Pylons ユーザのための Pyramid 移行ガイド

Author: Nozomu Kaneko

Publication: Django & Pyramid Con JP 2012
(in conjunction with Pycon JP 2012)

Date: 2012-09-16

お前誰よ

- Twitter: @knzm2011
- 勤務先: TriAx (Silver スポンサー)
- Pylons を使い始めて5年ぐらい
  - Python 歴もだいたい同じくらい
  - 入社後に Python に触り始めて、気がついたら CMS フレームワークを作ってた
- 翻訳とか
  - Pythonドキュメント日本語翻訳プロジェクト
  - Pylons プロジェクトドキュメント / Pyramid ドキュメント
  - PEP 333: Python Web Server Gateway Interface v1.0
  - Pylons 0.9.7 ドキュメント
- Pylons Project JP の中の人
- 実は Pyramid はあまり使ってない (これから...)

この発表について

- こんな人におすすめ
  - 今現在 Pylons を使っている
  - Pyramid に移行したいけど、どこから手を付けたらいいか分からない
  - Pyramid でオレオレフレームワークが作りたい
  - = 数ヶ月前の自分

Pyramid FAQ

- Pyramid と Pylons の関係は?
  - Pyramid は repoze.bfg と Pylons 1.0 が合流してできたウェブフレームワーク
  - Pyramid を開発しているプロジェクトの名前は Pylons プロジェクト
  - ウェブフレームワークとしての Pylons もまだ現役
- repoze.bfg とは何ですか?
  - repoze = Zope 由来のコンポーネントを WSGI アプリケーションで利用できるようにしたコンポーネント集
  - repoze.bfg = repoze のコンポーネントを再構成したフレームワーク
- Pylons 1.0 プロジェクトを Pyramid に移植すべきですか?
  - 後で詳しく説明します
- Python 3 で動きますか?
  - Pyramid 1.3 以降であれば Python 3.2 以上で動きます (thanks to @aodag)

Pylons に何が起こったか

- Pylons 1.0 までは順調に開発が進んだ
  - 特に、周辺ライブラリも含めた Pylons スタックは WSGI ベースのフレームワーク として代表的な存在に
- 拡張性に深刻な問題があることに気づいた
  - 2010年11月 Ben Bangert によるblog記事: “Why Extending Through Subclassing (a framework’s classes) is a Bad Idea”
- Pylons 2 の開発を進めるうちに repoze.bfg と似てきたことで、完全合併の方向へとシフト

Pylons 1.xの何が悪かったのか

- Pylons では、フレームワークの提供するベースクラスをサブクラス化することでプロジェクトを作成
  - ユーザはカスタマイズが必要なメソッドを自由にオーバーライドする
- フレームワークを改良したくても、すべての主要なメソッドは事実上の API として凍結されていた
- その他の理由
  - パフォーマンスの問題
  - 単体テストしづらい
  - 多重継承による奇妙な衝突が発生
- 結論: サブクラス化によるフレームワークの拡張はすべきでない

Pyramid の特徴

- Pylons と repoze.bfg それぞれに由来する豊富な機能 (※)
  - ルーティング: URLディスパッチ or トラバーサル
  - データベースエンジン: SQLAlchemy or ZODB
  - テンプレートエンジン: Mako or Chameleon
  - アクセス制御: ACL
  - scaffold
  - インタラクティブデバッガー
  - 様々な方法でアプリケーションを拡張可能なフック (後述)
- これまでの開発の教訓
  - 徹底したテストコード
  - 徹底したドキュメンテーション
  - サブクラス化に頼らない拡張方法

※ フレームワークが乱立することを防ぐため、Pyramid ではフレームワーク内である程度の機能の重複があることは想定内とされている

Pylons 1.0 プロジェクトを Pyramid に移植すべき3つの理由

- Pylons 1.0.x は「レガシー」扱いで今後はメンテナンスのみ
- Pyramid の方が (色々な意味で) 強力
  - 特に、拡張性が非常に高い
- Pyramid for Pylons Users が公開された (2012-06-12)
  - 翻訳済み: http://docs.pylonsproject.jp/projects/pyramid_cookbook-ja/en/latest/pylons/index.html

Pylons 1.0 プロジェクトを Pyramid に移植すべきでない理由

- Pyramid 自体が開発中
  - 例) 1.3.2 で Mako テンプレートの継承ができなくなるバグがあり、修正版 (1.3.3) がリリースされるまで 3 ヶ月近くまともに使えなかった
  - 全体的に Pylons ユーザ向けの機能はまだ弱い
- 情報が少ない
  - Pylons からの移行に関してはほとんど情報がない
  - 日本では Pyramid ユーザ自体が少ない (~30人ぐらい?)
  - Pylons Project JP http://www.pylonsproject.jp/ にぜひ参加を (宣伝)
- 結論
  - 移行はそれなりに大変なので、もう少し待った方がいいかも
  - まずは新規のプロジェクトで Pyramid を試してみる
  - 既存のプロジェクトに対しては今のうちから移行の準備をしておく

Pylons フレームワークと Pyramid の比較

Todo 詳しく書く

- paster コマンド -> p* コマンド
- scaffold
- ディレクトリレイアウト
- WSGI ミドルウェア -> tween
- main 関数
- ビューとルート (route)
- リソース
- 特殊グローバル変数
- レンダラー変数
- HTTP エラーとリダイレクト
- 静的ファイル
- セッション

- WSGI サーバ
- アクセス制御
- URL 生成

移植方法

- ゼロから Pyramid で書き直す場合
  - あまり変更せずに使用可能
    - モデル, テンプレート, 静的ファイル
  - 変更が必要
    - コントローラ, route マップ, グローバル変数

移植方法

- 一度に 1 つずつの URL を移植する場合: 移植された URL を Pyramid に、移植されていない URL を Pylons に処理させる
  1. Apache の中で Pyramid と Python の両方のアプリケーションを起動して、mod_rewrite を使用して異なる URL を異なるアプリケーションに送る。
  2. INI ファイルの中で paste.cascade を設定する。
     - 最初に片方の アプリケーションを実行してみて、“Not Found” が返る場合にはもう片方のアプリケーションを試す (Pylons が静的ファイルを返すのと同じ方法)
  3. Pyramid のビューで Pylons アプリケーションをラップする。

移植方法

- その他の注意
  - 複数のアプリケーションを同時に実行すると、データベース接続、セッション、データファイルなどで微妙な問題が表面化して調整が必要になる可能性がある
  - Pyramid アプリケーションを Python 2 と 3 のどちらで書くかを選択しなければならない

Pyramid の拡張方法

Todo 詳しく書く

- 設定ディレクティブ
- ビューマッパー
- リクエストファクトリ
- イベントシステム
- tween (Pyramid 内の WSGI ミドルウェアのようなもの)
- Zope コンポーネントアーキテクチャ (ZCA)