

ClickBlocks\Net\API

General information

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

This class can be used as a base class of the RESTful API class system. It allows to route the API request, handle errors and exceptions and convert the output to the necessary format.

Public static methods

process()

```
final public static void process()
```

Performs the requested API method and invokes the callback that associated with the given request. This method is the entry point of the API class system. Example:

```
...
class MyAPI extends API
{
    ...
}

// Start to process the API request.
MyAPI::process();
```

You cannot override this method in your classes.

error()

```
public static void error(Exception $e, array $info)
```

\$e	Exception	the caught exception.
\$info	array	the full information about the error.

The default error and exception handler of the API class system. The method stops the script execution and sets the response status code to 500.

Protected static properties

map

```
protected static array $map = []
```

Describes relations between all API requests and callbacks which should be invoked to process the particular request. Also **\$map** can contains different additional parameters for callbacks. The typical structure of **\$map** with all possible parameters is shown below:

```
$map = ['/url1' => ['GET|...' => ['callback' => 'some delegate',
```

```

        'component' => URL::PATH,
        'ssl' => TRUE|FALSE,
        'ignoreWrongDelegate' => TRUE|FALSE,
        'coordinateParameterNames' => TRUE|FALSE,
        'redirect' => 'URL for redirect'],
        'POST|...' => [...],
        ...],
        '/url2' => ['GET|...' => [...],
        'POST|...' => [...],
        ...],
        ...
        '/url-nth' => [...]];

```

Here the keys of the map array are the regular expression pattern (PCRE) for the relevant request URLs. The array values are also arrays. Their keys are the HTTP method of the specific request. You can specify several HTTP methods, each separated by symbol "|". The values of the nested arrays are the set of callback parameters, including callback itself:

- **callback** - any delegate which will be automatically invoked during processing of the request whose HTTP method and URL meet one of the HTTP methods and URL of the given callback.
- **component** - specifies which part of the request URL should meet the given regular expression pattern. The valid value of this parameter must be one of the constants of the class `ClickBlocks\Net\URL`. The default value (if the parameters is not set) is `URL::PATH`.
- **ssl** - if TRUE only requests with secured HTTP protocol can be processed. The default value is FALSE.
- **ignoreWrongDelegate** - boolean parameter that determines whether the wrong delegate should be ignored. The default value is FALSE.
- **coordinateParameterNames** - boolean parameter that determines whether to synchronize the URL template variables and parameters of the callback. The default value is FALSE.
- **redirect** - URL to redirect. Performs redirect to the specific URL if the request URL meets the given regular expression. The default value is empty.

request

```
protected static ClickBlocks\Net\Request $request
```

Represents the current HTTP request object.

response

```
protected static ClickBlocks\Net\Response $response
```

Represents the current HTTP response object.

contentType

```
protected static $contentType = 'json'
```

The response content type. It can be regular MIME-type or its alias (if exists).

outputCharset

```
protected static $outputCharset = 'UTF-8'
```

The output charset of the response body.

inputCharset

```
protected static $inputCharset = 'UTF-8'
```

The input charset of the response body.

namespace

```
protected static $namespace = '\\'
```

The default namespace of the all callbacks' delegates. It is useful in cases when all your callback methods are placed in the classes or functions of the single namespace.

convertOutput

```
protected static $convertOutput = false
```

Determines whether the response body should be converted according to the defined content type.

convertErrors

```
protected static $convertErrors = false
```

Determines whether any error information should be converted according to the defined content type.

Protected static methods

notFound()

```
protected static void notFound(mixed $content = null)
```

\$content	mixed	the body response.
------------------	-------	--------------------

This method is automatically called when the current request does not match any API methods (**\$map**'s callbacks). The method stops the script execution and sets the response status code to 404.

convert()

```
protected static string convert(mixed $content)
```

\$content	mixed	any data as a result of the request callback execution.
------------------	-------	---

Converts the execution result of the requested API method to the specified text format according to the output content type and charset. If the property **\$convertOutput** is FALSE the request result will be returned as is.

batch()

```
protected array batch()
```

The batch API method allowing to perform several independent between themselves API methods at once.

Protected non-static properties

before()

```
protected void before(array $resource, array &$amp;params)
```

\$resource	array	one of the values of the \$map property.
\$params	array	the URL template variables.

Automatically invokes before the API method call. You can use this method for all common actions such as creating connection with database,

authentication and etc. Within this method you can also change, if you need, the template variables which after that will be passed to the callback method.

after()

```
protected void after(array $resource, array $params, mixed &$result)
```

\$resource	array	one of the values of the \$map property.
\$params	array	the URL template variables.
\$result	mixed	the result of the API method execution.

Automatically invokes after the API method call. You can use this method for deleting temporary resources, closing connection with database or final processing of the execution result of the API method.