

# Simple Web Access

by Alberto Simões  
ams@di.uminho.pt



Unfortunately these last months I haven't had much time to learn new modules or to write about them. In this issue I'll present `LWP::Simple`, a well known module to retrieve documents from the Web. It's a simple wrapper around the `LWP` module, a full featured framework to develop any kind of web client I might need. The module is in the Perl `libwww` package.

## ■ Basic Usage -----

The most used function in `LWP::Simple` is `get`. It accepts a URL, retrieves it, and returns its content. As URLs are sometimes wrong, or the servers busy, I don't forget to test if the retrieval succeeded:

```
use LWP::Simple;

my $contents = get("http://use.perl.org/");

if( defined $contents ) {
    # ... do something with contents
}
else {
    # ... fail gracefully
}
```

While I might want to look to the contents of the URL and do something with it, I usually just want to make it available to the user by saving it (mostly if I am fetching a lot of pages), or by printing it. As these two tasks are very common, `LWP::Simple` provides two functions to help me performing these actions: `getstore` and `getprint`.

When I want to fetch a page and save it to a file I use `getstore`, a function that takes as arguments not just the URL to be fetched but the name of the file where I want to save its contents:

```
use LWP::Simple;

getstore(
    "http://use.perl.org/index.rss",
    "perlnews.rss"
);
```

With the same idea in mind, `LWP::Simple` gives me a `getprint` function, although I do not think it is that useful.

Its usage is straightforward: it just prints to the default file handle.

```
use LWP::Simple;

getprint( "http://use.perl.org/" );
```

While this code is simple, I like to give feedback to my users about the success of my operation. Although my first example checks the return contents from the `get` function, I am not doing the same with `getstore` or `getprint`. These two functions return the HTTP response code, but I don't know HTTP well enough to decide what the successful response codes are. `LWP::Simple` makes my life easier with the `is_success` and `is_error` functions that figure it out for me so I don't have to know:

```
use LWP::Simple;

my $code = getstore(
    "http://use.perl.org/index.rss",
    "perlnews.rss"
);

if( is_error($code) ) {
    die "Unfortunately Perl geeks have
    their server down!";
}
```

## ■ Looking to document headers -----

Web servers, when answering the HTTP protocol, give more information to clients than just the content. There are three main properties about web documents that I like to examine: the document MIME type (such as plain text, HTML, PDF or something else), the content size, and the last time the content was modified.

To access this information, I can ask `LWP::Simple` for the document headers with the `head` function. This function is mainly used in list context and returns a five item list that includes the MIME type, the content length, the last modification time, information about when the document expires (useful mostly for proxies), and the name (and version) of the web server: