TurboCMF

A lightweight framework for database–driven web applications.

Presentation Overview

High–Level Overview

Technical Overview

Gritty Details

High–Level Overview

It’s a framework.

Frameworks are good.

They make jobs easier.

(Leveraging existing code.)

Based on TurboGears 2.

Which is based on Pylons.

Which is based on Paster.

Which equals a lot of code you don’t have to write yourself.

It’s data-based, not code-based.

99% of things can be accomplished from the web interface.

No mucking about.

Getting off the ground is easy.

(One line in a terminal.)

Skinning is easy.

(A single template to modify.)

(Use CSS everywhere.)

Lots of standards to choose from.

XHTML 1.1

http://www.w3.org/TR/xhtml11/

XPath 1.0

http://www.w3.org/TR/xpath

XInclude 1.0

http://www.w3.org/TR/xinclude/

CSS

http://www.w3.org/Style/CSS/

RSS & Atom

http://en.wikipedia.org/wiki/RSS_(file_format)
http://en.wikipedia.org/wiki/Atom_(standard)

Lots of other acronyms.

None of which really matter.

Everything is modular.

Everything.

Modular.

Really.

Technical Overview

Everyone’s favorite new acronym.

WSGI

(We held this one in reserve earlier for greater impact.)

Framework on a framework.

Okay, yes, there’s a lot of code.

Offers flexibility.

Lets you choose the best of breed components for your application.

TurboCMF is very modular.

Everything on a site is an Asset.

Everything has a path.

Everything has an order.

Everything may be a container.

May. Some things aren’t.

Some things are singletons.

(Shopping carts.)

(Authentication mechanisms.)

Some sample Asset types include:

Page

A container for static content, usually shown as a concrete page.

Folder

A general purpose organizational container.

Account

A place to store authentication information e.g. a password.

Navigation

A configurable list of links.

Alias

A redirector and convenient click counter.

Cart

A shopping cart management and checkout interface.

Subscription

A recurring purchase.

Feed

A provider of RSS and Atom feeds for its containing asset.

Etc.

(There are a lot of them.)

Application Server

Application using Assets

Custom Application

CMF co–exists peacefully.

New & Old

Getting Started

Install TurboGears

Follow the online guide: http://www.turbogears.org/2.0/docs/

Install Desired Libraries

MySQL, PIL, ReST, &c.

Install TurboCMF & Components

Online guides are good: http://docs.turbocmf.org/

Create your site.

```
paster quickcmf
```

Answer a few questions.

Site name, database connection, default user…

Visit your new site.

http://localhost:8080/

Questions?

Gritty Detail

(Things to frighten children with.)

MVC

Model, View, Controller

Model

SQLAlchemy + Elixir

View

Genshi XHTML Templates

Controller

Standard Python Classes

Instance = Directory

Method = File

CMF Controllers

Special naming conventions.

_action_foo(…)

http://…/action:foo

_view_bar(…)

http://…/view:foo

Example Controller

```
from cmf.core import View
from cmf.components.asset import controller

class Foo(controller.Asset):
  bar = View("Bar", "My cool view.")

  @expose("genshi:...foo.views.hello")
  def _view_bar(self):
    return dict(message="Hello World")
```