

ずばびたテック

Quicity - Quick and Smarty

Webアプリ作成の効率化

PHPによるWebアプリケーションやオフライン処理プログラムの作成を劇的に効率化します。対話的にページの追加やテーブル構成の編集をすることで、Webアプリの土台を素早く作成できます。

見通しのよいMVCスタイル

Model、View、Controllerを分離したオブジェクト指向スタイルの開発環境を提供します。Modelは独自のO/Rマッパーを、ViewはSmartyとHTML_QuickFormをシームレスに連携させます。

ディレクトリ構成がサイトマップになる

テンプレートフォルダの中にHTMLファイルとフォルダを配置すると、それがサイトマップになります。

サイトマップに対応するクラスとメソッド

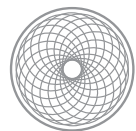
ライブラリフォルダ内には、サイトマップのサブディレクトごとに同名のクラスファイルを配置し、クラスファイル内には、HTMLファイルに対応するメソッドを記述します。URLを見れば、どのクラスのどのメソッドで処理しているかがすぐにわかります。

状況を理解し、判断する

QuicityのControllerは、リクエストされたURLやクエリを元に、登録、更新、表示、削除などのステータスを判断し、それに応じてViewとActionを決定します。定型的な処理はQuicityにまかせ、複雑で重要な処理はプログラマが自由に制御することができます。

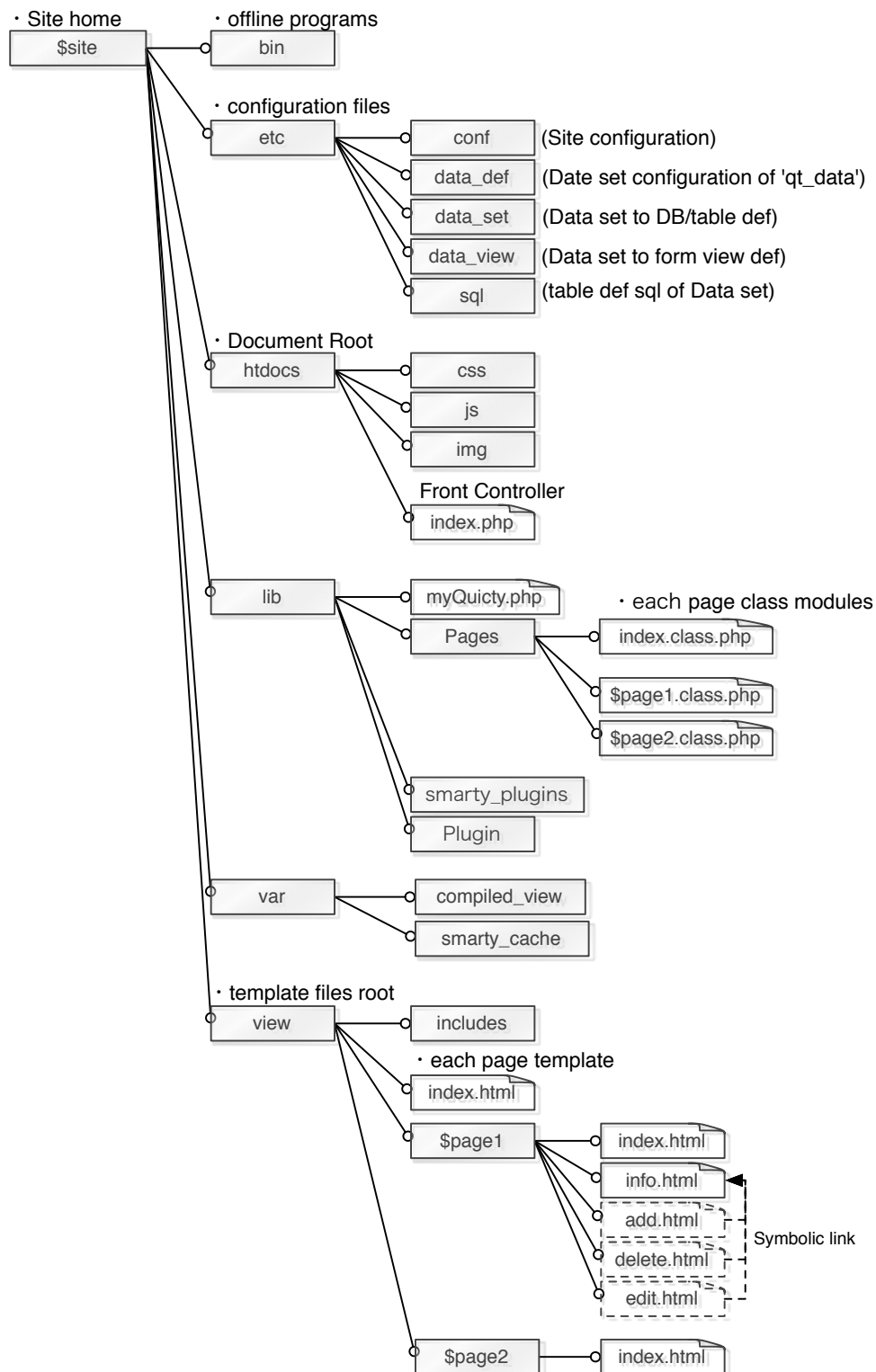
便利な機能を提供

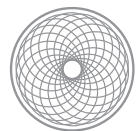
ページング、パンくず（パスリスト）、ページキャッシュと部分評価、ユーザー認証、メールの非同期送信、データの重複チェックなど、よくある処理を使いやすい形で提供します。またデフォルトのテンプレートは最初からSEOに最適化されています。



ずびたテック

Quicityサイトの基本ディレクトリ構成

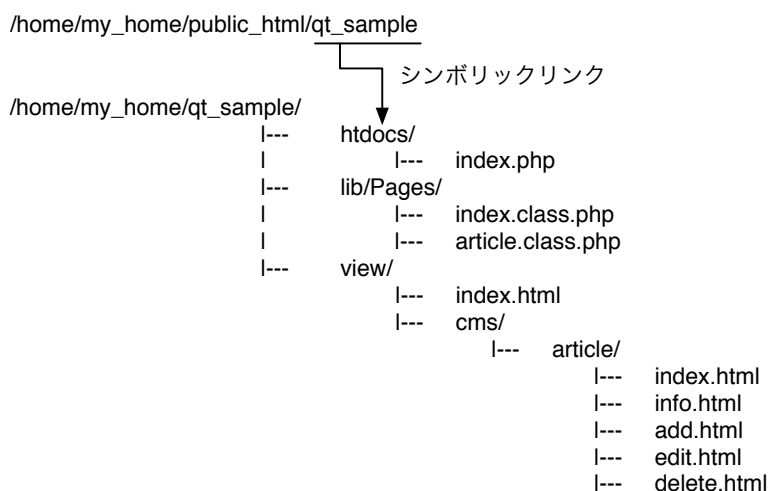




ずばびたテック

Quicityサイトの基本動作

ディレクトリ構成



URLへのリクエスト

http://www.my_hosy.com/~my_home/qt_sample/cms/article/add..html



シンボリックリンクとmod_rewriteにより、フロントコントローラに制御が渡される

/home/my_home/qt_sample/htdocs/index.php



URLから表示するテンプレートを決定

/home/my_home/qt_sample/view/cms/article/index.html



テンプレートのディレクトリ名から、クラスモジュールを決定

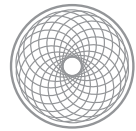
/home/my_home/qt_sample/lib/Pages/cms/article.class.php



テンプレート名に対応するクラスモジュール内のアクション（メソッド）を実行

```
class article extend myQuicity {  
  
    function dispatch_add() {  
        $FORM = $this->new_data_set('article');  
        ~  
        $this->assign_form('input_form',$FORM);  
        $this->display($this->current_template);  
    }  
}
```

ずばびたテック



ずばびたテック

インタラクティブにフォームとテーブルを設計する

Quicityでは、HTMLフォームとSQLテーブルの設計は、コマンドラインの編集プログラム「qt_data」を使ってインタラクティブに行えます。qt_dataで定義するHTMLフォームとSQLテーブルの組み合わせを「データセット」と呼びます。

qt_dataは、以下の5つのファイルを生成します。

view.confファイル

役割：HTMLフォームとデータ検証の設定

場所：etc/data_view/データセット名.conf

作成方法：qt_dataのw (write config)コマンド

set.confファイル

役割：view.confの定義をSQLのテーブルに割り当てる

場所：etc/data_set/データセット名.conf

作成方法：qt_dataのw (write config)コマンド

sqlファイル

役割：SQLテーブルの定義

場所：etc/sql/DBシステム名/データセット名.sql

※DBシステム名= sqlite | postgresql | mysql

作成方法：qt_dataのw (write config)コマンド

initファイル

役割：sqlファイルの実行（テーブルの初期化）バッチ

場所：bin/init_データセット名.def

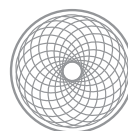
作成方法：qt_dataのw (write config)コマンド

defファイル

役割：qt_dataデータの保存

場所：etc/data_def/データセット名.def

作成方法：qt_dataのs (save data)コマンド



ずばびたテック

data_viewファイル

- data_viewファイルは、データの型と表示方法、デフォルト値や制約などを定義する

etc/data_view/sample.view.conf

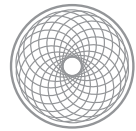
[id] type=id	[セクション名] HTMLの<input>タグのname属性に相当する。 SQLテーブルのカラム名でもある。 1 つめのセクションは必ず[id]、type=idでなければならない。
[name] type=text label=名称 attr=start size=50 maxlength=50 end_attr=true	type=タイプ名 ※フィールドの形式。HTMLの<input>タグのtype属性やHTML_QuickFormのaddElementのタイプ名を指定する。Quicity独自のタイプ名もある。
[url] type=url label=URL attr=start size=50 maxlength=128 end_attr=true	label=見出し ※HTMLで表示するときのフィールドの見出し。 <table>〜</table>内で<th>〜</th>に 使用される。
	attr=start 〜 end_attr=true ※<input>タグのオプションを「名前=値」の形式で列挙する。 1行に1属性。文字列の場合も、"は使わない。



このほか、

- ・ list=start ~ end_list=true : select型のoptionを指定。
 - ・ default_list=start ~ end_default_list=true : デフォルト値を指定。
 - ・ rule=start ~ end_start=true : HTML_QuickFormの検証ルールを指定。
 - ・ force_value_list=start ~ end_force_value=true : クラス変数や環境変数の値を割り当てる。
 - ・ reference=start ~ end_reference=true : 配列やDBのテーブルから値を取り出して割り当てる。
- などが指定できる。

名称	<input type="text"/>
URL	<input type="text"/>
<input type="button" value="登録"/>	



ずびたテック

data_setファイル

- data_setファイルは、data_viewファイルとsqlファイルのテーブル定義を対応づける。

etc/data_view/
sample.view.conf

```
[id]
type=id

[name]
type=text
label=名称
attr=start
size=50
maxlength=50
end_attr=true

[url]
type=url
label=URL
attr=start
size=50
maxlength=128
end_attr=true
```

etc/data_set/sample.set.conf

```
[dsn]
system=sqlite
dbname=sample
path=/tmp

[bind]
name:text=sample.name:text
url:url=sample.url:text
```

etc/sql/sqlite/sqmplesql

```
drop table sample;
create table sample(
  id INTEGER PRIMARY KEY,
  name text,
  url text
);
```

data_setファイルの構文

[dsn]

※使用するデータベースを指定するセクション

system=データベース種類

※pgsql、mysql、sqliteの3種類が指定できる。

dbname=データベース名

path=sqliteデータベースのあるディレクトリ名

※sqlite用

host=データベースサーバーのホスト名 or IPアドレス

※pgsql、mysql用

user=データベースのユーザー名

※pgsql、mysql用

password=データベースのパスワード

※pgsql、mysql用

persistent=0 or 1

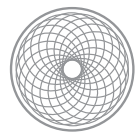
※1のとき持続的接続。pgsql、mysql用

[bind]

※HTMLフォームのフィールド（=view.confの[セクション]）とSQLテーブルのカラム名の対応を指定する。

フォーマット

セクション名:タイプ名=テーブル名:カラム名:カラムのタイプ



ずびたテック

Quictyのユーティリティコマンド

Quictyインストール後、以下のコマンドをコマンドラインで実行することにより、Quictyサイトを構築できます。

qt_new

機能：Quictyサイトの初期設定。

実行すると、Quictyサイトの基本的なディレクトリやファイルを設定します。htdocsをApacheのDocumentRootに設定しておけば、Webブラウザでアクセスすると、ひな形ページが表示されます。

qt_add_page

機能：ページとデータセットの追加。

実行するとメニューが表示されるので、希望するサイトマップ構成に従って、ページを選んで追加します。

ページと同名のデータセット（のひな形）を作成することもできます。ここで追加されるデータセットは、名前（text型）とURL（url型）の2つのカラムを持っています。

ページを追加後、テーブルの初期化バッチを実行すれば、データの追加、更新、削除、一覧表示の機能を持つページができあがります。

qt_data

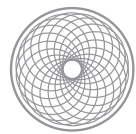
機能：データセットの作成と編集。

インタラクティブのデータセットを編集できます。qt_add_pageで作成したデータセットのひな形の定義（defファイル）を読み込んで、フィールド（カラム）の追加や削除、検証ルールの追加や削除などを行えます。

qt_template

機能：サイト・テンプレートのインストール。

簡易ゲストブックなど、作成済みのQuictyサイトを現在のディレクトリにインストールします。インストール後、初期化バッチを追加すれば、すぐに利用できます。



ずばびたテック

Quicityのインストール例

1. quicity.tar.gzを展開し、/usr/local/Quicityに配置
2. php.iniのinclude_pathに含まれたディレクトリが/usr/local/php/lib/phpのとき、以下のシンボリックリンクを張る

`/usr/local/php/lib/php/Quicity → /usr/local/Quicity`

例) `$ sudo ln -s /usr/local/Quicity /usr/local/php/lib/php/Quicity`

3. /usr/local/Quicity/binを実行PATHに含める

例) `$ PATH=/usr/local/Quicity/bin:$PATH`

4. 環境変数QUICITY_MODを設定

例) `$ export QUICITY_MOD=/usr/local/Quicity/modules`

※Quicityはphpのバイナリが/usr/local/php/bin/phpにあることを前提にしている。/usr/local/bin/phpにある場合は、

`$ sudo mkdir /usr/local/php`

`$ sudo mkdir /usr/local/php/bin`

`$ sudo ln -s /usr/local/bin/php /usr/local/php/bin/php`

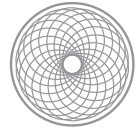
とシンボリックリンクを張るなどして、対応する。

Quicityに必要なPHPのライブラリ

- Smarty
- PAER::Auth
- PEAR::Config
- PEAR::DB (AuthとMail_Queueが使用)
- PEAR::HTML_QuickForm
- PEAR::Mail_Queue
- PEAR::Pager

使用が推奨されるPHPのextention

- PECL::apc (データキャッシュにも使用する)



ずばびたテック

Quicty用のPHP-5.1.xのコンフィグレーション例

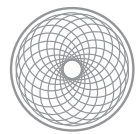
(Mac OS X 10.4 Tigerの場合)

```
# ./configure '--with-pgsql' '--with-mysql=/usr/local/mysql' '--with-pdo-pgsql' '--with-pdo-mysql=/usr/local/mysql' '--prefix=/usr/local/php' '--mandir=/usr/share/man' '--infodir=/usr/share/info' '--with-apxs' '--with-ldap=/usr' '--with-kerberos=/usr' '--enable-cli' '--with-zlib-dir=/usr' '--enable-trans-sid' '--with-xml' '--enable-exif' '--enable-ftp' '--enable-mbstring' '--enable-mbregex' '--enable-dbx' '--enable-sockets' '--with-iodbc=/usr' '--with-curl=/usr' '--with-config-file-path=/etc' '--sysconfdir=/private/etc' --with-libedit
# make
# make install
```

(Linuxの場合)

```
# ./configure '--with-pgsql' '--with-mysql=/usr/local/mysql' '--with-pdo-pgsql' '--with-pdo-mysql=/usr/local/mysql' '--prefix=/usr/local/php' '--with-apxs=/usr/local/apache/bin/apxs' '--enable-cli' '--with-xml' '--enable-exif' '--enable-ftp' '--enable-mbstring' '--enable-dbx' '--enable-sockets' --with-readline
# make
# make install
```

※オフラインプログラムでreadlineを使用する場合は、--with-readline もしくは --with-libeditを指定する。必須ではない。



ずびたテック

Quicityの最小構成プログラム例 (Single Level Single Page)

index.html

```
<!-- Header -->
{include file='includes/header.inc'}
</head>
<body>
<!-- Main -->
<table border=0 cellpadding=0 cellspacing=0 valign="top" id="container">
<tr><td valign="top" id="maincont">
    <h1>{$site.name}</h1>
    {include file='includes/auto_table_view.inc'}
    <hr>
    {include file='includes/auto_input_form.inc'}

</td>
<!-- Right -->
<td valign="top" id="mainright">
    {insert name='right'}
</td></tr>
</table>
<!-- Footer -->
{include file='includes/footer.inc'}
```

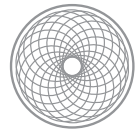
index.php

```
<?php
require_once '../lib/myQuicity.php';
$quicity = new myQuicity();
$quicity->dispatcher();
?>
```

myQuicity.php

```
<?php
require_once 'Quicity/Quicity.php';
class myQuicity extends Quicity {
    public function dispatch_index() {
        return $this->display($this->current_template);
    }
} // end of class myQuicity
?>
```

ずびたテック



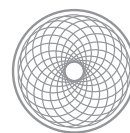
ずばびたテック

index.class.php

```
<?php
class index extends myQuicity {
    function dispatch_index() {
        // フォームの設定

        $_FORM = $this->new_data_set('single_single');
        $_FORM->set_base_link_addr(quicity_base_url().'');
        $_FORM->set_restart_link_addr(quicity_base_url().'');
        $_FORM->validate_and_freeze();
        $_FORM->set_button_status('use_confirm',false);
        $_FORM->bind_button_control();
        $result = $_FORM->automatic_form_handler($condition);
        if($this->QuicityStatus=='INSERT') redirect($_SERVER['REQUEST_URI']);
        // 表示

        $this->site['name'] = 'single_single';
        $this->assign('site',$this->site);
        $this->page['name'] = 'single_single';
        $this->page['base_url'] = quicity_base_url();
        $this->assign('page',$this->page);
        $this->assign_form('input_form',$_FORM);
        $table = $_FORM->select_table(' order by id');
        $this->assign('table_view',$_FORM->display_table_value($table));
        $result = $this->display($this->current_template,);
        return $result;
    }
} // end of index class
?>
```



ずばびたテック

白紙

ずばびたテック