

Facades

Overview

Facades serve as "static proxies" to underlying classes in the service container. This provides flexibility over traditional static methods with the benefit of terser syntax.

Use

In the context of a hub, a facade is a class that provides access to an object from the container. For this to work, all facades extend the base `HubzeroFacadesFacade` class.

A facade class only needs to implement a single method: `getAccessor`, which defines what to resolve from the container. The base `Facade` class makes use of the `__callStatic()` magic-method to defer calls from the facade to the resolved object.

Below is the facade for the `Filesystem` wherein the `getAccessor()` method returns the string `'filesystem'`, which is the key that the `Filesystem` service is registered with on the application.

```
class Filesystem extends Facade
{
    /**
     * Get the registered name.
     *
     * @return string
     */
    protected static function getAccessor()
    {
        return 'filesystem';
    }
}
```

In the example below, a call is made to `Filesystem` to check that a file exists. Looking quickly at the code, one might assume that the static method `exists()` is being called on the `Filesystem` class:

```
<?php

namespace ComponentsBlogSiteControllers;

use HubzeroComponentSiteController;
```

FACADES

```
use Filesystem;
use App;

class Media extends SiteController
{
    public function downloadTask()
    {
        //...

        if ( ! Filesystem::exists($file))
        {
            App::abort(404, 'File not found');
        }

        //...
    }
}
```

This facade serves as a proxy to accessing the underlying implementation of the `HubzeroFilesystemFilesystem` interface. So, when any static method on the facade is referenced, the application resolves the binding from the service container and runs the requested method against that object. In short, any calls made using the facade will be passed to the underlying instance of the filesystem service.

Class Reference

Below is a list of every facade, its underlying class, and the service container binding key where applicable.

Global (all client types)Facade	Class	Service Key	Client
App	HubzeroBaseApplication	app	all
Auth	HubzeroAuthManager	auth	admin, s
Cache	HubzeroCacheManager	cache	admin, s
Component	HubzeroComponentLoader	component	admin, s
Config	HubzeroConfigRepository	config	all
Date	HubzeroUtilityDate		all
Document	HubzeroDocumentManager	document	admin, s
Event	HubzeroEventsDispatcher	dispatcher	all

FACADES

Facade	Class	Service Key	Client	
		her		
	Filesystem	HubzeroFilesystemFile system	filesystem	all
	Html	HubzeroHtmlBuilder	html.builder	admin, s
	Lang	HubzeroLanguageTran slator	language	all
	Log	HubzeroLogWriter	log.debug	all
	Module	HubzeroModuleLoader	module	admin, s
	Notify	HubzeroNotificationHan dler	notification	admin, s
	Pathway	HubzeroPathwayTrail	pathway	site
	Plugin	HubzeroPluginLoader	plugin	all
	Request	HubzeroHttpRequest	request	all
	Response	HubzeroHttpResponse	response	all
	Router	HubzeroRoutingRouter	router	all
	Session	HubzeroSessionManag er	session	admin, s
	Toolbar	HubzeroHtmlToolbar	toolbar	admin
	Submenu	HubzeroHtmlToolbar	submenu	admin
	User	HubzeroUserManager	user	all