

Seasar Conference 2006 Autumn



Teedaで始めるPage駆動開発

Teedaプロジェクト/リーダー
電通国際情報サービス/Seasar2技術推進G

大谷 晋平



- Teedaとは
- Page駆動開発とは何か
- Teedaの特徴
 - Demo
- Teedaを支えるEclipseプラグイン「Dolteng」
- TeedaのRoadmap
- まとめ



自己紹介

- 名前：大谷 晋平 (おおたにしんぺい)
- Blog：<http://d.hatena.ne.jp/shot6/>
- 所属：株式会社電通国際情報サービス (<http://www.isid.co.jp>)
- コミッタ：Teeda、S2JSF、S2Struts、Seasar2
 - プレゼンテーション層で主に開発しています
- メール：shinpei.ohtani@gmail.com
- 特徴：めがね



- Teedaとは

- プレゼンテーション層の開発をよりシンプルに。

- Webアプリケーションフレームワーク
- JavaでPage駆動開発を実現

- Teedaとは沖縄の言葉で「太陽」



- <http://teeda.seasar.org/ja/>

- Teeda 1.0 リリース (^▽^)/

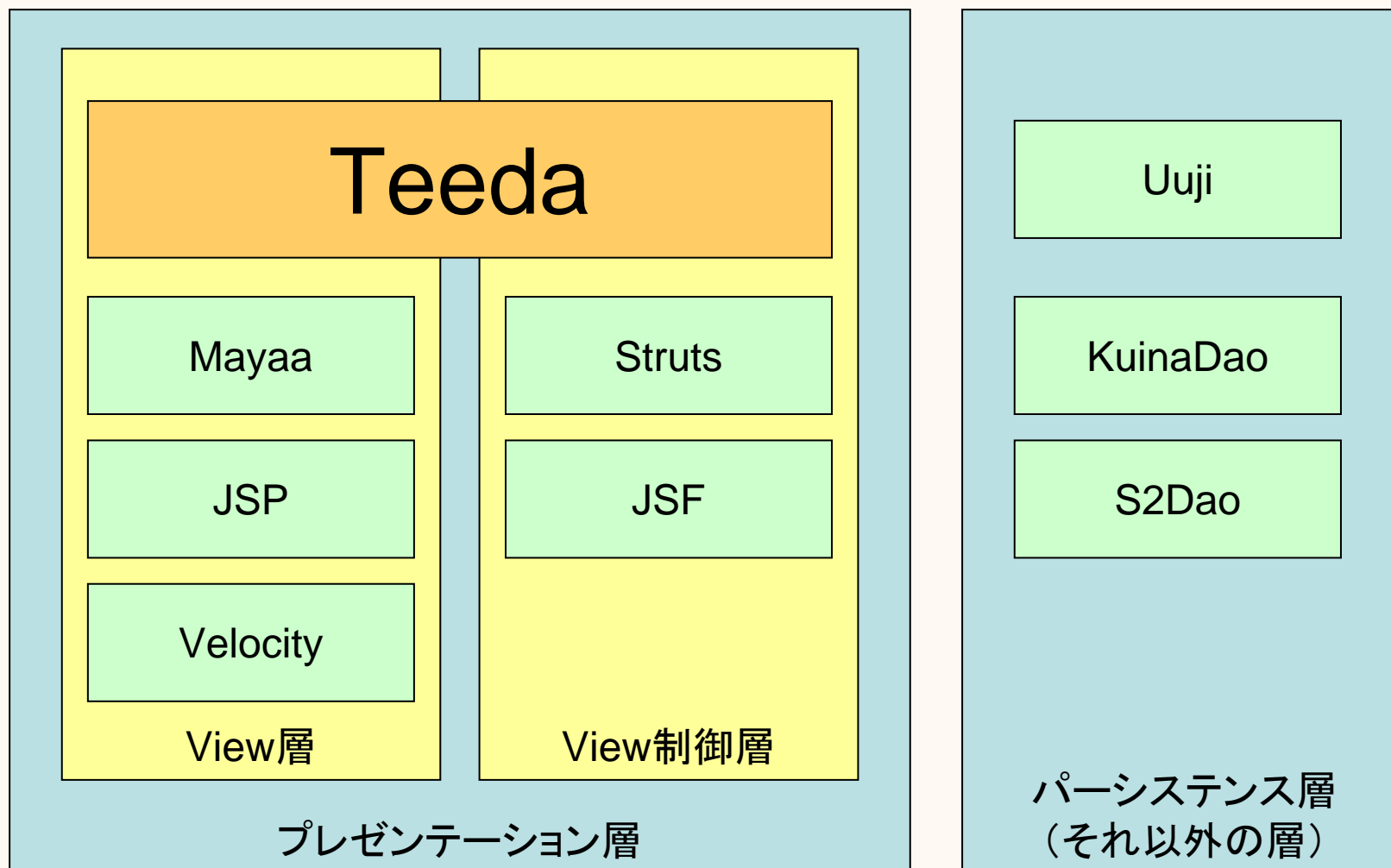
- 11月11日11時11分無事リリース



- Page駆動開発とは
 - HTMLをベースとした開発スタイル
 - HTMLにJavaクラスをマッピングする
 - 対応関係が明確なのでムリが少ない
 - 1Page = 1HTML + 1POJO(単なるJavaクラス)
 - TeedaではPOJOのことをPageクラスと呼びます
 - マッピングの設定が必要なの？
 - Teedaでは規約に従い、シンプルに名前をあわせるだけ。
 - » add.html → AddPage.java
 - » HTML名の頭を大文字 + Page



Teedaの位置づけ





- Webアプリケーションフレームワーク
 - さくさくWeb開発できる
 - **Super Agile**と**Easy Enterprise**の両面サポート
 - **Super Agile** = 独自仕様で開発生産性を重視
 - **Easy Enterprise** = 標準仕様に基づき機能拡張
- 設定ファイルをほぼ書かない
- PRGパターンの実装
- Stateless



WebアプリケーションFW、Teeda(1)

- Seasar2.4の機能を100%発揮できる
 - さくさく感のあるリズムのよい開発が可能に。
 - 最大のウリはHotDeployによる段階的な開発
 - 規約によって開発を効率化
 - フルスタック(WebからDB)で、まとめて一つのフレームワークとして提供される
 - 本来必要の無い設定を色々なところでやる必要もない
 - 一環したポリシーに基づいていて、違和感が少ない。
 - FWのポリシーの差を、使う人が吸収するのは時代遅れ



WebアプリケーションFW、Teeda(2)

- Super AgileとEasy Enterpriseの両面を支える
 - きちんと定義されたライフサイクルを持つ
 - プレゼンテーションの開発に必要な機能を持つ
 - Validation、Converterなど
 - 拡張性
 - Teedaは以下の要素から成り立つ
 - CoreはJSF仕様の実装部分
 - Extensionは拡張機能部分(今回はここ)
 - Ajaxな機能を実現する部分



設定ファイルを書かない(1)

- 設定ファイルをほぼ書かない
 - XML地獄からの脱却
 - 規約に基づき、必要なリソースに記述
 - 開発のリズムを崩さないためにも重要
 - 画面遷移の設定
 - HTMLに簡単な規約に基づいて書くだけで遷移
 - Pageクラス (Model) の登録
 - Seasar2.4のSmartDeploy機能でほぼ必要なし
 - 始めの一回で基点となるパッケージを指定するのみ



設定ファイルを書かない(2)

- 画面遷移は、HTMLに書くだけ
 - 非常に直感的
 - 開発に必要なリソース以外は修正しない

addResult.htmlに遷移する。

```
<input type="button" value="calculate" id="goAddResult" />
```

doCalculate()を呼んで、その結果で遷移する

```
<input type="button" value="calculate" id="doCalculate" />
```



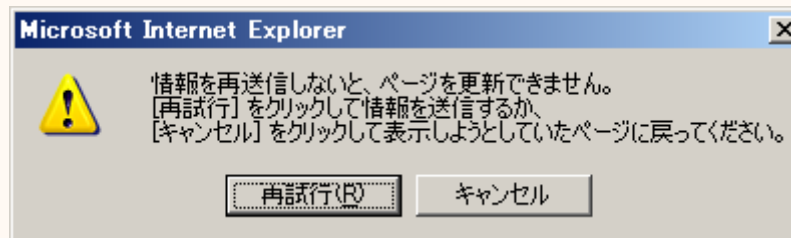
設定ファイルを書かない(3)

- Modelの登録
 - Rootパッケージ名を入力するだけ。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE components PUBLIC "-//SEASAR//DTD S2Container 2.4//EN"
"http://www.seasar.org/dtd/components24.dtd">
<components>
  <component class="org.seasar.framework.convention.impl.NamingConventionImpl">
    <initMethod name="addRootPackageName">
      <arg>ここにrootパッケージ名を入力</arg>
    </initMethod>
  </component>
</components>
```



- PRGパターンとは何か？
 - POST-REDIRECT-GETの組み合わせで画面遷移
 - 開発者を悩ませる4つの問題
 - ブラウザのRELOAD問題
 - ブラウザの戻るボタン問題
 - URLがずれる
 - この画面消せないんですか？というお客様の要望w



→この問題への解決方法、それが**PRGパターン**



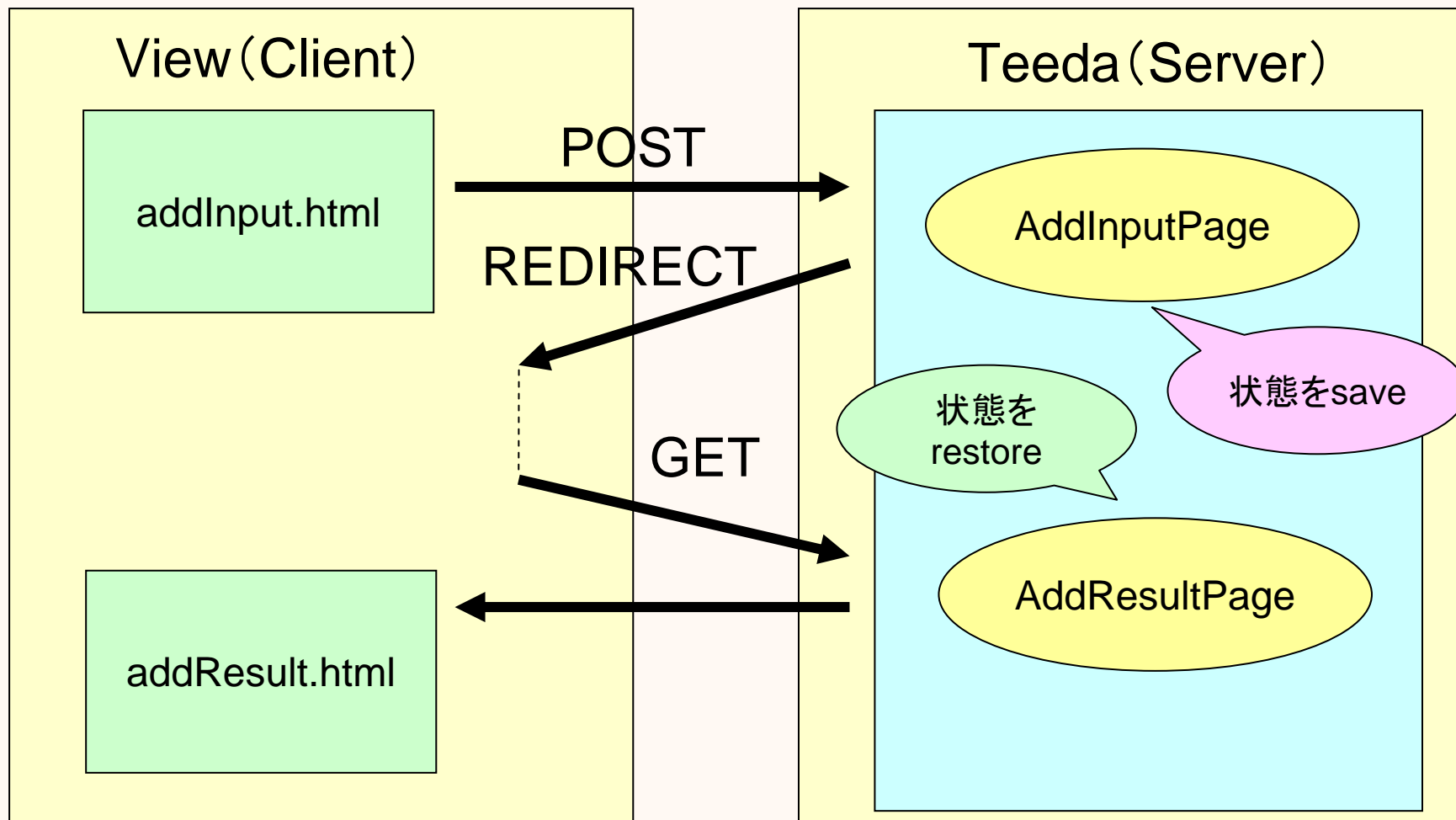
- 何が問題か
 - Forwardでの問題はPOSTしたresponseをそのまま使うため、reloadしたときも再度POSTされる。
 - REDIRECTを挟み、GETを使えば対処できる。
 - GETであれば、RELOADされてもModelが意図しない状態で更新されない
 - URLもずれない。
- PRGパターンは責務の分離
 - Modelの状態変更と、画面の描画の分離



PRGパターンの実装(3)

- Teedaでの実装は、
 - Viewをハンドリングする部分 (ViewHandler) で実装
 - REDIRECTする直前にPageのプロパティをsave
 - RESTOREされたときに、同一のプロパティ名があればrestoreする。
 - TAKE_OVERアノテーションで引き継ぐ・引き継がないを細かく制御可能。

```
@TakeOver(type = TakeOverType.NEVER)  
public String doFinish() {
```





- TeedaはStateless
 - Teedaでは極力状態を保持しない
 - Teedaでは明確な方針を出してます
 - 状態はHTML側に埋め込むか、DBに格納する
 - Teeda内部ではSessionを原則使わない
 - 限定して利用
 - PRGのPage間の遷移時に
 - ダブルサブMIT対策に
 - Sessionに状態を中途半端に保持しない
 - 通常のJSFでは、状態をひたすら保持

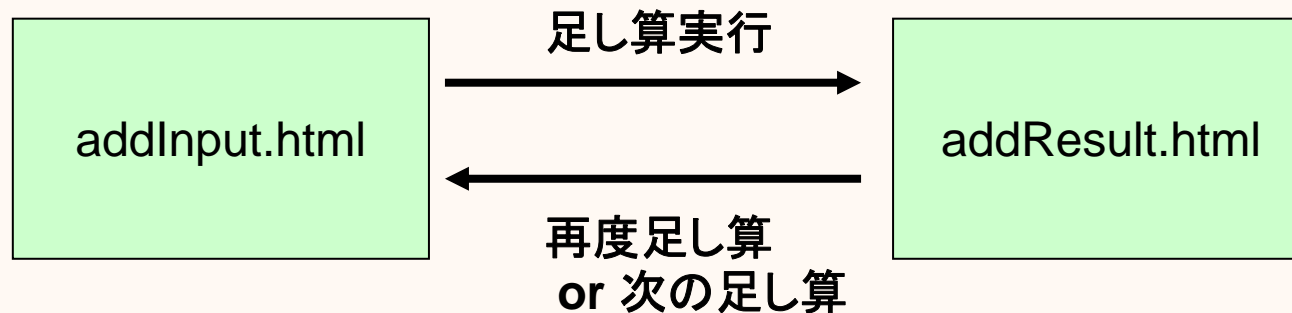


- ピュアなHTMLでViewを記述可能
 - HTMLテンプレートと呼びます
 - 設計の時に使ったHTMLのMockをそのまま実行環境に持っていける
 - Teedaではidを振るのみ。
これだけでコンポーネントとマッピングされる。
- ```
<input type="text" id="arg1" />
```
- マッピングの詳細規約などは最後に。



- HTMLテンプレートの分類
  - ポイントはコンポーネントの定義をどこに記述するか
    - どこにも書かない
      - **Teeda** (idベースの規約によって、自動設定する)
    - View (HTMLなど) に直接記述する
      - **S2JSF**
    - Viewとは別のファイルに記述する (直接も書ける)
      - **Mayaa**、**Tapestry4**
    - Javaで記述する
      - **Wicket**、**Click**など

- 足し算のサンプル
  - 足し算をして、結果を次画面に表示
  - 入力なし、または3桁未満ならValidationエラー



- まずはHTMLだけで動かしてみましよう。



## 足し算によるDemo(続き1)

- HTML (addInput.html)

```
<html>
(途中省略)
<form id="addForm">
<tr>
<td></td><td><input type="text" id="arg1"/></td><td></td>
</tr>
<tr><td> + </td>
<td><input type="text" id="arg2"/></td><td></td>
</tr>
</table>
<input type="button" value="calculate" id="doCalculate"
 onclick="location.href='./addResult.html'"/>
</form>
</body>
</html>
```

idによって、Pageクラスと  
マッピングする。



## 足し算によるDemo(続き2)

- Pageクラス(addInputPage.java)

```
public class AddInputPage {
 //各種使用するValidatorの設定(省略)
 private Integer arg1;
 private Integer arg2;
 private Integer result;

 public String doCalculate() {
 result = new Integer(arg1.intValue() + arg2.intValue());
 return "addResult"; // 次の画面遷移はaddResult.html
 }

 //各種getter/setter省略
}
```

idによって、Pageクラスとマッピングする。

ボタンが押されたら呼ばれるメソッド



## 足し算によるDemo(続き3)

---

- では実際に動かしてみましよう。
  - このDemoでは次のことがわかります
    - HotDeployによるサーバNonStopさくさく開発
    - Htmlテンプレートが実際に動く
    - Validation



## Ecilpseプラグイン「Dolteng」

- Dolteng
  - Teedaでの開発効率を促進
    - より、さくさく感のある開発へ。
    - HTMLとPageクラスのマッピング機能
      - QuickJUnitのように、Ctrl+5でHTML $\leftrightarrow$ Page間を遷移
    - 正しくマッピングされている場合はマーカーを表示
  - ではDemoをご覧ください





- Roadmap

- レイアウト機能 (TilesやS2JSFのと同様)
- ClientSideのValidation機能
  - サーバサイドでValidatorを設定すると、自動で出力
- Ajaxな機能の強化

2007年初頭(次のSeasarCon?)

- Flexとの連携(妄想^^; ?)
  - HTMLの限界を超える部分をFlexに委譲



- Teeda
  - プレゼンテーション層の開発をシンプルにする
  - Page駆動なWebアプリケーションフレームワーク
  - PRGパターンの実装
  - 設定いらず
    - 設定ファイルを書く必要は無い
    - 書く必要があるHTMLとPOJOを規約に沿って書く
  - HTMLテンプレート
    - 基本はidを書いて、Pageとマッピングするだけ。



- **Super Agile Web Development with Seasar2**のセッションでもっと凄いDemoをやります。
- リソース
  - Teedaを始めるには、、、
    - <http://www.seasar.org/wiki/index.php?Teeda%2FgettingStarted>
  - Teedaで問題があったら、
    - Seasar-userまでメールくださいm(\_\_ \_\_)m
    - または、<http://d.hatena.ne.jp/shot6/> にコメントください



- 感謝
  - 会場に来てくださった皆さんm(\_\_ \_\_)m
  - Teedaを使ってくださっているUserの人達
  - 一緒に議論し、創っているTeedaチームの皆さん
  - S2、S2JSFというカッコイイプロダクトを作ったひがさん
  - Seasarプロジェクトを支えている人達
  - 各種オープンソースに関わっている人達
    - 特にきっかけを作ってくれたMyFacesに感謝



Thanks for ALL!!!

---

ご清聴  
ありがとうございました。