Apache HTTP Server Version 2.4

Apache Module mod_dir

Description:	Provides for "trailing slash" redirects and serving directory index files
Status:	Base
Module Identifier:	dir_module
Source File:	mod_dir.c

Summary

The index of a directory can come from one of two sources:

- A file written by the user, typically called `index.html`. The `DirectoryIndex` directive sets the name of this file. This is controlled by `mod_dir`.
- Otherwise, a listing generated by the server. This is provided by `mod_autoindex`.

The two functions are separated so that you can completely remove (or replace) automatic index generation should you want to.

A "trailing slash" redirect is issued when the server receives a request for a URL `http://servername/foo/dirname` where `dirname` is a directory. Directories require a trailing slash, so `mod_dir` issues a redirect to `http://servername/foo/dirname/`.

DirectoryCheckHandler Directive

Description:	Toggle how this module responds when another handler is configured
Syntax:	`DirectoryCheckHandler On|Off`
Default:	`DirectoryCheckHandler Off`
Context:	server config, virtual host, directory, .htaccess
Override:	Indexes
Status:	Base
Module:	mod_dir
Compatibility:	Available in 2.4.8 and later. Releases prior to 2.4 implicitly act as if "DirectoryCheckHandler ON" was specified.

The `DirectoryCheckHandler` directive determines whether `mod_dir` should check for directory indexes or add trailing slashes when some other handler has been configured for the current URL. Handlers can be set by directives such as `SetHandler` or by other modules, such as `mod_rewrite` during per-directory substitutions.

In releases prior to 2.4, this module did not take any action if any other handler was configured for a URL. This allows directory indexes to be served even when a `SetHandler` directive is specified for an entire directory, but it can also result in some conflicts with modules such as `mod_rewrite`.

DirectoryIndex Directive

Description:	List of resources to look for when the client requests a directory
Syntax:	`DirectoryIndex disabled | *local-url* [*local-url*] ...`
Default:	`DirectoryIndex index.html`
Context:	server config, virtual host, directory, .htaccess
Override:	Indexes
Status:	Base
Module:	mod_dir

The `DirectoryIndex` directive sets the list of resources to look for, when the client requests an index of the directory by specifying a / at the end of the directory name. *Local-url* is the (%-encoded) URL of a document on the server relative to the requested directory; it is usually the name of a file in the directory. Several URLs may be given, in which case the server will return the first one that it finds. If none of the resources exist and the `Indexes` option is set, the server will generate its own listing of the directory.

Example

```
DirectoryIndex index.html
```

then a request for `http://example.com/docs/` would return `http://example.com/docs/index.html` if it exists, or would list the directory if it did not.

Note that the documents do not need to be relative to the directory;

```
DirectoryIndex index.html index.txt  /cgi-bin/index.pl
```

would cause the CGI script `/cgi-bin/index.pl` to be executed if neither `index.html` or `index.txt` existed in a directory.

A single argument of "disabled" prevents `mod_dir` from searching for an index. An argument of "disabled" will be interpreted literally if it has any arguments before or after it, even if they are "disabled" as well.

Note: Multiple `DirectoryIndex` directives within the *same context* (↗ ../sections.html) will add to the list of resources to look for rather than replace:

```
# Example A: Set index.html as an index page, then add index.php to that list as well.
<Directory "/foo">
    DirectoryIndex index.html
    DirectoryIndex index.php
</Directory>

# Example B: This is identical to example A, except it's done with a single directive.
<Directory "/foo">
    DirectoryIndex index.html index.php
</Directory>

# Example C: To replace the list, you must explicitly reset it first:
# In this example, only index.php will remain as an index resource.
<Directory "/foo">
    DirectoryIndex index.html
    DirectoryIndex disabled
    DirectoryIndex index.php
</Directory>
```

DirectoryIndexRedirect Directive

Description:	Configures an external redirect for directory indexes.
Syntax:	`DirectoryIndexRedirect on	off	permanent	temp	seeother	*3xx-code*`
Default:	`DirectoryIndexRedirect off`
Context:	server config, virtual host, directory, .htaccess
Override:	Indexes
Status:	Base
Module:	mod_dir
Compatibility:	Available in version 2.3.14 and later

By default, the `DirectoryIndex` is selected and returned transparently to the client. `DirectoryIndexRedirect` causes an external redirect to instead be issued.

The argument can be:

- `on`: issues a 302 redirection to the index resource.
- `off`: does not issue a redirection. This is the legacy behaviour of mod_dir.
- `permanent`: issues a 301 (permanent) redirection to the index resource.
- `temp`: this has the same effect as `on`
- `seeother`: issues a 303 redirection (also known as "See Other") to the index resource.
- *3xx-code*: issues a redirection marked by the chosen 3xx code.

Example

```
DirectoryIndexRedirect on
```

A request for `http://example.com/docs/` would return a temporary redirect to `http://example.com/docs/index.html` if it exists.

DirectorySlash Directive

Description:	Toggle trailing slash redirects on or off
Syntax:	`DirectorySlash On|Off`
Default:	`DirectorySlash On`
Context:	server config, virtual host, directory, .htaccess
Override:	Indexes
Status:	Base
Module:	mod_dir

The `DirectorySlash` directive determines whether `mod_dir` should fixup URLs pointing to a directory or not.

Typically if a user requests a resource without a trailing slash, which points to a directory, `mod_dir` redirects him to the same resource, but *with* trailing slash for some good reasons:

- The user is finally requesting the canonical URL of the resource
- `mod_autoindex` works correctly. Since it doesn't emit the path in the link, it would point to the wrong path.
- `DirectoryIndex` will be evaluated *only* for directories requested with trailing slash.
- Relative URL references inside html pages will work correctly.

If you don't want this effect *and* the reasons above don't apply to you, you can turn off the redirect as shown below. However, be aware that there are possible security implications to doing this.

```
# see security warning below!
<Location "/some/path">
    DirectorySlash Off
    SetHandler some-handler
</Location>
```

Security Warning

Turning off the trailing slash redirect may result in an information disclosure. Consider a situation where `mod_autoindex` is active (`Options +Indexes`) and `DirectoryIndex` is set to a valid resource (say, `index.html`) and there's no other special handler defined for that URL. In this case a request with a trailing slash would show the `index.html` file. **But a request without trailing slash would list the directory contents**.

Also note that some browsers may erroneously change POST requests into GET (thus discarding POST data) when a redirect is issued.

FallbackResource Directive

Description:	Define a default URL for requests that don't map to a file
Syntax:	`FallbackResource disabled	*local-url*`
Default:	`disabled - httpd will return 404 (Not Found)`
Context:	server config, virtual host, directory, .htaccess
Override:	Indexes
Status:	Base
Module:	mod_dir
Compatibility:	The `disabled` argument is available in version 2.4.4 and later

Use this to set a handler for any URL that doesn't map to anything in your filesystem, and would otherwise return HTTP 404 (Not Found). For example

```
FallbackResource /not-404.php
```

will cause requests for non-existent files to be handled by `not-404.php`, while requests for files that exist are unaffected.

It is frequently desirable to have a single file or resource handle all requests to a particular directory, except those requests that correspond to an existing file or script. This is often referred to as a 'front controller.'

In earlier versions of httpd, this effect typically required `mod_rewrite`, and the use of the `-f` and `-d` tests for file and directory existence. This now requires only one line of configuration.

```
FallbackResource /index.php
```

Existing files, such as images, css files, and so on, will be served normally.

Use the `disabled` argument to disable that feature if inheritance from a parent directory is not desired.

In a sub-URI, such as *http://example.com/blog/* this *sub-URI* has to be supplied as *local-url*:

```
<Directory "/web/example.com/htdocs/blog">
    FallbackResource /blog/index.php
</Directory>
<Directory "/web/example.com/htdocs/blog/images">
    FallbackResource disabled
</Directory>
```

A fallback handler (in the above case, `/blog/index.php`) can access the original requested URL via the server variable `REQUEST_URI`. For example, to access this variable in PHP, use `$_SERVER['REQUEST_URI']`.

Comments

Notice:
This is not a Q&A section. Comments placed here should be pointed towards suggestions on improving the documentation or server, and may be removed by our moderators if they are either implemented or considered invalid/off-topic. Questions on how to manage the Apache HTTP Server should be directed at either our IRC channel, #httpd, on Libera.chat, or sent to our mailing lists.