



Seasar Conference 2007

Spring

- 明日から始めるSeasar.PHP -



eXtreme Hardcore PHP Development

Agenda

- What's Seasar.PHP?
- S2Base.PHP5 with PRADO Framework
- S2Base.PHP5 with Zend Framework



自己紹介



eXtreme Hardcore PHP Development

id:yonekawa



eXtreme Hardcore PHP Development

ブログ最近書いてません

<http://d.hatena.ne.jp/yonekawa/>



Hatena::Diary

yonekawa

ようこそyonekawaさん

最新の日記 日記一覧 日記を書く 管理 ログアウト ヘルプ

yonekawaのはてなダイアリー

プロフィール



yonekawa
米川健一 プログラマ(Lv1)

日記の検索

yonekawa

検索

● 詳細 ○ 一覧

最新タイトル

- [Seasar][PHP]S2AnA.PHP5公開しました
- [Seasar][PHP]Seasar Conference 2007 Spring
- [PHP][Prhagger]Prhaggerブログのコードを読む
- [PHP][Prhagger]Prhaggerブログのコードを読む
- [Seasar][PHP]S2AnA.PHP5とPrhagger登場
- [seasar][php]とど整理した

Seasar Conference 2007 Spring

5/27 (SUN), Tokyo

<前の5日分

2007-05-18 編集

[Seasar][PHP]S2AnA.PHP5公開しました

だいぶ遅くなりましたが、無事公開できました。

<http://s2ana.php5.sandbox.seasar.org/>

ぜひ使っていただいて、要望とかいただけたらと思います。

S2AnA.PHP5についても27日にしゃべります。

Seasar Conference 2007 Spring

<http://event.seasarfoundation.org/sc2007spring/>

明日から始めるSeasar.PHP

講演者

講師

明日から始めるSeasar.PHP



eXtreme Hardcore PHP Development

最近はTwitterです

<http://twitter.com/yonekawa>



eXtreme Hardcore PHP Development

Seasar.PHPとの関わり



eXtreme Hardcore PHP Development

S2Dao.PHP5 (S2Pager.PHP5)



S2AnA.PHP5



eXtreme Hardcore PHP Development

What's Seasar.PHP



eXtreme Hardcore PHP Development

JavaのSeasar2を PHP5に移植



Seasar.PHP Products



eXtreme Hardcore PHP Development

S2Base.PHP5

S2Dao.PHP5

S2AnA.PHP5

S2Container.PHP5



S2Container.PHP5



eXtreme Hardcore PHP Development

DI × AOP



eXtreme Hardcore PHP Development

```
$container = S2ContainerFactory::crate('app.dicon');  
$server = $container->getComponent('server');
```

```
<component name="server" class="EchoServer">  
  <aspect>  
    <component class="S2Container_TraceInterceptor" />  
  </aspect>  
</component>
```



オートバイインデイング



```
class EchoServer
{
    private $message;

    public function __construct(Message $m)
    {
        $this->message = $m;
    }
}
```

```
<component name="impl" class="MessageImpl" />
```

```
<component class="EchoServer" autoBinding="auto">
```

```
<!-- <arg>impl</arg> -->
```

```
</component>
```



InterType



```
class Employee
{
    private $id;
    private $name;
    private $product;

    // no getter & no setter..
}
```



```
<component name="propertyInterType"  
    class="S2Container_PropertyInterType" />
```

```
<component name="Employee">  
    <interType>propertyInterType</interType>  
</component>
```

```
$emp = $container->getComponent('Employee');  
$emp->setName('yonekawa');  
echo $emp->getName(); // => yonekawa
```



S2Container.PHP5-1.1.3



S2CONTAINER_PHP5_ENV定数

```
// load foo.dicon
```

```
$container = S2ContainerFactory::create('foo.dicon');
```

```
define('S2CONTAINER_PHP5_ENV', 'test');
```

```
// load foo_test.dicon
```

```
$container = S2ContainerFactory::create('foo.dicon');
```



S2Dao.PHP5



eXtreme Hardcore PHP Development

1. Entityクラス定義

2. Daoインタフェース定義

3. コンポーネント登録



```
class Employee
{
    const TABLE = 'employee';
    private $id;
    private $name;
    private $product;

    public function getId() {
        return $this->id;
    }
    public function setId($id) {
        $this->id = $id;
    }
    ...
}
```



```
interface EmployeeDao
{
    const BEAN = 'Employee';

    public function insert(Employee $employee);
    public function update(Employee $employee);
    public function delete(Employee $employee);

    public function getAllEmployeesList();
}
```



```
<component name="EmployeeDao">  
    <aspect>dao.interceptor</aspect>  
</component>
```



2 Way SQL



```
interface EmployeeDao
{
    const BEAN = 'Employee';

    public function getEmployeeByIdList($id);
}
```

EmployeeDao_getEmployeeByIdList.sql

```
SELECT * FROM employee WHERE id = /*id*/ |
```



Return Type



Entity
Array
YAML
JSON




```
$result = $dao->getEmployeeByIdJson(1)  
echo $result;
```

```
[{"Employee":  
  {"id": "1",  
   "ID": "1",  
   "name": "yonekawa",  
   "NAME": "yonekawa",  
   "product": "S2AnA.PHP5",  
   "PRODUCT": "S2AnA.PHP5"}}]
```



```
/**  
 * @return json  
 */  
public function getEmployeeById($id);
```



S2Dao.PHP5-1.2.0-beta1



eXtreme Hardcore PHP Development

XML Return Type

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <Employee>
    <id>1</id>
    <name>yonekawa</name>
    <product>S2AnA.PHP5</product>
  </Employee>
</root>
```



ValueType

```
class Employee
{
    const TABLE = 'employee'

    private $bigstring;
    const bigstring_VALUE_TYPE
        = S2Dao_PHPType::LobType;

    ...
}
```



S2Base.PHP5



eXtreme Hardcore PHP Development

開発支援コマンドツール



\$ s2base project

\$ ls project

```
app          config      public      var          zf.xml  
build.xml   lib         test        vendor
```




```
Default (71,27)
yonekawa:~/sc2007 yonekawa$ phing
Buildfile: /Users/yonekawa/sc2007/build.xml
    [php] Evaluating PHP expression: ini_set('include_path','lib'.PA
SEPARATOR.ini_get('include_path'))
    [php] Evaluating PHP expression: require_once('S2Base/S2Base.php
php')

project > prepare:
    [php] Evaluating PHP expression: require_once('config/environmen
nc.php')
    [php] Evaluating PHP expression: require_once('S2Base/S2Base.cmd
p')

project > command:

[ Command list ]
0 : (exit)
1 : command
2 : dao
3 : dicon
4 : entity
5 : goya
6 : interceptor
7 : module
8 : service
9 : sqlite-cli
choice ? : █
cmoꝛc6 ; : █
ð : 2dJꝛf6-CJꝛ
8 : 26LΛꝛc6
λ : woqꝛꝛc6
```



```
Default (56,23)

columns ? (id,name,--,,) : id,name,age, memo

[ generate information ]
module name      : conference
dao interface name : EmployeeDao
dao test class name : EmployeeDaoTest
entity class name : EmployeeEntity
entity class extends : none
table name      : EMPLOYEE
columns         : id, name, age, memo
dao dicon file name : EmployeeDao.dicon

confirm ? (y/n) : y
[INFO ] create : /Users/yonekawa/sc2007/app/modules/employeeDao.class.php
[INFO ] create : /Users/yonekawa/sc2007/app/modules/employeeEntity.class.php
[INFO ] create : /Users/yonekawa/sc2007/test/modules/employeeDaoTest.class.php
[INFO ] create : /Users/yonekawa/sc2007/app/modules/employeeDao.dicon
```



Other Framework with Seasar.PHP



with **Symfony**
with **PRADO**
with **Zend Framework**



Eclipse Plugin for S2Dao.PHP5



eXtreme Hardcore PHP Development

ハタさんのブログ

<http://blog.xole.net/article.php?id=555>

The screenshot shows an IDE window titled 'hoge.php' containing the following PHP code:

```
<?php
class Hoge {
    private $hoge;
    private $foo;
    private $bar;

    public function getBar() {
        return $this->bar;
    }

    public function setBar($bar) {
        $this->bar = $bar;
    }

    public function getFoo() {
        return $this->foo;
    }

    public function setFoo($foo) {
        $this->foo = $foo;
    }

    public function getHoge() {
        return $this->hoge;
    }
}
```

A context menu is open over the `setBar` method. The menu items are:

- 元に戻す(U) 入力 (Ctrl+Z) - Undo
- ファイルを前回保存した状態に戻す(Y) (Ctrl+Z) - Revert to previous save state
- 保存(S) (Ctrl+S) - Save
- 表示(W) (Shift+Alt+W) - Show
- 切り取り(C) (Ctrl+X) - Cut
- コピー(C) (Ctrl+C) - Copy
- 貼り付け(P) (Ctrl+V) - Paste
- Source
- S2Dao.PHP5** (highlighted)
- PHP Help (Shift+Ctrl+H)
- Rename Local Variable in Functions/Methods
- Open Declaration/Include (F3)
- Open Included Files List
- Insert HTML (dl,ul,ol,table)
- 実行(R) - Run
- デバッグ(D) - Debug
- チーム(E) - Team

The sub-menu for **S2Dao.PHP5** is open, showing the following options:

- TABLEアノテーションの付加
- setter/getterの生成
- serialize/unserializeの生成
- __toStringの生成
- Entity用メソッドの生成
- 選択中のプロパティの_COLUMNアノテーションの付加

On the right side of the IDE, there is a 'Task List' panel and an 'アウトライン' (Outline) panel. The Outline panel shows the class structure:

- Hoge
 - \$hoge :
 - \$foo :
 - \$bar :** (highlighted)
 - getBar()
 - setBar()
 - getFoo()
 - setFoo()



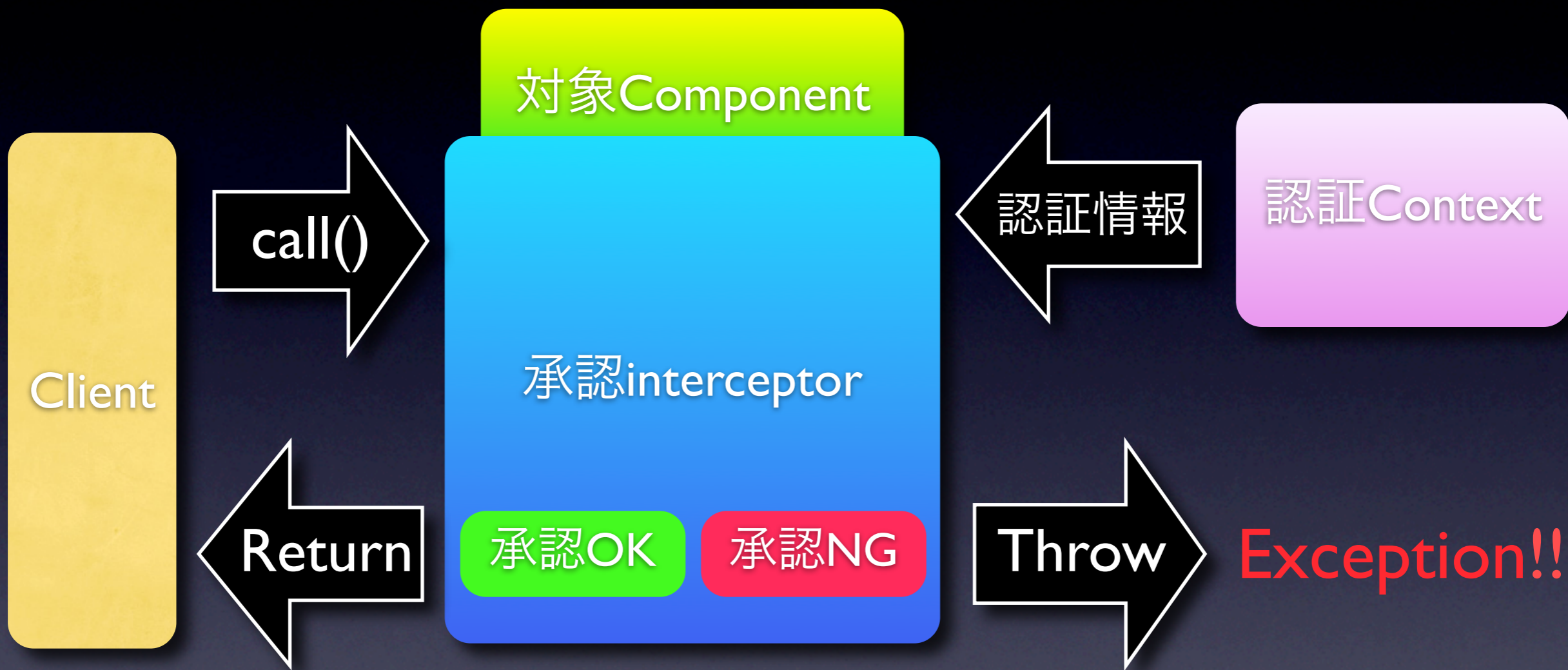
S2AnA.PHP5



eXtreme Hardcore PHP Development

Authentication and Authorization





More Info...



eXtreme Hardcore PHP Development

wiki

<http://wiki.s2php5.jp/>

[[s2container.php5]] wiki.s2php5.jp

現在位置: start » s2container.php5

ソースの表示 · 以前のリビジョン 最近の変更 · 索引 · ログイン

検索

Navigation

- Home
 - syntax
 - sandbox
- Projects
 - S2Container.PHP5
 - S2Dao.PHP5
 - S2Base.PHP5
 - S2Service.PHP5
 - S2AnA.PHP5
 - Committers
- PHP
 - PDO
 - Tips
 - Misc

S2Container.PHP5

- <http://s2container.php5.seasar.org/>

コンテンツ

- [__callメソッドにaspectする](#)
- [DIContainerBuilderの対応](#)
- [InterType](#)
- [コンテナとコンポーネントの組み立て](#)
- [ダイコンチップス](#)
- [ApplicationContext](#)

s2container.php5.txt · 最終更新: 2007/04/01 09:57 by klove

ソースの表示 · 以前のリビジョン 文書の先頭へ

CHIMERIC DE LICENSED W3C DOKUWIKI GET FIREFOX RSS XML FEED

W3C XHTML 1.0



eXtreme Hardcore PHP Development

labs

<http://labs.s2php5.jp/>

labs.s2php5.jp

[Home](#) [contents](#) [feeds](#) [info](#)

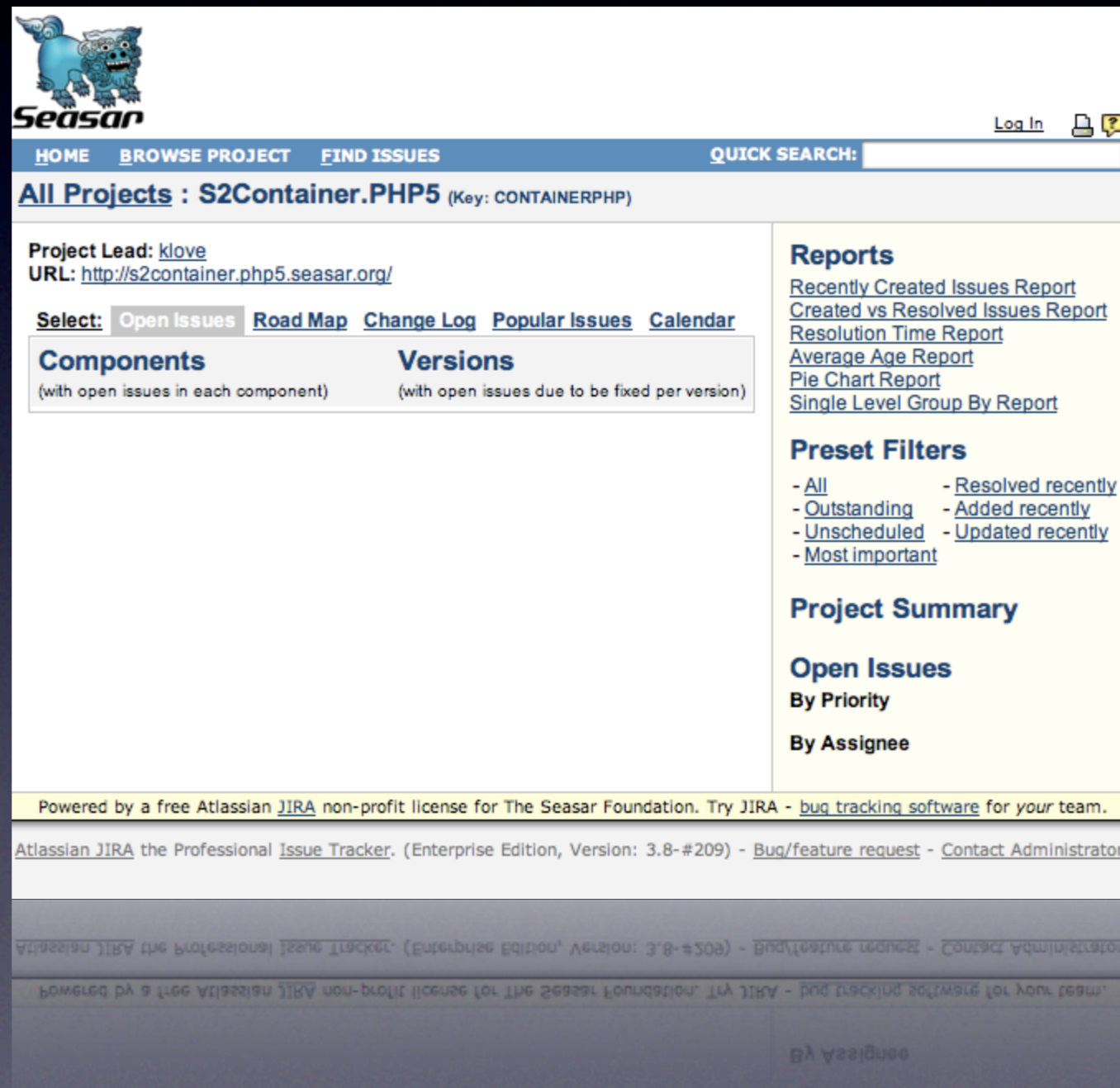
content

- [wiki.s2php5.jp](#)
Wiki docs
- [cl.s2php5.jp](#)
Changelog for s2php5 projects
- [update.s2php5.jp](#)
The Eclipse Plugin Update Site
- [labs.s2php5.jp/s2container](#)
- [labs.s2php5.jp/s2dao](#)
- [labs.s2php5.jp/s2base](#)
- [labs.s2php5.jp/s2service](#)
- [labs.s2php5.jp/misc](#)



Issue Tracker

<https://www.seasar.org/issues/>



The screenshot shows the Seasar Issue Tracker web interface. At the top left is the Seasar logo, a blue cartoon dog. The navigation bar includes links for HOME, BROWSE PROJECT, and FIND ISSUES, along with a QUICK SEARCH field. The main content area displays the project name "All Projects : S2Container.PHP5 (Key: CONTAINERPHP)". Below this, there is a section for Project Lead (klove) and URL (http://s2container.php5.seasar.org/). A "Select:" menu offers options like Open Issues, Road Map, Change Log, Popular Issues, and Calendar. Two main sections are highlighted: "Components" (with open issues in each component) and "Versions" (with open issues due to be fixed per version). On the right side, there are sections for Reports (Recently Created Issues Report, Created vs Resolved Issues Report, Resolution Time Report, Average Age Report, Pie Chart Report, Single Level Group By Report), Preset Filters (All, Outstanding, Unscheduled, Most important, Resolved recently, Added recently, Updated recently), Project Summary, and Open Issues (By Priority, By Assignee). At the bottom, there is a footer with text: "Powered by a free Atlassian JIRA non-profit license for The Seasar Foundation. Try JIRA - bug tracking software for your team." and "Atlassian JIRA the Professional Issue Tracker. (Enterprise Edition, Version: 3.8-#209) - Bug/feature request - Contact Administrators".



コミッタ募集集中!!



eXtreme Hardcore PHP Development

Seasar Conference 2007 Spring



S2BaseとPRADO Framework

2007/05/27 ueyama

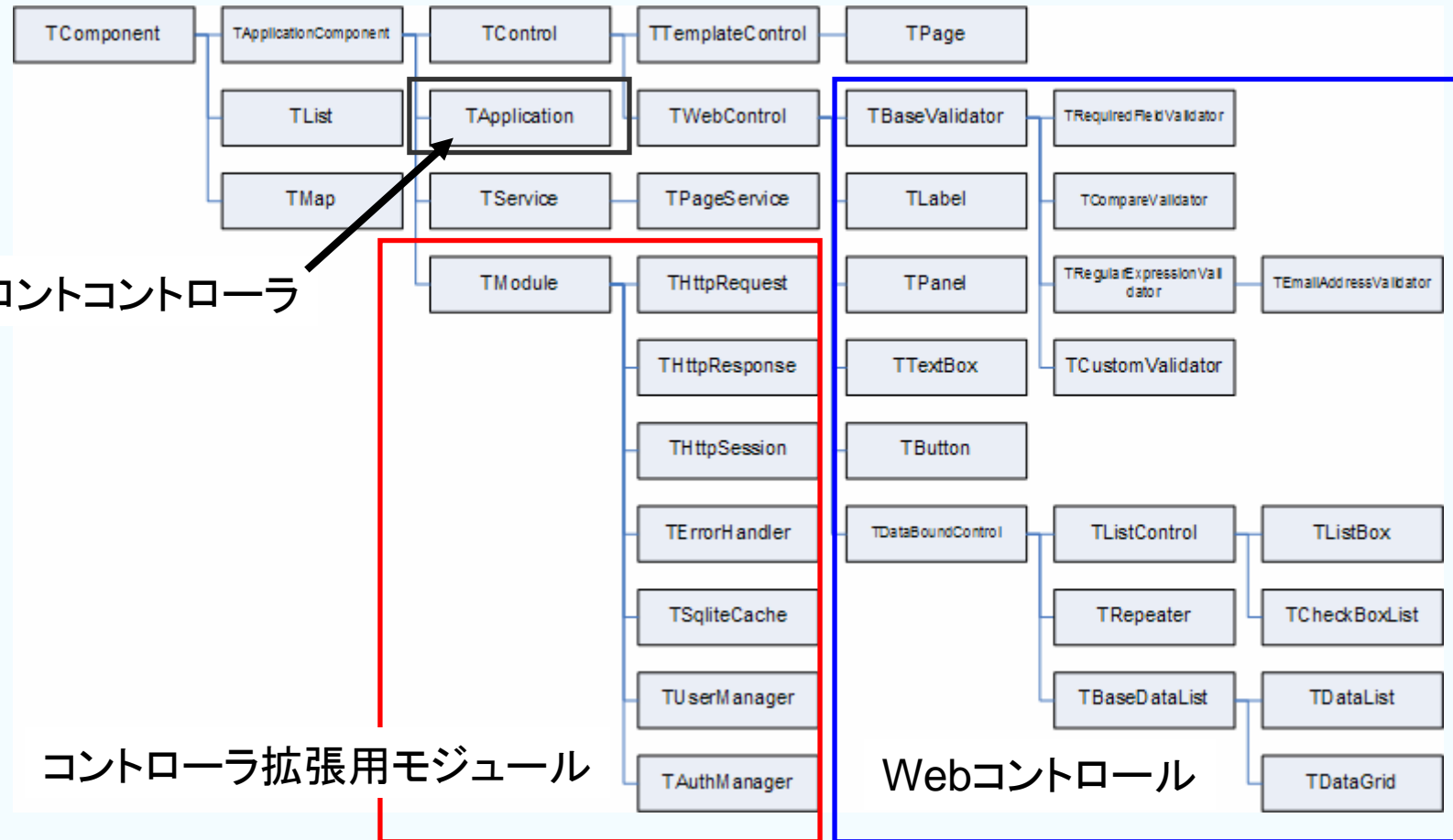
- PRADO Frameworkについて
- S2Baseとの連携について
- PRADO 3.1.0 RC 新機能について
- S2Base PRADO pluginの今後について
- 最後に

- Zendコンテストグランプリ作品
- Tapestry, ASP.NET, Delphiをインスパイア
- PHP5専用のプレゼンテーション層フレームワーク
- 現在のバージョンは、3.0.7
- 次期バージョンの3.1は、RC版がリリース済み

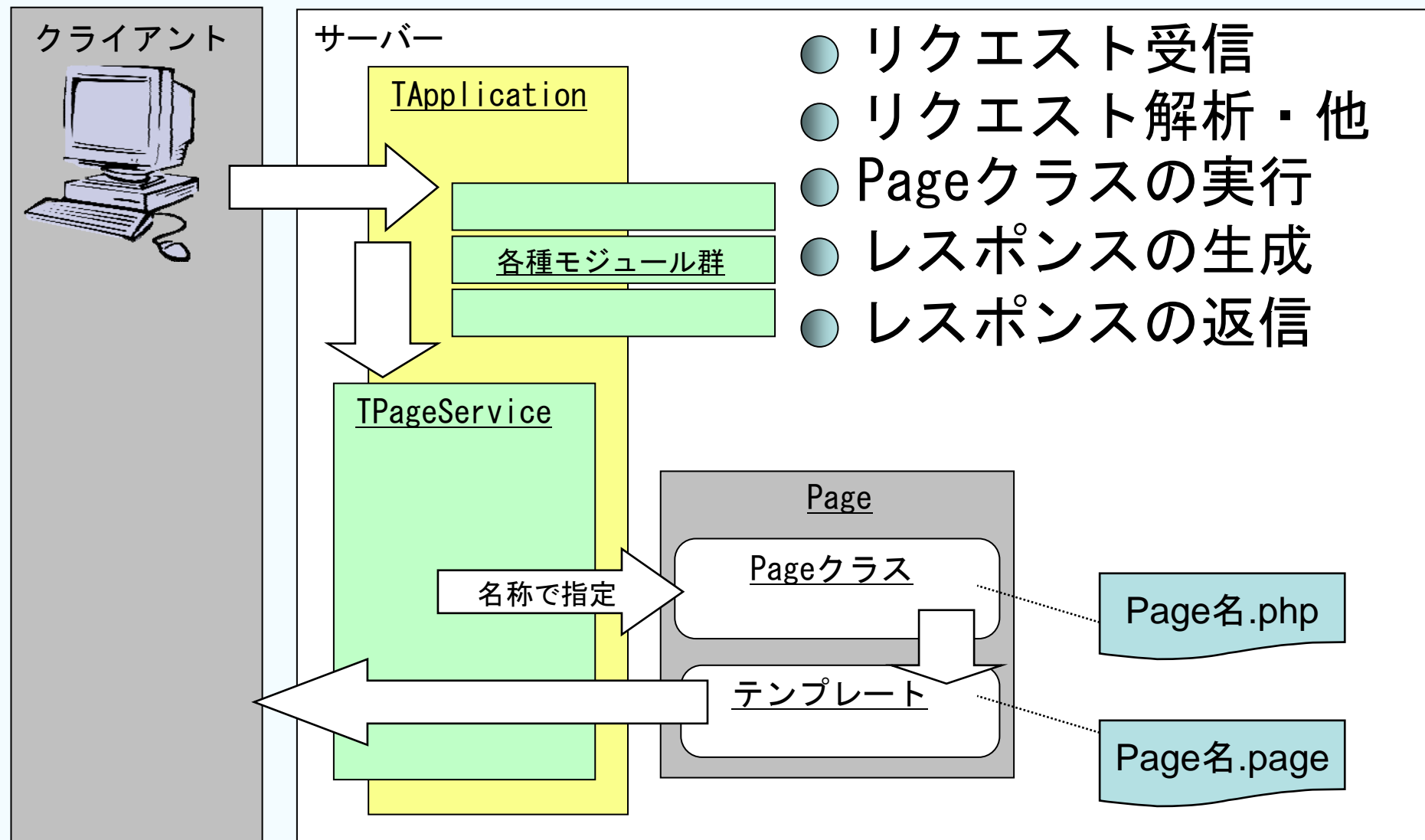


PRADO Frameworkの特徴

- ページ指向、コンポーネント指向
- イベントドリブなプログラミング
- プレゼンテーションと処理の分離
- 豊富なWebコントロール群による画面の作成



- これらを組み合わせてアプリを構築

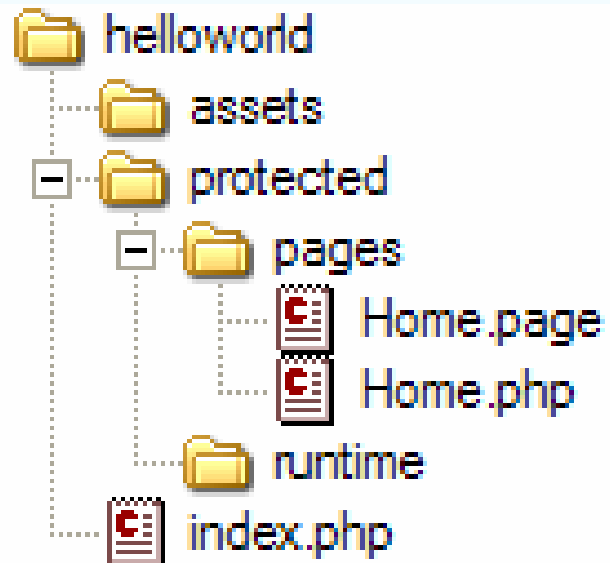


● ボタンのみの簡単なアプリケーションを構築

<http://サーバーアドレス/helloworld/> にてアクセス

*)デフォルトのページ名は、Home

● ディレクトリ構成



● 実行結果

Click me

クリック

Hello World!

● エントリースクリプト 【index.php】

```
require_once('path/to/prado.php');  
$application=new TApplication;  
$application->run();
```

- PRADO Frameworkの読み込み
- フロントコントローラ (TApplication) の生成
- フロントコントローラの実行

● ページテンプレート 【Home. page】

```
<html>
  <body>
    <com:TForm>
      <com:TButton Text="Click me" OnClick="buttonClicked" />
    </com:TForm>
  </body>
</html>
```

デザインと処理の分離
【デザイン記述部】

- コンポーネント (TForm, TButton) を配置することによる画面の作成

コンポーネント指向

- ユーザ操作 (OnClick) 時の処理を指定

イベントドリブン

● ページクラス 【Home.php】

```
class Home extends TPage{  
    public function buttonClicked($sender, $param) {  
        $sender->Text="Hello World!";  
    }  
}
```

デザインと処理の分離
【処理記述部】

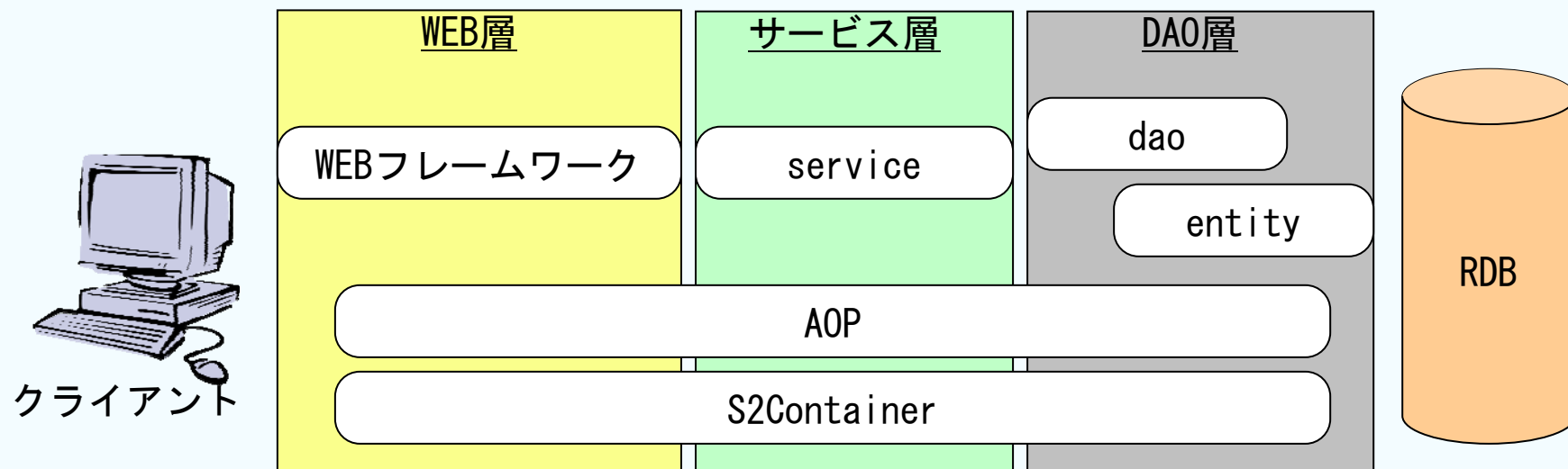
● 画面の処理は対応するPageクラスに記述

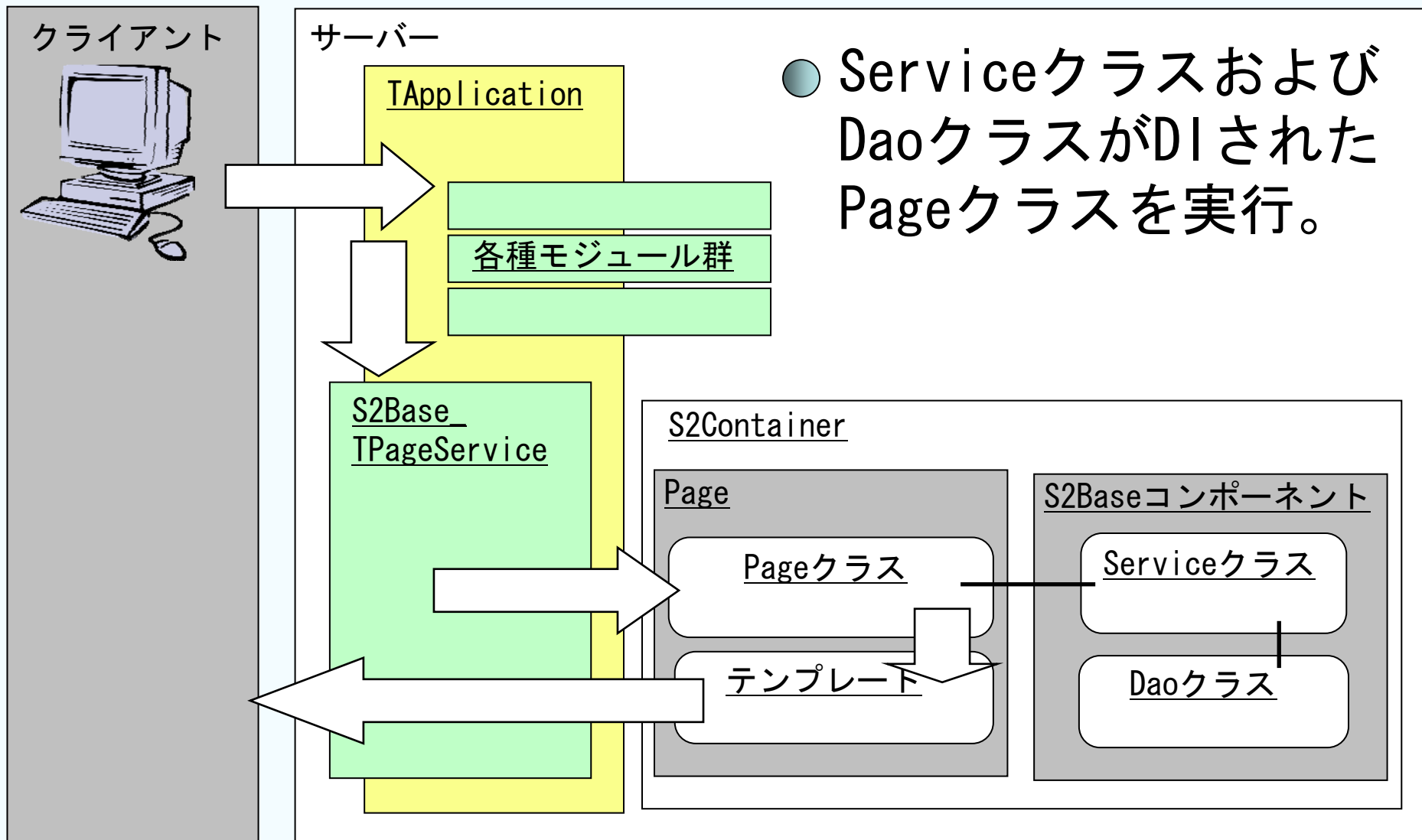
ページ指向

● ユーザ操作 (OnClick) 時の処理を記述

イベントドリブン

- S2ContainerとS2Daoを用いた開発を支援するコマンドツール
- WEBフレームワークとの連携をサポート
- GOYAの内部設計を取り入れている





- Serviceクラスおよび DaoクラスがDIされた Pageクラスを実行。



PRADO Framework 3.1.0新機能

- AJAXのサポート
- DBアクセス機能の充実
- 新規Serviceコンポーネント
- コマンドラインアプリケーション構築対応

● Active ControlによるAJAXアプリ構築

Javascriptによるクライアントサイドのスク립トを自動生成する
Webコントロール群を標準提供

アプリケーションの設定変更とWebコントロールの変更で
PRADOによって構築したWebアプリケーションのAJAX化が可能

```
<com:TActiveButton Text="Click me" OnClick="buttonClicked" />
```

● prototype.jsによるAJAXアプリ構築

公式サイトにprototype.js利用ガイドを掲載。
一般的なAJAXアプリケーション構築を支援

- PDOをベースとした2種類のDBアクセス機能
 - Active Record
データベースのテーブル／ビューの行をラップするオブジェクト (Active Record)を中核としたDBアクセス機能
 - SQLMap Data Mapper
PHPオブジェクトとデータベースのテーブル／ビューのマッピング……O/Rマッピングを行うフレームワーク
- TScaffoldViewコンポーネント
Active RecordによるDBアクセス機能を用いてScaffold機能を簡単に実現するWebコントロール

```
<com:TScaffoldView RecordClass="UserRecord" />
```

- TSOAPService

PHP のSOAP拡張を利用。
PHPクラスをSOAPサービスとして公開することができる。

- TJSONService

JSONフォーマットのレスポンスを返却するサービスを構築
することができる。

http://サーバーアドレス/helloworld/index.php?json=get_article 等とアクセス

- TFeedService

ユーザーにFeed情報を提供することができる。

<http://サーバーアドレス/helloworld/index.php?feed=ch2> 等とアクセス

- TShellApplicationコンポーネント

PHPによるコマンドラインアプリケーション構築の際のフロントコントローラ。

- 適用例 【prado-cli.php】

PRADOアプリケーション構築を支援するコマンドラインツール。下記機能を提供

1. アプリケーションの雛形作成機能
2. シェル機能(デバッグ等に利用)
3. 単体テスト・機能テストの雛形作成機能
4. データベースのメタ情報を基にしたActive Recordクラス生成機能

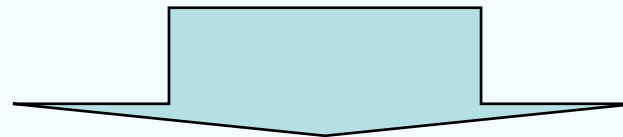
● プレゼンテーションフレームワーク

+

● DBアクセス機能の充実

● Webサービス対応

● AJAX対応 等



All in OneなWebフレームワークへ

- PRADO 3.1新機能への対応
- prado-cli.phpの拡張
- Teeda. PHP5構想





S2Base PRADO pluginの今後【3.1対応】

● 新規サービスとS2Baseとの連携

新規に追加されたサービスコンポーネントへのS2Baseとの連携機能の追加。TPageServiceの拡張と同様に実装予定。

● S2Daoを用いたScaffold機能の実現

T ScaffoldView Webコンポーネントを参考にして、S2Daoを利用したScaffold機能を実現するWebコンポーネントを実装

【イメージ】

```
<com:TS2DaoScaffoldView DaoClass="UserDao" />
```



S2Base PRADO pluginの今後【prado-cli】

- S2Baseの機能をprado-cli.phpベースで構築
 - Prado-cli.phpは、標準のアプリケーション構築支援ツール。
 - PRADOコミュニティへS2Containerを広めることも念頭に。

● Teeda Extention (HTMLテンプレート) をPHPに移植

ページ指向フレームワークであるPRADOは
基盤として最適

● PRADO

Pageテンプレート

HTMLによる画面デザイン
Webコンポーネントの配置

ページクラス

イベントハンドラーの記述

● Teeda Extention

HTMLテンプレート

HTMLによる画面デザイン
Spanタグ、IDによるコンポーネント部指定

ページクラス

アノテーションによるWebコンポーネント
の配置
イベントハンドラーの記述



- S2Prado
PradoとS2Containerが連携するコア部
- Teeda Extension.PHP5
S2Prado向けHTMLテンプレートエンジン
- 開発支援
prado-cli.php拡張
- S2Base PRADO plugin

- PRADO公式フォーラムにて情報交換を！

<http://www.pradosoft.com/forum/> にアクセス！

- PRADO公式サイトの日本語化を進めています。

http://www.pradosoft.com/wiki/index.php/Main_Page にアクセス！

- S2Base PRADO pluginの開発に協力を！

<http://www.seasar.org/> にアクセス！

Seasar Conference 2007 Spring



S2BaseとZend Framework

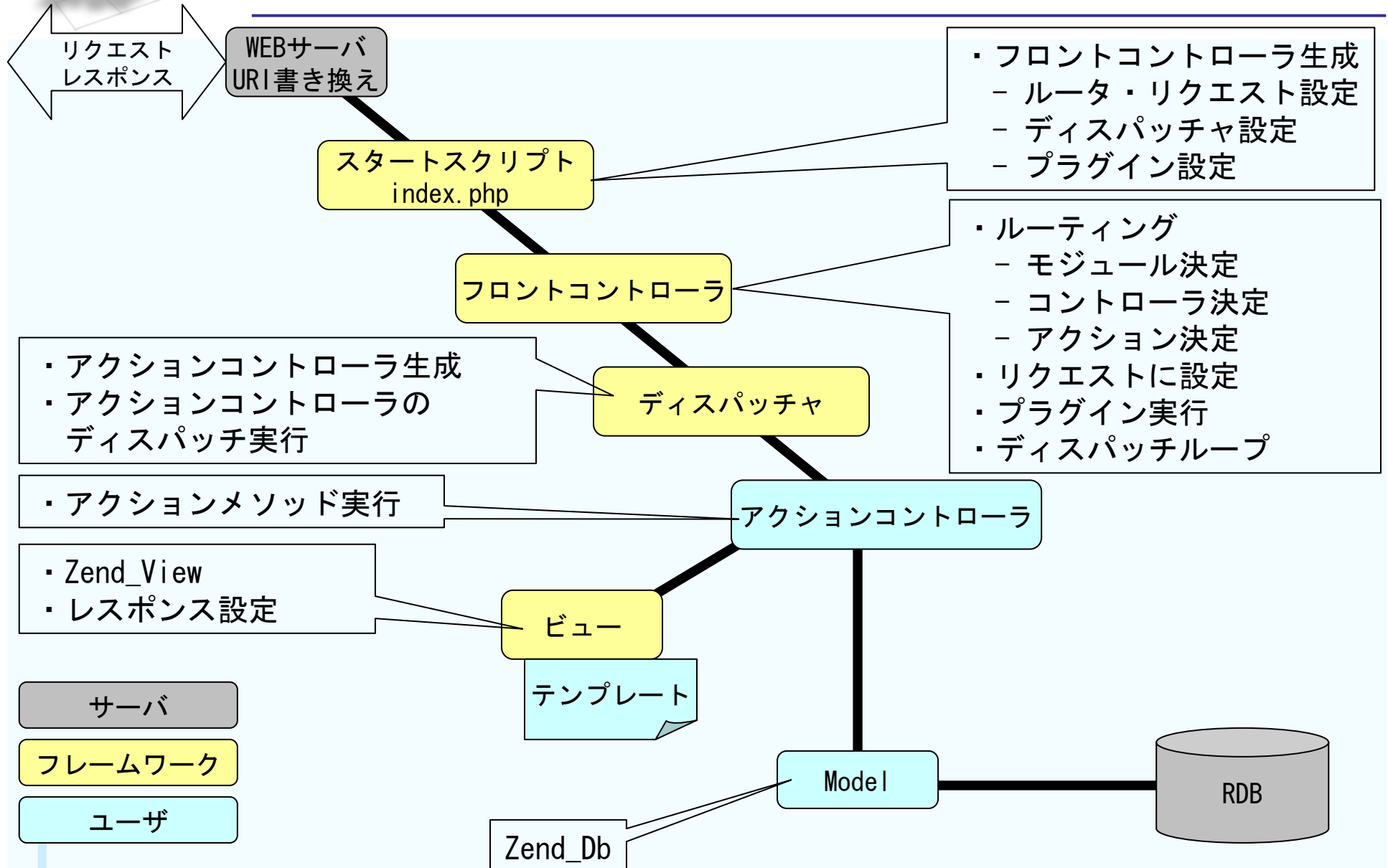
2007/05/27 klove

- Zend Frameworkについて
- S2Baseとの連携について
- S2Base_Zf デモ

• ライブラリ群

Acl	Gdata	Request
Auth	Http	Rest
Cache	Json	Search
Config	Locale	Server
Console	Log	Service
Controller	Mail	Session
Date	Measure	Translate
Db	Memory	Uri
Feed	Mime	Validate
Filter	Pdf	View
		XmlRpc

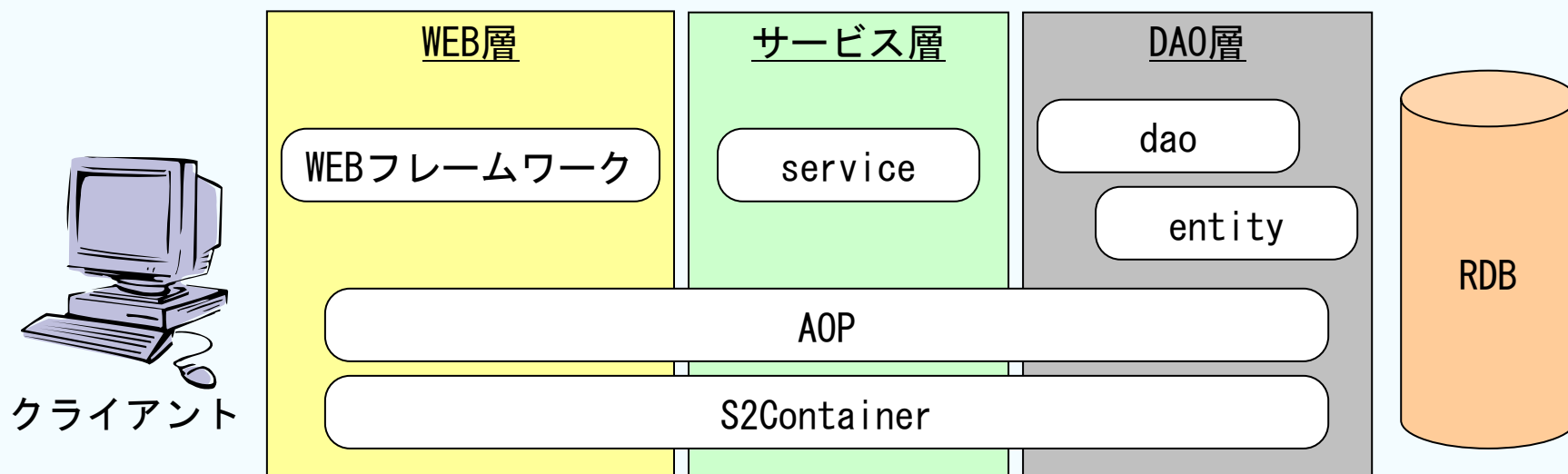
Zend_Controller 処理の流れ



- `http://localhost/index/search` の場合
モジュール名 : なし (default)
コントローラ名 : index
アクション名 : search

- `http://localhost/admin/index/login` の場合
モジュール名 : admin
コントローラ名 : index
アクション名 : login

- S2ContainerとS2Daoを用いた開発を支援するコマンドツール
- WEBフレームワークとの連携をサポート
- GOYAの内部設計を取り入れている





プロジェクトディレクトリ

```
s2base project/  
+-- app/  
|   +-- modules/ ← アプリケーション作成ディレクトリ  
+-- config/  
+-- lib/  
+-- public/ ← WEBサーバで公開  
+-- test/  
|   +-- modules/  
+-- var/  
+-- vendor
```



Moduleについて

```
s2base project/  
+-- app/  
  +-- modules/  
    +-- default/  
      +-- IndexController.php  
        searchAction メソッド
```

<http://localhost/index/search>

モジュール名 : なし (default)
コントローラ名 : index
アクション名 : search

モジュールを追加すると・・・

```
s2base project/
```

```
+-- app/
```

```
+-- modules/
```

```
+-- default/
```

```
+-- IndexController.php
```

```
searchAction メソッド
```

<http://localhost/index/search>

モジュール名 : なし (default)

コントローラ名 : index

アクション名 : search

```
+-- admin/
```

```
+-- IndexController.php
```

```
loginAction メソッド
```

<http://localhost/admin/index/login>

モジュール名 : admin

コントローラ名 : index

アクション名 : login



Controllerについて

```
s2base project/  
+-- app/  
  +-- modules/  
    +-- default/  
      +-- IndexController.php  
      +-- index/  
        +-- dao/  
        +-- dicon/  
        +-- entity/  
        +-- interceptor/  
        +-- model/  
        +-- service/  
        +-- validate/  
        +-- view/
```

index コントローラディレクトリ

コントローラを追加すると・・・



Controllerについて

```
s2base project/  
+-- app/  
  +-- modules/  
    +-- default/  
      +-- IndexController.php  
      +-- index/  
        | +-- dao/  
        | +-- dicon/  
        | +-- entity/  
        | +-- interceptor/  
        | +-- model/  
        | +-- service/  
        | +-- validate/  
        | +-- view/  
      +-- LoginController.php  
      +-- login/  
        +-- dao/  
        +-- . . .
```

} index コントローラディレクトリ

} login コントローラディレクトリ

- プロジェクトディレクトリが作成できる
- いろいろ生成コマンドが付いてくる
- S2ContainerとS2Daoが使える
- Zend_ControllerとS2Containerの連携がついてくる
- その他、いろいろ . . .
(ビューにSmartyとか、ValidatePlugin とか、)

- プロジェクトディレクトリの作成
- モジュールを使用する場合と使用しない場合
- GOYAコマンドを使ってみる

WEBサーバにApache、データベースにMySQLを使用します。
S2BaseはPEARでインストールしています。



おわり