



# Ruby on Rails

Lightweight Language Day & Night2005 フレームワーク対決資料

## ■ Rails の特長

Dave “達人プログラマー” Thomas による Rails が好きな理由ベスト10(の超訳)

1. Web 開発をアジャイルにする
2. ちょっと小粋なエフェクト付きの Web ページが作れる
3. 「フレームワークのお守り」ではなく、アプリの作成に集中できる
4. 規模が大きくなってもメンテナンス可能でありつづける
5. 顧客に対して「できます」という答えがもっと言えるようになる
6. テスティングが組み込まれていて簡単に使える
7. 即時のフィードバック: コードを編集して、再読み込みすると、その変更がブラウザで確認できる
8. メタプログラミングのおかげで高いレベルでプログラミングできる
9. コード生成のおかげで手早く始められる
10. XML 不要!

## ■ Rails のキーワード

- DRY (Don't Repeat Yourself): 重複は悪、同じことを違うところで書かない
- Convention Over Configuration: 細かく設定するより適切な規約で無設定に
- Agile: アジャイル開発しやすいように全てが設計
- DSL (Domain Specific Language): ドメイン特化言語風味の構文
- Active Record: シンプルで便利な O/R マッピング
- (Functional/Unit) Testing Framework: テストがかなり書きやすくなる
- Code Generation: 何も書かなくてもコードの雛型を生成
- Ajax: prototype.js を始めとして Ajax には妙に熱心
- Rake (Ruby Make): これまた DSL 風味な Makefile が Ruby で書ける
- Rubyish: 「The Ruby Way」「Rubyらしさ」にわりとこだわる
- fun!: 「たのしさ」を重視

## ■ Rails の MVC

- モデル: ActiveRecord(専用の O/R マッパーライブラリ)
- コントローラ: ActionController(アクションをクラス単位ではなくメソッド単位で管理)
- ビュー: ActionView+ERb(ヘルパーライブラリをつけた埋め込み Ruby)

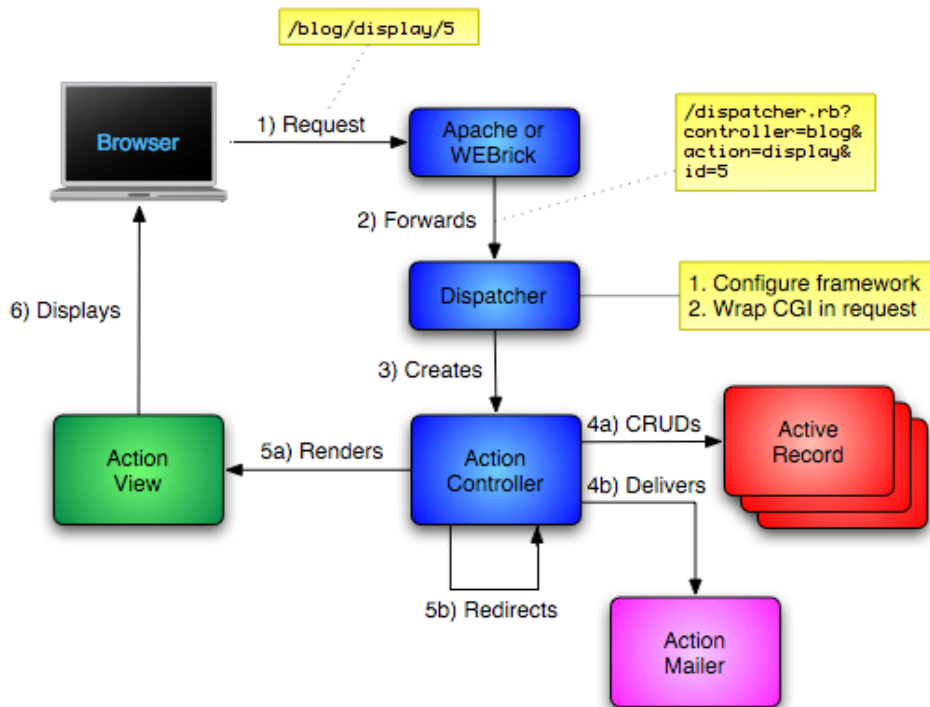


図 1: Rails のリクエスト処理の流れ

## ■ その他の Rails のポイント

- dRuby を使った実行時のリモートデバッグ
- 標準で Development/Production/Test の 3 モードの実行環境を提供
- コード行数等がすぐわかるメトリクス出力機能(役に立つかは微妙)
- 一目瞭然、15 分で blog を作ったりするデモムービー
- とにかく強力なマーケティングセンス
- Web サーバには Apache よりも Lighttpd+FastCGI が人気
- Ajax といい Lighttpd といいなんとなく新しい物好き
- 一方、継続(Continuation)には興味なさそう(Ruby なら Borges が有名)
- TemplateEngine が標準でないのは賛否両論(非標準ならある)
- 標準クラスにいきなりメソッドを追加することも辞さない漢気あふれる実装
- 日本人には不評気味の単数形・複数形自動変換
- 最初はみんな唾然とする色つきログファイル(もちろん off にもできる)
- 日本語・多言語対応はまだ発展途上(内部は UTF-8 が無難?)