

Singleton Classes in PHP

By Quentin Zervaas, 22 March 2010, tagged in OOP, PHP

In this PhpRiot Snippet I will show you how to create and use singleton classes. A singleton class is a class that can be instantiated once only. While it may not come up frequently, it can be a useful technique when it simply doesn't make sense to have more than one instance of a class.

To achieve this we cannot instantiate it directly using the `new` keyword. Instead, we must use a static class method to access the create and access the instance.

The key steps in creating singleton classes are as follows:

- Prevent direct instantiation by making the constructor private
- Store the only instance of the class as a static property of the class.
- Provide a static method to access the instance. The instance is created the first time this method is called.

Let's say we want to create a singleton class called `MySingleton`. The following code defines this class, including the private constructor so it cannot be directly instantiated.

Listing 1. Defining the `MySingleton` class (`listing-1.php`)

```
<?php
class MySingleton
{
    private function __construct()
    {
        // put normal constructor code.
        // it will only ever be called once
    }

    public function someMethod()
    {
        echo "Doing something!";
    }
}
?>
```

Next we add a static property called `$_instance`. We make this private so it cannot be accessed except through the static method we'll also create. The convention for the singleton accessor method is to name it `GetInstance` (or `getInstance`, depending on your preference).

Listing 2. Turning `MySingleton` into a singleton class (`MySingleton.php`)

```
<?php
class MySingleton
{
    private static $_instance;

    public static function GetInstance()
    {
        if (!self::$_instance instanceof self) {
            self::$_instance = new self();
        }

        return self::$_instance;
    }

    private function __construct()
    {
        // put normal constructor code.
        // it will only ever be called once
    }

    public function someMethod()
    {
        echo "Doing something!";
    }
}
?>
```

The `GetInstance` method first checks if the `$_instance` property is a `MySingleton`. If it's not, the one and only instance is created. Once it has been created it can be returned.

The following listing shows how you can access and use the object.

Singleton Classes in PHP

By Quentin Zervaas, 22 March 2010, tagged in OOP, PHP

Listing 3. Using the MySingleton class (index.php)

```
<?php
    require_once('MySingleton.php');

    $obj = MySingleton::GetInstance();
    $obj->someMethod();
?>
```