



# Ktai Library勉強会@名古屋

## 2:実践Ktai Library



滝下 真玄(MASA-P)

<http://www.ecworks.jp/>

[info@ecworks.jp](mailto:info@ecworks.jp)



MENU

- 「Ktai Library」を動かしてみよう
- 「一括変換」を利用した  
携帯サイトの制作
- PCサイトと携帯サイトを  
「なう」なやり方で振り分けてみる

# 【何故「Ktai Library」?】

ここから先は  
「だいたいセルフ」な  
ワークショップ形式です



# 【何故「Ktai Library」?】

「自分のペースで」  
制作に取り組んでみてください

分からない点がありましたら  
遠慮無くどうぞ(^^)





# 「Ktai Library」を 動かしてみよう

# 【「Ktai Library」を動かしてみよう】

基本的には「PHPで作る  
携帯サイト デベロッパーズ  
ガイド」の第4章(P173～)に  
書かれています。

一応ざっくりと  
手順を説明します



PHPで作る  
携帯サイト  
デベロッパーズガイド  
Mobile Site  
by PHP

滝下真玄 [著]

人気フレームワークCakePHPと  
オリジナルのKtai Libraryで一  
携帯サイト開発が「仕様の違い」から解放される!!

完成サイトのソースコードはダウンロードサービス!!

西和システム



# 【「Ktai Library」を動かしてみよう】

## 1. CakePHP(1.3)のインストール

- オールグリーンにする(お約束)

## 2. Ktai Libraryのインストール

- github(<http://github.com/MASA-P/KtaiLibrary>)からアーカイブ一式を入手
- 解凍したものをCakeに上書きインストール



# 【「Ktai Library」を動かしてみよう】

## 3. 絵文字画像のインストール

- TypePad(<http://start.typepad.jp/typecast/>)からアーカイブ一式を入手し「app/webroot/img/emoticons」へコピー

## 4. Ktai Libraryの設定(controller)

- `var $components = array('Ktai');`
- `var $helpers = array('Ktai');`
- `var $ktai = array(~);`





# 【「Ktai Library」を動かしてみよう】

## 5. ビューの作成

- 携帯用layoutを作成
- ビューを作成(携帯向けで普通に)

## 6. テスト表示

**手順としてはこれだけ!**



# 【「Ktai Library」を動かしてみよう】

## 注意点

- バージョン0.3からは「KtaiAppController」を継承するようになった(AppControllerにコピペするのでもOK)
- ‘input\_encoding’ => ‘UTF8’
- ‘output\_encoding’ => ‘UTF8’
- ‘use\_img\_emoji’ => true



# 【「Ktai Library」を動かしてみよう】

できるかな？





# 「一括変換」を利用した 携帯サイトの制作

## 【「一括変換」を利用した携帯サイトの制作】

**Cakeは基本的にUTF-8で  
エンコーディングされている  
しかし携帯サイトは  
SJISで表示したい場合が  
ある**



# 【「一括変換」を利用した携帯サイトの制作】

そんな場合はKtai Libraryの  
「一括変換」機能を使おう!



# 【「一括変換」を利用した携帯サイトの制作】

## ■ポイント1: controller

```
App::import('Vendor', 'ecw/Lib3gkDef');
App::import('Controller', 'KtaiApp');
class MypagesController extends KtaiAppController {
var $ktai = array(
//  'input_encoding' => KTAI_ENCODING_UTF8,
//  'output_encoding' => KTAI_ENCODING_SJISWIN,
    'input_encoding' => 'UTF-8',
    'output_encoding' => 'SJIS-win',
    'use_img_emoji' => true,

    'output_auto_convert_emoji' => true,           //自動絵文字変換
    'output_auto_encoding' => true,              //自動エンコーディング
    'output_convert_kana' => 'knr',             //mb_convert_kana()
);
```



# 【「一括変換」を利用した携帯サイトの制作】

## ■ポイント2:layout

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ShiftJIS">
<?php if($ktai->is_iphone()){ ?><meta name="viewport"
content="width=260"><?php } ?>
<title><?php echo $title_for_layout; ?></title>
</head>
```





# 【「一括変換」を利用した携帯サイトの制作】

できるかな？



# 【「一括変換」を利用した携帯サイトの制作】

## 注意点



# 【「一括変換」を利用した携帯サイトの制作】

基本的に自動変換は  
**処理が重い!**



# 【「一括変換」を利用した携帯サイトの制作】

出来る限り  
目的のエンコーディングで  
サイト構築をしよう!





# PCサイトと携帯サイトを 「なう」なやり方で振り分けてみる

## 【PCサイトと携帯サイトを「なう」なやり方で振り分けてみる】

従来では、  
Ktaiコンポーネントや  
ヘルパーを登録して  
コントローラやビューで  
使うのが一般的だけど



↑これは「ナウい」人

## 【PCサイトと携帯サイトを「なう」なやり方で振り分けてみる】

サブクラス化を  
したことで  
「なう」な使い方が  
できるようになった



↑これは「ナウい」人

【PCサイトと携帯サイトを「なう」なやり方で振り分けてみる】

## 本日の目玉

CakePHP 1.3の新機能と  
Ktai Libraryの「なう」な使い方で  
PCサイトと携帯サイトを実現する





# 【PCサイトと携帯サイトを「なう」なやり方で振り分けてみる】

## ■ポイント1:bootstrap.php

```
//CakePHP 1.3の新機能
```

```
//任意のディレクトリパスの指定
```

```
App::import('Vendor', 'ecw/Lib3gkCarrier');
```

```
$carrier = Lib3gkCarrier::get_instance();
```

```
if($carrier->is_ktai()){
```

```
    App::build(array(
```

```
        'views' =>
```

```
        array('/home/samba/ктаistudy2/app/views/mobile/'),
```

```
    ));
```

```
}
```



# 【PCサイトと携帯サイトを「なう」なやり方で振り分けてみる】

## ■ポイント2: roots.php

```
Router::connectNamed(array(), array('argSeparator' => '~'));
```

```
Router::connect('/', array(  
    'controller' => 'mypages', 'action' => 'index'));
```

```
Router::connect('/m/', array(  
    'prefix' => 'm', 'controller' => 'mypages', 'action' => 'index'));
```

```
Router::connect('/m/:controller/:action/*', array('prefix' => 'm'));
```



## 【PCサイトと携帯サイトを「なう」なやり方で振り分けてみる】

### ■ポイント3: 携帯用のコントローラ・ビューの設置

```
+ app/  
  +- controllers/  
    | +- mypages_controller.php      //コントローラ  
  +- views/  
    +- mobile/  
      | +- m_mypages/  
      |   +- index.ctp              //携帯用のビュー  
    +- mypages/  
      +- index.ctp                  //PC用のビュー
```



## 【PCサイトと携帯サイトを「なう」なやり方で振り分けてみる】

### ■ポイント4: mypages\_controller.php

```
function beforefilter(){
    if($this->Ktai->is_ktai() && !isset($this->params['prefix'])){
        $this->redirect(array('m' => true));
        return;
    }
    if(!$this->Ktai->is_ktai() && isset($this->params['prefix'])){
        $this->redirect(array('m' => false));
        return;
    }
    parent::beforeFilter();
}
```



## 【PCサイトと携帯サイトを「なう」なやり方で振り分けてみる】

### ■ポイント4:mypages\_controller.php

```
function index(){
    //ここに処理を書く
    $this->set('title', 'PC用ページ');
}

function m_index(){
    $this->layout = 'm_'. $this->layout;

    //ここに処理を書く
    $this->set('title', '携帯用ページ');
}
```



## 【PCサイトと携帯サイトを「なう」なやり方で振り分けてみる】

できるかな？



お疲れ様でした

