NAME

Catalyst::Manual::Deployment - Deploying Catalyst

DEPLOYMENT OPTIONS

Catalyst applications are most often deployed as a FastCGI or mod_perl application (with FastCGI being the recommended option). However, as Catalyst is based on the PSGI specification, any web handler implementing that specification can be used to run Catalyst applications.

This documentation most thoroughly covers the normal and traditional deployment options, but will mention alternate methods of deployment, and we welcome additional documentation from people deploying Catalyst in non-standard environments.

Deployment in a shared hosting environment

Almost all shared hosting environments involve deploying Catalyst as a FastCGI application on Apache. You will usually want to have a set of libraries specific to your application installed on your shared host.

Full details of deploying Catalyst in a shared hosting environment are at Catalyst::Manual::Deployment::SharedHosting.

FastCGI

FastCGI is the most common Catalyst deployment option. It is documented generally in Catalyst::Manual::Deployment::FastCGI, and there are specific instructions for using FastCGI with common web servers below:

Apache

Catalyst::Manual::Deployment::Apache::FastCGI

nginx

Catalyst::Manual::Deployment::nginx::FastCGI

lighttpd

Catalyst::Manual::Deployment::lighttpd::FastCGI

Microsoft IIS

Catalyst::Manual::Deployment::IIS::FastCGI

mod_perl

Traditionally a common deployment option for dedicated applications, mod_perl has some advantages and disadvantages over FastCGI. Use of mod_perl is documented in Catalyst::Manual::Deployment::Apache::mod_perl.

Development Server

It is possible to deploy the Catalyst development server behind a reverse proxy. This may work well for small-scale applications which are in an early development phase, but which you want to be able to show to people. See Catalyst::Manual::Deployment::DevelopmentServer.

PSGI

Catalyst can be deployed with any PSGI-compliant handler. See Catalyst::PSGI for more information; a list of possible deployment servers are shown below:

Starman

Starman is a high-performance Perl server implementation, which is designed to be used directly (rather than behind a reverse proxy). It includes HTTP/1.1 support, chunked requests and responses, keep-alive, and pipeline requests.

Starlet

Starlet is a standalone HTTP/1.0 server with keep-alive support which is suitable for running HTTP application servers behind a reverse proxy.

Twiggy

Twiggy is a high-performance asynchronous web server. It can be used in conjunction with Catalyst, but there are a number of caveats which mean that it is not suitable for most deployments.

Chef

Chef is an open-source systems integration framework built specifically for automating cloud computing deployments. A Cookbooks demonstrating how to deploy a Catalyst application using Chef is available at http://community.opscode.com/cookbooks/catalyst and http://github.com/melezhik/cookbooks/wiki/Catalyst-cookbook-intro.

AUTHORS

Catalyst Contributors, see Catalyst.pm

COPYRIGHT