Search

Getting Started
- Introduction
- A simple tutorial

Language Reference
- Basic syntax
- Types
- Variables
- Constants
- Expressions
- Operators
- Control Structures
- Functions
- Classes and Objects
- Namespaces
- Enumerations
- Errors
- Exceptions
- Fibers
- Generators
- Attributes
- References Explained
- Predefined Variables
- Predefined Exceptions
- Predefined Interfaces and Classes
- Predefined Attributes
- Context options and parameters
- Supported Protocols and Wrappers

Security
- Introduction
- General considerations
- Installed as CGI binary
- Installed as an Apache module
- Session Security
- Filesystem Security
- Database Security
- Error Reporting
- User Submitted Data
- Hiding PHP
- Keeping Current

Features
- HTTP authentication with PHP
- Cookies
- Sessions
- Dealing with XForms
- Handling file uploads
- Using remote files
- Connection handling
- Persistent Database Connections
- Command line usage

Garbage Collection
DTrace Dynamic Tracing

Function Reference
Affecting PHP's Behaviour
Audio Formats Manipulation
Authentication Services
Command Line Specific Extensions
Compression and Archive Extensions
Cryptography Extensions
Database Extensions
Date and Time Related Extensions
File System Related Extensions
Human Language and Character Encoding Support
Image Processing and Generation
Mail Related Extensions
Mathematical Extensions
Non-Text MIME Output
Process Control Extensions
Other Basic Extensions
Other Services
Search Engine Extensions
Server Specific Extensions
Session Extensions
Text Processing
Variable and Type Related Extensions
Web Services
Windows Only Extensions
XML Manipulation
GUI Extensions

Keyboard Shortcuts

?
: This help

j
: Next menu item

k
: Previous menu item

g p
: Previous man page

g n
: Next man page

G
: Scroll to bottom

g g
: Scroll to top

g h
: Goto homepage

g s
: Goto search
(current page)

/
: Focus search box

Объявление классов атрибутов »
« Синтаксис атрибутов

- Руководство по PHP
- Справочник языка
- Атрибуты

Change language: Russian

Чтение атрибутов с помощью Reflection API

Для доступа к атрибутам классов, методов, функций, параметров, свойств и констант класса в Reflection API существует метод **getAttributes()**, который определен для каждого из перечисленных объектов рефлексии. Этот метод возвращает массив объектов ReflectionAttribute, у каждого из которых можно запросить имя и аргументы, а также создать объект класса, представляющего атрибут.

Отделение полученного через рефлексию представления атрибута от явного создания объекта даёт программисту более полный контроль над обработкой ошибок, связанных с отсутствующими классами атрибутов, опечатками или отсутствующими аргументами. Объект класса атрибута будет создан и проверен на корректность аргументов только после вызова метода ReflectionAttribute::newInstance(), не раньше.

Пример #1 Чтение атрибутов средствами Reflection API

```
<?php

#[Attribute]
class MyAttribute
{
public $value;

public function __construct($value)
{
$this->value = $value;
}
}

#[MyAttribute(value: 1234)]
class Thing
{
}

function dumpAttributeData($reflection) {
$attributes = $reflection->getAttributes();

foreach ($attributes as $attribute) {
var_dump($attribute->getName());
var_dump($attribute->getArguments());
var_dump($attribute->newInstance());
}
}

dumpAttributeData(new ReflectionClass(Thing::class));
/*
string(11) "MyAttribute"
array(1) {
["value"]=>
int(1234)
}
object(MyAttribute)#3 (1) {
["value"]=>
int(1234)
}
*/
```

Чтобы получить атрибуты только нужного класса, вместо последовательного перебора всех атрибутов объекта рефлексии в метод **getAttributes()** передают в качестве аргумента имя искомого класса атрибута.

Пример #2 Чтение конкретных атрибутов средствами Reflection API

```
<?php
```

```
function dumpMyAttributeData($reflection) {
$attributes = $reflection->getAttributes(MyAttribute::class);

foreach ($attributes as $attribute) {
var_dump($attribute->getName());
var_dump($attribute->getArguments());
var_dump($attribute->newInstance());
}
}

dumpMyAttributeData(new ReflectionClass(Thing::class));
```


User Contributed Notes 1 note

6
Hirusha Sharma ¶
2 years ago

```
Fetch properties from functions:

----------------------------------------
Function definition with attributes:
----------------------------------------
#[ReadOnly]
#[Property(type: 'function', name: 'Hello')]
function Hello()
{
return "Hello";
}

----------------------------------------
Gather attributes from the function
----------------------------------------
function getAttributes(Reflector $reflection)
{
$attributes = $reflection->getAttributes();
$result = [];
foreach ($attributes as $attribute)
{
$result[$attribute->getName()] = $attribute->getArguments();
}
return $result;
}

$reflection = new ReflectionFunction("Hello");
print_r(getAttributes($reflection));

----------------------------
OUTPUT
----------------------------
Array
(
[ReadOnly] => Array
(
)

[Property] => Array
(
[type] => function
```

```
[name] => Hello
)

)
```