

社内勉強会

# **symfony tips その壱**

**2007-8-9(thu.) TAKAHARA Yoshihiro**

# 今日話す事

- まずやる事
- 知っておくべき事
- ~~良くある問題と対策~~

→時間がないので延期

# 対象レベル

- symfony開発で100h以上やっている人
  - 初級者(not 初心者)
    - askeet よんだー
    - symfony book読んだー
    - 仕事でやってみたー

# まずやること

- セキュリティに関する事はすぐにやる
  - Output Escapingの設定
  - Routingの設定

これらの事は他人任せにせずプロジェクトメンバー全員で確認する

# Output Escapingの設定

- アプリケーションを作ったら
  - `apps/{app}/config/settings.yml`を確認

# Output Escapingって何 だっけ？

- アクションからの変数を自動でエスケープしてくれる仕組み。
- `{$text | escape }`  昔はこんなの  
一つ一つ書いてた
- クロスサイトスクリプティング対策

# settings.yml

```
web_debug:      on
cache:          off
no_script_name: off
etag:           off

test:
  .settings:
    # E_ALL | E_STRICT & ~E_NOTICE = 2047
    error_reporting: 2047
    cache:           off
    web_debug:       off
    no_script_name: off
    etag:            off

all:
  .settings:
    escaping_strategy: bc
    escaping_method:   ESC_ENTITIES

...
#all:
# .actions:
#   default_module:   default # Default
```

bc → bothに

下を切る

# bcって？

- backward compatible
  - 以前のOutput Escapingが無かった頃の互換モード
  - つまりデフォルトではエスケープされない！

# bothにすると...

- 全てのテンプレート変数が**問答無用**にエスケープされる。
- エスケープされたくない時はどうすんの？

# HTMLを表示する

- エスケープからエスケープする
  - 変数の場合(e.g. \$html)
    - `$sf_data->getRaw('html');`
  - オブジェクトの場合
    - `$content->getHtml(ESC_RAW)`

# routingの設定

- アプリケーションを作ったら
  - `apps/{app}/config/routing.yml`を確認

# routing.yml

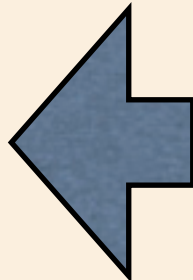
```
# default rules
homepage:
  url: /
  param: { module: default, action: index }

default_symfony:
  url: /symfony/:action/*
  param: { module: default }

default_index:
  url: /:module
  param: { action: index }

default:
  url: /:module/:action/*
```

defaultの3つ  
いらねー



# なぜ消すか？

- 意図しないアクションが呼ばれてしまい、誤動作するのを防ぐ

# 例えば...

```
public function executeSend() {  
    //既に組み込まれている、sfComponent::sendEmail()を利用します。  
    $this->sendEmail("mymodule", "mail");  
}  
public function executeMail() {  
    $this->mail = new sfMail();  
    $this->mail->initialize();  
    $this->mail->setMailer('sendmail');  
    $this->mail->setCharset('utf-8');  
  
    // definition of the required parameters  
    $this->mail->setSender("sender@example.com");  
    $this->mail->setFrom("from@example.com");  
    $this->mail->addReplyTo("replyto@example.com");  
    $this->mail->addAddress("to@example.com");  
    $this->mail->setSubject("メールですよ");  
}
```

# routingは必要な分だけ 追加しよう

```
homepage:
  url: /
  param: { module: page, action: show, content: homepage }

company:
  url: /company/
  param: { module: page, action: show, content: company }

company_profile:
  url: /company/profile
  param: { module: page, action: show, content: company_profile }

company_philosophy:
  url: /company/philosophy
  param: { module: page, action: show, content: company_philosophy }

business:
  url: /business/
  param: { module: page, action: show, content: business }

business_foundry:
  url: /business/foundry
  param: { module: page, action: show, content: business_foundry }
```

propel-generate-crud  
つかえないじゃん！

答え：使わなくていいです

# 復習：まずやること

- セキュリティに関する事はすぐにやる

- Output Escapingの設定

- Routingの設定

これらの事は他人任せにせずプロジェクトメンバー全員で確認する

# 知っておくべき事

- schema.ymlの書き方
- アクションの減量法

# schema.ymlの書き方

- schema.ymlがどこまで楽させてくれるか
  - id
  - created\_at, updated\_at
  - テーブル名\_id
  - \_idn

# アクションの減量法

- 基本戦略
  - ロジックはモデルに書く
  - メール本文はsfMailViewに
  - 1/3 小技関数を使う
  - getRequestParameter()をなんとかする
  - sfPageFlowで制御する

# ロジックはモデルに

- Criteriaはアクション内で使わない原則

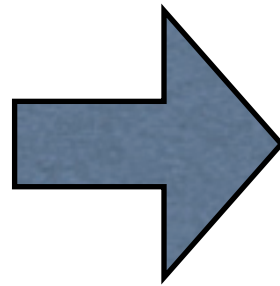
# メール本文は sfMailViewに

- 以前やったので飛ばします（下を参考に）
- <http://blog.symfony.jp/2007/08/01/how-to-send-mail/>

# 1/3 小技関数を使う

- 意外と使われていない 1/3小技関数

```
function executeIndex(){  
    if($ok){  
        $this->forward("@homepage");  
    }  
  
    if($err){  
        $this->forward("@homepage");  
    }  
}
```



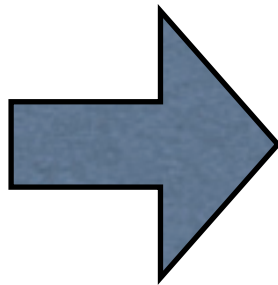
```
function executeIndex(){  
  
    $this->forwardIf("@homepage",$ok);  
  
    $this->forwardUnless("@homepage",$ok);  
}
```

forward404,redirectにもあるよ

# getRequestParameter() をなんとかする

- hydrateObjectでなんとかする！

```
$model->setValue1($this->getRequestPamameter('value1'));  
$model->setValue2($this->getRequestPamameter('value2'));  
$model->setValue3($this->getRequestPamameter('value3'));  
(以下続く)
```



```
$this->hydrateObject($model,array('value1','value2','value3'...));
```

[http://blog.symfony.jp/2007/02/17/hydrate\\_object/](http://blog.symfony.jp/2007/02/17/hydrate_object/)

# sfPageFlowで制御する

- 複雑なフォーム遷移系はsfPageFlowで
  - 公式プラグインとして提案中
  - それなりに複雑なので説明は後で

<http://blog.symfony.jp/2007/08/03/sfpageflow/>

アクションの減量法を  
マスターして楽しよう

# 今日話した事

- まずやる事
- 知っておくべき事
- 良くある問題と対策 → 次の機会に

**ご清聴ありがとうございました**