее уже нельзя менять на другую)

Conf.php

```
return array(
```

```
    2 => "ScheduledJobActionSetObjectProp",
```

```
    3 => "MyNewAction",
```

```
);
```

После этого новое действие должно появиться в списке выбора.