Teeda

- JSFからのTeeda -

自己紹介

- 名前
  - 米林　正明
- ID
  - id:yone098
- 所属
  - 株式会社Abby　代表取締役社長
- BLOG
  - よねのはてな http://d.hatena.ne.jp/yone098

自己紹介

- Seasarとの関わり
  - Teeda committer
  - S2JSF Project leader
- Teeda執筆活動
  - JavaExpert#01
  - JavaExpert#02
  - Teeda本 5月末～6月初

自己紹介

- 宣伝
  - 株式会社ヌーラボ
    - http://www.nulab.co.jp/
    - 業務提携しました
  - Backlog
    - http://www.backlog.jp/
  - キュアル
    - 日本初(世界初？)プリクラ動画
    - LIVE PARK in AKIBAにて3台稼動中

Agenda

- What is Teeda?
- JSF and Teeda Core
- Teeda Extension
- Teeda Ajax
- Teeda Test
- Teeda Tips

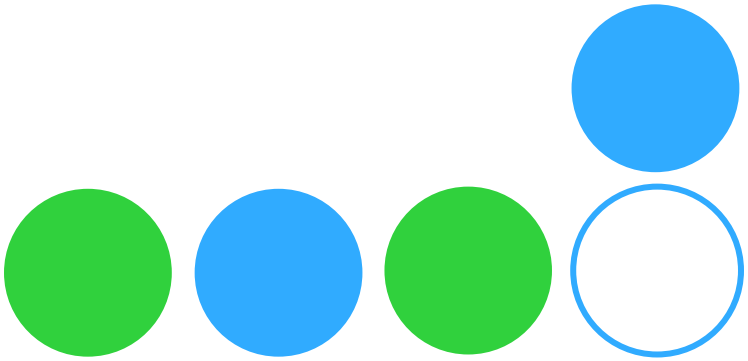
- What is Teeda?

Teeda

- “てぃーだ”
  - 沖縄の言葉で「太陽」
- 実装Webアプリケーションフレームワーク
  - Java実装
- Seasarプロジェクトにて開発
  - http://teeda.seasar.org/ja/
- 日本初JSF実装
  - コンセプト JSF meets DI × AOP

Teeda

- Teedaの構成
  - **Teeda Core**
    - JSF1.1の実装
    - JSFにおけるUIコンポーネントの管理にSeasar2を利用
  - **Teeda Extension**
    - Teeda Coreをベースに HTMLテンプレートと規約に基づいた拡張機能
  - **Teeda Ajax**
    - Ajaxに特化したライブラリ
    - 単独利用も可能



Teeda

- What is Teedaのまとめ
  - Teedaは3構成
    - JSF実装部分
    - 拡張Extention部分
    - Ajax部分

- JSF and Teeda Core

Teeda

- JSFとは
  - JavaServer Facesの頭文字とった略語
    - 「Faces」 → MVCにおける顔(UIを表現)
  - Servlet API は原則意識しない
  - 画面の入力値やイベント処理をManaged Bean(POJO)に指定する
    - 開発やテストがし易いメリット
  - 規格(JSR-127)であり実装は無い
    - 実装は各ベンダや各組織が行う

Teeda

- JSF実装(海外)
  - MyFaces
    - 2009/1/30 Core1.2.6 release
    - Tomahawk等のコンポーネントが充実
    - http://myfaces.apache.org/
    - S2JSF1.0系のJSF実装

Teeda

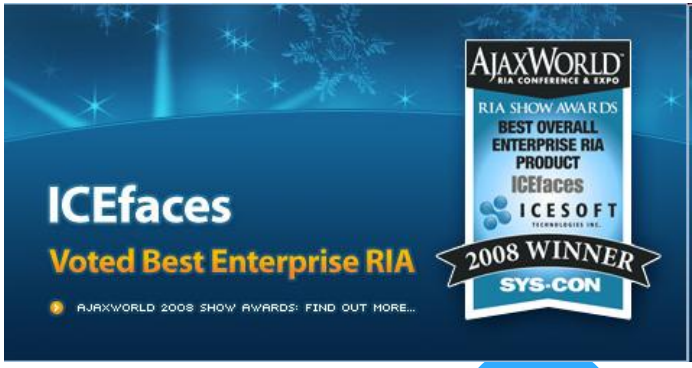
- **JSF実装(海外)**
  - Mojarra(the JSF RI)
    - 2008/12/19 Mojarra 1.2_11 release
    - 2009/2/6 Mojarra 2.0.0 PR2 release
    - やっぱりSun信頼性が高そう
    - https://javaserverfaces.dev.java.net/

Teeda

- **JSF実装(海外)**
  - ICEFaces
    - 2009/12/02 1.8.0-RC1 release

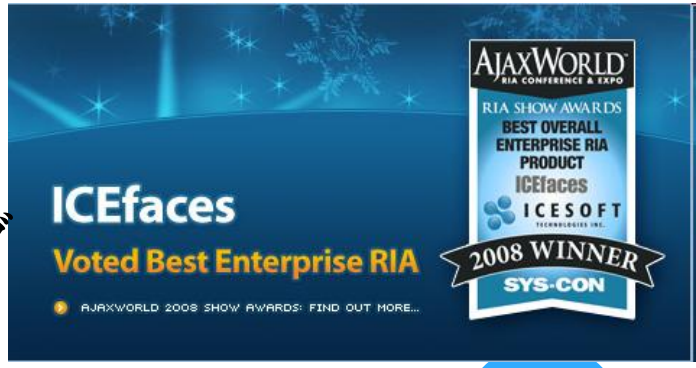
Teeda

- **JSF実装(海外)**
  - ICEFaces
    - 2009/12/02 1.8.0-RC1 release
    - AJAXWorld RIA Conference
      - Best Overall Enterprise RIA Product



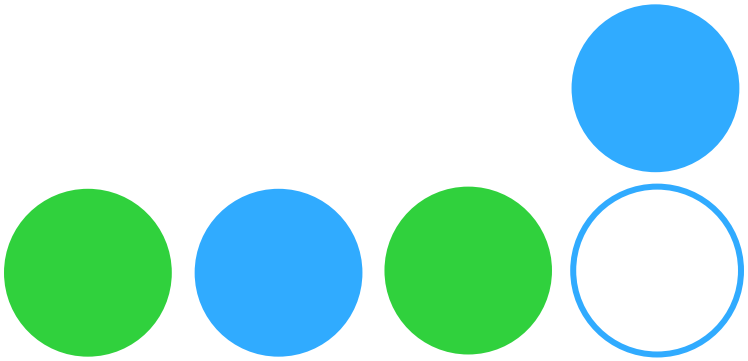
Teeda

- **JSF実装(海外)**
  - ICEFaces
    - 2009/12/02 1.8.0-RC1 release
    - AJAXWorld RIA Conference
      - Best Overall Enterprise RIA Product
    - http://www.icefaces.org/
    - 特徴
      - D2D(Direct-toDOM)レンダリング
      - partialSubmit
    - 生産性が高い(**NetBeans半端ないです**)
    - 実績多数
      - Bank Of America/SIEMENTS/JP MORGAN/RIM
      - ANZ/EDS/EFD/KION/FedEX/SYBASE…他多数



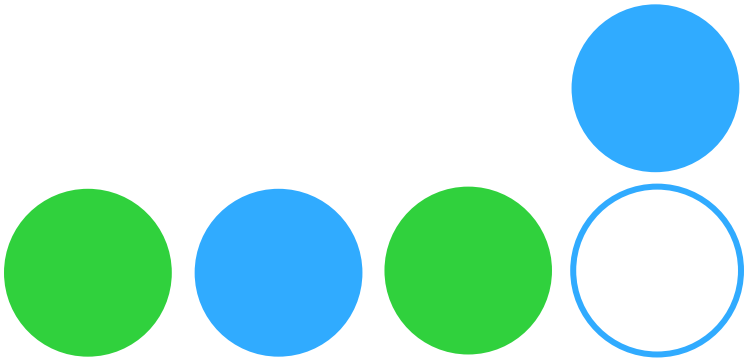
Teeda

- **JSF実装(国内)**
  - Teeda Core
    - 2009/2/13 Teeda 1.0.13-sp6 release
    - 日本初のJSF実装
      - Teeda Coreが出るまでは国内には無かった
    - S2JSF1.1系のJSF実装として採用
      - S2JSF1.0系のJSF実装にはMyFacesを採用
    - Teeda Coreのみでの国内案件実績



Teeda

- JSFとTeeda Coreを
  コードを見ながら紹介

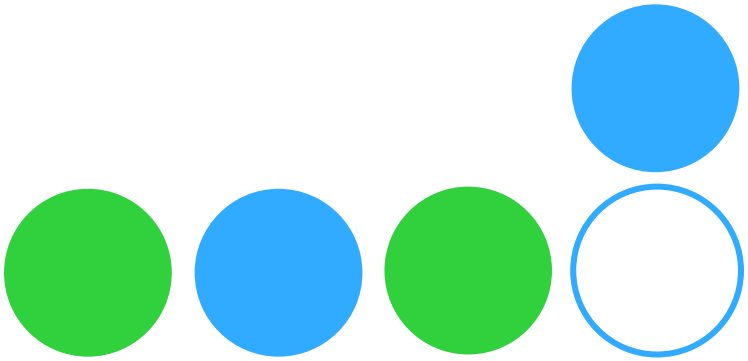


Teeda

コード見て
理解深める
SeasarCon❤

Teeda

- JSFの動きを確認
  - 簡単な足し算アプリ



Teeda

- 足し算サンプル(calculate.jsp)

```
<%@ page contentType="text/html; charset=UTF-8" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<html>
<head><title>calculate</title></head>
<body>
<f:view>
  <h:messages globalOnly="false" showDetail="true"/>
  <h:form>
    <h:inputText value="#{calcPage.arg1}"/> +
    <h:inputText value="#{calcPage.arg2}"/> =
    <h:outputText value="#{calcPage.result}"/><br/>
    <h:commandButton action="#{calcAction.add}" value="calculate"/>
  </h:form>
</f:view>
</body>
</html>
```

Teeda

- 足し算サンプル(faces-config.xml)

```
<managed-bean>
    <managed-bean-name>calcAction</managed-bean-name>
    <managed-bean-class>samples.calculate.CalcAction</managed-bean-class>
    <managed-bean-scope>request</managed-bean-scope>
    <managed-property>
        <property-name>calcPage</property-name>
        <value>#{calcPage}</value>
    </managed-property>
</managed-bean>

<managed-bean>
    <managed-bean-name>calcPage</managed-bean-name>
    <managed-bean-class>samples.calculate.CalcPage</managed-bean-class>
    <managed-bean-scope>request</managed-bean-scope>
</managed-bean>
```

Teeda

- 足し算サンプル(CalcAction.java/CalcPage.java)

```java
package samples.calculate;
public class CalcAction {
    private CalcPage calcPage;
    public CalcPage getCalcPage() {
        return calcPage;
    }
    public void setCalcPage(CalcPage calcPage) {
        this.calcPage = calcPage;
    }
    public String add() {
        this.calcPage.setResult(this.calcPage.getArg1()
                + this.calcPage.getArg2());
        return null;
    }
}
```

```java
package samples.calculate;
public class CalcPage {

    private int arg1;

    private int arg2;

    private int result;

    // setter getter省略
```

Teeda

- まとめ
  - ActionはPOJO
  - ManagedBeanの定義による簡易DI機能
  - スコープ管理はfaces-config.xml

Teeda

- Teeda Coreの動きを確認
  - 簡単な足し算アプリ

Teeda

- JSFの足し算サンプル(teedaCalculate.jsp)

```
<%@ page contentType="text/html; charset=UTF-8" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<html>
<head><title>calculate</title></head>
<body>
<f:view>
  <h:messages globalOnly="false" showDetail="true"/>
  <h:form>
    <h:inputText value="#{teedaCalcPage.arg1}"/> +
    <h:inputText value="#{teedaCalcPage.arg2}"/> =
    <h:outputText value="#{teedaCalcPage.result}"/><br/>
    <h:commandButton action="#{teedaCalcAction.add}" value="calculate"/>
  </h:form>
</f:view>
</body>
</html>
```

Teeda

- 足し算サンプル(sample.dicon)

```
<component
  class="org.seasar.framework.container.autoregister.FileSystemComponentAutoRegister">
 <property name="instanceDef">
  @org.seasar.framework.container.deployer.InstanceDefFactory@REQUEST
 </property>
 <initMethod name="addClassPattern">
  <arg>"samples.teeda"</arg>
  <arg>".*Action|.*Page"</arg>
 </initMethod>
</component>
```

Teeda

- 足し算サンプル(TeedaCalcAction.java/TeedaCalcPage.java)

```java
package samples.teeda.calculate;
public class TeedaCalcAction {
    private TeedaCalcPage teedaCalcPage;
    public TeedaCalcPage getTeedaCalcPage() {
        return teedaCalcPage;
    }
    public void setTeedaCalcPage(TeedaCalcPage teedaCalcPage) {
        this.teedaCalcPage = teedaCalcPage;
    }
    public String add() {
        this.teedaCalcPage.setResult(this.teedaCalcPage.getArg1()
                + this.teedaCalcPage.getArg2());
        return null;
    }
}
```

```java
package samples.teeda.calculate;

public class TeedaCalcPage {

    private int arg1;

    private int arg2;

    private int result;

    // setter getter 省略
```

Teeda

- まとめ
  - ActionはPOJO
  - Seasar2によるDI機能
  - スコープ管理はSeasar2
    - faces-config.xmlに定義しなくても良い

Teeda

- Seasar2で管理するメリット
  - 自動DI機能
    - JSF実装の簡易DIはManagedBeanのみ対象
      - バリデータ・コンバータも対象に出来る
  - AOP機能
    - Seasar2のAOP機能を利用
  - 自動登録機能
    - Seasar2の自動登録機能を利用
      - ManagedBeanが増えても設定ファイル変更無し

Teeda

- Teeda Coreのまとめ
  - 基盤はJSF実装
  - DIとAOPにSeasar2採用

- 重要ポイント
  - JSFのライフサイクル

Teeda

- JSFライフサイクル
  - Restore View
  - Apply Request Value
  - Process Validation
  - Update Model Value
  - Invoke Application
  - Render Response

Teeda

- JSFライフサイクル
  - UIコンポーネント復元
  - コンポーネントへリクエスト値を格納
  - バリデータ・コンバータ適用
  - コンポーネント値をManaged Beanに反映
  - 関連付けられたアクション実行
  - レスポンスのレンダリング

Teeda

- JSFライフサイクル
  - Restore View フェーズ
    - ページのコンポーネントビューであるUIViewRootを作成。JSFからのアクセスの場合にはサーバにUIViewRootが格納されているので復元される。

      初めて画面にアクセスした場合に空のUIViewRootを作成し、残りのフェーズをスキップしRender ResponseフェーズでHttpレスポンスを返します。

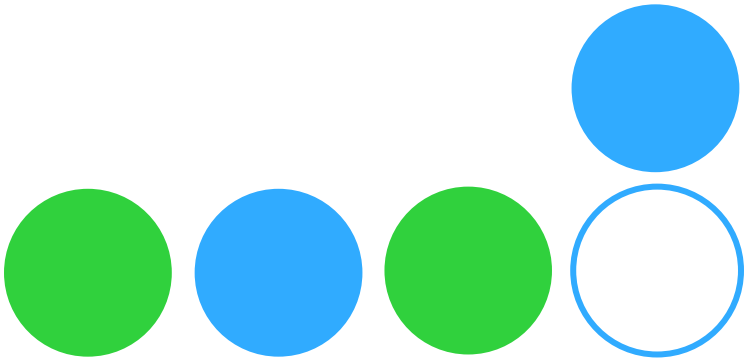
Teeda

- JSFライフサイクル
  - Apply Request Value フェーズ
    - ブラウザから送信された値をRestore Viewフェーズで復元されたコンポーネントに反映します。

Teeda

- JSFライフサイクル
  - Process Validation フェーズ
    - コンポーネントに登録された全てのバリデーションを実行します。

      バリデーションエラー・コンバージョンエラーがあった場合は残りのフェーズをスキップしてRender Responseフェーズを実行します。



Teeda

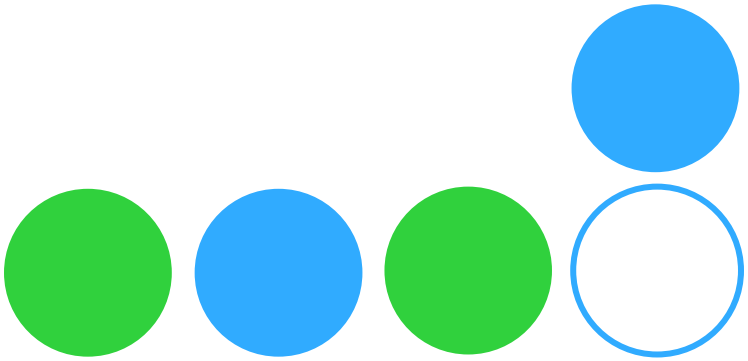
- JSFライフサイクル
  - Update Model Value フェーズ
    - コンポーネントの値をManaged Beanに反映します。

Teeda

- JSFライフサイクル
  - Invoke Application フェーズ
    - ボタンやリンクに関連づけられたアクションイベントを処理します。

Teeda

- JSFライフサイクル
  - Renderer Response フェーズ
    - レスポンスを生成します。レンダリングが終了するとコンポーネントの状態がサーバに保存されます。設定によってはクライアントに保存。



Teeda

- JSFライフサイクルのまとめ
  - ライフサイクルを理解しておくことによってTeedaでエラーが発生した場合に、どのような動きをしているのか把握し易い。Teeda CoreだけでなくTeeda Extentionでも同様です。

Teeda Extension

Teeda

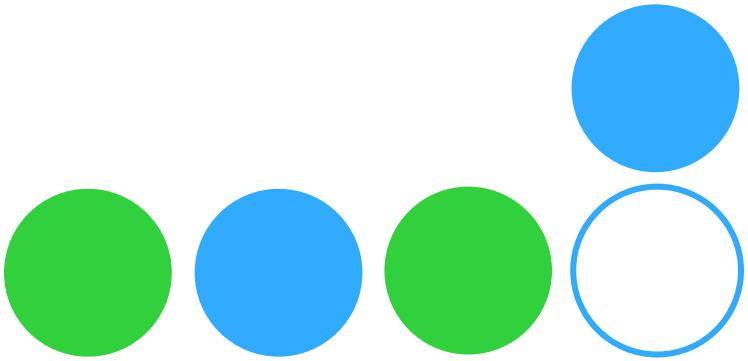
- Teeda Coreをベースに HTMLと規約に基づいた拡張を提供

Teeda

- 何故Extension？

Teeda

- JSFで改善したかった部分
  - viewがJSP
  - faces-config.xmlの肥大化
  - forwardベースのアーキテクチャ
    - URLがずれる



Teeda

- 改善

Teeda

- Teeda Extensionの特徴
  - HTMLテンプレート
  - レスコンフィグレーション
  - PRGパターン
  - SMART Deploy

Teeda

- HTMLテンプレート
  - Page駆動開発
    - HTMLを中心に開発するスタイル
    - HTMLに対応するPageクラスを作成
      - PageクラスはPOJO
      - EclipseプラグインであるDoltengを使えば自動生成可能
    - HTMLのid属性⇔Pageクラスのプロパティ

Teeda

- Pageクラス
  - 命名規則
    - HTML名の先頭を大文字にして Pageというサフィックスをつける

test.html→TestPage.java

Teeda

- test.html

```
<form id="testForm">
<input type="text" id="arg1" />
 <input type="text" id="arg2" />
 <input type="submit" id="doExec" />
</form>
```

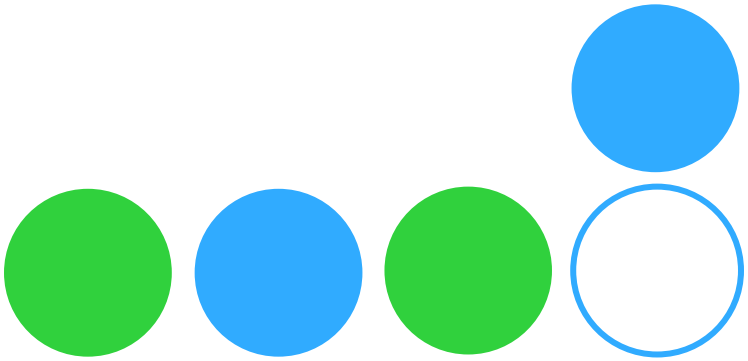
- TestPage.java

```
public class TestPage {

    public int arg1;

    public int arg2;

    public Class<?> doExec() {
        return null;
    }
}
```

復習

Teeda

- Teedaの基盤はJSF
  - HTMLテンプレートからJSPを連想してみる



Teeda

●内部理解の近道

```
<form id="testForm">
<input type="text" id="arg1" />
 <input type="text" id="arg2" />
 <input type="submit" id="doExec" />
</form>
```

```
<%@ page contentType="text/html; charset=UTF-8" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<html>
<head><title>calculate</title></head>
<body>
<f:view>
  <h:messages globalOnly="false" showDetail="true"/>
  <h:form>
    <h:inputText value="#{teedaCalcPage.arg1}"/> +
    <h:inputText value="#{teedaCalcPage.arg2}"/> =
    <h:outputText value="#{teedaCalcPage.result}"/><br/>
    <h:commandButton action="#{teedaCalcAction.add}" value="calculate"/>
  </h:form>
</f:view>
</body>
</html>
```

Teeda

- レスコンフィグレーション
  - Struts
    - struts-config.xml
  - JSF
    - faces-config.xml
  - Teeda
    - 遷移先の設定を書かない

Teeda

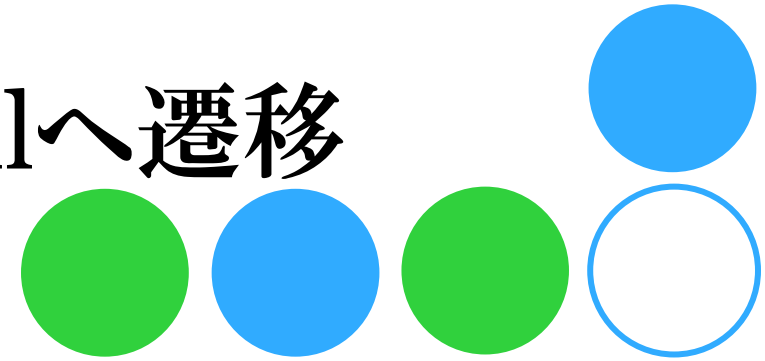
- test.html

```
<a href="link.html">リンク</a>
```

- TestPage.java

```
public Class<?> doExec() {
    return LinkPage.class;
}
```

ともにlink.htmlへ遷移



Teeda

- convention.dicon

```
<?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>"examples.teeda"</arg>
    </initMethod>
  </component>
</components>
```

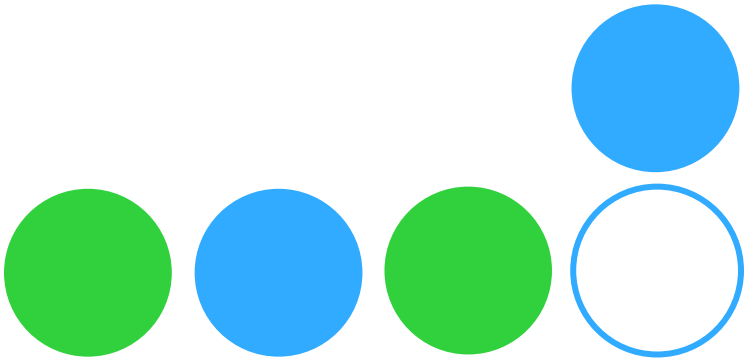
ルートパッケージを登録するだけ

Teeda

- PRGパターン
  - POST
  - REDIRECT
  - GET

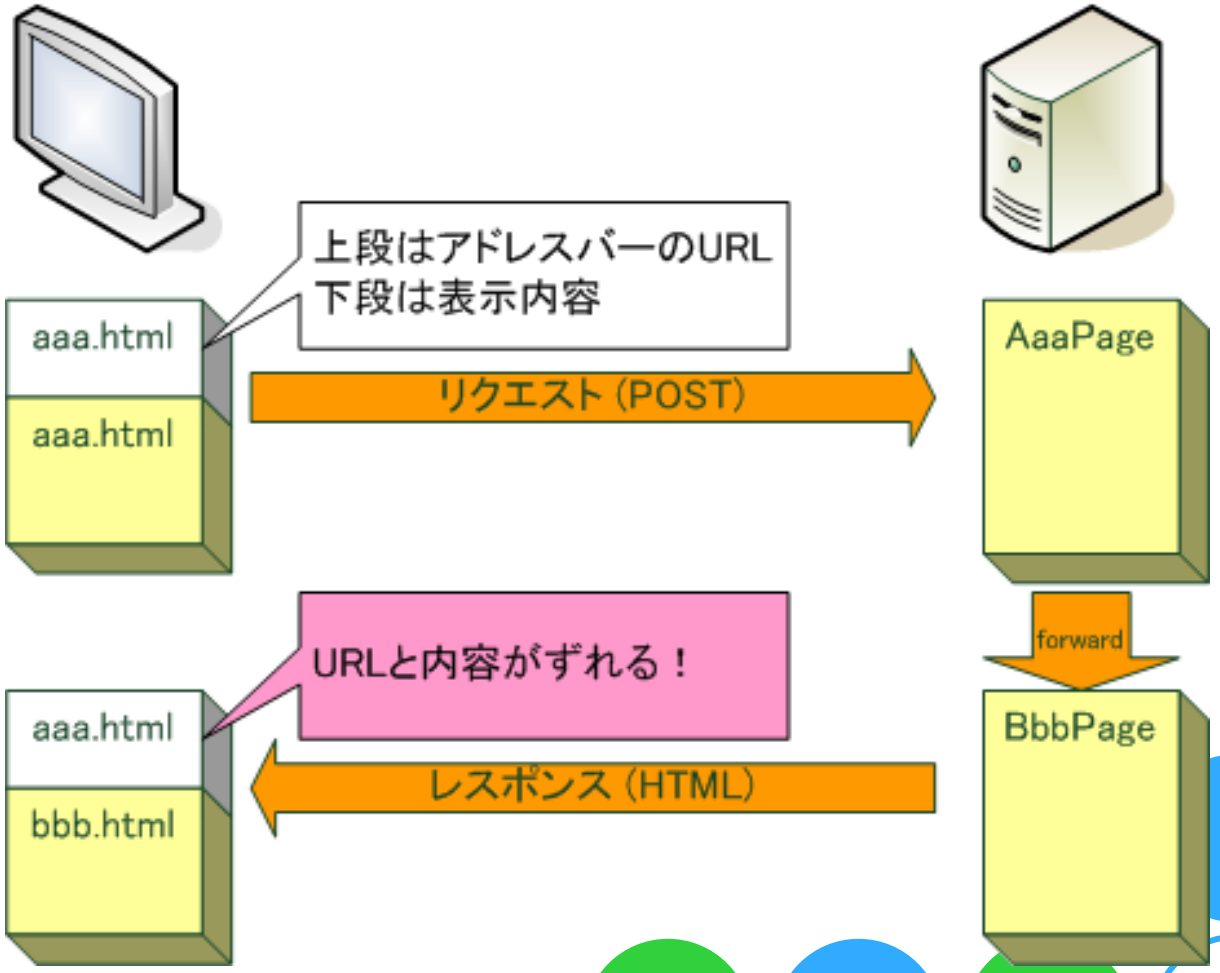
Teeda

- JSFの動作原理
  - forwardベースのアーキテクチャ
    - URLがずれる



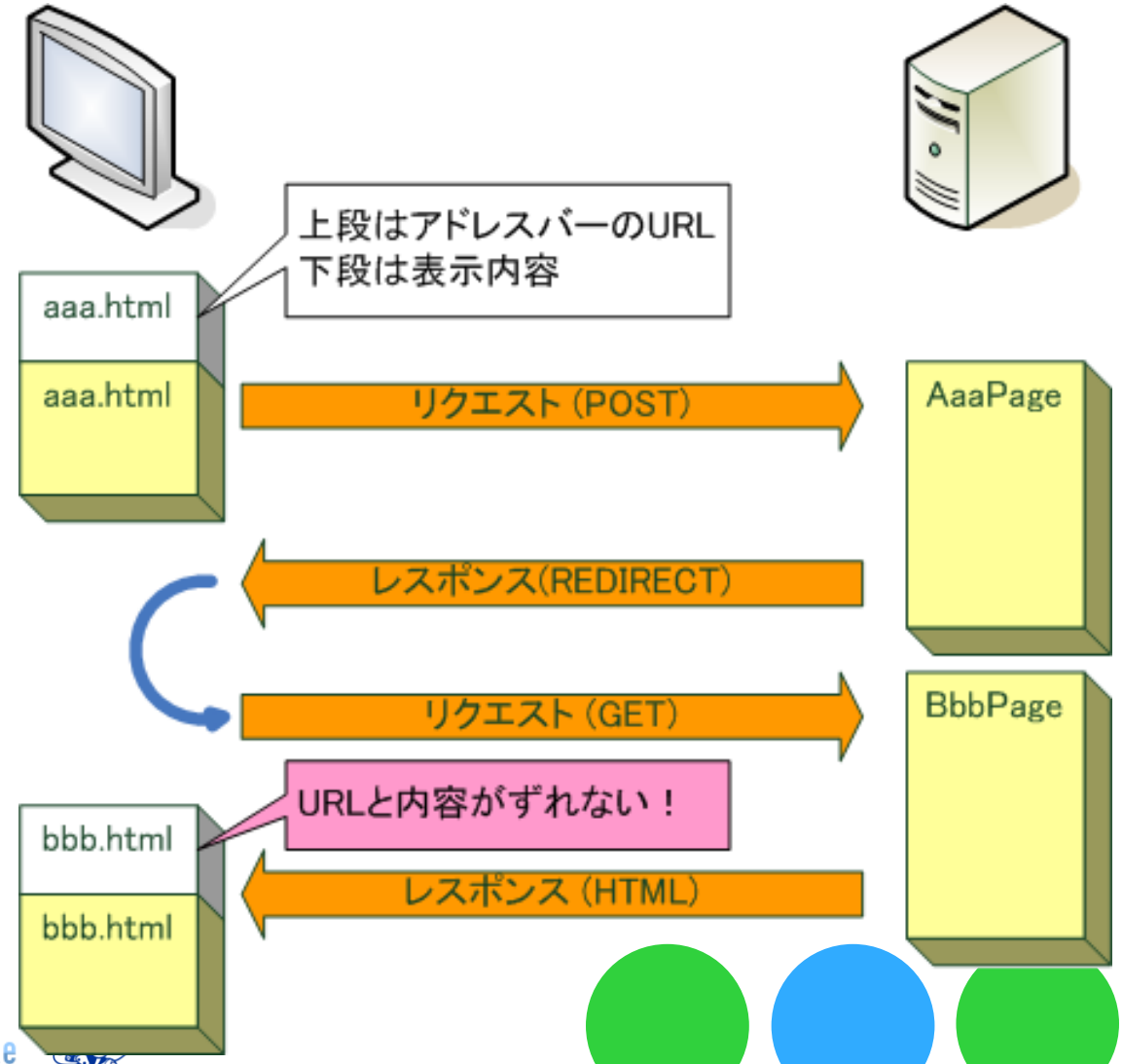
Teeda

- JSFの場合



Teeda

- PRGの場合



Teeda

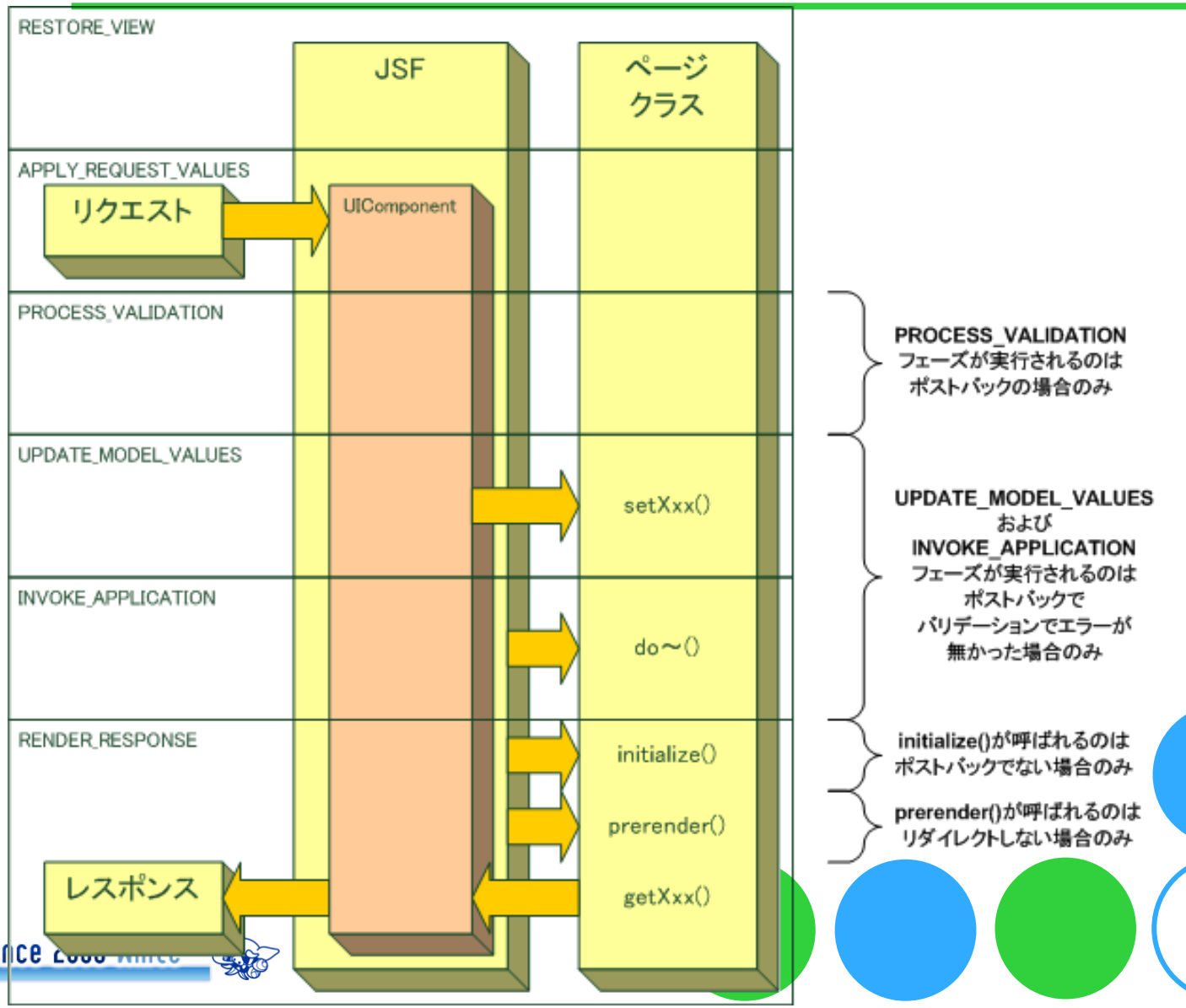
- PRGパターンのメリット
  - ブラウザのリロードによる予期せぬ更新を防げる
  - ブラウザの戻るボタン対応
  - 表示されるURLがずれない

Teeda

- SMART Deploy
  - Seasar2.4の機能
    - Tomcat等のWebコンテナを起動したまま画面の追加変更を行い即座に確認可能

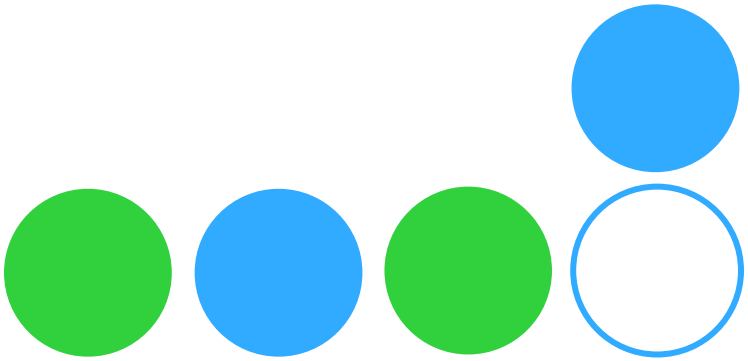
Teeda

- JSFライフサイクルとの関係



Teeda

- Teeda Extensionまとめ
  - JSF単体よりも開発しやすい



- Teeda Ajax

Teeda

- Teeda Ajax
  - 簡単にPageクラスのメソッドをAjaxで呼び出せる
  - JSの関数名により呼び出すコンポーネント解決

サブアプリケーション名＋”_”
Pageクラス名＋”_”
メソッド名

Teeda

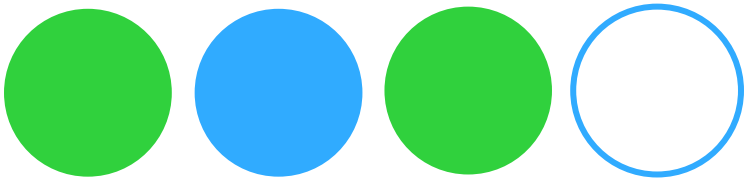
- ajax.html

```
<input type="button" value="test" onclick="test();" />
```

```
function ajax_ajaxPage_ajaxHello(response) {
  alert(response); // HelloAjax!が表示される
}
function test() {
  Kumu.Ajax.executeTeedaAjax(
    ajax_ajaxPage_ajaxHello, []); //callback関数設定
}
```

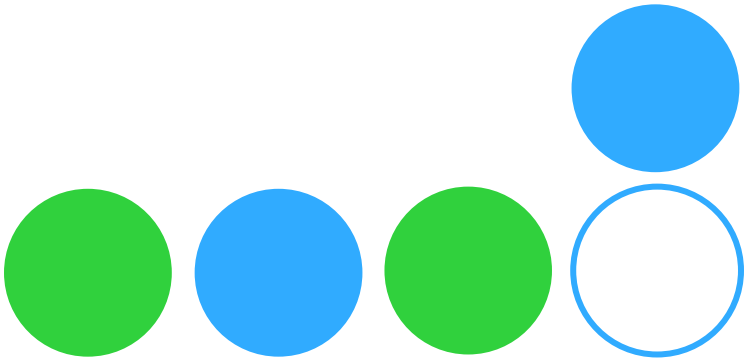
- AjaxPage.java

```
package examples.teeda.web.ajax;

public class AjaxPage {

  public String ajaxHello() {
    return "HelloAjax";
  }
```



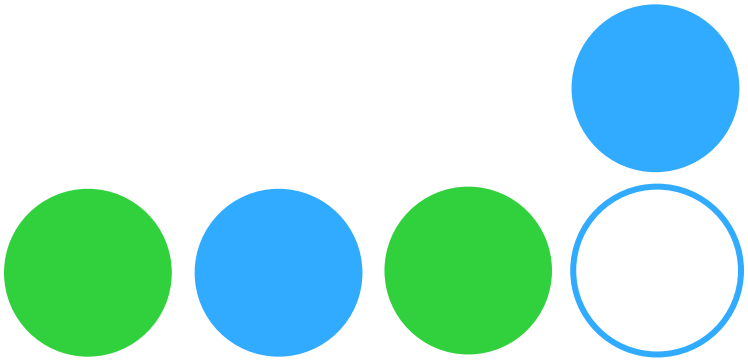
Teeda

- Teeda Ajax補足
  - Kumuを使わなくても可能
    - prototype.js
    - jQuery
    - 独自ライブラリ



Teeda

- Teeda Ajaxまとめ
  - Seasar2配下コンポーネントがJavaScriptから手軽に利用できる



- Teeda Test

Teeda

- Teeda Test
  - TeedaTestCase
    - JUnitの拡張テストケース
    - FacesContextなどのモックを標準で用意
  - TeedaWebTester
    - APサーバにデプロイした状態でテスト
    - 画面からの操作を自動化
    - jWebUnit/HtmlUmitで構築

Teeda

- TeedaTestCase

```
public class DownloadActionTest extends TeedaTestCase {
    DownloadAction action = new DownloadAction();
    action.setResponse(getResponse());
    assertEquals(false, getFacesContext().getResponseComplete());
    action.download();
    assertEquals(true, getFacesContext().getResponseComplete());
}
```

Teeda

●TeedaWebTester

```
public class AddWebTest extends TestCase {
  public void testAdd() throws Exception {
    final TeedaWebTester tester = new TeedaWebTester();
    tester.beginAt("http://localhost:8080/teeda-html-example", "view/add/add.html");
    tester.setTextById("arg1", ""); // @Required
    tester.setTextById("arg2", "aaa"); // converter
    tester.submitById("doCalculate");
    tester.dumpHtml(); // for debug

    tester.assertTextEqualsById("arg1Message", "値を入力してください(INPUT1)");
    tester.assertTextEqualsById("arg2Message", "¥"INPUT2¥" : 値(aaa)は適切な型に変換できません。");
  }
}
```

Teeda Tips

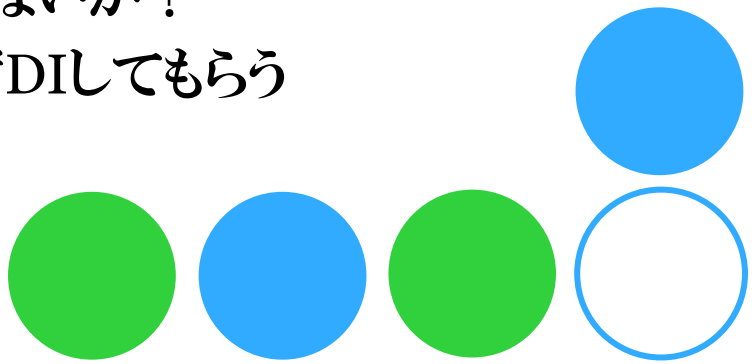
Teeda

- Teeda Tips
  - 動かない場合疑うポイント
    - HTMLに対応するPageクラスが存在するか？
    - Pageクラスが正常にS2に登録されているか？
    - バリデーション・コンバージョンエラーが発生していないか？
    - HTMLのFormのidがxxxFormになっているか？
    - Formがネストしていないか？

Teeda Tips

SMART Deployで嵌るポイント

- COOL Deploy時に動作しない
  - Pageクラスの継承元が抽象クラスになっているか？
- 独自クラスがSMART Deploy対象にならない
  - diconファイルに自分で登録していないか？
- ClassCastExceptionが発生
  - HttpSessionを直接使用していないか？
  - 自分でgetComponentを行わずDIしてもらう



Teeda

- Teeda Tips
  - 拡張ポイント
    - カスタムコンポーネント作成
      - UIコンポーネント, レンダラ, Tagクラス
    - カスタムバリデーション作成
    - カスタムコンバータ作成
    - TeedaAjax拡張
      - prototype.js, jQuery, ExtJS
    - 各ライフサイクルフェーズへの処理の差込
      - カスタムPhaseListener

Teeda

- Teeda振り返り
  - Teeda Core
  - Teeda Extension
  - Teeda Ajax
  - Teeda Test
  - Teeda Tips

- Teedaの今後

Teeda

- Teedaの今後
  - 1.0系
    - 小さな修正をService Packとして提供
  - 1.1系
    - ネストしたプロパティ対応
    - forwardによる画面遷移対応
    - コマンドリンク対応

- Teedaの今後
  - Seasarのメインストリームから外れた？
    - 他プロダクトが注目されている時こそ、使ってくれるユーザへのMLでの対応やBLOGでの対応が必要

Teeda

助け合う
そんな気持ちで
OSS❤

Teeda

- Enjoy Teeda!
- Enjoy OSS!

ご清聴
ありがとう
ございました