

Service Management Tabs

Table of Contents

- 1 Summary
- 2 Methods
 - 2.1 allowsServiceTabs()
 - 2.2 getAdminServiceTabs(\$service)
 - 2.3 getClientServiceTabs(\$service)
- 3 Example Plugin

Summary

As of Blesta version 4.3.0, plugins may make tabs available when managing a service in the admin or client interface. In order to do so, plugins must be registered for the package by setting it under *Plugin Integration* when adding or updating a package. Once set on the package, the plugin's defined service management tabs will be visible when managing a service that uses that package.

Methods

allowsServiceTabs()

Your plugin must define that it supports service management tabs or else they will not be available to be set on packages and the tabs will not be visible to services.

```
class MyPlugin extends Plugin
{
    ...
    /**
     * Returns whether this plugin provides support for setting admin or client service tabs
     * @see Plugin::getAdminServiceTabs
     * @see Plugin::getClientServiceTabs
     *
     * @return bool True if the plugin supports service tabs, or false otherwise
     */
    public function allowsServiceTabs()
    {
        return true;
    }
    ...
}
```

getAdminServiceTabs(\$service)

This method returns a list of key/value pairs representing tab names to display when an admin goes to manage a service.

```

class MyPlugin extends Plugin
{
...
    /**
    * Returns all tabs to display to an admin when managing a service
    *
    * @param stdClass $service An stdClass object representing the selected service
    * @return array An array of tabs in the format of method => array where array contains:
    *   - name (required) The name of the link
    *   - href (optional) use to link to a different URL
    *   Example: array('methodName' => "Title", 'methodName2' => "Title2")
    *   array('methodName' => array('name' => "Title", 'href' => "https://blesta.com"))
    */
    public function getAdminServiceTabs(stdClass $service)
    {
        return [
            // Define a tab, 'tabOne' that references a method on this class by the same name,
MyPlugin::tabOne
            'tabOne' => [
                'name' => 'Tab One'
            ],
            // Define a tab, 'google' that links to another URL
            'google' => [
                'name' => 'Google',
                'href' => 'https://google.com'
            ]
        ];
    }

    /**
    * Displays the custom tab defined by ::getAdminServiceTabs
    *
    * @param stdClass $service An stdClass object representing the service
    * @param array $get Any GET parameters
    * @param array $post Any POST parameters
    * @param array $files Any FILES parameters
    * @return string The content of the tab
    */
    public function tabOne(stdClass $service, array $get = null, array $post = null, array $files = null)
    {
        $this->view = new View();

        // Make the service available to the view
        $this->view->set('service', $service);

        // Load the view at /plugins/my_plugin/views/default/tab_one.pdt
        $this->view->setView('tab_one', 'MyPlugin.default');

        // Return the content of the view
        return $this->view->fetch();
    }
...
}

```

getClientServiceTabs(\$service)

This method returns a list of key/value pairs representing tab names to display when a client goes to manage their service.

```

class MyPlugin extends Plugin
{
...
/**
 * Returns all tabs to display to a client when managing a service
 *
 * @param stdClass $service A stdClass object representing the selected package
 * @return array An array of tabs in the format of method => array where array contains:
 *   - name (required) The name of the link
 *   - icon (optional) use to display a custom icon
 *   - href (optional) use to link to a different URL
 *   Example: array('methodName' => "Title", 'methodName2' => "Title2")
 *   array('methodName' => array('name' => "Title", 'icon' => "icon"))
 */
public function getClientServiceTabs(stdClass $service)
{
    return [
        // Define a tab, 'tabTwo' that references a method on this class by the same name,
MyPlugin::tabTwo
        'tabTwo' => [
            'name' => 'Tab Two'
        ],
        // Define a tab, 'google' that links to another URL
        'google' => [
            'name' => 'Google',
            'href' => 'https://google.com'
        ]
    ];
}

/**
 * Displays the custom tab defined by ::getClientServiceTabs
 *
 * @param stdClass $service An stdClass object representing the service
 * @param array $get Any GET parameters
 * @param array $post Any POST parameters
 * @param array $files Any FILES parameters
 * @return string The content of the tab
 */
public function tabTwo(stdClass $service, array $get = null, array $post = null, array $files = null)
{
    $this->view = new View();

    // Make the service available to the view
    $this->view->set('service', $service);

    // Load the view at /plugins/my_plugin/views/default/tab_one.pdt
    $this->view->setView('tab_two', 'MyPlugin.default');

    // Display a notice message on this tab page
    $this->setMessage('notice', 'This is a test tab');

    // Return the content of the view
    return $this->view->fetch();
}
...
}

```

Example Plugin

Download the attached sample plugin that implements management tabs.



plugin-test-management-tabs.zip