NAME

Catalyst::Manual::Deployment::Apache::FastCGI - Deploying Catalyst with FastCGI on Apache

Setup

1. Install Apache with mod_fastcgi

mod_fastcgi for Apache is a third-party module, and can be found at http://www.fastcgi.com/. It is also packaged in many distributions (for example, libapache2-mod-fastcgi in Debian). You will also need to install the FCGI module from CPAN.

Important Note! If you experience difficulty properly rendering pages, try disabling Apache's mod_deflate (Deflate Module), e.g. 'a2dismod deflate'.

Apache 1.x, 2.x

Apache requires the mod_fastcgi module. The same module supports both Apache 1 and 2.

There are three ways to run your application under FastCGI on Apache: server, static, and dynamic.

Standalone server mode

```
FastCgiExternalServer /tmp/myapp.fcgi -socket /tmp/myapp.socket
Alias /myapp/ /tmp/myapp.fcgi/

# Or, run at the root
Alias / /tmp/myapp.fcgi/

# Optionally, rewrite the path when accessed without a trailing slash
RewriteRule ^/myapp$ myapp/ [R]
```

The FastCgiExternalServer directive tells Apache that when serving /tmp/myapp to use the FastCGI application listening on the socket /tmp/mapp.socket. Note that /tmp/myapp.fcgi **MUST NOT** exist -- it's a virtual file name. With some versions of `mod_fastcgi` or `mod_fcgid`, you can use any name you like, but some require that the virtual filename end in `.fcgi`.

It's likely that Apache is not configured to serve files in /tmp, so the Alias directive maps the url path /myapp/ to the (virtual) file that runs the FastCGI application. The trailing slashes are important as their use will correctly set the PATH_INFO environment variable used by Catalyst to determine the request path. If you would like to be able to access your app without a trailing slash (http://server/myapp), you can use the above RewriteRule directive.

Static mode

The term 'static' is misleading, but in static mode Apache uses its own FastCGI Process Manager to start the application processes. This happens at Apache startup time. In this case you do not run your application's fastcgi.pl script -- that is done by Apache. Apache then maps URIs to the FastCGI script to run your application.

```
    FastCgiServer /path/to/myapp/script/myapp_fastcgi.pl -processes 3
    Alias /myapp/ /path/to/myapp/script/myapp_fastcgi.pl/
```

FastCgiServer tells Apache to start three processes of your application at startup. The Alias command maps a path to the FastCGI application. Again, the trailing slashes are important.

Dynamic mode

In FastCGI dynamic mode, Apache will run your application on demand, typically by requesting a file with a specific extension (e.g. .fcgi). ISPs often use this type of setup to provide FastCGI support to many customers.

In this mode it is often enough to place or link your *_fastcgi.pl script in your cgi-bin directory with the extension of .fcgi. In dynamic mode Apache must be able to run your application as a CGI script so ExecCGI must be enabled for the directory.

```
    AddHandler fastcgi-script .fcgi
```

The above tells Apache to run any .fcgi file as a FastCGI application.

Here is a complete example:

```
    <VirtualHost *:80>
        ServerName www.myapp.com
        DocumentRoot /path/to/MyApp

        # Allow CGI script to run
        <Directory /path/to/MyApp>
            Options +ExecCGI
        </Directory>

        # Tell Apache this is a FastCGI application
        <Files myapp_fastcgi.pl>
            SetHandler fastcgi-script
        </Files>
    </VirtualHost>
```

Then a request for /script/myapp_fastcgi.pl will run the application.

For more information on using FastCGI under Apache, visit
http://www.fastcgi.com/mod_fastcgi/docs/mod_fastcgi.html

Authorization header with mod_fastcgi or mod_cgi

By default, mod_fastcgi/mod_cgi do not pass along the Authorization header, so modules like `Catalyst::Plugin::Authentication::Credential::HTTP` will not work. To enable pass-through of this header, add the following mod_rewrite directives:

```
    RewriteCond %{HTTP:Authorization} ^(.+)
    RewriteRule ^(.*)$ $1 [E=HTTP_AUTHORIZATION:%1,PT]
```

2. Configure your application

```
    # Serve static content directly
    DocumentRoot  /var/www/MyApp/root
    Alias /static /var/www/MyApp/root/static

    FastCgiServer /var/www/MyApp/script/myapp_fastcgi.pl -processes 3
    Alias /myapp/ /var/www/MyApp/script/myapp_fastcgi.pl/

    # Or, run at the root
    Alias / /var/www/MyApp/script/myapp_fastcgi.pl/
```

The above commands will launch 3 app processes and make the app available at /myapp/

Standalone server mode

While not as easy as the previous method, running your app as an external server gives you much more flexibility.

First, launch your app as a standalone server listening on a socket.

```
    script/myapp_fastcgi.pl -l /tmp/myapp.socket -n 5 -p /tmp/myapp.pid -d
```

You can also listen on a TCP port if your web server is not on the same machine.

```
    script/myapp_fastcgi.pl -l :8080 -n 5 -p /tmp/myapp.pid -d
```

You will probably want to write an init script to handle starting/stopping of the app using the pid file.

Now, we simply configure Apache to connect to the running server.

```
    # 502 is a Bad Gateway error, and will occur if the backend server is down
    # This allows us to display a friendly static page that says "down for
    # maintenance"
    Alias /_errors /var/www/MyApp/root/error-pages
    ErrorDocument 502 /_errors/502.html

    FastCgiExternalServer /tmp/myapp.fcgi -socket /tmp/myapp.socket
    Alias /myapp/ /tmp/myapp.fcgi/
```

```
    # Or, run at the root
    Alias / /tmp/myapp.fcgi/
```

More Info

Catalyst::Manual::Deployment::FastCGI.

AUTHORS

Catalyst Contributors, see Catalyst.pm

COPYRIGHT