NAME

PSP::CGI - CGI class for the Perl Script Pages

SYNOPSIS

```
 use PSP::CGI;
 my $cgi = new PSP::CGI();

 my $value = $cgi->param($key);
 my @values = $cgi->multiparam($key);

 my $param = $cgi->param();
 my $value = $param->{$key};

 my @keys = $cgi->params();

 my $upload_item = $cgi->upload($key);

 my $upload = $cgi->upload();
 my $upload_item = $upload->{$key};

 my $filename = $upload_item->{filename};
 my $size = $upload_item->{size};
 my $content = $upload_item->{content};

 my @keys = $cgi->uploads();

 my $value = $cgi->cookie($key);

 my $cookie = $cgi->cookie();
 my $value = $cookie->{$key};

 my @keys = $cgi->cookies();

 my $success = $cgi->setcookie($key, $value, $lifetime,
    domain => $domain,
   path   => $path,
   secure => $bool
  );

 $cgi->setheader(
   HeaderOne => ’foo’,
   HeaderTwo => ’bar’
  );

 my $text = $cgi->flushHeaders();
```

DESCRIPTION

This module provides object oriented CGI capabilities for the Perl Script Pages (PSP) embedded Perl engine. See *PSP::Engine* for more information.

OVERVIEW

Object Constructors

new

Reads GET and POST variables (CGI variables and file uploads) and Cookies, creates a new object of the PSP::CGI class. The new constructor is the only object constructor for this module. See *new object constructor* for details.

Object Methods

param

Returns either the valie of a named CGI parameter or a reference to the internal parameter hash. See *param method* for details.

multiparam

Returns the values of a named CGI parameter as an array, allowing direct access to parameters that were transmitted twice or more (i.e. from form elements with "multiple option" set). See *multiparam method* for details.

params

Returns an array of CGI parameter names supplied by the web browser. See *params method* for details.

upload

Returns either an upload item for a named HTTP upload or a reference to the internal uploads hash. Upload items are hashs containing more information about the upload, see syntax and $item description for *upload method* for details.

uploads

Returns an array of HTTP file upload names supplied by the web browser. See *uploads method* for details.

cookie

Returns either the value of a named cookie or a reference to the internal cookies hash. See *cookie method* for details.

cookies

Returns an array of cookie names supplied by the web browser. See *cookies method* for details.

setcookie

Places a cookie of given name and value for a defined lifetime on the HTTP header queue. Defining domain, path or HTTPS restrictions is optional. See *setcookie method* for details.

setheader

Places one or more headers of given names and values on the HTTP header queue. See *setheader method* for details.

flushHeaders

Returns the object's headers and cookie data in HTTP headers format and resets the data storage. See *flushHeaders method* for details.

SYNTAX

new Object Constructor

Reads GET and POST variables (CGI variables and file uploads) and Cookies, creates a new object of the PSP::CGI class.

```
use PSP::CGI;
my $cgi = new PSP::CGI();
```

Required Parameters

None.

Optional Parameters

None.

param Method

Returns either the value of a named CGI parameter or a reference to the internal parameter hash.

```
my $value = $cgi->param($key);

my $param = $cgi->param();
my $value = $param->{$key};
```

If a requested parameter was transmitted twice or more (i.e. for a multi select input form), the values are returned separated by ASCII NUL (0x00) characters. You convert these parameters to an array as follows:

```
my @array = split(/\x00/, $cgi->param($keyOfMultiselectInput));
```

This is done automatically by the *multiparam method*. See below.

Required Parameters

None.

Optional Parameters

$key

The name of the CGI parameter to evaluate.

Return Values

$value

The value of the named CGI parameter or undef if the named parameter did not exists.

$param

Returned only when no $key parameter was supplied or was undef. A reference to the internal parameter hash, allowing direct access to CGI parameters.

multiparam Method

Returns the values of a named CGI parameter as an array, allowing direct access to parameters that were transmitted twice or more (i.e. from form elements with "multiple option" set).

```
my @values = $cgi->multiparam($key);
```

Required Parameters

$key

The name of the CGI parameter to evaluate.

Optional Parameters

None.

Return Values

@values

An array containing one element per value found for the named CGI parameter. If the named parameter did not exist, an empty array is returned.

params Method

Returns an array of CGI parameter names supplied by the web browser.

```
 my @keys = $cgi->params();
```

Required Parameters

None.

Optional Parameters

None.

Return Values

@keys

Array of CGI parameter names/hash search keys.

upload Method

Returns either an upload item for a named HTTP upload or a reference to the internal uploads hash. Upload items are hashs containing more information about the upload, see syntax and $item description below.

```
 my $item = $cgi->upload($key);

 my $upload = $cgi->upload();
 my $item = $upload->{$key};

 my $filename = $item->{filename};
 my $size = $item->{size};
 my $content = $item->{content};
```

Required Parameters

None.

Optional Parameters

$key

The name of the HTTP upload to evaluate.

Return Values

$item

A reference to a hash structure containing the following elements: "filename", the name of the file uploaded; "size", size of the uploaded data in bytes; "content", the actual data uploaded.

$upload

Returned only when no $key parameter was supplied or was undef. A reference to the internal uploads hash, allowing direct access to HTTP uploads.

uploads Method

Returns an array of HTTP file upload names supplied by the web browser.

```
my @keys = $cgi->uploads();
```

Required Parameters

None.

Optional Parameters

None.

Return Values

@keys

Array of HTTP file upload names/hash search keys.

cookie Method

Returns either the value of a named cookie or a reference to the internal cookies hash.

```
my $value = $cgi->cookie($key);

my $cookie = $cgi->cookie();
my $value = $cookie->{$key};
```

Required Parameters

None.

Optional Parameters

$key

The name of the cookie to evaluate.

Return Values

$value

The value of the named cookie or undef if the named cookie did not exist.

$cookie

Returned only when no $key parameter was supplied or was undef. A reference to the internal cookies hash, allowing direct access to cookies.

cookies Method

Returns an array of cookie names supplied by the web browser.

```
my @keys = $cgi->cookies();
```

Required Parameters

None.

Optional Parameters

None.

Return Values

@keys

Array of cookie names/hash search keys.

setcookie Method

Places a cookie of given name and value for a defined lifetime on the HTTP header queue. Defining domain, path or HTTPS restrictions is optional.

```
my $success = $cgi->setcookie($key, $value, $lifetime,
  domain => $domain,
  path   => $path,
  secure => $bool
 );
```

Required Parameters

$key

The name of the cookie's key/value pair. If a cookie of the same name was set before, it will get overwritten.

$value

The value of the cookie's key/value pair.

$lifetime

Lifetime of the cookie in seconds (How long the browser shall store the cookie).

Optional Parameters

The following parameters are hash key/value parameters.

domain

A domain/host name to restrict the cookie to. By default the cookie will restricted to the server's hostname by the browser if te domain parameter is omitted.

path

Path to restrict the cookie to. By default the cookie will not be restricted to a path, allowing all files on the host name to receive the cookie.

secure

Boolean parameter, tells the browser to submit the cookie only if HTTPS is used. By default the cookie will be submitted regardless of the connection being encrypted or not.

Return Values

$success

Returns 1 on success or undef on error.

setheader Method

Places one or more headers of given names and values on the HTTP header queue.

```
$cgi->setheader(
  HeaderOne => ’foo’,
  HeaderTwo => ’bar’
 );
```

If the a header value contains \r or \n characters, everything beyond the first \r or \n will be removed. This prevents header injections.

Required Parameters

The method accepts a hash of one or more HTTP headers and corresponding values. Existing headers of same name will be overwritten. See example above.

Optional Parameters

None.

Return Values

None.

flushHeaders method

Returns the object's headers and cookie data in HTTP headers format and resets the data storage.

```
 my $text = $cgi->flushHeaders();
```

Required Parameters

None.

Optional Parameters

None.

Return Values

$text

A text string containing the headers and cookies set in HTTP headers format.

AUTHOR

Veit Wahlich

E-Mail: cru |at| zodia |dot| de

WWW: *http://home.ircnet.de/cru/*

VERSION

v0.7 Wednesday, 18 January 2006

COPYRIGHT/LICENSE