Messengers

This document describes how to implement Messengers. A Messenger allows sending notifications through different methods of communication, like SMS and voice calls.

> **Compatibility check**
>
> Messengers are only available in Blesta v4.12.0 onwards.

Getting Started with Messengers

For the purpose of this manual, the messenger name we'll refer to is **my_messenger**. Your messenger name will, of course, differ.

> **Messenger names must be unique**
>
> A user will not be able to install two messengers with the same name.

File Structure

Messengers are fully contained within their named messenger directory and placed in the */installpath/components/messengers/* directory. Each messenger is only required to contain a single class, representing the messenger, but the following is the recommended structure:

- */messengers/*
  - */my_messenger/*
    - */views/*
    - */language/*
    - **my_messenger.php**

Install Logic

If your messenger requires any code to execute when installed, place that logic in your messenger's **install()** method. This method can also return an array of key => value pairs of metadata, which will be stored in the database for future use.

/messengers/my_messenger/my_messenger.php

```
<?php
class MyMessenger extends Messenger {

    ...

    public function install() {
        #
        # TODO: Place installation logic here
        #
    }
}
?>
```

Uninstall Logic

If your messenger required code to install, it likely requires code to uninstall. Place that logic in your messenger's **uninstall()** method.

There are two parameters passed to the uninstall() method. The first (**$messenger_id**) is the ID of the messenger being uninstalled. Because a messenger can be installed independently under different companies you may need to perform uninstall logic for a single messenger instance. The second parameter (**$last_instance**) identifies whether or not this is the last instance of this messenger in the system. If **true**, be sure to remove any remaining remnants of the messenger.

/messengers/my_messenger/my_messenger.php

```
<?php
class MyMessenger extends Messenger {

    ...

    public function uninstall($messenger_id, $last_instance) {
        #
        # TODO: Place uninstallation logic here
        #
    }
}
?>
```

Upgrade Logic

When the version of your code differs from that recorded within Blesta a user may initiate an upgrade, which will invoke your messenger's **upgrade()** method.

The **$current_version** is the version currently installed. That is, the version that will be upgraded from.

/messengers/my_messenger/my_messenger.php

```
<?php
class MyMessenger extends Messenger {

    ...

    public function upgrade($current_version) {
        #
        # TODO: Place upgrade logic here
        #
    }

}
?>
```

Error Passing

Blesta facilitates error message passing through the use of the **Input** component. Simply load the Input component into your messenger and set any errors you encounter using **Input::setErrors()**. The setErrors() method accepts an array of errors. This can be a multi-dimensional array (in the case of multiple errors triggered for the same input or criteria) or a single dimensional array. The first dimension should be the name of the input that triggered the error. The second dimension, if necessary, is used to identify the type of error encountered.

/messengers/my_messenger/my_messenger.php

```
<?php
class MyMessenger extends Messenger {

    public function __construct() {
        Loader::loadComponents($this, ['Input']);
    }

    ...

    public function upgrade($current_version) {
        // Ensure new version is greater than installed version
        if (version_compare($this->version, $current_version) < 0) {
            $this->Input->setErrors([
                'version' => ['invalid' => 'Downgrades are not allowed.']
            ]);
            return;
        }
    }
}
?>
```

Messenger Methods

Now that we've looked at some of the basic of creating a messenger, let's take a look at all of the required methods each messenger must implement: Messenger Methods.