* [Welcome to CodeIgniter](http://docs.google.com/general/welcome.html)
* [Installation Instructions](http://docs.google.com/installation/index.html)
  + [Downloading CodeIgniter](http://docs.google.com/installation/downloads.html)
  + [Installation Instructions](http://docs.google.com/installation/index.html)
  + [Upgrading From a Previous Version](http://docs.google.com/installation/upgrading.html)
  + [Troubleshooting](http://docs.google.com/installation/troubleshooting.html)
* [CodeIgniter Overview](http://docs.google.com/overview/index.html)
  + [Getting Started](http://docs.google.com/overview/getting_started.html)
  + [CodeIgniter at a Glance](http://docs.google.com/overview/at_a_glance.html)
  + [Supported Features](http://docs.google.com/overview/features.html)
  + [Application Flow Chart](http://docs.google.com/overview/appflow.html)
  + [Model-View-Controller](http://docs.google.com/overview/mvc.html)
  + [Architectural Goals](http://docs.google.com/overview/goals.html)
* [Tutorial](http://docs.google.com/tutorial/index.html)
  + [Static pages](http://docs.google.com/tutorial/static_pages.html)
  + [News section](http://docs.google.com/tutorial/news_section.html)
  + [Create news items](http://docs.google.com/tutorial/create_news_items.html)
  + [Conclusion](http://docs.google.com/tutorial/conclusion.html)
* [Contributing to CodeIgniter](http://docs.google.com/contributing/index.html)
  + [Writing CodeIgniter Documentation](http://docs.google.com/documentation/index.html)
  + [Developer’s Certificate of Origin 1.1](http://docs.google.com/DCO.html)
* [General Topics](http://docs.google.com/general/index.html)
  + [CodeIgniter URLs](http://docs.google.com/general/urls.html)
  + [Controllers](http://docs.google.com/general/controllers.html)
  + [Reserved Names](http://docs.google.com/general/reserved_names.html)
  + [Views](http://docs.google.com/general/views.html)
  + [Models](http://docs.google.com/general/models.html)
  + [Helpers](http://docs.google.com/general/helpers.html)
  + [Using CodeIgniter Libraries](http://docs.google.com/general/libraries.html)
  + [Creating Libraries](http://docs.google.com/general/creating_libraries.html)
  + [Using CodeIgniter Drivers](http://docs.google.com/general/drivers.html)
  + [Creating Drivers](http://docs.google.com/general/creating_drivers.html)
  + [Creating Core System Classes](http://docs.google.com/general/core_classes.html)
  + [Creating Ancillary Classes](http://docs.google.com/general/ancillary_classes.html)
  + [Hooks - Extending the Framework Core](http://docs.google.com/general/hooks.html)
  + [Auto-loading Resources](http://docs.google.com/general/autoloader.html)
  + [Common Functions](http://docs.google.com/general/common_functions.html)
  + [Compatibility Functions](http://docs.google.com/general/compatibility_functions.html)
  + [URI Routing](http://docs.google.com/general/routing.html)
  + [Error Handling](http://docs.google.com/general/errors.html)
  + [Caching](http://docs.google.com/general/caching.html)
  + [Profiling Your Application](http://docs.google.com/general/profiling.html)
  + [Running via the CLI](http://docs.google.com/general/cli.html)
  + [Managing your Applications](http://docs.google.com/general/managing_apps.html)
  + [Handling Multiple Environments](http://docs.google.com/general/environments.html)
  + [Alternate PHP Syntax for View Files](http://docs.google.com/general/alternative_php.html)
  + [Security](http://docs.google.com/general/security.html)
  + [PHP Style Guide](http://docs.google.com/general/styleguide.html)
* [Libraries](http://docs.google.com/index.html)
  + [Benchmarking Class](http://docs.google.com/benchmark.html)
  + [Caching Driver](http://docs.google.com/caching.html)
  + [Calendaring Class](http://docs.google.com/calendar.html)
  + [Shopping Cart Class](http://docs.google.com/cart.html)
  + [Config Class](http://docs.google.com/config.html)
  + [Email Class](http://docs.google.com/email.html)
  + [Encrypt Class](http://docs.google.com/encrypt.html)
  + [Encryption Library](http://docs.google.com/encryption.html)
  + [File Uploading Class](http://docs.google.com/file_uploading.html)
  + [Form Validation](http://docs.google.com/form_validation.html)
  + [FTP Class](http://docs.google.com/ftp.html)
  + [Image Manipulation Class](http://docs.google.com/image_lib.html)
  + [Input Class](http://docs.google.com/input.html)
  + [Javascript Class](http://docs.google.com/javascript.html)
  + [Language Class](http://docs.google.com/language.html)
  + [Loader Class](http://docs.google.com/loader.html)
  + [Migrations Class](http://docs.google.com/migration.html)
  + [Output Class](http://docs.google.com/output.html)
  + [Pagination Class](http://docs.google.com/pagination.html)
  + [Template Parser Class](http://docs.google.com/parser.html)
  + [Security Class](http://docs.google.com/security.html)
  + [Session Library](http://docs.google.com/sessions.html)
  + [HTML Table Class](http://docs.google.com/table.html)
  + [Trackback Class](#gjdgxs)
  + [Typography Class](http://docs.google.com/typography.html)
  + [Unit Testing Class](http://docs.google.com/unit_testing.html)
  + [URI Class](http://docs.google.com/uri.html)
  + [User Agent Class](http://docs.google.com/user_agent.html)
  + [XML-RPC and XML-RPC Server Classes](http://docs.google.com/xmlrpc.html)
  + [Zip Encoding Class](http://docs.google.com/zip.html)
* [Database Reference](http://docs.google.com/database/index.html)
  + [Quick Start: Usage Examples](http://docs.google.com/database/examples.html)
  + [Database Configuration](http://docs.google.com/database/configuration.html)
  + [Connecting to a Database](http://docs.google.com/database/connecting.html)
  + [Running Queries](http://docs.google.com/database/queries.html)
  + [Generating Query Results](http://docs.google.com/database/results.html)
  + [Query Helper Functions](http://docs.google.com/database/helpers.html)
  + [Query Builder Class](http://docs.google.com/database/query_builder.html)
  + [Transactions](http://docs.google.com/database/transactions.html)
  + [Getting MetaData](http://docs.google.com/database/metadata.html)
  + [Custom Function Calls](http://docs.google.com/database/call_function.html)
  + [Query Caching](http://docs.google.com/database/caching.html)
  + [Database Manipulation with Database Forge](http://docs.google.com/database/forge.html)
  + [Database Utilities Class](http://docs.google.com/database/utilities.html)
  + [Database Driver Reference](http://docs.google.com/database/db_driver_reference.html)
* [Helpers](http://docs.google.com/helpers/index.html)
  + [Array Helper](http://docs.google.com/helpers/array_helper.html)
  + [CAPTCHA Helper](http://docs.google.com/helpers/captcha_helper.html)
  + [Cookie Helper](http://docs.google.com/helpers/cookie_helper.html)
  + [Date Helper](http://docs.google.com/helpers/date_helper.html)
  + [Directory Helper](http://docs.google.com/helpers/directory_helper.html)
  + [Download Helper](http://docs.google.com/helpers/download_helper.html)
  + [Email Helper](http://docs.google.com/helpers/email_helper.html)
  + [File Helper](http://docs.google.com/helpers/file_helper.html)
  + [Form Helper](http://docs.google.com/helpers/form_helper.html)
  + [HTML Helper](http://docs.google.com/helpers/html_helper.html)
  + [Inflector Helper](http://docs.google.com/helpers/inflector_helper.html)
  + [Language Helper](http://docs.google.com/helpers/language_helper.html)
  + [Number Helper](http://docs.google.com/helpers/number_helper.html)
  + [Path Helper](http://docs.google.com/helpers/path_helper.html)
  + [Security Helper](http://docs.google.com/helpers/security_helper.html)
  + [Smiley Helper](http://docs.google.com/helpers/smiley_helper.html)
  + [String Helper](http://docs.google.com/helpers/string_helper.html)
  + [Text Helper](http://docs.google.com/helpers/text_helper.html)
  + [Typography Helper](http://docs.google.com/helpers/typography_helper.html)
  + [URL Helper](http://docs.google.com/helpers/url_helper.html)
  + [XML Helper](http://docs.google.com/helpers/xml_helper.html)

Toggle Table of Contents

[CodeIgniter](http://docs.google.com/index.html)

* [Welcome to CodeIgniter](http://docs.google.com/general/welcome.html)
* [Installation Instructions](http://docs.google.com/installation/index.html)
  + [Downloading CodeIgniter](http://docs.google.com/installation/downloads.html)
  + [Installation Instructions](http://docs.google.com/installation/index.html)
  + [Upgrading From a Previous Version](http://docs.google.com/installation/upgrading.html)
  + [Troubleshooting](http://docs.google.com/installation/troubleshooting.html)
* [CodeIgniter Overview](http://docs.google.com/overview/index.html)
  + [Getting Started](http://docs.google.com/overview/getting_started.html)
  + [CodeIgniter at a Glance](http://docs.google.com/overview/at_a_glance.html)
  + [Supported Features](http://docs.google.com/overview/features.html)
  + [Application Flow Chart](http://docs.google.com/overview/appflow.html)
  + [Model-View-Controller](http://docs.google.com/overview/mvc.html)
  + [Architectural Goals](http://docs.google.com/overview/goals.html)
* [Tutorial](http://docs.google.com/tutorial/index.html)
  + [Static pages](http://docs.google.com/tutorial/static_pages.html)
  + [News section](http://docs.google.com/tutorial/news_section.html)
  + [Create news items](http://docs.google.com/tutorial/create_news_items.html)
  + [Conclusion](http://docs.google.com/tutorial/conclusion.html)
* [Contributing to CodeIgniter](http://docs.google.com/contributing/index.html)
  + [Writing CodeIgniter Documentation](http://docs.google.com/documentation/index.html)
  + [Developer’s Certificate of Origin 1.1](http://docs.google.com/DCO.html)
* [General Topics](http://docs.google.com/general/index.html)
  + [CodeIgniter URLs](http://docs.google.com/general/urls.html)
  + [Controllers](http://docs.google.com/general/controllers.html)
  + [Reserved Names](http://docs.google.com/general/reserved_names.html)
  + [Views](http://docs.google.com/general/views.html)
  + [Models](http://docs.google.com/general/models.html)
  + [Helpers](http://docs.google.com/general/helpers.html)
  + [Using CodeIgniter Libraries](http://docs.google.com/general/libraries.html)
  + [Creating Libraries](http://docs.google.com/general/creating_libraries.html)
  + [Using CodeIgniter Drivers](http://docs.google.com/general/drivers.html)
  + [Creating Drivers](http://docs.google.com/general/creating_drivers.html)
  + [Creating Core System Classes](http://docs.google.com/general/core_classes.html)
  + [Creating Ancillary Classes](http://docs.google.com/general/ancillary_classes.html)
  + [Hooks - Extending the Framework Core](http://docs.google.com/general/hooks.html)
  + [Auto-loading Resources](http://docs.google.com/general/autoloader.html)
  + [Common Functions](http://docs.google.com/general/common_functions.html)
  + [Compatibility Functions](http://docs.google.com/general/compatibility_functions.html)
  + [URI Routing](http://docs.google.com/general/routing.html)
  + [Error Handling](http://docs.google.com/general/errors.html)
  + [Caching](http://docs.google.com/general/caching.html)
  + [Profiling Your Application](http://docs.google.com/general/profiling.html)
  + [Running via the CLI](http://docs.google.com/general/cli.html)
  + [Managing your Applications](http://docs.google.com/general/managing_apps.html)
  + [Handling Multiple Environments](http://docs.google.com/general/environments.html)
  + [Alternate PHP Syntax for View Files](http://docs.google.com/general/alternative_php.html)
  + [Security](http://docs.google.com/general/security.html)
  + [PHP Style Guide](http://docs.google.com/general/styleguide.html)
* [Libraries](http://docs.google.com/index.html)
  + [Benchmarking Class](http://docs.google.com/benchmark.html)
  + [Caching Driver](http://docs.google.com/caching.html)
  + [Calendaring Class](http://docs.google.com/calendar.html)
  + [Shopping Cart Class](http://docs.google.com/cart.html)
  + [Config Class](http://docs.google.com/config.html)
  + [Email Class](http://docs.google.com/email.html)
  + [Encrypt Class](http://docs.google.com/encrypt.html)
  + [Encryption Library](http://docs.google.com/encryption.html)
  + [File Uploading Class](http://docs.google.com/file_uploading.html)
  + [Form Validation](http://docs.google.com/form_validation.html)
  + [FTP Class](http://docs.google.com/ftp.html)
  + [Image Manipulation Class](http://docs.google.com/image_lib.html)
  + [Input Class](http://docs.google.com/input.html)
  + [Javascript Class](http://docs.google.com/javascript.html)
  + [Language Class](http://docs.google.com/language.html)
  + [Loader Class](http://docs.google.com/loader.html)
  + [Migrations Class](http://docs.google.com/migration.html)
  + [Output Class](http://docs.google.com/output.html)
  + [Pagination Class](http://docs.google.com/pagination.html)
  + [Template Parser Class](http://docs.google.com/parser.html)
  + [Security Class](http://docs.google.com/security.html)
  + [Session Library](http://docs.google.com/sessions.html)
  + [HTML Table Class](http://docs.google.com/table.html)
  + [Trackback Class](#gjdgxs)
  + [Typography Class](http://docs.google.com/typography.html)
  + [Unit Testing Class](http://docs.google.com/unit_testing.html)
  + [URI Class](http://docs.google.com/uri.html)
  + [User Agent Class](http://docs.google.com/user_agent.html)
  + [XML-RPC and XML-RPC Server Classes](http://docs.google.com/xmlrpc.html)
  + [Zip Encoding Class](http://docs.google.com/zip.html)
* [Database Reference](http://docs.google.com/database/index.html)
  + [Quick Start: Usage Examples](http://docs.google.com/database/examples.html)
  + [Database Configuration](http://docs.google.com/database/configuration.html)
  + [Connecting to a Database](http://docs.google.com/database/connecting.html)
  + [Running Queries](http://docs.google.com/database/queries.html)
  + [Generating Query Results](http://docs.google.com/database/results.html)
  + [Query Helper Functions](http://docs.google.com/database/helpers.html)
  + [Query Builder Class](http://docs.google.com/database/query_builder.html)
  + [Transactions](http://docs.google.com/database/transactions.html)
  + [Getting MetaData](http://docs.google.com/database/metadata.html)
  + [Custom Function Calls](http://docs.google.com/database/call_function.html)
  + [Query Caching](http://docs.google.com/database/caching.html)
  + [Database Manipulation with Database Forge](http://docs.google.com/database/forge.html)
  + [Database Utilities Class](http://docs.google.com/database/utilities.html)
  + [Database Driver Reference](http://docs.google.com/database/db_driver_reference.html)
* [Helpers](http://docs.google.com/helpers/index.html)
  + [Array Helper](http://docs.google.com/helpers/array_helper.html)
  + [CAPTCHA Helper](http://docs.google.com/helpers/captcha_helper.html)
  + [Cookie Helper](http://docs.google.com/helpers/cookie_helper.html)
  + [Date Helper](http://docs.google.com/helpers/date_helper.html)
  + [Directory Helper](http://docs.google.com/helpers/directory_helper.html)
  + [Download Helper](http://docs.google.com/helpers/download_helper.html)
  + [Email Helper](http://docs.google.com/helpers/email_helper.html)
  + [File Helper](http://docs.google.com/helpers/file_helper.html)
  + [Form Helper](http://docs.google.com/helpers/form_helper.html)
  + [HTML Helper](http://docs.google.com/helpers/html_helper.html)
  + [Inflector Helper](http://docs.google.com/helpers/inflector_helper.html)
  + [Language Helper](http://docs.google.com/helpers/language_helper.html)
  + [Number Helper](http://docs.google.com/helpers/number_helper.html)
  + [Path Helper](http://docs.google.com/helpers/path_helper.html)
  + [Security Helper](http://docs.google.com/helpers/security_helper.html)
  + [Smiley Helper](http://docs.google.com/helpers/smiley_helper.html)
  + [String Helper](http://docs.google.com/helpers/string_helper.html)
  + [Text Helper](http://docs.google.com/helpers/text_helper.html)
  + [Typography Helper](http://docs.google.com/helpers/typography_helper.html)
  + [URL Helper](http://docs.google.com/helpers/url_helper.html)
  + [XML Helper](http://docs.google.com/helpers/xml_helper.html)

[CodeIgniter](http://docs.google.com/index.html)

* [Docs](http://docs.google.com/index.html) »
* [Libraries](http://docs.google.com/index.html) »
* Trackback Class

classic layout

Trackback Class[**¶**](#30j0zll)

The Trackback Class provides functions that enable you to send and receive Trackback data.

If you are not familiar with Trackbacks you’ll find more information [here](http://en.wikipedia.org/wiki/Trackback).

* [Using the Trackback Class](#1fob9te)
  + [Initializing the Class](#3znysh7)
  + [Sending Trackbacks](#2et92p0)
  + [Receiving Trackbacks](#tyjcwt)
  + [Your Ping URL](#3dy6vkm)
  + [Creating a Trackback Table](#1t3h5sf)
  + [Processing a Trackback](#4d34og8)
    - [Notes:](#2s8eyo1)
* [Class Reference](#17dp8vu)

[**Using the Trackback Class**](#3rdcrjn)[**¶**](#1fob9te)

[**Initializing the Class**](#26in1rg)[**¶**](#3znysh7)

Like most other classes in CodeIgniter, the Trackback class is initialized in your controller using the $this->load->library() method:

$this->load->library('trackback');

Once loaded, the Trackback library object will be available using:

$this->trackback

[**Sending Trackbacks**](#lnxbz9)[**¶**](#2et92p0)

A Trackback can be sent from any of your controller functions using code similar to this example:

$this->load->library('trackback');  
  
$tb\_data = array(  
 'ping\_url' => 'http://example.com/trackback/456',  
 'url' => 'http://www.my-example.com/blog/entry/123',  
 'title' => 'The Title of My Entry',  
 'excerpt' => 'The entry content.',  
 'blog\_name' => 'My Blog Name',  
 'charset' => 'utf-8'  
);  
  
if ( ! $this->trackback->send($tb\_data))  
{  
 echo $this->trackback->display\_errors();  
}  
else  
{  
 echo 'Trackback was sent!';  
}

Description of array data:

* **ping\_url** - The URL of the site you are sending the Trackback to. You can send Trackbacks to multiple URLs by separating each URL with a comma.
* **url** - The URL to YOUR site where the weblog entry can be seen.
* **title** - The title of your weblog entry.
* **excerpt** - The content of your weblog entry.
* **blog\_name** - The name of your weblog.
* **charset** - The character encoding your weblog is written in. If omitted, UTF-8 will be used.

Note

The Trackback class will automatically send only the first 500 characters of your entry. It will also strip all HTML.

The Trackback sending method returns TRUE/FALSE (boolean) on success or failure. If it fails, you can retrieve the error message using:

$this->trackback->display\_errors();

[**Receiving Trackbacks**](#35nkun2)[**¶**](#tyjcwt)

Before you can receive Trackbacks you must create a weblog. If you don’t have a blog yet there’s no point in continuing.

Receiving Trackbacks is a little more complex than sending them, only because you will need a database table in which to store them, and you will need to validate the incoming trackback data. You are encouraged to implement a thorough validation process to guard against spam and duplicate data. You may also want to limit the number of Trackbacks you allow from a particular IP within a given span of time to further curtail spam. The process of receiving a Trackback is quite simple; the validation is what takes most of the effort.

[**Your Ping URL**](#1ksv4uv)[**¶**](#3dy6vkm)

In order to accept Trackbacks you must display a Trackback URL next to each one of your weblog entries. This will be the URL that people will use to send you Trackbacks (we will refer to this as your “Ping URL”).

Your Ping URL must point to a controller function where your Trackback receiving code is located, and the URL must contain the ID number for each particular entry, so that when the Trackback is received you’ll be able to associate it with a particular entry.

For example, if your controller class is called Trackback, and the receiving function is called receive, your Ping URLs will look something like this:

http://example.com/index.php/trackback/receive/entry\_id

Where entry\_id represents the individual ID number for each of your entries.

[**Creating a Trackback Table**](#44sinio)[**¶**](#1t3h5sf)

Before you can receive Trackbacks you must create a table in which to store them. Here is a basic prototype for such a table:

CREATE TABLE trackbacks (  
 tb\_id int(10) unsigned NOT NULL auto\_increment,  
 entry\_id int(10) unsigned NOT NULL default 0,  
 url varchar(200) NOT NULL,  
 title varchar(100) NOT NULL,  
 excerpt text NOT NULL,  
 blog\_name varchar(100) NOT NULL,  
 tb\_date int(10) NOT NULL,  
 ip\_address varchar(45) NOT NULL,  
 PRIMARY KEY `tb\_id` (`tb\_id`),  
 KEY `entry\_id` (`entry\_id`)  
);

The Trackback specification only requires four pieces of information to be sent in a Trackback (url, title, excerpt, blog\_name), but to make the data more useful we’ve added a few more fields in the above table schema (date, IP address, etc.).

[**Processing a Trackback**](#2jxsxqh)[**¶**](#4d34og8)

Here is an example showing how you will receive and process a Trackback. The following code is intended for use within the controller function where you expect to receive Trackbacks.:

$this->load->library('trackback');  
$this->load->database();  
  
if ($this->uri->segment(3) == FALSE)  
{  
 $this->trackback->send\_error('Unable to determine the entry ID');  
}  
  
if ( ! $this->trackback->receive())  
{  
 $this->trackback->send\_error('The Trackback did not contain valid data');  
}  
  
$data = array(  
 'tb\_id' => '',  
 'entry\_id' => $this->uri->segment(3),  
 'url' => $this->trackback->data('url'),  
 'title' => $this->trackback->data('title'),  
 'excerpt' => $this->trackback->data('excerpt'),  
 'blog\_name' => $this->trackback->data('blog\_name'),  
 'tb\_date' => time(),  
 'ip\_address' => $this->input->ip\_address()  
);  
  
$sql = $this->db->insert\_string('trackbacks', $data);  
$this->db->query($sql);  
  
$this->trackback->send\_success();

[**Notes:**](#z337ya)[**¶**](#2s8eyo1)

The entry ID number is expected in the third segment of your URL. This is based on the URI example we gave earlier:

http://example.com/index.php/trackback/receive/entry\_id

Notice the entry\_id is in the third URI segment, which you can retrieve using:

$this->uri->segment(3);

In our Trackback receiving code above, if the third segment is missing, we will issue an error. Without a valid entry ID, there’s no reason to continue.

The $this->trackback->receive() function is simply a validation function that looks at the incoming data and makes sure it contains the four pieces of data that are required (url, title, excerpt, blog\_name). It returns TRUE on success and FALSE on failure. If it fails you will issue an error message.

The incoming Trackback data can be retrieved using this function:

$this->trackback->data('item')

Where item represents one of these four pieces of info: url, title, excerpt, or blog\_name

If the Trackback data is successfully received, you will issue a success message using:

$this->trackback->send\_success();

Note

The above code contains no data validation, which you are encouraged to add.

[**Class Reference**](#3j2qqm3)[**¶**](#17dp8vu)

*class* CI\_Trackback[¶](#1y810tw) $data = array('url' => '', 'title' => '', 'excerpt' => '', 'blog\_name' => '', 'charset' => '')

Trackback data array.

$convert\_ascii = TRUE

Whether to convert high ASCII and MS Word characters to HTML entities.

send(*$tb\_data*)[¶](#4i7ojhp)

|  |  |
| --- | --- |
| Parameters: | * **$tb\_data** (*array*) – Trackback data |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Send trackback.

receive()[¶](#2xcytpi)

|  |  |
| --- | --- |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

This method simply validates the incoming TB data, returning TRUE on success and FALSE on failure. If the data is valid it is set to the $this->data array so that it can be inserted into a database.

send\_error([*$message = 'Incomplete information'*])[¶](#1ci93xb)

|  |  |
| --- | --- |
| Parameters: | * **$message** (*string*) – Error message |
| Return type: | void |

Responses to a trackback request with an error message.

Note

This method will terminate script execution.

send\_success()[¶](#3whwml4)

|  |  |
| --- | --- |
| Return type: | void |

Responses to a trackback request with a success message.

Note

This method will terminate script execution.

data(*$item*)[¶](#2bn6wsx)

|  |  |
| --- | --- |
| Parameters: | * **$item** (*string*) – Data key |
| Returns: | Data value or empty string if not found |
| Return type: | string |

Returns a single item from the response data array.

process(*$url*, *$data*)[¶](#qsh70q)

|  |  |
| --- | --- |
| Parameters: | * **$url** (*string*) – Target url * **$data** (*string*) – Raw POST data |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Opens a socket connection and passes the data to the server, returning TRUE on success and FALSE on failure.

extract\_urls(*$urls*)[¶](#3as4poj)

|  |  |
| --- | --- |
| Parameters: | * **$urls** (*string*) – Comma-separated URL list |
| Returns: | Array of URLs |
| Return type: | array |

This method lets multiple trackbacks to be sent. It takes a string of URLs (separated by comma or space) and puts each URL into an array.

validate\_url(*&$url*)[¶](#1pxezwc)

|  |  |
| --- | --- |
| Parameters: | * **$url** (*string*) – Trackback URL |
| Return type: | void |

Simply adds the *http://* prefix it it’s not already present in the URL.

get\_id(*$url*)[¶](#49x2ik5)

|  |  |
| --- | --- |
| Parameters: | * **$url** (*string*) – Trackback URL |
| Returns: | URL ID or FALSE on failure |
| Return type: | string |

Find and return a trackback URL’s ID or FALSE on failure.

convert\_xml(*$str*)[¶](#2p2csry)

|  |  |
| --- | --- |
| Parameters: | * **$str** (*string*) – Input string |
| Returns: | Converted string |
| Return type: | string |

Converts reserved XML characters to entities.

limit\_characters(*$str*[, *$n = 500*[, *$end\_char = '…'*]])[¶](#147n2zr)

|  |  |
| --- | --- |
| Parameters: | * **$str** (*string*) – Input string * **$n** (*int*) – Max characters number * **$end\_char** (*string*) – Character to put at end of string |
| Returns: | Shortened string |
| Return type: | string |

Limits the string based on the character count. Will preserve complete words.

convert\_ascii(*$str*)[¶](#3o7alnk)

|  |  |
| --- | --- |
| Parameters: | * **$str** (*string*) – Input string |
| Returns: | Converted string |
| Return type: | string |

Converts high ASCII text and MS Word special characterss to HTML entities.

set\_error(*$msg*)[¶](#23ckvvd)

|  |  |
| --- | --- |
| Parameters: | * **$msg** (*string*) – Error message |
| Return type: | void |

Set an log an error message.

display\_errors([*$open = '<p>'*[, *$close = '</p>'*]])[¶](#ihv636)

|  |  |
| --- | --- |
| Parameters: | * **$open** (*string*) – Open tag * **$close** (*string*) – Close tag |
| Returns: | HTML formatted error messages |
| Return type: | string |

Returns error messages formatted in HTML or an empty string if there are no errors.

[Next](http://docs.google.com/typography.html)   [Previous](http://docs.google.com/table.html)

© Copyright 2014 - 2018, British Columbia Institute of Technology. Last updated on Jun 12, 2018.

Built with [Sphinx](http://sphinx-doc.org/) using a [theme](https://github.com/snide/sphinx_rtd_theme) provided by [Read the Docs](https://readthedocs.org).