

יכולות שיקוף

[יכולות שיקוף](#)

[ReflectionFunction](#) המחלקה

[ReflectionParameter](#) המחלקה

[ReflectionClass](#) המחלקה

[ReflectionMethod](#) המחלקה

[ReflectionProperty](#) המחלקה

[Reflection Extension](#) המחלקה

יכולות שיקוף

בדומה לשפות תיכנות אחרות, שפת התיכנות PHP כוללת את ה-Reflection API אשר מאפשר לנו לבדוק בזמן ריצה קוד ב-PHP ולקבל עליו מידע אשר יכול לעזור לנו ליצירה אוטומטית של תיעוד, לבצע Reverse Engineering ולבדוק אם פונקציונליות מסוימת קיימת או לא.

ה-Reflection API התווסף ל-PHP בגירסה 5. כאשר משתמשים ב-Reflection API הקוד שאותו אנו בודקים מתבצע מאחורי הקלעים ובכך ה-Reflection API מצליח לאסוף עליו מידע. מסיבה זו יש להיזהר מלבצע reflection על קוד שלא נכתב על ידינו.

אחד השימושים הבולטים ב-Reflection API הוא בביצוע בדיקות.

המחלקה ReflectionFunction

כל אובייקט מהמחלקה ReflectionFunction מייצג פונקציה מסויימת. כשאשר אנו מקבלים אובייקט מהמחלקה ReflectionFunction אנו יכולים לקרוא להפעלת של מתודות על אותו אובייקט ובאמצעותן לקבל מידע נוסף על הפונקציה שהוא מייצג.

באמצעות קריאה להפעלת הפונקציה get_defined_functions אנו מקבלים מערך של אובייקטים מטיפוס ReflectionFunction אשר מייצגים את הפונקציות הזמינות להפעלה. באמצעות הפעלת המתודות השונות שמוגדרות במחלקה ReflectionFunction על כל אחד מהאובייקטים האלה אנו יכולים לקבל מידע מגוון על כל אחת מהפונקציות שהם מייצגים.

דוגמת הקוד הבאה מציגה שימוש בסיסי במחלקה ReflectionFunction.

```
<?php
function doSomething()
{
    echo "<B>Do Something</B><BR>";
}
function sayHello($ob = "Students")
{
    echo "<B>Hello "+$ob+"</B>";
}
function sayGoodMorning($ob = "Students")
{
    echo "<B>Good Morning "+$ob+"</B>";
}
$funcs = get_defined_functions();
foreach($funcs['user'] as $function)
{
    try
    {
        $ob = new ReflectionFunction($function);
        echo "<BR>function name... ".$ob->getName();
        echo "<BR>file name... ".$ob->getFileName();
        echo "<BR>start line... ".$ob->getStartLine();
        echo "<BR>end line... ".$ob->getEndLine();
        echo "<BR>";
    }
}
```

```
}  
catch(ReflectionException $exception)  
{  
    echo "<BR><B>Exception Caught</B><BR>";  
}  
}  
?>
```

הפלט שנקבל הוא:

function name... doSomething

file name... E:\PHP_DEVELOPMENT_TOOLS\workspace\hello_proj\reflection_function_demo.php

start line... 2

end line... 5

function name... sayHello

file name... E:\PHP_DEVELOPMENT_TOOLS\workspace\hello_proj\reflection_function_demo.php

start line... 6

end line... 9

function name... sayGoodMorning

file name... E:\PHP_DEVELOPMENT_TOOLS\workspace\hello_proj\reflection_function_demo.php

start line... 10

end line... 13

המחלקה ReflectionParameter

כל אובייקט מהמחלקה ReflectionParameter מייצג פרמטר מסויים בפונקציה נתונה. באמצעות הפעלת המתודות שהוגדרו במחלקה ReflectionParameter על אובייקט שנוצר ממנה ניתן לקבל מידע נוסף על הפרמטר שהוא מייצג.

התכנית הבאה מציגה דוגמא לשימוש במחלקה ReflectionParameter.

```
<?php
class Rectangle
{

}

function doSomething($varA, &$varB, Rectangle $varC, $varD=12)
{
    echo "<br>something<br>";
}

$obj = new ReflectionFunction(doSomething);

foreach ($obj->getParameters() as $index => $param)
{
    echo "<br>parameter name: ".$param->getName();
    echo "<br>parameter class: ".$param->getClass();
    echo "<br>parameter allows null: ".$param->allowsNull();
    echo "<br>passed by reference: ".$param->isPassedByReference();
    echo "<br>";
}
?>
```

הפלט שנקבל מהפעלת התכנית הזו הוא:

```
parameter name: varA
parameter class:
parameter allows null: 1
passed by reference:
```

```
parameter name: varB  
parameter class:  
parameter allows null: 1  
passed by reference: 1
```

```
parameter name: varC  
parameter class: Class [ class Rectangle ] { @@ E:\PHP_DEVELOPMENT_TOOLS\workspace\  
hello_proj\reflection_parameter_demo.php 2-5 - Constants [0] { } - Static  
properties [0] { } - Static methods [0] { } - Properties [0] { } - Methods [0]  
{ } }  
parameter allows null:  
passed by reference:
```

```
parameter name: varD  
parameter class:  
parameter allows null: 1  
passed by reference:
```

המחלקה ReflectionClass

אובייקט שנוצר מהמחלקה ReflectionClass מייצג מחלקה. באמצעות הפעלת מתודות על האובייקט ניתן לקבל מידע נוסף על המחלקה שהוא מייצג.

הדוגמא הבאה מציגה שימוש בסיסי במחלקה זו.

```
<?php
class Rectangle
{
    private $width;
    private $height;
    function Rectangle($valW,$valH)
    {
        $this->setHeight($valH);
        $this->setWidth($valW);
    }
    function area()
    {
        return $this->width * $this->height;
    }
    function setWidth($val)
    {
        if($val>0)
        {
            $this->width = $val;
        }
    }
    function setHeight($val)
    {
        if($val>0)
        {
            $this->height = $val;
        }
    }
}
```

```
$obj = new ReflectionClass(Rectangle);

echo "<br>\$obj->isAbstract()=" . $obj->isAbstract();
echo "<br>\$obj->isFinal()=" . $obj->isFinal();
echo "<br>\$obj->isInterface()=" . $obj->isInterface();
echo "<br>\$obj->getName()=" . $obj->getName();
echo "<br>\$obj->getStartLine()=" . $obj->getStartLine();
echo "<br>\$obj->getEndLine()=" . $obj->getEndLine();
?>
```

הפלט שנקבל:

```
$obj->isAbstract()=
$obj->isFinal()=
$obj->isInterface()=
$obj->getName()=Rectangle
$obj->getStartLine()=2
$obj->getEndLine()=29
```


המחלקה ReflectionMethod

אובייקט ממחלקה זו מייצג מתודה מסויימת (מתודה ב-class מסויים). באמצעות הפעלת מתודות על אובייקט זה ניתן לקבל מידע נוסף על המתודה שאותה הוא מייצג.

הדוגמא הבאה מציגה שימוש בסיסי במחלקה זו.

```
<?php
class Rectangle
{
    private $width;
    private $height;
    function Rectangle($valW,$valH)
    {
        $this->setHeight($valH);
        $this->setWidth($valW);
    }
    function area()
    {
        return $this->width * $this->height;
    }
    function setWidth($val)
    {
        if($val>0)
        {
            $this->width = $val;
        }
    }
    function setHeight($val)
    {
        if($val>0)
        {
            $this->height = $val;
        }
    }
}
```

```
$obj = new ReflectionMethod('Rectangle','setHeight');

echo "<br>\$obj->isAbstract()=" . $obj->isAbstract();
echo "<br>\$obj->isFinal()=" . $obj->isFinal();
echo "<br>\$obj->isPublic()=" . $obj->isPublic();
echo "<br>\$obj->isPrivate()=" . $obj->isPrivate();
echo "<br>\$obj->getName()=" . $obj->getName();
echo "<br>\$obj->getStartLine()=" . $obj->getStartLine();
echo "<br>\$obj->getEndLine()=" . $obj->getEndLine();

?>
```

הפלט שמתקבל הוא:

```
$obj->isAbstract()=
$obj->isFinal()=
$obj->isPublic()=1
$obj->isPrivate()=
$obj->getName()=setHeight
$obj->getStartLine()=22
$obj->getEndLine()=28
```

המחלקה ReflectionProperty

אובייקט מטיפוס ReflectionProperty מייצג property שמוגדר במחלקה מסויימת. באמצעות הפעלת מתודות על אובייקט זה ניתן לקבל מידע נוסף על ה-property המיוצג.

```
<?php
class Rectangle
{
    private $width;
    private $height;
    function Rectangle($valW,$valH)
    {
        $this->setHeight($valH);
        $this->setWidth($valW);
    }
    function area()
    {
        return $this->width * $this->height;
    }
    function setWidth($val)
    {
        if($val>0)
        {
            $this->width = $val;
        }
    }
    function setHeight($val)
    {
        if($val>0)
        {
            $this->height = $val;
        }
    }
}

$obj = new ReflectionProperty('Rectangle','width');
```

```
echo "<br>\$ob->isPublic()=".$ob->isPublic();  
echo "<br>\$ob->isPrivate()=".$ob->isPrivate();  
echo "<br>\$ob->getName()=".$ob->getName();
```

```
?>
```

הפלט שיתקבל הוא:

```
$ob->isPublic()=  
$ob->isPrivate()=1  
$ob->getName()=width
```

המחלקה ReflectionExtension

אובייקטים שנוצרים ממחלקה זו מייצגים extension שמוותקנים בסביבת ההרצה שלנו. באמצעות הפעלת מתודות על אובייקטים אלה ניתן לקבל מידע נוסף על ה-extensions שמוותקנים בסביבה שלנו.

הדוגמא הבאה מציגה אופן שימוש בסיסי במחלקה זו.

```
<?php
$extensions = get_loaded_extensions();

foreach($extensions as $key => $value)
{
    $obj = new ReflectionExtension($value);
    echo "<br>extension name: ".$value;
    echo "<br>extension version: ".$obj->getVersion();
    echo "<br>";
}
?>
```

הפלט שנקבל הוא:

```
extension name: com_dotnet
extension version: 0.1

extension name: ctype
extension version:

extension name: session
extension version:

extension name: filter
extension version: 0.11.0

extension name: hash
extension version: 1.0
...
```