ClickBlocks\MVC\Page

General information

Inheritance	no
Subclasses	no
Interfaces	no
Source	Framework/mvc/page.php

Page is analogue of controller in MVC model. Also it is the base class for all page classes. The purpose of a page class is execution of ajax requests and control of UI on the server side. So, any page class can be treated as a container for event handlers of the page UI. **Page** also has the means for caching HTML of the entire page.

Public static properties

cache

```
public static ClickBlocks\Cache\Cache $cache
```

Default cache object for caching the rendering of page HTML.

cacheGroup

```
public static string $cacheGroup = 'pages'
```

Cache group of the default cache for page classes. If the value of this property is NULL, the cache group is automatically turns out to "pages".

cacheExpire

```
public static integer $cacheExpire = 0
```

Default cache expire of the default cache for page classes.

current

```
public static ClickBlocks\MVC\Page $current
```

An instance of the page class, whose workflow executes in the moment.

Public non-static properties

view

```
public ClickBlocks\Web\POM\View $view
```

An instance of the view object that represents the view (HTML/CSS/JS) of the web page.

noAccessURL

```
public string $noAccessURL
```

The URL for redirect if the current page is not accessible.

noSessionURL

```
public string $noSessionURL
```

The URL for redirect if the user session is expired.

Public non-static methods

__construct()

```
public void __construct(string $template = null)
```

$template	string	HTML template of the page or the path to a file of the page template.

Class constructor. It creates unique identifier of the page based on page template, page class and site unique ID. This UID will be used for caching of the page rendering.

getPageID()

```
public string getPageID()
```

Returns the unique identifier of the page.

setPageID()

```
public void setPageID(string $UID)
```

$UID	string	the new unique identifier of the page.

Sets up the new unique identifier of the page.

getCache()

```
public ClickBlocks\Cache\Cache getCache()
```

Returns the page cache object. This cache object is determined by the following way: if static property `Page::$cache` is defined then this property value becomes the page cache, otherwise, the default cache of the framework becomes the page cache.

isExpired()

```
public boolean isExpired()
```

Returns FALSE if the page is not cached or its cache is expired. Otherwise, it returns TRUE.

restore()

```
public string restore()
```

Returns HTML of the page from its cache or NULL if the page cache is not set or expired.

getSequenceMethods()

```
public array getSequenceMethods(string $first = true)
```

$first	string	type of method sequence.

Returns list of workflow methods. if **$first** is TRUE, the list of methods for the first visit to the page is returned, otherwise it returns the list of methods for all next visits.

get()

```
public ClickBlocks\Web\POM\Control|boolean get(string $id, boolean $searchRecursively = true)
```

$id	string	unique or logic identifier of a control.
$searchRecursively	boolean	the sign of recursive search of a control.

This method is alias of method **get()** of class **ClickBlocks\Web\POM\View** and returns control object by its unique or logic ID. If a control with such ID is not found, it returns FALSE.

access()

```
public boolean access()
```

Checks accessibility of the page. By default this method always returns TRUE. But you can change such behaviour in your page class by overriding this method.
The method is automatically invoked once after constructor and before any other methods.

parse()

```
public void parse()
```

Parses the page template. This method is automatically invoked.

init()

```
public void init()
```

Initializes the page view. This method is automatically invoked once for GET non-Ajax request to the page.

load()

```
public void load()
```

Prepares the page view. This method is automatically invoked each time you visit the page.

render()

```
public void render()
```

Renders the page HTML. This method is automatically invoked once for GET non-Ajax request.

assign()

```
public void assign()
```

Assigns the changes that received from the client side, to controls. This method is automatically invoked during Ajax-request.

process()

```
public void process()
```

Performs the Ajax request.

unload()

```
public void unload()
```

Completes the page workflow. This method is automatically invoked each time you visit the page. You can use this method to free previously allocated resources.

Protected non-static properties

expire

```
protected integer $expire = 0
```

The expiration time (in seconds) of the page cache. If it is 0, the page rendering is not cached. If this property is not defined, the value of static property `Page::$cacheExpire` will be used as a cache lifetime.

ajaxPermissions

```
protected $ajaxPermissions =
          ['permitted' => ['/^ClickBlocks\\\\(MVC|Web\\\\POM)\\\\[^\\\\]*$/i'],
           'forbidden' => ['/^ClickBlocks\\\\Web\\\\POM\\\\[^\\\\]*\[\d*\]->offset(Set|Get|Unset|Exists)$/i']];
```

The list of regular expressions that restrict the area of permitted delegates.

sequenceMethods

```
protected array $sequenceMethods = ['first' => ['parse', 'init', 'load', 'render', 'unload'],
                                    'after' => ['assign', 'load', 'process', 'unload']]
```

The sequence of class methods that determines the class workflow. The sequence should consist of two parts: for the first visit to the page (GET non-Ajax request) and for other visits.