

Capítulo 31. Zend_Registry

Traduzido por Flávio Gomes da Silva Lisboa (Zend Framework 1.5.1)

Sumário

31.1. Usando o Registro.....	1
31.1.1. Configurando Valores no Registro.....	1
31.1.2. Obtendo Valores do Registro.....	1
31.1.3. Construindo um Objeto Registry.....	2
31.1.4. Acessando o Registro como uma Matriz.....	2
31.1.5. Acessando o Registro como um Objeto.....	3
31.1.6. Consultando se um índice existe.....	3
31.1.7. Estendendo o Registro.....	4
31.1.8. Desconfigurando o Registro Estático.....	4

31.1. Usando o Registro

O registro é um container para armazenar objetos e valores no espaço da aplicação. Por armazenar o valor no registro, o mesmo objeto está sempre disponível ao longo de sua aplicação. Esse mecanismo é uma alternativa para o uso de armazenamento global.

O uso típico do registro é através de métodos estáticos na classe `Zend_Registry`. Alternativamente, a classe é um objeto array¹, assim você pode acessar elementos armazenados dentro dele com uma interface convenientemente parecida com um matriz.

31.1.1. Configurando Valores no Registro

Para armazenar uma entrada no registro, use o método estático `set()`.

Exemplo 31.1. Exemplo do método `set()`

```
<?php
Zend_Registry::set('index', $value);
```

O valor pode ser um objeto, uma matriz, ou um escalar. Você pode alterar o valor armazenado em uma entrada específica do registro usando `set()` para configurá-la para um novo valor.

O índice pode ser um escalar, string ou inteiro, como um vetor comum.

31.1.2. Obtendo Valores do Registro

Para recuperar uma entrada do registro, use o método estático `get()`.

Exemplo 31.2. Exemplo do método `get()`

¹ Matriz, ou vetor, de dados.

```
<?php
$value = Zend_Registry::get('index');
```

O método `getInstance()` retorna o objeto de registro estático.

Um objeto registro é iterável.

Exemplo 31.3. Exemplo de iteração sobre o registro

```
<?php
$registry = Zend_Registry::getInstance();
foreach ($registry as $index => $value) {
    echo "O índice de registro $index contém:\n";
    var_dump($value);
}
```

31.1.3. Construindo um Objeto Registry

Além do acesso ao registro estático através de métodos estáticos, você pode criar uma instância diretamente e usá-la como um objeto.

A instância do registro que você acessa através de métodos estáticos é simplesmente uma instância única, e é por conveniência que é armazenada estaticamente, assim você pode acessá-la de qualquer lugar em sua aplicação.

Use um construtor `new` tradicional para criar uma instância do registro. Ele dá a você a oportunidade de inicializar as entradas no registro como uma matriz associativa.

Exemplo 31.4. Exemplo de construção de um registro

```
<?php
$registry = new Zend_Registry(array('index' => $value));
```

Depois de construir essa instância, você pode manipulá-la com métodos de objetos e matrizes, ou pode configurar essa instância para tornar-se a instância estática através do método `setInstance()`.

Exemplo 31.5. Exemplo de inicialização do registro estático

```
<?php
$registry = new Zend_Registry(array('index' => $value));
Zend_Registry::setInstance($registry);
```

O método `setInstance()` lança uma exceção `Zend_Exception` se o registro estático já foi inicializado pelo seu primeiro acesso.

31.1.4. Acessando o Registro como uma Matriz

Se você tem diversos valores para obter ou configurar, pode achar conveniente acessar o registro com uma notação de matriz.

Exemplo 31.6. Exemplo de acesso por matriz

```
<?php
$registry = Zend_Registry::getInstance();
$registry['index'] = $value;
var_dump( $registry['index'] );
```

31.1.5. Acessando o Registro como um Objeto

Você pode também achar conveniente acessar o registro de modo mais na moda da orientação a objetos, usando nomes de índices como propriedades de objeto. Para fazer isso, você precisa especificamente construir o objeto registro usando a opção `ArrayObject::ARRAY_AS_PROPS`, e inicializar a instância estática. Você deve fazer isso antes do registro estático ter sido acessado pela primeira vez. Tome cuidado ao usar essa opção, uma vez que algumas versões de PHP tem bugs quando o registro é usado com essa opção.

Exemplo 31.7. Exemplo de acesso por objeto

```
<?php
// Na inicialização de sua aplicação:
$registry = new Zend_Registry(array(), ArrayObject::ARRAY_AS_PROPS)
Zend_Registry::setInstance($registry);
$registry->tree = 'apple';
.
.
.
// em uma função diferente, em qualquer outro lugar de sua aplicação:
$registry = Zend_Registry::getInstance();
echo $registry->tree; // exibe "apple"
$registry->index = $value;
var_dump($registry->index);
```

31.1.6. Consultando se um índice existe

Para descobrir se um índice particular no registro tem um valor, use o método estático `isRegistered()`.

Exemplo 31.8. Exemplo do método `isRegistered()`

```
<?php
if (Zend_Registry::isRegistered($index)) {
    $value = Zend_Registry::get($index);
}
```

Para descobrir se um índice particular em um objeto registro matriz-objeto tem um valor, use `isset()` como você faria com uma matriz comum.

Exemplo 31.9. Exemplo do método `isset()`

```
<?php
$registry = Zend_Registry::getInstance();
// usando a sintaxe de acesso por matriz
if (isset($registry['index'])) {
    var_dump( $registry['index'] );
}
// usando a sintaxe de acesso por objeto, se habilitada
if (isset($registry->index)) {
    var_dump( $registry->index );
}
```

31.1.7. Estendendo o Registro

O registro estático é uma instância da classe `Zend_Registry`. Se você quiser adicionar funcionalidade ao registro, você pode criar uma classe que estenda `Zend_Registry`, e então pode especificar essa classe como a classe de uso para o registro estático. Use o método estático `setClassName()` para especificar a classe. A classe deve estender `Zend_Registry`.

Exemplo 31.10. Exemplo de especificação do nome da classe de registro estático

```
<?php
Zend_Registry::setClassName('My_Registry');
Zend_Registry::set('index', $value);
```

O registro lança uma exceção `Zend_Exception` se você tentar configurar o nome da classe depois do registro ter sido acessado pela primeira vez. É recomendado que você especifique o nome da classe para seu registro estático na inicialização de sua aplicação.

31.1.8. Desconfigurando o Registro Estático

Embora isso não seja normalmente necessário, você pode desconfigurar a instância estática do registro. Use o método estático `_unsetInstance()`.



Risco de perda de dados

Quando você usa `_unsetInstance()`, todos os dados no registro estático são descartados e não podem ser recuperados.

Você pode usar esse método, por exemplo, se você quiser fazer uso de `setInstance()` ou `setClassName()` depois que o objeto de registro estático tiver sido inicializado. A desconfiguração da instância estática permite que você use esses métodos.

Exemplo 31.11. Exemplo do método `_unsetInstance()`

```
<?php
Zend_Registry::set('index', $value);
Zend_Registry::_unsetInstance();
// troca a classe
Zend_Registry::setClassName('My_Registry');
Zend_Registry::set('index', $value);
```