

Plugin Events

Some plugins may need to be invoked to perform certain tasks immediately before or after an action has occurred. For this reason plugins may register plugin events. These [Events](#) automatically initialize the plugin and invoke the desired event handler (i.e. callback method) when triggered. All event handlers accept a single `\Blesta\Core\Util\Events\Common\EventInterface` (previously an [EventObject](#)). parameter.

For performing tasks at regular intervals see [Plugin Cron Tasks](#).



The EventObject is deprecated

Blesta v4.3.0 has deprecated and replaced the *EventObject* \$event that is passed to your event callback. Plugins will still be passed an *EventObject* for backward compatibility, but this will be replaced by a `\Blesta\Core\Util\Events\Common\EventInterface` in the future, which supports the same methods and data. This means that your plugin implementation should not check for an *EventObject* type unless you also support a type check on `\Blesta\Core\Util\Events\Common\EventInterface`.

Adding Plugin Events

To add plugin events, simply return them from your plugin's `getEvents()` method.

/plugins/my_plugin/my_plugin_plugin.php

```
<?php
class MyPluginPlugin extends Plugin
{
    ...

    public function getEvents()
    {
        return [
            [
                'event' => "Appcontroller.preAction",
                'callback' => array("this", "run")
            ]
            // Add additional events here
        ];
    }

    public function run($event)
    {
        // This will execute when the "Appcontroller.preAction" event is triggered
    }
}
```

In the example above you'll notice the [callback](#) is referencing a class named "this". Blesta automatically maps the string "this" to the plugin when the event is triggered. This is because we want the instance of the class loaded at the time of the event to be triggered, not the instance at the time the event was registered. You can specify here any valid callback, though it is best to use "this" or static class methods where possible as there is limited space to store class instances.