

OSC2009 Tokyo/spring



Usagi Project
Kunitsuji

<http://usagi-project.org>

**SNSのカスタマイズ方法をズバリ解説！
MyNETS、OpenPNEの触り方**

ファイルの構成

SNSのアプリケーション関連のファイルは、
基本的にすべて次の場所に保存されています。



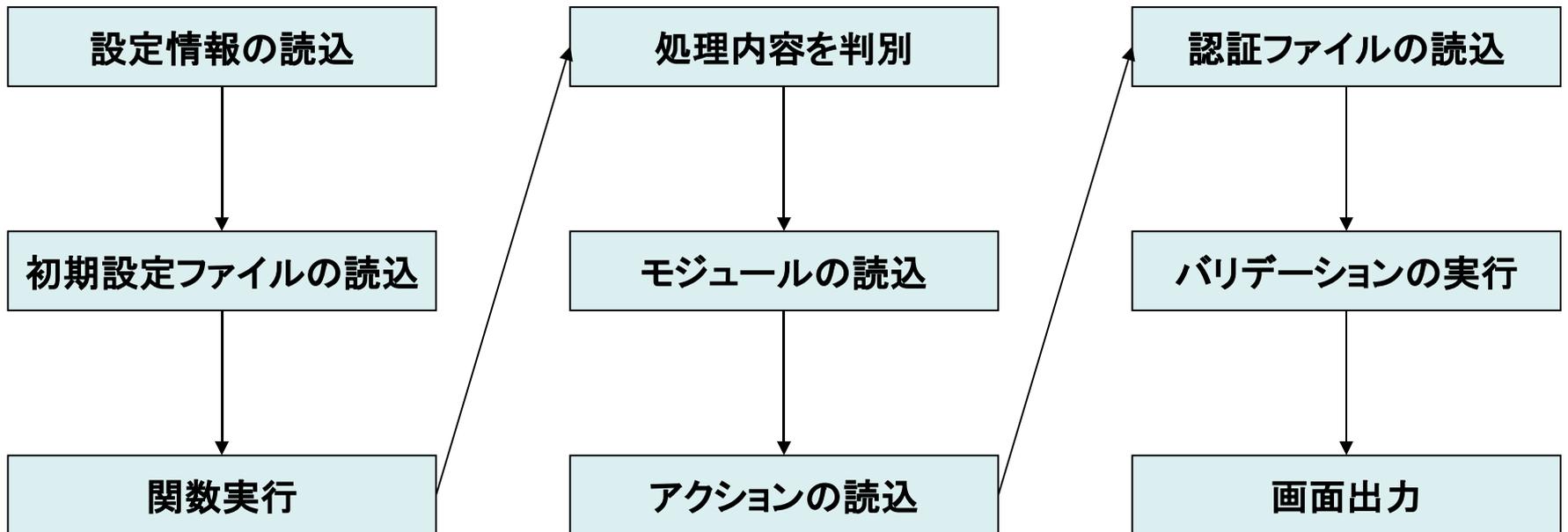
ウェブアプリケーションディレクトリ

この下に、次の内容のものが保存されています。

ライブラリ関連 モジュール関連 コンポーネント(MyNETSのみ)

処理の流れ

リクエストは、次の手順で処理がされていきます。



カスタマイズする部分

カスタマイズを行う一番多い場所

- 1・DBからデータを取得する処理部分
- 2・アクション処理実行部分
- 3・テンプレート表示部分

カスタマイズする部分

DBからデータを取得する処理部分

DBからデータを取り出す部分は、
webapp/lib/db/read,またはwrite
OpenPNE2.12ではwebapp/lib/db

SQLを実行してDBからデータを取得しています。

この関数内を修正することで取り出すデータを変更することができます。

カスタマイズする部分

アクション処理実行部分

実際のリクエストに対するアクションの部分は
webapp/modules/{モジュール名}/page
または
webapp/modules/{モジュール名}/do
で行われています。

この時のpageやdoを「タイプ」と呼んで区別しています。
pageは表示を伴う処理、doはDB等にデータを書き込む
処理を主に担当します。

http://hoge hoge.com/?m=pc&a=page_h_home

この部分がタイプとなります。
URLのリクエストパラメータから判断しています。

カスタマイズする部分

アクション処理実行部分

http://hoge hoge.com/?m=pc&a=page_h_home

上記のURLの場合、m=pcの「pc」がモジュール
a=page_h_homeの
「page」がタイプ
「h_home」がアクション名(クラス名)となります。

PCモジュールの、PAGEタイプでh_homeクラスを実行する
という流れになります。

モジュールはmodules/ディレクトリの下にあるディレクトリ名をさします。
タイプはmodules/pc/page等の、モジュールディレクトリの下にpageか
doディレクトリをさします。

カスタマイズする部分

テンプレート表示部分

テンプレートの表示は、必ずアクションクラス名と同じテンプレートが取得、表示されるようになっていきます。

例) h_homeアクションの場合、取得されるテンプレートは、次のようになります。

webapp/modules/pc/templates/h_home.tpl

カスタマイズする部分

テンプレートを指定して取得できるようにするには？
実際にカスタマイズしてみましょう。

アクションの終了からテンプレートの表示に対しては、次のきっかけで行われます。

return 'success';

これにより、アクションが正常に終了しテンプレートを取得表示するトリガーになります。

MyNETSの場合

```
if ($result == 'success') {  
    send_nocache_headers();  
    $smarty->ext_display("{ $action2 }.tpl");  
}
```

OpenPNE2.12の場合

```
if ($result == 'success') {  
    send_nocache_headers();  
    if ($smarty->ext_search($smarty->templates_dir . '/common/layout.tpl', $place)) {  
        $smarty->assign('op_content', $smarty->ext_fetch("{ $action }.tpl"));  
        $smarty->ext_display('common/layout.tpl');  
    } else {  
        $smarty->ext_display("{ $action }.tpl");  
    }  
}
```

```
if ($result == 'success')
{
    send_nocache_headers();
    $smarty->ext_display("{$_action2}.tpl");
}
else if ($result != "")
{
    send_nocache_headers();
    $smarty->ext_display("{$_result}.tpl");
}
else
{
    openpne_display_error('template表示指定エラー', true);
}
```

カスタマイズする部分

アクションクラスのファイルを詳しく見ておきましょう

アクションクラスでは、上から順番に様々な処理がされています。
基本的にはパラメータを受け取り、DBから取得してテンプレートにセットしています。

※POST、GETデータを取得する際に必須な処理
バリデーションを行った結果を受け取る。

`$requests['hgogehoge']`

`$_POST['hogehoge']`はダメ

`modules/pc/validation/{type}/{アクションファイル名}.ini`が必要

カスタマイズする部分

一つのリクエストで必要なファイルとして

pageタイプ page/アクション名.php
 templates/アクション名.tpl
 validate/page/アクション名.ini(リクエストパラメータを受け取る場合)

doタイプ do/アクション名.php
 validate/do/アクション名.ini(リクエストパラメータを受け取る場合)

アクションクラスでの決まりごと

```
class pc_page_c_event_edit extends OpenPNE_Action
```

モジュール、タイプ、アクション名をアンダースコア()で連結すること

OpenPNE_Actionクラスを継承(エクステンド)すること

```
function isSecure()
```

```
{  
    return false;
```

```
}
```

を記述すると、認証を行わないアクションとなる。

カスタマイズする部分

決まりごと

`class pc_page_c_event_edit extends OpenPNE_Action`
モジュール、タイプ、アクション名をアンダースコア(`_`)で連結すること
OpenPNE_Actionクラスを継承(エクステンド)すること

```
function isSecure()  
{  
    return false;  
}
```

を記述すると、認証を行わないアクションとなる。認証を行うアクション(ログイン後に使うもの)には記述しない。

`function execute($requests)` メソッドを必ず実装すること。
その中に処理を記述すること。
メソッドの最後に必ず `return 'success';` を記述すること。

Smartyの活用について

Smartyとは、テンプレート表示のためのライブラリ。
MyNETS、OpenPNEではこのSmartyを使用しています

。

管理画面でテンプレート挿入でPHPロジックを使いたい！場合

たとえば次のようなものがあります。

自分のHOMEを表示した際、携帯アドレスの登録をしているかどうかを判断し、していない場合はプロフィール画像の下に携帯アドレス登録のリンクやQRを取り付けたい場合。

Smartyの関数を作成します！

```
function smarty_function_t_mobile_qr($params, &$smarty)
```

管理画面で

```
{t_mobile_qr view=true}
```

等を挿入すると、PHP処理を行った結果が表示されます。

セキュリティ上注意すること

カスタマイズを行う場合、脆弱性等の対応や各種セキュリティを考慮しなければなりません

本来MyNETSやOpenPNEではMVCに沿って作ることで各種セキュリティ対策が行われます。

※直接画面出力を行ったり、Smartyでsmarty:nofaultsを指定する場合、
htmlspecialchars()

関数を利用して表示させるようにしましょう。

※マイネッツでは扱いやすいように

h()

関数として記述できるようにしています。

\$_REQUESTや\$_GET、\$_POSTでパラメータを受け取らないようにします。

アクションに処理を記述する際、想定がいの値を受け取った場合を考慮してエラー処理を記述するようにします。

カスタマイズをした場合

webapp/ディレクトリのファイルを上書きしないようにする

直接カスタマイズしたファイルを上書き更新せず、できるだけwebapp_ext/ディレクトリを活用すること。

※OpenPNEの場合、DBディレクトリ以下のファイルはwebapp_extに用意しても読み込むことができませんので、その場合は直接修正する以外ありません。。

※MyNETSでは、webapp_ext/lib/db/以下にread/としてdiary.phpを用意し、そこに既存の関数名を記述しても構いません。

例)function db_diary_get_c_diary4id(\$c_diary_id)をカスタマイズした場合webapp_ext/lib/db/read/diary.phpを作成

そこに上記の関数名を記述。

関数の処理をカスタマイズしたものを記述。

この場合、webapp_extが有効と設定されている場合、そちらの関数を読み込み、もとのfunction db_diary_get_c_diary4id()関数は読み込みません

バージョン管理アプリを利用

ファイルの修正などを管理するSVN等を利用する

SVN等のアプリを利用することで、ファイルの修正、バージョン管理を行うことが可能となります。

カスタマイズする以外にも利用価値は高く、ご利用をお勧めします。

※WindowsアプリでTortoisSVNというアプリがあります。
非常に便利です。

その他可能なこと

MyNETSでのカスタマイズでは次のようなことも行えます

MyNETSでは、フック(HOOK)というものを用意しています。
処理の途中で特殊な処理を実行することができます。

たとえば、リクエストの処理を実行する最初に何かを行いたい
最終出力の後に何かを行いたい
等の場合、フックを使うことで簡単に実現できます。
利用の仕方は、[conf/hooks.php](#)を参照してください。

見通しの良いカスタマイズを行う

カスタマイズを行う際に注意したいこと

アクションクラスにバンバンコードを書いてしまうと、あとでメンテナンスする際、非常に見通しの悪いコードになってしまいます。

処理は基本的に別のファイルに分散し、関数、クラスで定義する
アクションからはその関数、クラスメソッドを呼び出すだけにし、結果のみを取得するようにする。

DBデータの取得は基本的にlib/db/以下にまとめられています。
同じように何か新しいものを作る場合、できるだけアクションのファイル内に記述せず、別のファイルに分け、アクションからはその呼び出しのみを行うようにしましょう。

例) 対象のメンバーが、フレンドのフレンドかどうかを判定するロジックを作った場合。

db/以下のファイルに記述するか、MyNETSの場合components/にクラスとして保存する。

アクションからは、「is_friend_friend()」などを判定するようにのみ記述する

カスタマイズのご相談、その他

カスタマイズをしたい、技術指導を受けたい

Usagi Project では、パートナー企業をご紹介します。
MyNETSが中心ですが、OpenPNEも対応できる企業ばかりです。

株式会社エムズリンクでもカスタマイズのご依頼、ご相談、技術指導、技術サポートを行っておりますので、お気軽にご相談ください。

緊急告知！！

117万円で販売していたソフトを
オープンソース化！

concrete5、日本語化プロジェクトスタート！

詳しくはUsagi Project のブースでお聞きください！