

# PHPフレームワーク入門

## Zend Framework 2

---

コネクト株式会社

# アジェンダ

---

## PHPビルトイン関数とPDOによるデータベース処理

- PHP環境の確認
- SQLITEのインストールとテーブル作成
- PHPビルトイン関数によるデータベース処理(例題解説)
- PDOによるデータベース処理(例題解説)

## Zend Framework 2 入門(補足資料)

- Zend Framework 2 のインストールと動作確認
- Zend Framework 2 を使用したシンプルな例題

## Zend Framework 2 データベース処理

- データベース処理の下準備
- 例題解説

# MySQLのテーブル作成

---

Zend Framework 2 の例題で使用する  
MySQLのテーブルを作成します

# テーブルmeiboの定義

---

```
CREATE TABLE meibo (  
  sno int AUTO_INCREMENT,  
  name varchar(30),  
  memo text,  
  updata timestamp,  
  PRIMARY KEY (sno) );
```

例題で使用するテーブル'meibo'を作成します。

Nameとmemoは、utf8\_binの指定です。

# テストデータ

---

```
mysql> select * from meibo;
```

sno	name	memo	updata
1	testName	testmemo	2015-02-17 18:50:30
2	佐藤	メモ	2015-02-17 18:51:41
3	田中	ダミーのデータです	2015-02-17 19:18:22

```
4 rows in set (0.05 sec)
```

# Zend Framework 2 インストールと動作確認

---

ZEND FRAMEWORK 2 の配置と動作確認  
ようにスケルトンプロジェクトの配置を  
行います。

# ベース環境の構築

---

## Zend Framework 2 の必要条件

- Webサーバ
  - リライトエンジン(mod\_rewriteなど)
- PHP 5.3.x
- HTML5対応ブラウザ

# インストール手順①

---

## WebサーバとPHP 5.3環境を用意

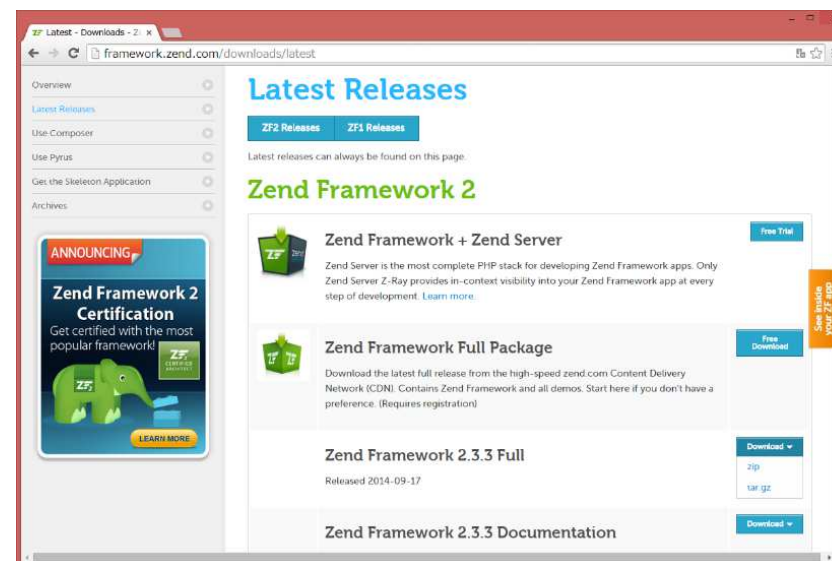
- Zend Server 6 以上の場合
  - Zend Framework と Zend Framework 2 インストール済
  - 手順③へ
- XAMPPなどの場合
  - Zend Framework 2 独自にインストール



# インストール手順②

## Zend Framework 2を入手

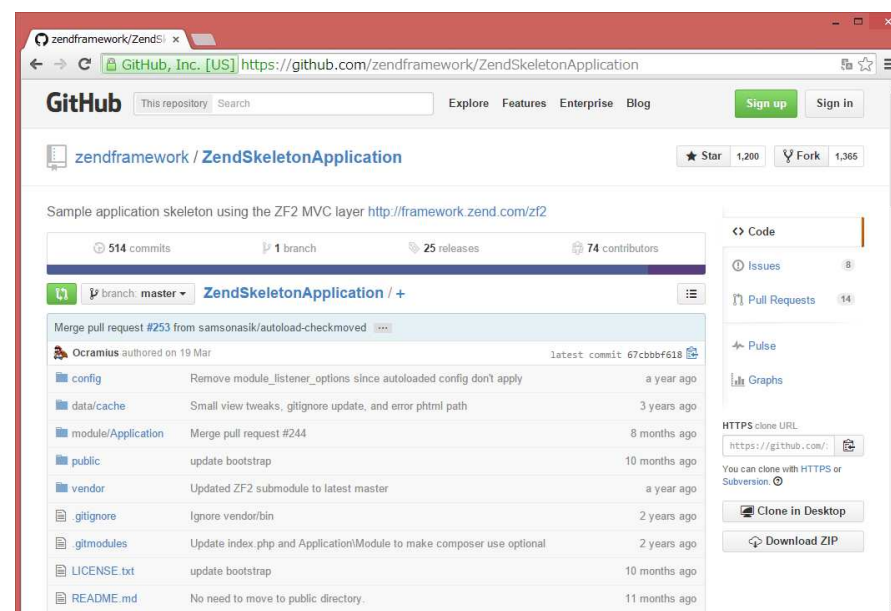
- <http://framework.zend.com/downloads/latest>
- ZendFramework-2.3.4.zip \*2/15現在
- 適当に展開(別途説明)



# インストール手順③

## スケルトンプロジェクトの入手

- <https://github.com/zendframework/ZendSkeletonApplication>
- ZendSkeletonApplication-master.zip



# インストール手順④

---

## スケルトンプロジェクトの配置

- 適当に展開します
- publicディレクトリをドキュメントルートへ
  - ディレクトリ名を ZendSkeleton に変更
- 残りを適当に配置します
  - ディレクトリ名を ZendSkeleton に変更

具体例を次ページで解説します

# インストール手順④ 補完

## 圧縮ファイル内部

```
¥
├─config
│  └─autoload
├─data
│  └─cache
├─module
│  └─Application
│     ├──config
│     ├──language
│     ├──src
│     │  └─Application
│     │     └─Controller
│     └─view
│        ├──application
│        │  └─index
│        ├──error
│        └─layout
├─public
│  ├──css
│  ├──fonts
│  ├──img
│  └─js
└─vendor
```

## エントリーポイントの配置(Webサーバのドキュメントルート)

- XAMPPの場合
- c:¥xampp¥htdocs¥ZendSkeleton

## アプリ実態の配置(Webサーバのドキュメントルート以外)

- XAMPPの場合
- c:¥xampp¥ZendSkeleton

## Zend Framework 2 の配置(Webサーバのライブラリ置き場)

- XAMPPの場合
- C:¥xampp¥Library

# インストール手順⑤

---

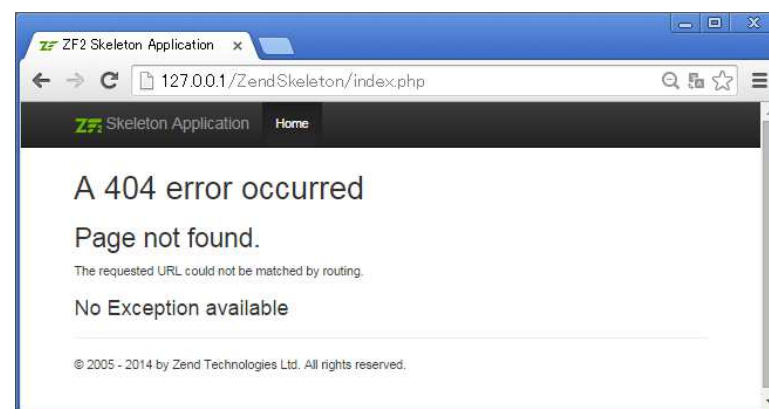
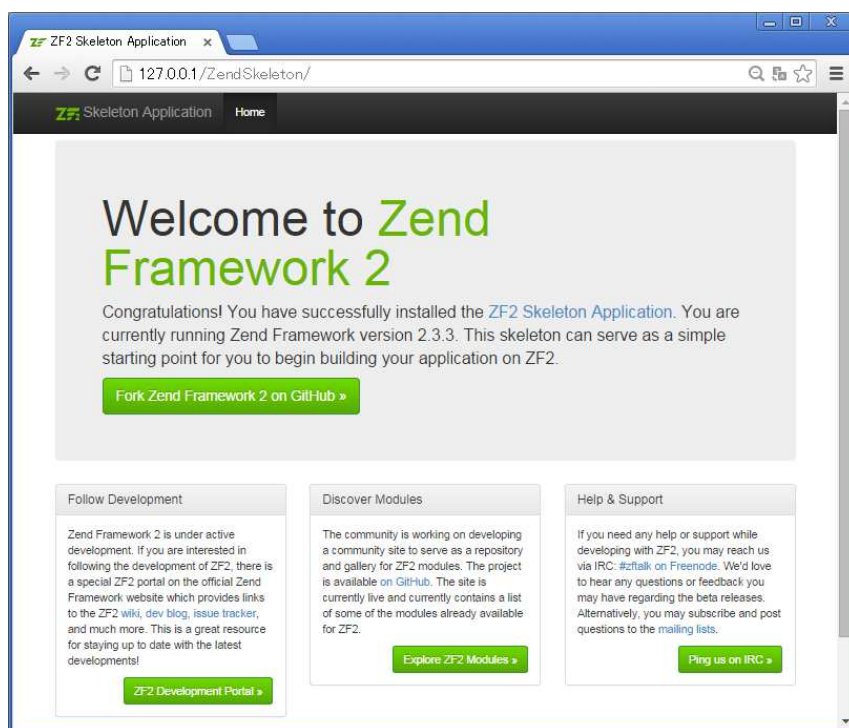
## エントリーポイントindex.phpの設定

- ディレクトリ C:¥xampp¥htdocs¥ZendSkeleton
- ファイル名 index.php
- 場所 6行目と8行目付近
  - Zend Framework 2 へのパス設定
  - カレントディレクトリをアプリ実態へ
- index.phpの内容(XAMPPの例)
  - `putenv("ZF2_PATH=" . 'C:¥xampp¥Library¥ZendFramework-2.3.3¥library');`
  - `chdir('c:¥xampp¥ZendSkeleton');`

# インストール手順⑥

## スケルトンプロジェクトの動作確認

- URL `http://127.0.0.1/ZendSkeleton/`



# Zend Framework 2

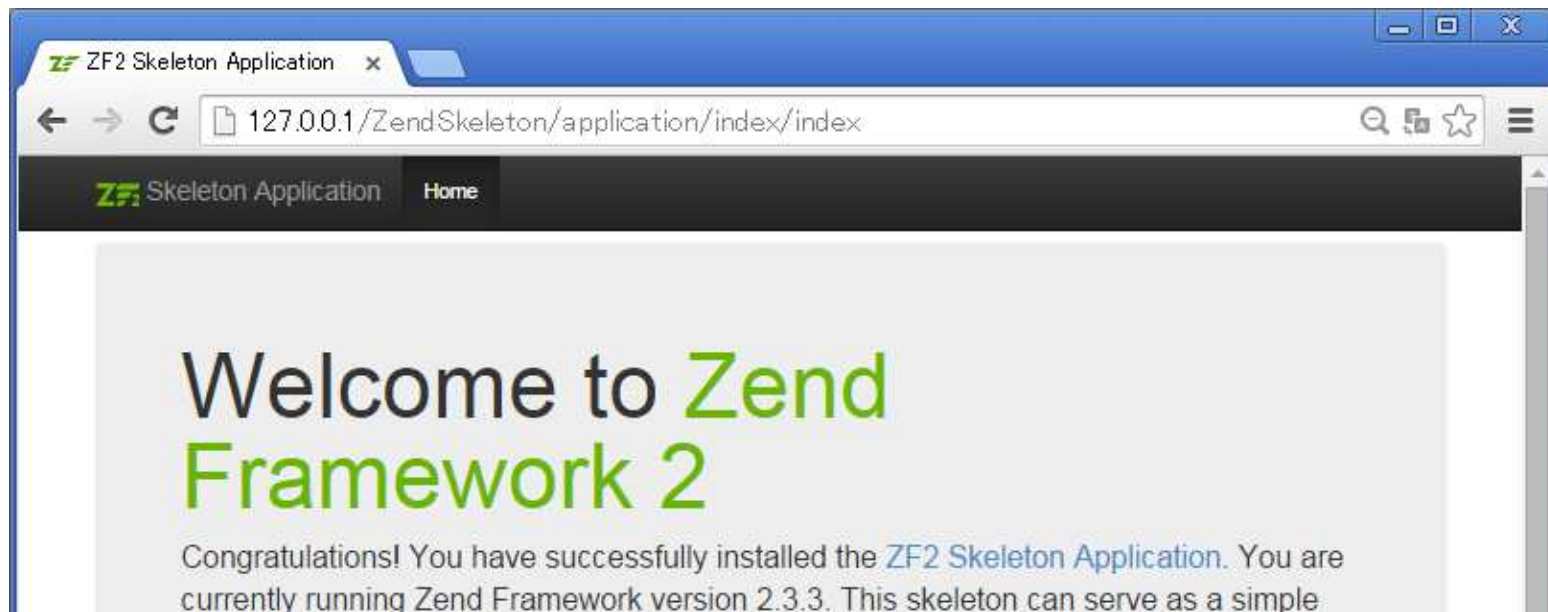
## 基本概念

---

ZEND FRAMEWORK 2 の基本的な概念  
を説明します。

# リクエストURLとデフォルト

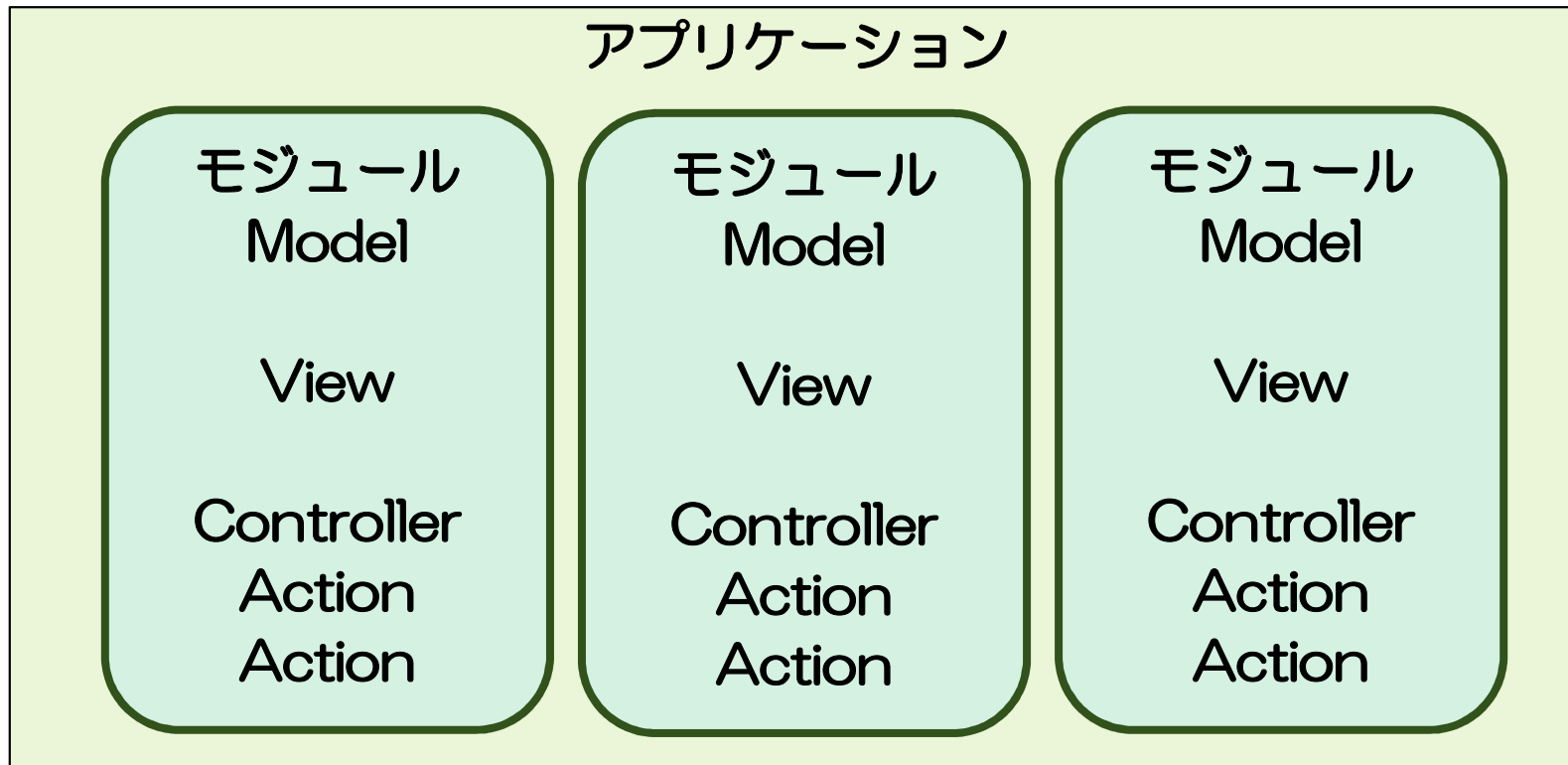
- `http://ドメイン/モジュール/コントローラ/アクション`
- `http://127.0.0.1/ZendSkeleton/`
- `http://127.0.0.1/ZendSkeleton/application/index/index`





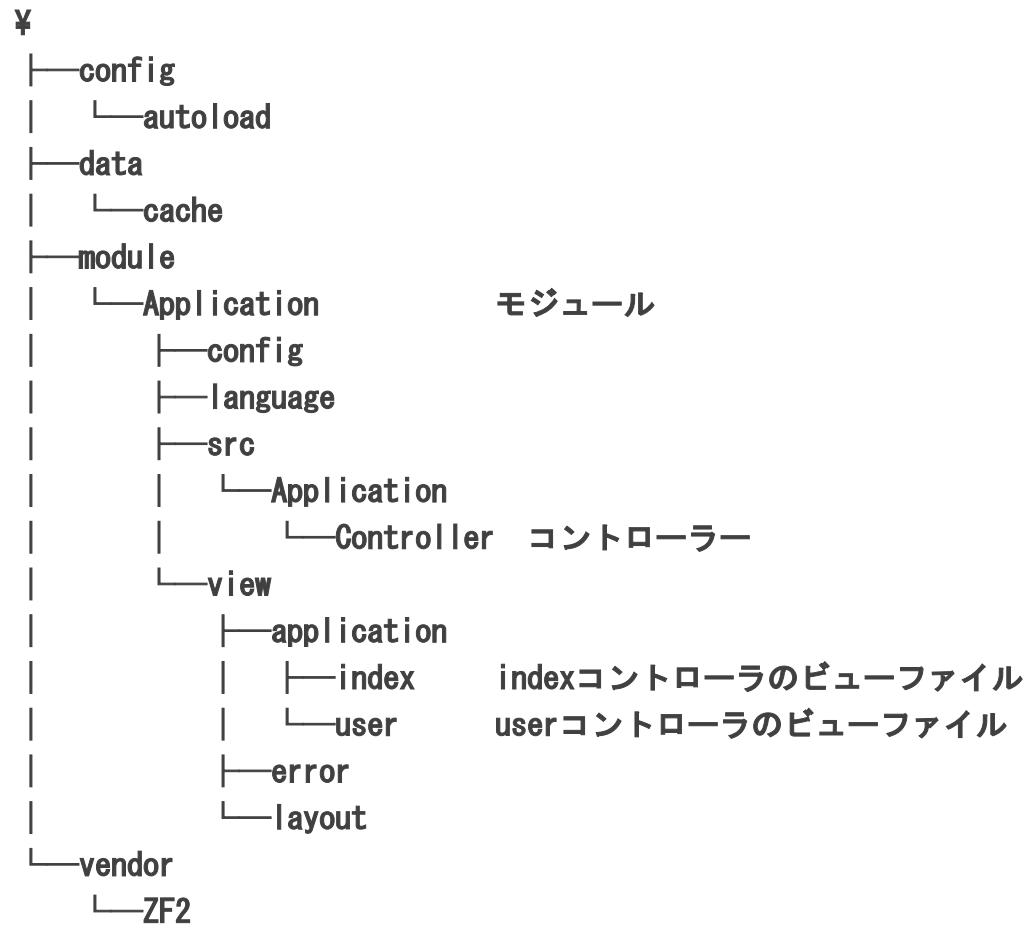
# モジュール概念

---



# スケルトンプロジェクトで確認

## 圧縮ファイル内部



# Zend Framework 2 例題による入門

---

簡単な例題を通して ZEND FRAMEWORK 2  
の基本動作を説明します

# Zend Framework 2 例題

---

コントローラーおよびアクションの作成  
アクションからビューで値の受け渡し  
FORMから値を取得

Captchaによる認証処理

# 例題1

## コントローラー/アクションの作成

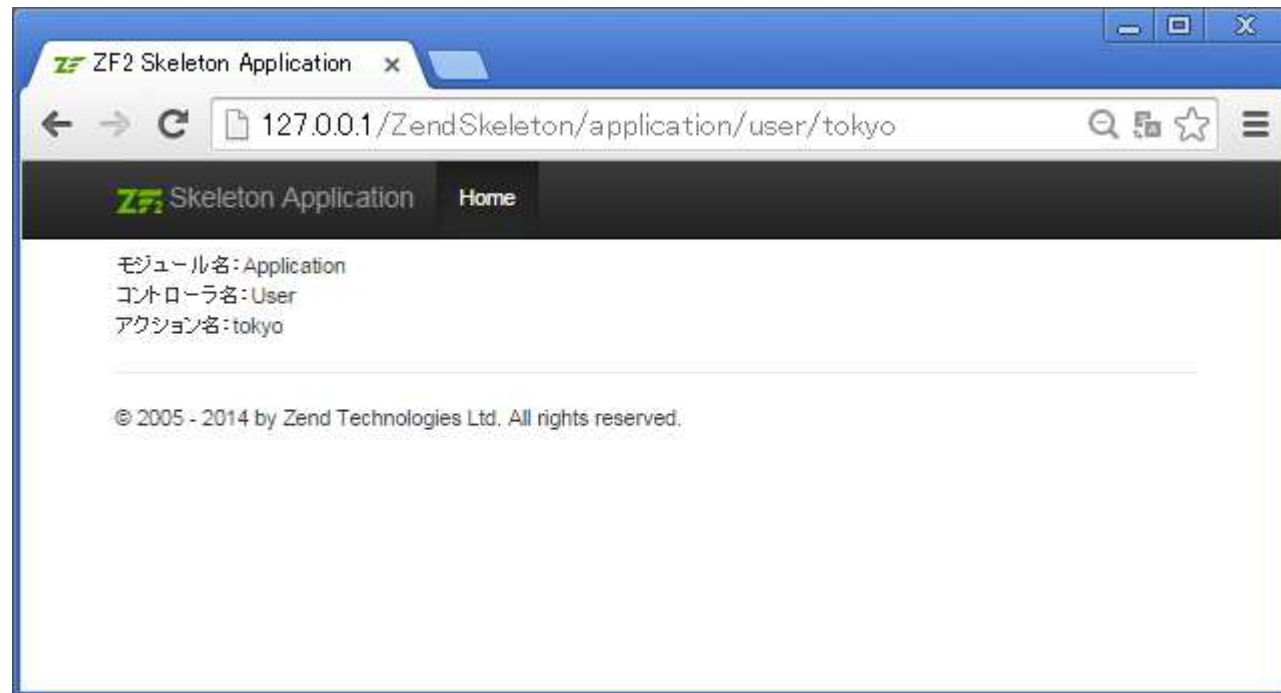
---

- コントローラーおよびアクションを作成
- C:¥xampp¥ZendSkeleton¥module¥Application¥src¥Application¥ControllerにUserController.phpをコピー
  - コントロールを登録  
C:¥xampp¥ZendSkeleton¥module¥Application¥configのmodule.config.phpを修正
  - ビューを配置  
C:¥xampp¥ZendSkeleton¥module¥Application¥view¥application¥userにtokyo.phtmlをコピー
- 以下のURLにアクセスして出力結果を確認しましょう。
  - <http://127.0.0.1/ZendSkeleton/application/user/tokyo>

# 例題1

## コントローラー/アクションの作成

- <http://127.0.0.1/ZendSkeleton/application/user/tokyo>



# 例題2

## 値の受け渡し

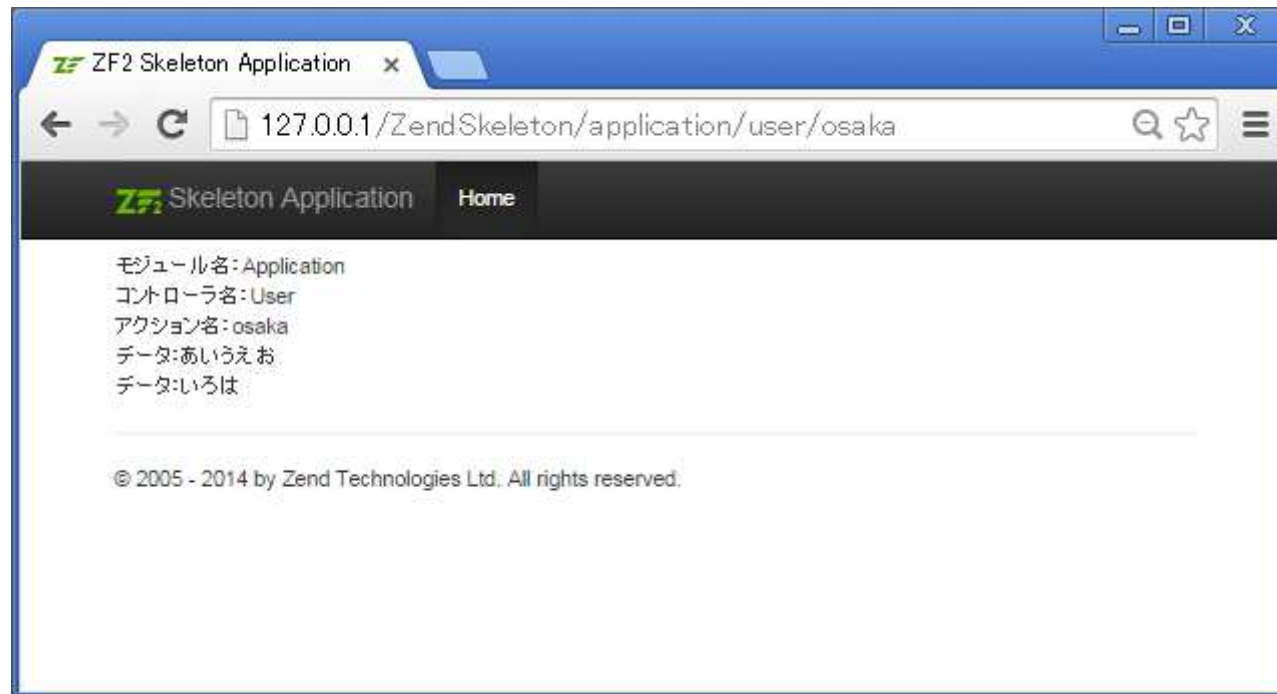
---

- ビューを配置  
C:¥xampp¥ZendSkeleton¥module¥Application¥view¥application¥userにosaka.phtmlをコピー
- 以下のURLにアクセスして出力結果を確認しましょう。
  - <http://127.0.0.1/ZendSkeleton/application/user/osaka>

# 例題2

## 値の受け渡し

- <http://127.0.0.1/ZendSkeleton/application/user/osaka>





# 例題3

## FORMから値を取得

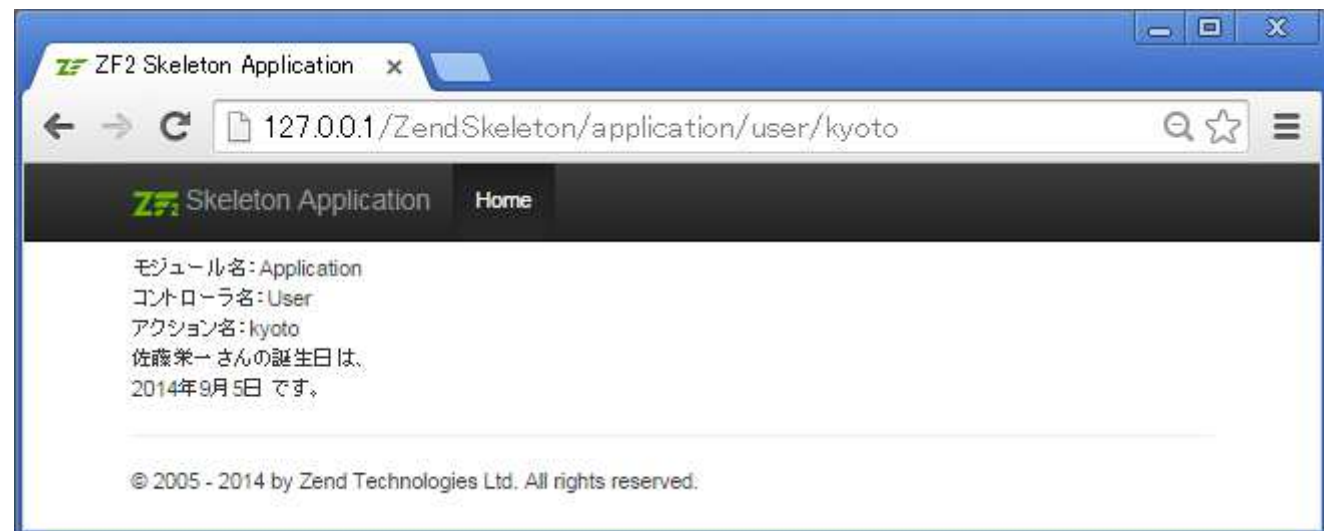
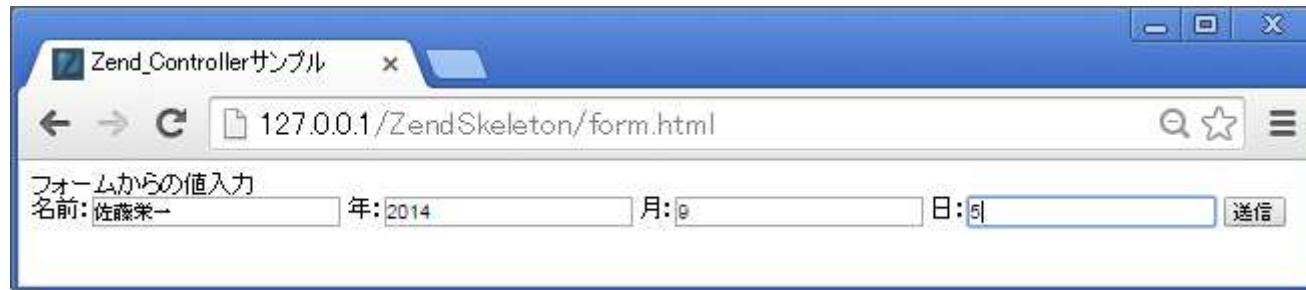
---

- フォームを配置
- C:\xampp\htdocs\ZendSkeleton\にform.htmlをコピー
  - ビューを配置  
C:\xampp\ZendSkeleton\module\Application\view\application\userにkyoto.phtmlをコピー
- 以下のURLにアクセスして出力結果を確認しましょう。
  - <http://127.0.0.1/ZendSkeleton/application/user/kyoto>

# 例題3

## FORMから値を取得

- <http://127.0.0.1/ZendSkeleton/form.html>



# 例題コースコード

簡単な例題を通してZEND FRAMEWORK 2  
の基本動作を説明します

---

# エントリーポイント

---

## Index.php

```
<?php
/**
 * This makes our life easier when dealing with paths. Everything is relative
 * to the application root now.
 */
putenv("ZF2_PATH=" . 'C:\xampp\Library\ZendFramework-2.3.3\library');

//chdir(dirname(__DIR__));
chdir('c:\xampp\ZendSkeleton');

// Decline static file requests back to the PHP built-in webserver
if (php_sapi_name() === 'cli-server' && is_file(__DIR__ . parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH))) {
    return false;
}

// Setup autoloading
require 'init_autoloader.php';

// Run the application!
Zend\Mvc\Application::init(require 'config/application.config.php')->run();
```

# コントローラーの登録

---

## module.config.phpの76行目

- 'controllers' => array(
  - 'invokables' => array(
    - 'Application¥Controller¥Index' => 'Application¥Controller¥IndexController',
    - 'Application¥Controller¥User' => 'Application¥Controller¥UserController',
    - ),
    - ),

# コントローラーとアクション

---

## IndexController.php

```
<?php
/**
 * Zend Framework (http://framework.zend.com/)
 *
 * @link http://github.com/zendframework/ZendSkeletonApplication for the canonical source repository
 * @copyright Copyright (c) 2005-2014 Zend Technologies USA Inc. (http://www.zend.com)
 * @license http://framework.zend.com/license/new-bsd New BSD License
 */

namespace Application\Controller;

use Zend\Mvc\Controller\AbstractActionController;
use Zend\View\Model\ViewModel;

class IndexController extends AbstractActionController
{
    public function indexAction()
    {
        return new ViewModel();
    }
}
```

# コントローラーとアクション

---

```
◦class UserController extends ActionController
◦{
◦ public function tokyoAction()
◦ {
◦     return new ViewModel();
◦ }
◦
◦ public function osakaAction()
◦ {
◦     $data = array(
◦         'key1' => 'あいうえお',
◦         'key2' => 'いろは'
◦     );
◦     return new ViewModel($data);
◦ }
◦
◦ public function kyotoAction()
◦ {
◦     $data = array(
◦         'name' => $this->params()->fromPost( 'name' ),
◦         'year' => $this->params()->fromPost( 'year' ),
◦         'month' => $this->params()->fromPost( 'month' ),
◦         'day' => $this->params()->fromPost( 'day' )
◦     );
◦     return new ViewModel($data);
◦ }
◦}
```

# FORM

---

## form.html

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Zend_Controllerサンプル</title>
  </head>
  <body>
    フォームからの値入力<br>
    <form action="http://127.0.0.1/ZendSkeleton/application/user/kyoto" method="POST">
      名前:<input type="text" name="name" value="お名前">
      年:<input type="text" name="year" value="2000">
      月:<input type="text" name="month" value="12">
      日:<input type="text" name="day" value="31">
      <input type="submit" value="送信">
    </form>
    <br>
  </body>
</html>
```



# ビューファイル

---

## tokyo.phtml

```
<p>
モジュール名 : Application<br>
コントローラ名 : User<br>
アクション名 : tokyo<br>
</p>
```

## osaka.phtml

```
<p>
モジュール名 : Application<br>
コントローラ名 : User<br>
アクション名 : osaka<br>
データ : <?php echo $key1; ?> <br>
データ : <?php echo $key2; ?> <br>
</p>
```

## kyoto.phtml

```
<p>
モジュール名 : Application<br>
コントローラ名 : User<br>
アクション名 : kyoto<br>
<?php echo $name; ?>さんの誕生日は、<br>
<?php echo $year."年".$month."月".$day."日"; ?> です。<br>
</p>
```

# PHPフレームワークのDB処理

---

Zend Framework 2にてデータベース処理

# データベース処理での モジュール構造

---

アプリケーション

Autoload [アダプタ/IDおよびパスワード]

モジュール

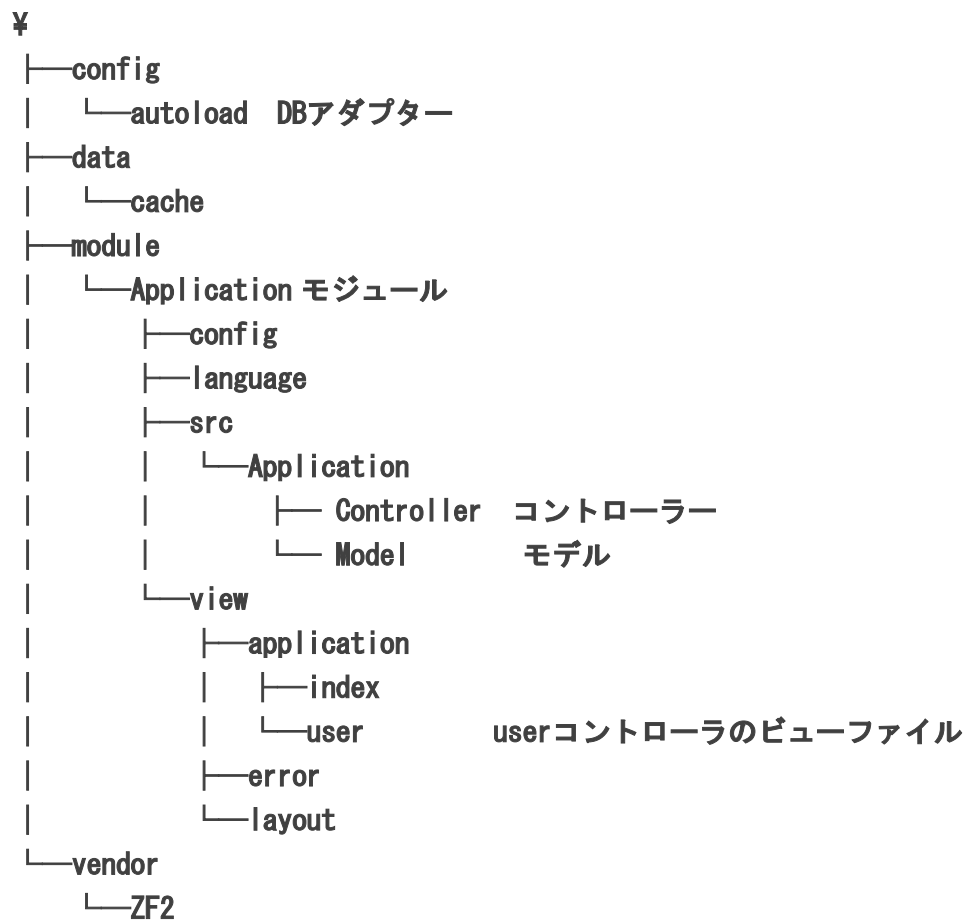
Model [テーブル構造/基本処理]

View [テンプレート]

Controller [個別処理]

# スケルトンプロジェクトで確認

## 圧縮ファイル内部



# 例題ファイル

役割	ディレクトリ	ファイル名
データベースアクセス	config¥autoload¥	global.php
ユーザ名とパスワード		local.php
モデル	module¥Applicaion¥src¥Application¥Model	User.php
連携モデル		UserTable.php
テーブル管理クラス インスタンス生成	module¥Applicaion¥src¥Application	Module.php
コントローラー	module¥Applicaion¥src¥Application¥Controller	UserController.php
テンプレート	module¥Applicaion¥src¥Application¥view	add.phtml sel.phtml sel2.phtml

# global.phpに追記

---

```
return array(
    'db' => array(
        'driver'      => 'Pdo',
        'dsn'         => 'mysql:dbname=test;host=localhost',
        'driver_options' => array(
            PDO::MYSQL_ATTR_INIT_COMMAND => 'SET NAMES ¥'UTF8¥'
        ),
    ),
    'service_manager' => array(
        'factories' => array(
            'Zend¥Db¥Adapter¥Adapter'
                => 'Zend¥Db¥Adapter¥AdapterFactory',
        ),
    ),
);
```

# local.phpを作成

---

```
return array(  
    'db' => array(  
        'username' => 'root',  
        'password' => '',  
    ),  
);
```

# モデルを作成 User.php

---

```
<?php
namespace Application¥Model;
class User
{
    public $sno;
    public $name;
    public $memo;

    public function exchangeArray($data)
    {
        $this->sno      = (isset($data['sno'])) ? $data['sno'] : 0;
        $this->name     = (isset($data['name'])) ? $data['name'] : "";
        $this->memo     = (isset($data['memo'])) ? $data['memo'] : "";
    }
}
```



# モデルを作成 UserTable.php①

---

```
<?php
namespace Application¥Model;
use Zend¥Db¥TableGateway¥TableGateway;
use Zend¥Db¥Sql¥Sql;
class UserTable
{
    protected $tableGateway;
    public function __construct(TableGateway $tableGateway)
    {
        $this->tableGateway = $tableGateway;
    }

    各処理を記述
}
```

# モデルを作成 UserTable.php①

---

```
public function fetchAll()  
{  
    $resultSet = $this->tableGateway->select();  
    return $resultSet;  
}
```

# モデルを作成 UserTable.php②

---

```
public function getUser($sno)
{
    $id    = (int) $sno;
    $rowset = $this->tableGateway->select(array('sno' => $sno));
    $row    = $rowset->current();
    if (!$row) {
        throw new \Exception("Could not find row sno");
    }
    return $row;
}
```

# モデルを作成 UserTable.php③

---

```
public function saveUser(User $user)
{
    $data = array(
        'name'    => $user->name,
        'memo'    => $user->memo,
    );
    $sno = (int)$user->sno;
    if ($sno == 0) {
        $this->tableGateway->insert($data);
    } else {
        if ($this->getUser($sno)) {
            $this->tableGateway->update($data, array('sno' => $sno));
        } else {
            throw new \Exception('Form sno does not exist');
        }
    }
}
```

# モデルを作成 UserTable.php④

---

```
public function deleteUser($sno)
{
    $this->tableGateway->delete(array('sno' => $sno));
}
```

# インスタンスを作成 Module.php

---

```
public function getServiceConfig()
{
    return array(
        'factories' => array(
            'Application\Model\UserTable' => function($sm) {
                $tableGateway = $sm->get('UserTableGateway');
                $table = new UserTable($tableGateway);
                return $table;
            },
            'UserTableGateway' => function ($sm) {
                $dbAdapter = $sm->get('Zend\Db\Adapter\Adapter');
                $resultSetPrototype = new ResultSet();
                $resultSetPrototype->setArrayObjectPrototype(new User());
                return new TableGateway(
                    'meibo', $dbAdapter, null, $resultSetPrototype
                );
            },
        )
    );
}
```

# コントローラーに処理を追加①

```
public function addAction()  
{  
    // 新規追加  
    $user = new User();  
    $user->name = '佐藤栄一';  
    $user->memo = 'ダミーのデータです';  
  
    $this->getUserTable()->saveUser( $user );  
  
    // ビューへ渡す値を配列にて定義  
    $values = array(  
        'key1' => 'INSERT',  
        'key2' => '1件追加',  
    );  
    $view = new ViewModel( $values );  
  
    return $view;  
}
```

モデルへのデータ引渡

DB処理です

テンプレートへのデータ引渡

# コントローラーに処理を追加②

---

```
public function selAction()
{
    $ret=$this->getUserTable()->fetchAll( );

    // ビューへ渡す値を配列にて定義
    $values = array(
        'key1' => 'SELECT',
        'key2' => 'テーブル全体',
        'ret' => $ret,
    );
    $view = new ViewModel( $values );

    return $view;
}
```



# コントローラーに処理を追加③

---

```
public function sel2Action()
{
    $ret=$this->getUserTable()->getUser( 2 );
    // ビューへ渡す値を配列にて定義
    $values = array(
        'key1' => 'SELECT',
        'key2' => '条件指定',
        'ret' => $ret,
    );
    $view = new ViewModel( $values );
    return $view;
}
```

# コントローラーに処理を追加④

```
public function getUserTable()
```

```
{
```

```
    if (!$this->userTable) {
```

```
        $sm = $this->getServiceLocator();
```

```
        $this->userTable = $sm->get('Application¥Model¥UserTable');
```

```
    }
```

```
    return $this->userTable;
```

```
}
```



サービスの生成



インスタンスを返す

# ビューを作成

---

add.phtml

```
<p>
モジュール名 : Application<br>
コントローラ名 : User<br>
アクション名 : add<br>
データ:<?php echo $key1; ?> <br>
データ:<?php echo $key2; ?> <br>
</p>
```

sel2.phtml

```
<p>
モジュール名 : Application<br>
コントローラ名 : User<br>
アクション名 : sel2<br>
データ:<?php echo $key1; ?> <br>
データ:<?php echo $key2; ?> <br>
オブジェクト:<?php var_dump($ret); ?>
<br>
</p>
```

sel2.phtml

```
<p>
モジュール名 : Application<br>
コントローラ名 : User<br>
アクション名 : sel<br>
データ:<?php echo $key1; ?> <br>
データ:<?php echo $key2; ?> <br>
<table class="table">
<tr>
  <th>sno</th>
  <th>name</th>
  <th>memo</th>
</tr>
<?php foreach ($ret as $user) : ?>
<tr>
  <td><?php echo $this->escapeHtml($user->sno);?></td>
  <td><?php echo $this->escapeHtml($user->name);?></td>
  <td><?php echo $this->escapeHtml($user->memo);?></td>
</tr>
<?php endforeach; ?>
</table>
</p>
```

# 参考資料

---

## 書籍

- Zend Framework 2徹底解説
  - 著者: 濱田優
  - 定価: 3,600円(税別)
  - ISBN978-4-7980-4011-0



5年振りにバージョンアップした  
注目PHPフレームワークの全貌!!

