ClickBlocks\Core\Delegate

General information

Inheritance	no
Child classes	no
Interfaces	ClickBlocks\Core\IDelegate
Source	Framework/core/delegate.php

Delegate class extends the concept of a callback function in php. Instead, the notion of a delegate that represents the object class **Delegate** or a string in a certain format, which is a description of the plan call a function or a class method.

A delegate is a kind of alias some method of a class or function. You can work with it in the same way as with the function it describes.

There are seven basic types of lines delegate:

1. **function** - function call.
2. **class::method** - call of a class static method.
3. **class->method** - call of a class non-static method. The class has constructor without mandatory parameters.
4. **class[]** - call of a class constructor that has no mandatory parameters.
5. **class[n]** - call of a class constructor that has **n** mandatory parameters.
6. **class[n]->method** - call of a class non-static method. Class constructor has **n** mandatory parameters.
7. **class@cid->method** - call of non-static method of some web-control.

Here. **function** - name of some function, **class** - name of some class, **method** - class method name, **n** - number of mandatory parameters of a class constructor, **cid** - unique or logical identifier of some web-control.

Besides of these basic types of delegates, there are two context-dependent cases of delegates:

- **::method** - if the page class is defined (variable `ClickBlocks\MVC\Page::$page`), then a delegate is similar to calling the appropriate static method of paging class. If the page class is not defined, the appropriate method of class **CB** will be called.
- **->method** - like the first case, only applied to non-static methods.

Public non-static methods

__construct()

```
public void __construct(mixed $callback)
```

$callback	mixed	delegate or any callable object.

Class constructor. Parses a delegate string and prepare a delegate to invoking.

__toString()

```
public string __toString()
```

Returns a string representation of the delegate. If a closure is used instead of a delegate, the method returns the string "Closure".

__invoke()

```
public mixed __invoke(mixed $arg1, mixed $arg2, ...)
```

Allows to call an object of delegate as a class method or function.
The method takes parameters of function or method that are invoked by delegate.
Example:

```
// Creates delegate for method "foo" of class "Test".
// Class constructor takes two parameters.
$d = new Delegate('Test[2]->foo');
// Invokes method "foo" and passes parameter 'test' to it.
// Passes parameters 'a' and 'b' to the class constructor.
$d('a', 'b', 'test');
```

call()

```
public mixed call(array $args = null)
```

$args	array	array of parameters of delegated method or function.

Calls delegated method of a class or a function. Example:

```
// Creates delegate for method "foo" of class "Test".
// Class constructor takes two parameters.
$d = new Delegate('Test[2]->foo');
// Invokes method "foo" and passes parameter 'test' to it.
// Passes parameters 'a' and 'b' to the class constructor.
$d->call(['a', 'b', 'test']);
```

isPermitted()

```
public function isPermitted(array $permissions)
```

$permissions	array	permissions to call the delegate.

Checks whether or not the delegate can be invoked according to permissions. Permissions array have the following structure:

```
[
  'permitted' => ['regexp1', 'regexp2', ... ],
  'forbidden' => ['regexp1', 'regexp2', ...]
]
```

If string representation of the delegate matches at least one of **permitted** regular expressions and none of **forbidden** regular expressions, the method returns TRUE. Otherwise it returns FALSE.

isCallable()

```
public boolean isCallable(boolean $autoload = true)
```

$autoload	boolean	determines whether to include delegated class if it is not defined.

The method returns TRUE, if the delegate can be called and FALSE otherwise.

getInfo()

```
public array getInfo()
```

Returns the full details of the delegate. The format of the returned array is as follows:

```
['class'   => ... [string] ...,
  'method'  => ... [string] ...,
  'static'  => ... [boolean] ...,
  'numargs' => ... [integer] ...,
  'type'    => ... [string] ...,
  'cid'     => ... [string] ...]
```

here, **class** - class name of delegated method, **method** - name of delegated method or function, **static** - sign of static method, **numargs** - number of mandatory parameters of constructor of delegated class, **type** - delegate type, **cid** - unique or logical identifier of web-control.

getClass()

```
public string getClass()
```

Returns name of delegated class.

getMethod()

```
public string getMethod()
```

Returns name of delegated class method or function.

getType()

```
public string getType()
```

Returns delegate type. It can be one of the following values: "function", "closure", "class" or "control".

getParameters()

```
public array|boolean getParameters()
```

Returns array of parameters of delegated class method, function or closure or FALSE on failure.

getClassObject()

```
public object getClassObject(array $args = null)
```

$args	array	array of arguments of constructor of the delegated class.

Returns object of the delegated class. If delegate type is "function" the method returns NULL. For delegate type "control" the method returns object of a web-control (or FALSE, if control with such ID does not exist).

Protected non-static properties

callback

```
protected string $callback
```

String representation of the delegate.

class

```
protected string $class
```

Name of delegated class.

method

```
protected string $method
```

Name of delegated class method.

static

```
protected boolean $static
```

Contains TRUE, if the delegated method is static. If delegate type is "function" or "closure", then the value of the property is NULL.

numargs

```
protected integer $numargs
```

Number of mandatory parameters of constructor of delegated class. For delegate type "function" or "closure" property value is NULL.

cid

```
protected string $cid
```

Unique or logical identifier of delegated web-control.

type

```
protected string $type
```

Type of the delegate.