

NAME

Lighttpd

Standalone server mode

Static mode

Non-root configuration

Static file handling

AUTHORS

COPYRIGHT

NAME

Catalyst::Manual::Deployment::lighttpd::FastCGI - Deploying Catalyst with lighttpd

Lighttpd

These configurations were tested with Lighttpd 1.4.7.

Standalone server mode

```
server.document-root = "/var/www/MyApp/root"

fastcgi.server = (
    "" => (
        "MyApp" => (
            "socket"      => "/tmp/myapp.socket",
            "check-local" => "disable"
        )
    )
)
```

Static mode

```
server.document-root = "/var/www/MyApp/root"

fastcgi.server = (
    "" => (
        "MyApp" => (
            "socket"      => "/tmp/myapp.socket",
            "check-local" => "disable",
            "bin-path"    => "/var/www/MyApp/script/myapp_fastcgi.pl",
            "min-procs"   => 2,
            "max-procs"   => 5,
            "idle-timeout" => 20
        )
    )
)
```

Note that in newer versions of lighttpd, the min-procs and idle-timeout values are disabled. The above example would start 5 processes.

Non-root configuration

You can also run your application at any non-root location with either of the above modes. Note the required `mod_rewrite` rule.

```
url.rewrite = ( "myapp\$" => "myapp/" )
fastcgi.server = (
    "/myapp" => (
        "MyApp" => (
            # same as above
        )
    )
)
```

For more information on using FastCGI under Lighttpd, visit <http://www.lighttpd.net/documentation/fastcgi.html>

Static file handling

Static files can be served directly by lighttpd for a performance boost.

```
$HTTP["url"] !~ "^/(?:(img/|static/|css/|favicon.ico$)" {
    fastcgi.server = (
        "" => (
            "MyApp" => (
                "socket"      => "/tmp/myapp.socket",
                "check-local" => "disable",
            )
        )
    )
}
```

This will serve everything in the `img`, `static`, and `css` directories statically, as well as the `favicon` file.

AUTHORS

Catalyst Contributors, see `Catalyst.pm`

COPYRIGHT

This library is free software. You can redistribute it and/or modify it under the same terms as Perl itself.

syntax highlighting: no syntax highlighting ▼