



Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'Apache2::Module.3pm'

\$ man Apache2::Module.3pm

libapache2-mod-perl2-2.0.12::doUserClibapache2-mod-perl2-2.0.12::docs::api::Apache2::Module(3pm)

NAME

Apache2::Module - Perl API for creating and working with Apache modules

Synopsis

```
use Apache2::Module ();

#Define a configuration directive

my @directives = (

    {

        name => 'MyDirective',

    }

);

Apache2::Module::add(__PACKAGE__, \@directives);

# iterate over the whole module list

for (my $modp = Apache2::Module::top_module(); $modp; $modp = $modp->next) {

    my $name          = $modp->name;

    my $index         = $modp->module_index;

    my $ap_api_major_version = $modp->ap_api_major_version;

    my $ap_api_minor_version = $modp->ap_api_minor_version;

    my $commands      = $modp->cmds;

}

# find a specific module

my $module = Apache2::Module::find_linked_module('mod_ssl.c');

# remove a specific module
```

```

$module->remove_loaded_module();

# access module configuration from a directive
sub MyDirective {
    my ($self, $parms, $args) = @_;
    my $srv_cfg = Apache2::Module::get_config($self, $parms->server);
    [...]
}

# test if an Apache module is loaded
if (Apache2::Module::loaded('mod_ssl.c')) {
    [...]
}

# test if a Perl module is loaded
if (Apache2::Module::loaded('Apache2::Status')) {
    [...]
}

```

Description

"Apache2::Module" provides the Perl API for creating and working with Apache modules

See [Apache Server Configuration Customization in Perl](#).

API

"Apache2::Module" provides the following functions and/or methods:

"add"

Add a module's custom configuration directive to Apache.

```
Apache2::Module::add($package, $cmds);
```

arg1: \$package (string)

the package of the module to add

arg2: \$cmds (ARRAY of HASH refs)

the list of configuration directives to add

ret: no return value

since: 2.0.00

See also [Apache Server Configuration Customization in Perl](#).

"ap_api_major_version"

Get the httpd API version this module was build against, not the module's version.

```
$major_version = $module->ap_api_major_version();
```

obj: \$module ("Apache2::Module object")

ret: \$major_version (integer)

since: 2.0.00

This method is used to check that module is compatible with this version of the server before loading it.

"ap_api_minor_version"

Get the module API minor version.

```
$minor_version = $module->ap_api_minor_version();
```

obj: \$module ("Apache2::Module object")

ret: \$minor_version (integer)

since: 2.0.00

"ap_api_minor_version()" provides API feature milestones.

It's not checked during module init.

"cmds"

Get the "Apache2::Command" object, describing all of the directives this module defines.

```
$command = $module->cmds();
```

obj: \$module ("Apache2::Module object")

ret: \$command ("Apache2::Command object")

since: 2.0.00

"get_config"

Retrieve a module's configuration. Used by configuration directives.

```
$cfg = Apache2::Module::get_config($class, $server, $dir_config);
```

```
$cfg = Apache2::Module::get_config($class, $server);
```

```
$cfg = $self->get_config($server, $dir_config);
```

```
$cfg = $self->get_config($server);
```

obj: \$module ("Apache2::Module object")

arg1: \$class (string)

The Perl package this configuration is for

arg2: \$server ("Apache2::ServerRec object")

The current server, typically "\$r->server" or "\$parms->server".

opt arg3: \$dir_config ("Apache2::ConfVector object")

By default, the configuration returned is the server level one. To retrieve the per directory configuration, use "\$r->per_dir_config" as a last argument.

ret: \$cfg (HASH reference)

A reference to the hash holding the module configuration data.

since: 2.0.00

See also Apache Server Configuration Customization in Perl.

"find_linked_module"

Find a module based on the name of the module

```
$module = Apache2::Module::find_linked_module($name);
```

arg1: \$name (string)

The name of the module ending in ".c"

ret: \$module ("Apache2::Module object")

The module object if found, "undef" otherwise.

since: 2.0.00

For example:

```
my $module = Apache2::Module::find_linked_module('mod_ssl.c');
```

"loaded"

Determine if a certain module is loaded

```
$loaded = Apache2::Module::loaded($module);
```

name: \$module (string)

The name of the module to search for.

If \$module ends with ".c", search all the modules, statically compiled and dynamically loaded.

If \$module ends with ".so", search only the dynamically loaded modules.

If \$module doesn't contain a ".", search the loaded Perl modules (checks %INC).

ret: \$loaded (boolean)

Returns true if the module is loaded, false otherwise.

since: 2.0.00

For example, to test if this server supports ssl:

```
if (Apache2::Module::loaded('mod_ssl.c')) {  
    [...]  
}
```

To test is this server dynamically loaded mod_perl:

```
if (Apache2::Module::loaded('mod_perl.so')) {  
    [...]
```

```
}
```

To test if "Apache2::Status" is loaded:

```
if (Apache2::Module::loaded('Apache2::Status')) {  
    [...]  
}
```

"module_index"

Get the index to this modules structures in config vectors.

```
$index = $module->module_index();  
obj: $module ( "Apache2::Module object" )  
ret: $index ( integer )  
since: 2.0.00
```

"name"

Get the name of the module's .c file

```
$name = $module->name();  
obj: $module ( "Apache2::Module object" )  
ret: $name ( string )  
since: 2.0.00
```

For example a mod_perl module, will return: mod_perl.c.

"next"

Get the next module in the list, "undef" if this is the last module in the list.

```
$next_module = $module->next();  
obj: $module ( "Apache2::Module object" )  
ret: $next_module ( "Apache2::Module object" )  
since: 2.0.00
```

"remove_loaded_module"

Remove a module from the list of loaded modules permanently.

```
$module->remove_loaded_module();  
obj: $module ( "Apache2::Module object" )  
ret: no return value  
since: 2.0.00
```

"top_module"

Returns the first module in the module list. Useful to start a module iteration.

```
$module = Apache2::Module::top_module();
```

ret: \$module ("Apache2::Module object")

since: 2.0.00

See Also

[mod_perl 2.0 documentation.](#)

Copyright

mod_perl 2.0 and its core modules are copyrighted under The Apache Software License, Version 2.0.

Authors

The mod_perl development team and numerous contributors.

perl v5.34.0

libapache2-mod-perl2-2.0.12::docs::api::Apache2::Module(3pm)