ClickBlocks\Net\Headers

General information

Inheritance	no
Child classes	no
Interfaces	no
Source	Framework/net/headers.php

Class provides access to HTTP-headers of the current HTTP request or response.

Public static methods

getRequestHeaders()

```
public static ClickBlocks\Net\Headers getRequestHeaders()
```

Returns an object of class `ClickBlocks\Net\Headers` that allows access to HTTP-headers of the current HTTP-request.

getResponseHeaders()

```
public static ClickBlocks\Net\Headers getResponseHeaders()
```

Returns an object of class `ClickBlocks\Net\Headers` that allows access to HTTP-headers of the last server HTTP-response.

normalizeHeaderName()

```
public static string normalizeHeaderName(string $name)
```

$name	string	unnormalized name of HTTP-header.

Returns normalized name of HTTP-header.

Public non-static methods

getHeaders()

```
public array getHeaders()
```

Returns array of HTTP-headers.

setHeaders()

```
public void setHeaders(array $headers)
```

$headers	array	array of HTTP-headers.

Sets HTTP-headers.

merge()

```
public void merge(array $headers)
```

$headers	array	array of HTTP-headers.

Performs merging of the current HTTP-headers with HTTP-headers that passed as a method parameter.

clean()

```
public void clean()
```

Removes all HTTP-headers.

has()

```
public boolean has(string $name)
```

$name	string	the header name.

Checks whether the given header exists or not. Returns TRUE if the HTTP header is defined and FALSE otherwise.

get()

```
public string get(string $name)
```

$name	string	name of the HTTP-header.

Returns value of HTTP-header or FALSE if such header is not defined.

set()

```
public void set(string $name, string $value)
```

$name	string	name of the HTTP-header.
$value	string	value of the HTTP-header.

Sets value of HTTP-header.

remove()

```
public void remove(string $name)
```

$name	string	name of the HTTP-header.

Removes HTTP-header by its name.

getContentType()

```
public array|string getContentType(boolean $withCharset = false)
```

$withCharset	boolean	if TRUE the method returns an array of the following structure `['type' => ..., 'charset' => ...]`, otherwise only content type will be returned.

Returns the content type and/or response charset.

setContentType()

```
public void setContentType(string $type, string $charset = null)
```

$type	string	the content type or its alias.
$charset	string	the content charset.

Sets content type header. You can use aliases of several content types. Here are all content types that have aliases:

- **text** - text/plain
- **html** - text/html
- **json** - application/json
- **xml** - application/xml
- **css** - text/css
- **js** - application/javascript

getCharset()

```
public string getCharset()
```

Returns the response charset. If the "Content-Type" header with charset is not set the method returns NULL.

setCharset()

```
public void setCharset(string $charset = 'UTF-8')
```

$charset	string	the response charset.

Sets the response charset. If the "Content-Type" header is not set the "text/html" content type will be used.
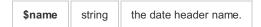
getAcceptableContentTypes()

```
public array getAcceptableContentTypes()
```

Returns a list of content types acceptable by the client browser. If header "Accept" is not set the methods returns an empty array.

getDate()

```
public DateTime|boolean getDate(string $name)
```

$name	string	the date header name.

Returns the given date header as a DateTime instance. If the date header is not set or not parseable the method returns FALSE.

setDate()

```
public void setDate(string $name, string|DateTime $date = 'now')
```

$name	string	the date header name.
$date	string, DateTime	the date header value.

Sets value of the given date header.

hasCacheControlDirective()

```
public boolean hasCacheControlDirective(string $directive)
```

$directive	string	the cache control directive name.

Checks whether the given cache control directive is set.

getCacheControlDirective()

```
public mixed getCacheControlDirective(string $directive)
```

$directive	string	the cache control directive name.

Returns the value of the given cache control directive. If the directive is not set the method returns NULL.

setCacheControlDirective()

```
public void setCacheControlDirective(string $directive, mixed $value = true)
```

$directive	string	the cache control directive name.
$value	mixed	the cache control directive value.

Sets the value of the given cache control directive.

removeCacheControlDirective()

```
public void removeCacheControlDirective(string $directive)
```

$directive	string	the cache control directive name.

Removes the given cache control directive.

__toString()

```
public string __toString()
```

Returns the headers as a string. This method allows to use the class Headers as a string.

Protected non-static properties

headers

```
protected array $headers = []
```

Array of HTTP-headers.

contentTypeMap

```
protected array $contentTypeMap = ['text' => 'text/plain',
                                   'html' => 'text/html',
                                   'json' => 'application/json',
                                   'xml'  => 'application/xml',
```

```
                           'css'  => 'text/css',
                           'js'   => 'application/javascript']
```

Array of content types that have aliases. Keys of this array are aliases and its values are content types.

cacheControl

```
protected array $cacheControl = []
```

Directives of the "Cache-Control" header.

Protected non-static methods

formCacheControlHeader()

```
protected string formCacheControlHeader()
```

Forms and returns the "Cache-Control" header value.