Accelerated Pyramid

Syllabus for Summer 2017

https://iotopia-solutions.com/training/AcceleratedPyramid

Course Objectives

This course will get you productive with Pyramid in just three days. After this course you'll be well-equipped to create a Pyramid-based site backed with either the ZODB or a SQL database. You'll also learn how to use Pyramid to make a ReSTful API that can safely receive and share JSON data, and you'll learn good server-side testing methodology.

Course Content

This course will focus on using Pyramid by example. It will teach you how to make both page and JSON resources and how to attach them to templates. It will teach you how to make templates using Chameleon, Mako, or Jinja2, and how to use both traversal and URL dispatch. It will teach you how to configure basic authentication and authorization. Pyramid is extremely lightweight and fast and is one of the most thoroughly tested Web frameworks available today, and after taking this course you'll be able to put it to work for you.

Student Background

Before attending this course you should have some familiarity with Python. Pyramid is a Python framework, and we will not be trying to replicate the content of our other course, Accelerated Python.

Computer Requirements

Modern Python can be run on most hardware in use today. For the purposes of this class we'd recommend you use a reasonably new laptop running something like Mac OS X (10.6+), Microsoft Windows 7+, or any recent version of Linux.

Suggested Reading

The ultimate Pyramid reference is maintained by the Pyramid team themselves online at:
https://docs.pylonsproject.org/projects/pyramid/en/latest/#narrative-documentation
If you like the feel of a book in your hand, *The repoze.bfg Web Application Framework* by Chris McDonough is really dated (repoze.bfg was the old name for Pyramid and quite a bit has changed) but probably your best paper option. It's strongly recommended that you follow the online documentation instead.

Class Schedule

This class will run for three eight-hour days. Each day will be broken up with a break in the morning, afternoon, and at lunch time. Work periods will consist of a lecture on a topic followed by exercises to reinforce it.

Day 1: Routes & Resources

Models and backing stores. Views. Resources. Requests, responses, query strings, and cookies. Routes using traversal and/or URL dispatch. ZODB vs. SQL vs. other. Static resources. Separation of concerns. Addition of CSS and JavaScript.

Day 2: Templates & Rendering

JSON resource rendering and ReST. Fine tuning headers and response codes. Rendering via templates with Chameleon, Mako, or Jinja2. Syntax for ZPT and Mako. Event handling. Hooks. Unit testing.

Day 3: For Deeper Understanding

More in-depth coverage of database connectivity. System interrogation examples. Authentication & authorization. ACLs. Sessions. Deployment. Virtualenv. SupervisorD. Working with Apache or NginX. Caching. Compression. SSL. Further separation of concerns: configuration via .INI files.