

・ 第 38 回 PHP 勉強会 2008-12-20
於:トライコーン株式会社 1F セミナールーム

題: テンプレート・エンジンの役割を考える ～MVC 論争に一席～

* いしだなおと

* naoto@isnot.jp

* <http://isnot.jp/>

* <http://profile.typekey.com/isnotjp/>

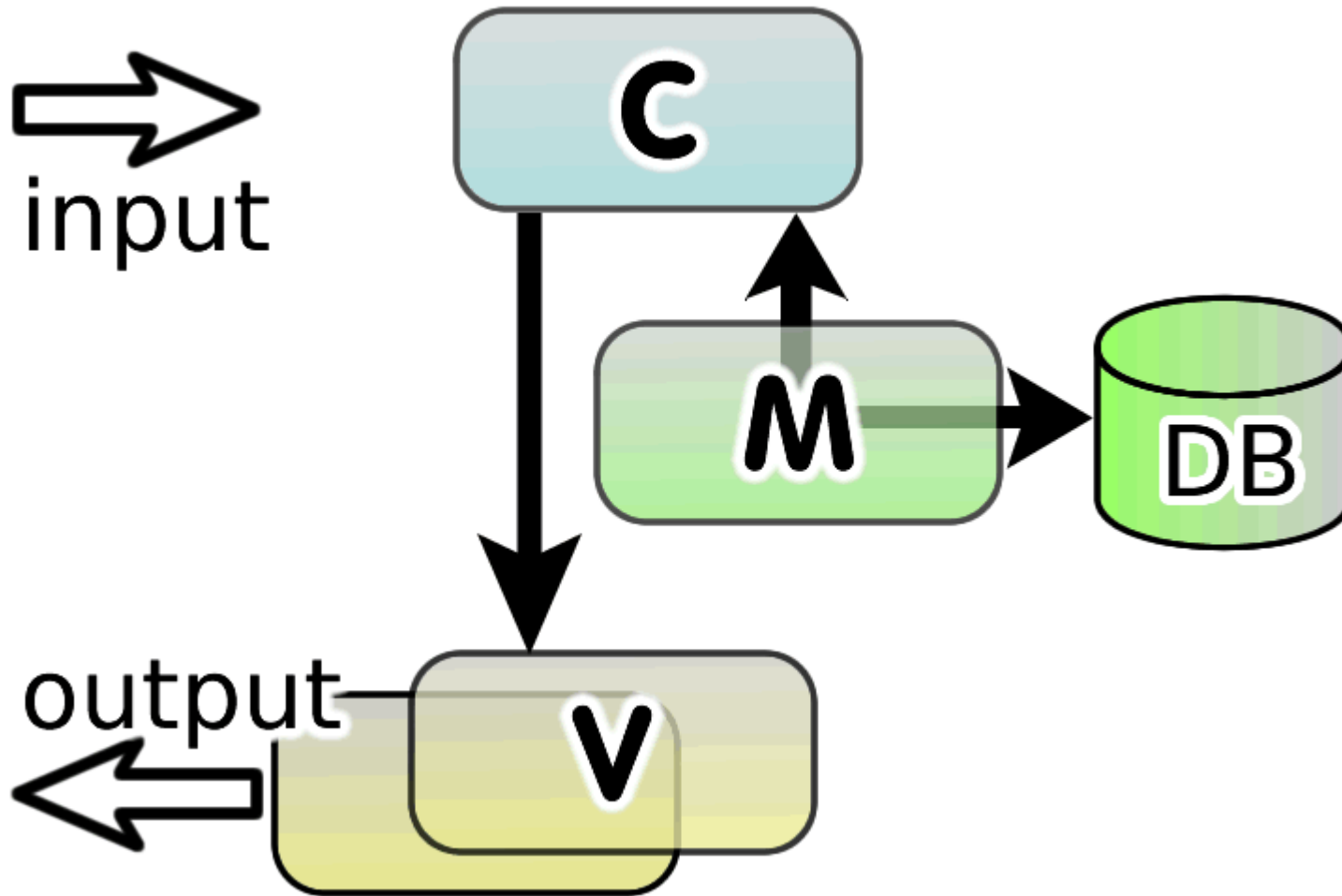
* 株式会社 RYUS <http://ryus.co.jp/>

自己紹介

概要

- * 自己紹介
- * 「MVC」とは
- * これまでの流れ・おさらい
- * 比較
- * 検討
- * まとめ

「MVC」とは(図)



Model

- Object
- ORM (Object-relational mapping)
 - DB 抽象化レイヤ
 - DB のテーブル

- アプリケーションのために設計した、オブジェクト・モデルそのもの
- ORM=アプリケーションのために設計したオブジェクト・モデルと、データ・ストア(テーブル上の表現)との間のインピーダンス・ミスマッチを補完する
 - データ・ストアとは、永続化されたデータのこと、テーブル定義やリレーションなどによって束縛される
 - ようするに DB にあるデータ

※インピーダンス・ミスマッチ=位相のずれ

みなさんは Model については
良く考えていらっしゃるようです

View

- Template
 - plugin/filter
- 多様な表現

• 多様な表現

- html/html
- json
- xml
 - Atom PP/Atom Feeds
 - RSS
 - ASX (ストリーミング用メタ)
- CSV
- mail の文面
- 帳票など
- ...

- データセットと出力の表現の間のインピーダンス・ミスマッチを補完する
 - 出力の表現は、規格や定義、妥当性、顧客の要求によって束縛される
- あとでくわしく

Controller

- FrontController
- Router
 - uri とアクション等との関連付け
- Action
- Filter

- リクエストに従って、Model からコレクション(オブジェクトの集合)を得る
 - 出力の表現を決定し、コレクションを View に渡す
 - View がレンダリングした結果を、レスポンスとして返す
 - 合間合間に各種処理(Filter)をはさむこともできる
-
- 今回は Controller については取りあげません

このトークのねらい

- MVC における **View** について検討します
- テンプレート・エンジンの意義を再考します
- View と Template の役割分担について、**考えるきっかけ**になる話題を提示します

このトークの前提

- テンプレート・エンジン = **Smarty**
ということで話します

このトークのおもな対象

- Smarty をある程度使ったことがある方
- PHP のフレームワークを使った、「見通しのよい設計」について考えたことがある方

これまでの流れ

おさらい

2008-07-06 PHP 勉強会 SIDE-B(第1回) 「PHP フレームワークと Smarty」(いしだなおと)

PHP 勉強会 SIDE-B(第1回) 2008-07-06

題: PHP フレームワークと Smarty

- * いしだなおと
- * naoto@isnot.jp
- * <http://isnot.jp/>
- * 求職中

1/33

実際、オブジェクト・モデリング/テーブル設計がうまく
いっていれば、View(クラス)ですべきことというのはそ
う多くはない(はず)

比較的多数の PHP フレームワークでは、基本的に
View は **POPT**(Plain Old PHP Template)が採用されて
います

Smarty は DSL (Domain Specific Language) です

Controller/View から Smarty インスタンスに渡すデータセットは、HTML ではなく、素の文字列/数値もしくはそれらのハッシュ/配列/データ構造を渡すほうがすっきりいく

HTML タグは Template で処理するか、Smarty plugin を活用するとカッコいい(セキュリティ的な意味で)

Smarty plugin で、各ヘルパのラッパーを作ったほうがよいということです

2008-09-26 設計勉強会

MVC とは何か。おまえら MVC わからずにフレームワーク使うな (halt)



Ethna の場合

- MVC といいながらなんかおかしい
- View にロジックをもりもり
- ViewClass で setAttribute して Smarty に値を渡して
るんだけどその値をまた Smarty で加工 → それなら ViewClass とか言ってわかる意味なくね？
- 自分は ViewClass は必要ないと思ったので ViewClass を使わないでコードをかいていた

CakePHP の MVC

- ロジックはモデルに格納
- アクション(Controller のメソッド)はそれをよんで view にデータをセットするだけ
- テンプレートファイルが View

スッキリしてない？

まとめ

- フレームワーク使ってる人は一度自分のコード設計がまちがってないかよく考えたほうがいいんじゃないかな？

そう思うよ！

(by いしだなおと)

ただし、まとめについて

Template がない？！

- View の実装に、テンプレート・エンジンに相当する部分を持っている
- PHP のシンタックスで書ける
- でも結局は、いろいろと(それぞれに独自の)書き方のルールが存在する
- つまり独自のテンプレート・エンジンを実装しているに過ぎない

「それ以前」の件については割愛
そもそも PHP が～の件については割愛

心地よくアプリケーションをコーディングしたいように、
心地よく Template を編集したい

好きなやり方でやったらいいんじゃないかな？

ORM との対比(1/3)

Model

1. Model の中に **SQL** 文を直に書いて、mysql (i) 関数を使ってコンテンツ/データの出し入れをすることができる

View

1. PHP コードに **html の断片**を埋め込んで書くことができる

ORM との対比(2/3)

Model

2. **せめて**、DB 抽象化レイヤを使いたくなる

View

2. **せめて**、Template 部分は別ファイルに分離しておきたくなる

ORM との対比(3/3)

Model

3. ORM を使うことによって、より抽象度の高いコーディングが可能に(見通しのよさ)

View

3. テンプレート・エンジンを使うことによって、より抽象度の高いコーディングが可能に(なるような使い方をしたいところ)

ORM との対比(まとめ)

Model

1. Model の中に SQL 文を直に書いて、mysql (i) 関数を使ってコンテンツ/データの入し入れをすることができる
2. せめて、DB 抽象化レイヤを使いたくなる
3. ORM を使うことによって、より抽象度の高いコーディングが可能に(見通しのよさ)

View

1. PHP コードに html の断片を埋め込んで書くことができる
2. せめて、Template 部分は別ファイルに分離しておきたくなる
3. テンプレート・エンジンを使うことによって、より抽象度の高いコーディングが可能に(なるような使い方をしたいところ)

- ↑これができないでいるために、テンプレート・エンジンが不要と結論付けている

↑これができないでいるために、テンプレート・エンジンが不要と結論付けている

もっと使いこなせば
いいと思うよ！

- 個々の View は(ほぼ)変えずに、多様な表現(html 以外)に対応する
 - 表現別に、FW が用意した View を継承した、それぞれの View クラスを使い分ける
- 出力に対して、透過的なフィルター処理を書く(pre、post)
- フレームワークの実装に関わらず、いつも同じシンタックスで、ヘルパを利用する

- 体験談; Ethna に Smarty plugin が揃っていたので、それらを Code Igniter に移植して使いました
- CakePHP でもなんでも、Smarty plugin の実装の中から、ヘルパ(相当)を呼び出せばよい

公知のテンプレート・エンジン(Smarty)に、独自の拡張(plugin)を加えながら使いこなそう

Smarty のおいしいところ

- Smarty を入れたとして、通り一辺の使い方をするだけでも、一定程度の効果がある
 - 変数を展開する、条件分岐、ループ
- Smarty のおいしいところは、そこだけではない。
 - むしろ、上記の使い方なら、たしかに PHP (POPT) だけで足りることかもしれない

悪しき慣わしは排除せよ

- 各種便利関数＝ヘルパを Template から使いこなせ
- [PHP] ～ [/PHP] は禁じ手(基本的に)
- オブジェクトに備わっている「副作用のある」メソッドは、Template から使い出すと、いろいろアレ
 - DB に update がかかる処理を Template の中から引き金を引くのは避けておこう

∴できる限り、Template のレンダリングが走り出す直前までに、その Template で必要なデータセットをお膳立てしておく = View クラス側の処理

私が知っている Smarty plugin の活用しどころのいくつかについて話そう

いつもある処理/場合わけによってする処理のいくつかは、フィルターで解決

- prefilter、outputfilter でエンコーディングを変換(携帯サイト向け)
- XSS 対策のエスケープは、設定/規約やフィルター系の処理を使ってトータルに処理しよう
 - バリデーションは View の仕事ではないよ(お約束として)

常に埋め込むべき内容を、簡単に呼び出せるようにしておこう

- CSRF 対策/2 重 POST 対策
- 国際化(i18n)
- html、form の要素を組み立てる

「見た目」的な何かは、Template で

- 数値のカンマネーション、フォーマット
 - 3桁区切りのカンマ、桁揃え、ゼロパディング
- 文字列のトランケーション
 - 長いタイトルを 25 文字で切って、...(続き)、とか
- 文字列に prefix、postfix を付加する
 - 12,345 円?、¥12,345?
- 日付、時刻のフォーマット

サーバサイドでしなくてもいい仕事は、Javascript を使ってもいい

- 要素の並び替え、絞込み
 - どこがそれに当たるかは……場合によるとしか言いようがない

(このトークの主題)

(Smarty の)本当の使いどころは、カスタマイズしたプラグインを使って View の仕事を Template に移譲できることにある(、だろう)

資料 1

2008-07-06 PHP 勉強会 SIDE-B(第 1 回)

<http://events.php.gr.jp/events/show/47>

「PHP フレームワークと Smarty」(いしだなおと)

いくつかの PHP の Web アプリケーション・フレームワークにおいて、テンプレート・エンジンに Smarty を使う方法が提供されています。
だいたい感想程度に 軽〜く紹介します。

* 発表資料

* Ustream

資料 2

2008-09-26 設計勉強会

<http://events.php.gr.jp/events/show/52>

MVC とは何か。おまえら MVC わからずにフレームワーク使うな(halt)

アクション。と名のつくものを持つフレームワークを使っている人はたいてい MVC がなにかわからずにコードを書いています。

MVC も理解せずにコードを書いているから「これはどこにかけばいいんだろう」とか連呼する事になるわけです。

Mojavi からひきつがれてきた勘違いコーディングの例と、どうやったら正しく MVC できるのかと紹介します。

? action にロジック? プンプ wwwwww はずかしくないの wwwwww

お知らせ

株式会社 RYUS では、PHP での開発経験のあるエンジニアを募集しております。

<http://ryus.co.jp/>

お気軽にご相談ください。面接など前提にしない場合でも、お話伺います。

おわり

ご静聴ありがとうございました。

いしだなおと

- naoto@isnot.jp
- <http://isnot.jp/>
- この資料の URL
 - http://isnot.jp/pub/20081220_php38_naoto.pdf