

1 Apache -- A ghost mod_perl 2.0 class

1.1 Synopsis

1.2 Description

There is no Apache class per se.

There are several modules that put their functions into the `Apache::` namespace. For example `ModPerl::Util` defines a function `Apache::current_callback()`:

```
use ModPerl::Util;
my $callback = Apache::current_callback();
```

There are several modules that require the *Apache* class as the first argument to the class methods that they define. For example `Apache::Server` defines a class method `Apache->server`:

```
use Apache::Server;
my $server = Apache->server;
```

There are several modules that install constants into the `Apache::` namespace. For example `Apache::ServerUtil` defines a constant `Apache::Server::server_root`:

```
use Apache::ServerUtil;
my $server_root = Apache::Server::server_root;
```

To use this functions and methods you need to load the module that defines them. If you aren't sure which module contains the symbol you are after, use the helper module `ModPerl::MethodLookup`.

1.3 See Also

`mod_perl 2.0` documentation.

1.4 Copyright

`mod_perl 2.0` and its core modules are copyrighted under The Apache Software License, Version 1.1.

1.5 Authors

The `mod_perl` development team and numerous contributors.

Table of Contents:

| | | |
|-----|--------------------------------------|---|
| 1 | Apache -- A ghost mod_perl 2.0 class | 1 |
| 1.1 | Synopsis | 2 |
| 1.2 | Description | 2 |
| 1.3 | See Also | 2 |
| 1.4 | Copyright | 2 |
| 1.5 | Authors | 2 |