

Framework Core

allow(string *set*);

Allow functions defined in PHP extensions to be used in templates. The default set is 'standard | date | pcre'.

cached(string *variable*);

Returns an array containing the data size and time the variable was stored in cache. If variable is not present in cache, FALSE is returned.

call(mixed *processes*);

Convenience method for executing sandboxed processes. Used for programmatically calling an anonymous or named function, a single or daisy-chained string of named functions/imported files similar to route() and input() handlers.

clear(string *variable*);

Removes **variable** from the framework's symbol table and cache.

config(string *ini_file*);

Set up the framework according to the contents of specified configuration file.

exists(string *variable*);

Return TRUE if framework **variable** has been assigned a value, or FALSE if **variable** has not been defined.

get(string *variable*);

Return value of framework variable. Fetches value from cache if persisted by the set() method.

http404(void);

Trigger an HTTP 404 error.

httpstatus(integer *status_code*);

Send HTTP status header. Return text equivalent of **status_code**.

map(string *uri*, string *class*);

Assign **class** as route handler for incoming HTTP request. **class** should contain the appropriate methods for handling the **uri** pattern: get(), post(), put(), head() or delete().

mock(string *pattern*);

Simulate an incoming HTTP request. Refer to route() for pattern syntax.

mset(array *kv_pairs*);

Similar to set() but used for performing multiple assignments at once. mset() binds values in the associative array **kv_pairs** to keys representing framework variables.

profile(void);

Returns an array of runtime performance analysis data.

realIP(void);

Sniff HTTP headers for remote user's real IP address.

reroute(string *location*);

Reroute to specified URI **location**.

resolve(string *expression*);

Evaluate **expression** containing template tokens.

route(string *pattern*, mixed *handler*, [integer *timeout*], [boolean *allow*]);

Assign **handler** to an HTTP request **pattern** consisting of a GET, HEAD, POST, PUT or DELETE method (or a pipe-separated combination), a space and a URI. If cache **timeout** is specified, the framework retrieves any previously-generated output from cache (if timer has not expired). Otherwise, the cache is refreshed by calling **handler**. Output is not cached if **timeout** is not specified. If **allow** is specified and evaluates to FALSE (default is TRUE), direct links (also known as hot links) are blocked and the framework redirects the user to the URL pointed to by the **HOTLINK** global variable.

run(void);

Run the application and match incoming HTTP request to defined routes.

serve(string *template*, [boolean *is_html*], [string *path*]);

Returns a string representing a rendered HTML/XML template. If **is_html** is TRUE (default), template is interpreted as a regular HTML/text file. If FALSE, rendering follows strict XML parsing rules. Specify **path** if template is not in the framework's GUI folder.

set(string *variable*, mixed *value*, boolean *persist*);

Bind **value** to specified framework **variable**. Framework stores **variable** in cache (persists between sessions) if the **persist** argument is specified and evaluates to TRUE.

Framework Variables

string **AUTOLOAD**

Location(s) of user-defined PHP classes that the framework will autoload at run-time. If onLoad method is defined in an autoloader class, it is automatically executed. Default is 'autoload/' folder under the Web root. Accepts a pipe-separated list of folders.

string **BASE**

Location of index.php main controller.

string **CACHE**

Cache back-end. Framework auto-detects presence of apc, memcache, and xcache and uses the first available PHP extension. If none found, framework uses the (slower) file system.

array **COOKIE, GET, POST, REQUEST, SESSION, FILES, SERVER, ENV**

Framework equivalents of PHP globals of the same names. These variables may be used directly in templates.

array **DB**

Default database identifier for various SQL/NoSQL objects. Assign Data Source Name (DSN) to DB.dsn to establish a database connection. Set the values of DB.user, DB.password, DB.options and DB.attributes if needed. DB.pdo contains PDO instance representing the connection. For MongoDB, use DB.mdb. Result of the last query is saved in DB.result.

boolean **DEBUG**

Toggle switch for including framework internals in stack traces. Default is FALSE.

string **DNSBL**

Pipe-separated list of public blacklist services for looking up IP addresses of spammers.

string **E404**, string **E500**

Contains file names of custom HTTP 404 and 500 error templates. File should be in the path pointed to by the GUI framework variable.

string **ENCODING**

Character set used for document encoding. Default value is UTF-8.

array **ERROR**

Profile of the last HTTP error that occurred. ERROR.code is the HTTP error number (404 or 500). ERROR.title contains a brief description of the error. ERROR.text provides greater detail. For HTTP 500 errors, use ERROR.stack to display a stack trace.

string **EXEMPT**

Pipe-separated list of IP addresses that should not be cross-checked against the DNS blacklist of spammers. Default value is NULL. Framework automatically excludes all private IP address ranges from spam checks.

string **FONTS**

Path to font files used by the captcha() method. Default is the Web root.

array **FUNCS**

Contains a list of all functions permitted in templates .

string **GUI**

Location of graphical user interface files used by serve() and minify() methods. Default is the Web root folder.

array **HEADERS**

List of HTTP headers received by the http() method.

string **HOTLINK**

The URI for forwarding attempts to link directly to specific routes, images, media files, etc. Useful for preventing bandwidth theft.

string **IMPORTS**

Location of imported files used in the route(), input() and call() methods. Default is the Web root folder.

integer **LOGS**

Location of custom logs. Default value is the 'logs/' folder under the Web root.

integer **MAXSIZE**

Maximum file upload size in bytes derived from php.ini upload_max_filesize setting.

array **PARAMS**

Holds the captured values of tokens defined in a route() pattern. PARAMS.O contains the captured URL relative to the Web root.

boolean **QUIET**

Toggle switch for suppressing display output and error messages (useful for unit testing).

boolean **RELEASE**

Toggle switch for indicating release or development version of application. If FALSE (default), stack trace appears in both HTML error page and server's error log. If TRUE, only the error log contains the stack trace.

array **RESPONSE**

The body of the last HTTP response (particularly useful when output is suppressed by the QUIET flag).

string **SPAM**

The URI for redirecting a user identified by a DNS blacklist as a spammer.

string **SYNC**

Cache timeout for Axon sync() method. Default value is 60 seconds.

array **TEST**

Array containing the results of each expect() statement.

integer **THROTTLE**

The minimum number of milliseconds to serve a Web page (for regulating speed of HTTP response in limited bandwidth environments and/or during a denial-of-service attack).

float **TIME**

Contains a Unix timestamp (floating point) representing the instance when the framework was started.

Template Directives

@token

Replace token with global variable of the same name.

{ expression }

Evaluate expression and replace with result converted to string. expression may include template tokens, constants, operators (unary, arithmetic, ternary and relational), parentheses, data type converters, and PHP functions. See `allow()` method for restrictions.

<F3:include href="string filename"/>

Get contents of filename and insert at current position in template.

<F3:exclude> *text-block* </F3:exclude>

Remove text-block at runtime.

<F3:check if="boolean condition">

<F3:true> *true-block* </F3:true>

<F3:false> *false-block* </F3:false>

</F3:check>

Evaluate condition. If TRUE, true-block replaces the F3:check block. Otherwise, false-block is evaluated.

<F3:repeat [key="scalar @key"]

index="mixed @index" group="array @group">

text-block

</F3:repeat>

Repeat text-block as many times as there are elements in the array variable group.

{* comment *}

Similar to F3::exclude but should not contain HTML/XML tags.