

## Web Beans

1. Web Beans とは .....	2
2. 策定状況 .....	3
3. 参照実装のセットアップ .....	4
3.1. JDK のインストール.....	4
3.2. JBoss AS のインストール.....	4
3.3. Apache Ant のインストール.....	4
3.4. Web Beans のインストール.....	5
3.5. JBoss AS の起動.....	5
3.6. サンプルのデプロイ .....	5
4. タイプセーフな依存性注入.....	7
4.1. 使用方法.....	7
5. 統合式言語との統合 .....	8
5.1. 使用方法.....	8
6. スコープとコンテキスト .....	11
6.1. 対話スコープ .....	11
6.2. 使用方法.....	12
7. JSF における EJB コンポーネントの使用 .....	15
7.1. 使用方法.....	15
8. まとめ.....	17
9. 参考資料.....	17

## 1.Web Beans とは

Web Beans は、JCP ( Java Community Process ) において JSR-299 として標準化が進められている Dependency Injection ( 依存性注入 ) およびコンテキストに関する仕様です。JBoss Seam、Google Guice、および、Spring の成果を元に標準化が進められています。スペックリードは、JBoss Seam の創始者である Gavin King 氏が務めています。現在策定中の Java EE 6 に追加される予定で、Java EE 6 に新たに追加される仕様の中で最も注目されている仕様のひとつです。

Web Beans は、下記の機能により Web アプリケーション開発における Web 層とエンタープライズサービスの結合が簡単になります。

- タイプセーフな依存性注入
- Unified EL ( 統合式言語 ) との統合
- 対話スコープ

これらの機能をアノテーションにより使用することができます。

また、Web Beans により EJB コンポーネントを JSF マネージド Bean として使用することができます。

サポート環境は、Java EE 5、Java EE 6、および、Java SE です。なお、名称に“ Web ”と冠していますが、Web アプリケーションだけでなく、トランザクション管理や永続化のような Java EE サービスを提供する embeddable EJB Lite container ( 組み込み可能な EJB 軽量コンテナ ) と連動させることで Java SE 環境でも使用することができます。

改訂公開レビュー草案で「Contexts and Dependency Injection for Java」に変更されましたが、このドキュメントでは引き続き「Web Beans」という名称を使用します。

## 2.策定状況

2009年4月22日現在も策定中です。ステータスとしては公開レビュー投票が終わった段階となります。

参照実装を JBoss Seam の Web サイト (<http://www.seamframework.org/>) からダウンロードすることができます。最新バージョンは、1.0.0.BETA1 です。

なお、この章以降では、実装方法の説明に参照実装に付属するサンプルを使用します。

### 3.参照実装のセットアップ

参照実装を使用するには以下の環境が必要です。

- J2SE 5.0 以降
- JBoss AS 5.0.0.GA 以降
- Apache Ant 1.7.0 以降

以下、Windows 環境におけるセットアップを説明します。

#### 3.1. JDK のインストール

説明を割愛します。

#### 3.2. JBoss AS のインストール

- ① <http://www.jboss.org/jbossas/downloads/> から JBoss AS をダウンロードする。

Java SE 6 用とそれ以外の JDK 用のパッケージが用意されています。たとえば、Java SE 6 用のパッケージは、jboss-5.0.1.GA-jdk6.zip です。

- ② ①で取得した zip ファイルを任意のフォルダに解凍する。  
たとえば、C:¥Applications に解凍すると、フォルダ構成は、C:¥Applications¥jboss-5.0.1.GA となります。

#### 3.3. Apache Ant のインストール

- ① <http://ant.apache.org/> から Apache Ant をダウンロードする。  
たとえば、apache-ant-1.7.1-bin.zip をダウンロードします。

- ② ①で取得した zip ファイルを任意のフォルダに解凍する。  
たとえば、C:¥Applications に解凍すると、フォルダ構成は、C:¥Applications¥apache-ant-1.7.1 となります。

- ③ 環境変数を設定する。  
環境変数「ANT\_HOME」を作成し Ant のインストールフォルダ (C:¥Applications¥apache-ant-1.7.1) を設定します。また、Path に「%ANT\_HOME%¥bin」を追加します。

### 3.4. Web Beans のインストール

- ① <http://www.seamframework.org/> から Web Beans をダウンロードする。  
たとえば、webbeans-1.0.0.BETA1.zip をダウンロードします。
- ② ①で取得した zip ファイルを任意のフォルダに解凍する。  
たとえば、C:¥Applications に解凍すると、フォルダ構成は、C:¥Applications¥webbeans-1.0.0.BETA1 となります。
- ③ JBoss AS のインストールフォルダを設定する。  
jboss-as¥build.properties の jboss.home に JBoss AS のインストールフォルダを設定します。  
たとえば、「jboss.home=C:¥Applications¥jboss-5.0.1.GA」となります。
- ④ JBoss AS に Web Beans をインストールする。  
コマンドプロンプトを実行し、jboss-as フォルダに移動します。そして、「ant update」を実行します。

#### リスト 3-1 Web Beans の JBoss AS へのインストールの実行例

```
> cd C:¥Applications¥webbeans-1.0.0.BETA1¥jboss-as  
> ant update
```

### 3.5. JBoss AS の起動

run.bat を実行する。

たとえば、C:¥Applications¥jboss-5.0.1.GA¥bin¥run.bat を実行します。

### 3.6. サンプルのデプロイ

参照実装に付属するサンプルのビルド方法およびデプロイについて説明します。  
以下、例として conversations サンプルのデプロイ方法を説明します。

コマンドプロンプトを起動する。  
conversations フォルダに移動する。  
ant deploy コマンドを実行する。

#### リスト 3-2 サンプルのビルドの実行例



```
> cd C:\Applications\webbeans-1.0.0.BETA1\examples\conversations
> ant deploy
```

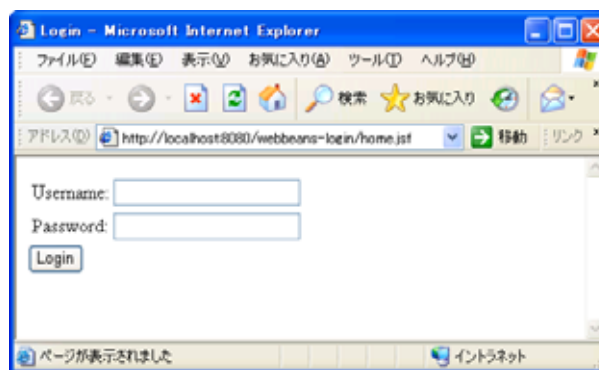
## 4. タイプセーフな依存性注入

Web Beans の依存性注入は、タイプセーフであるため注入するクラスを間違えることがありません。

### 4.1. 使用方法

参照実装の login サンプルで使用方法を説明します。

図 4-1 login サンプル



依存性を注入するには `javax.inject.Current` アノテーションを設定します。

リスト 4-1 `org.jboss.webbeans.examples.login.Login` クラス

```
...
public class Login implements Serializable {

    @Current Credentials credentials;

    ...
}
```

`org.jboss.webbeans.examples.login.Login` クラスの `credentials` フィールドに `javax.inject.Current` アノテーションが設定されています。Login クラスがインスタンス化されたときに自動的に `org.jboss.webbeans.examples.login.Credentials` クラスが注入されます。

## 5. 統合式言語との統合

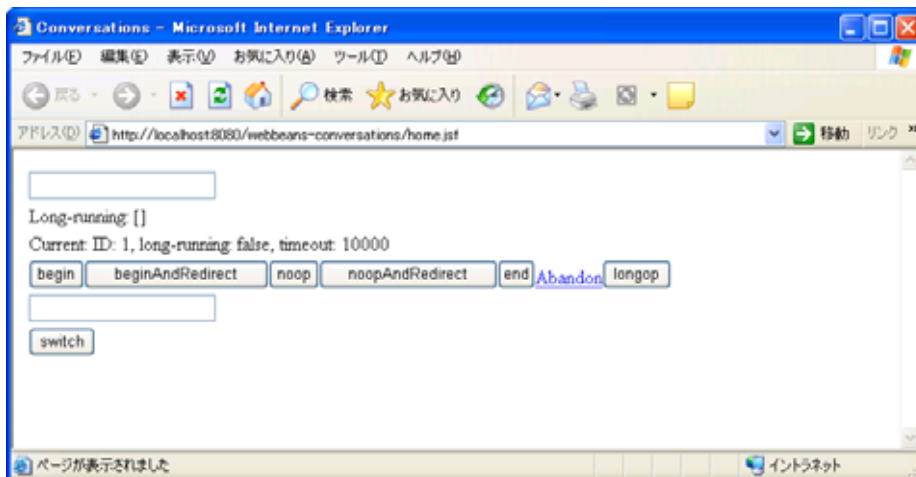
javax.annotation.Named アノテーションを使用し、Bean に名前を付けることによって JSF から Bean を使用することができます。Web Beans を使用しない JSF では faces-config.xml にマネージド Bean を設定する必要がありますが、Web Beans を使用した JSF では Named アノテーションにより faces-config.xml の記述が不要となります。

### 5.1. 使用方法

参照実装の conversations サンプルで使用方法を説明します。

conversations サンプルでは JSF に対応した Facelets というビュー技術を使用しています。

図 5-1 conversations サンプル



Bean に名前を付けるにはクラスに Named アノテーションを設定します。引数の文字列が Bean の名前となります。

リスト 5-1 org.jboss.webbeans.examples.conversations.Conversations

```
...  
@Named("conversations")  
public class Conversations implements Serializable {  
...  
}
```



org.jboss.webbeans.examples.conversations.Conversations クラスは、  
「@Named("conversations")」とアノテーションが設定されているので名前は  
「conversations」となります。

JSF で使用するには、統合式言語で Bean の名前を指定します。

#### リスト 5-2 home.xhtml

```
...  
<h:commandButton action="#{conversations.begin}" value="begin"/>  
...
```

home.xhtml のように Conversations クラスの名前である「conversations」が  
指定されています。

Named アノテーションは、引数を省略することができます。引数を省略した場  
合は、クラス名の最初の文字を小文字に変換した文字列が名前となります。

#### リスト 5-3 org.jboss.webbeans.examples.conversations.Data

```
...  
@Named  
public class Data implements Serializable  
{  
...  
}
```

org.jboss.webbeans.examples.conversations.Data クラスは、引数を省略した  
Named アノテーションが設定されています。したがって、名前は、「Data」の「D」  
を小文字に変換した「data」となります。

#### リスト 5-4 home.xhtml

```
...  
<h:inputText value="#{data.data}" />  
...
```

Named アノテーションを使用することにより faces-config.xml の記述が不要になりました。

#### リスト 5-5 faces-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>  
<faces-config version="1.2" xmlns="http://java.sun.com/xml/ns/javaee"  
  xmlns:xi="http://www.w3.org/2001/XInclude"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  
http://java.sun.com/xml/ns/javaee/web-facesconfig_1_2.xsd">  
  
  <navigation-rule>  
    <from-view-id>/home.xhtml</from-view-id>  
    <navigation-case>  
      <from-outcome>home</from-outcome>  
      <to-view-id>/home.xhtml</to-view-id>  
      <redirect/>  
    </navigation-case>  
  </navigation-rule>  
  
  <application>  
    <view-handler>com.sun.facelets.FaceletViewHandler</view-handler>  
  </application>  
</faces-config>
```

conversations サンプルの faces-config.xml にマネージド Bean の設定の記述がありません。

## 6. スコープとコンテキスト

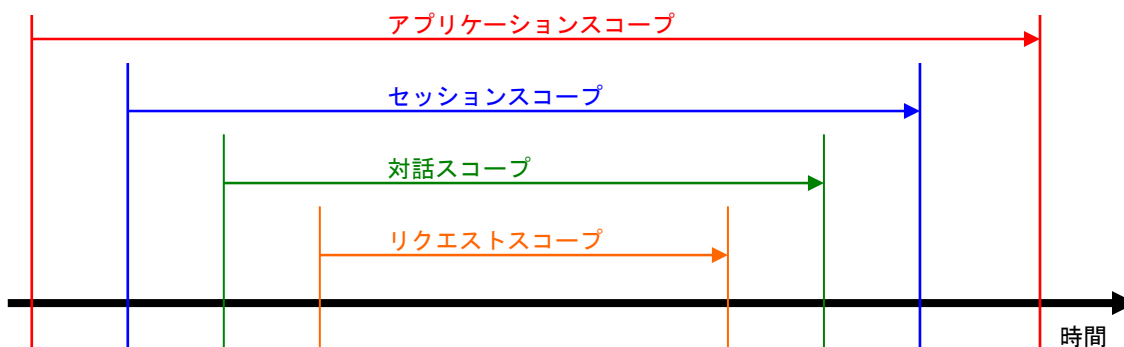
スコープには対応するコンテキストがあります。表 1 は、スコープとコンテキストの対応を示します。

表 6-1 スコープとコンテキストの対応

スコープ	コンテキスト	コンテキスト・クラス
リクエストスコープ	リクエストコンテキスト	HttpServletRequest
セッションスコープ	セッションコンテキスト	HttpSession
アプリケーションスコープ	アプリケーションコンテキスト	ServletContext
対話スコープ	対話コンテキスト	javax.context.Conversation

リクエストスコープ、セッションスコープ、アプリケーションスコープは、サーブレットの仕様で定義されたスコープです。対話スコープは、Web Beans で新しく導入されるスコープです。リクエスト、セッション、アプリケーションの 3 つのスコープは、業務的には何の意味も持ちません。そこで、業務的な意味を持たせたスコープとして対話スコープが導入されます。

図 6-1 スコープの生存時間の概念図



### 6.1. 対話スコープ

マスターデータを登録するために[登録画面] [確認画面] [完了画面]と遷移する機能を開発することになったとします。

Web Beans を使用しない場合では、[登録画面]の入力データをセッションに格納し、[完了画面]を表示する後処理としてセッションから不要なデータを削除する処理を実装することになると思います。もし、不要なデータを削除し忘れるとリソースを無駄遣いすることになります。

Web Beans を使用した場合は、[登録画面] [確認画面] [完了画面]をひとつの対話として管理します。対話が終了すると対話に関連付けられたデータは自動的に破棄されます。セッションから不要なデータを削除する処理を実装する必要がなくなり、リソースを無駄遣いすることもなくなります。

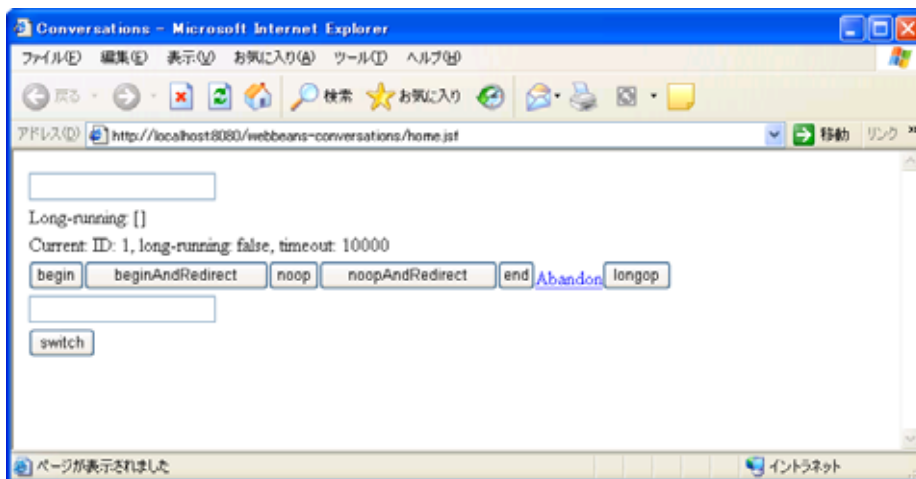
対話は、必ず 1 つ JSF リクエストと関連付けられます。また、「transient」と「long-running」のどちらか一方の状態を持ちます。JSF リクエストを受け取ったときに前のリクエストから引き継がれた対話がある場合は、対話が継続していることを表します。関連付けられた対話がない場合は、「transient」の状態に対話を開始します。そして、レスポンスを返すときに「transient」の状態である場合は、対話が終了し対話コンテキスト、および、対話スコープのオブジェクトが破棄されます。「long-running」の状態である場合は、対話を継続し次のリクエストに引き継がれます。

対話を「long-running」の状態するには、`javax.context.Conversation` クラスの `begin` メソッドを実行します。また、「transient」の状態するには `Conversation` クラスの `end` メソッドを実行します。

## 6.2. 使用方法

参照実装の `conversations` サンプルで使用方法を説明します。

図 6-2 conversations サンプル



リスト 6-1 `org.jboss.webbeans.examples.conversations.Conversations`

```
@SessionScoped
@Named("conversations")
public class Conversations implements Serializable {
    ...

    public String end()
    {
        conversation.end();
        return "home";
    }

    public void begin()
    {
        conversation.begin();
    }

    ...
}
```

org.jboss.webbeans.examples.conversations.Conversations クラスは、javax.context.SessionScoped アノテーションによりセッションスコープのコンポーネントとなります。したがって、Conversations クラスのインスタンスは、セッションが終了するまで破棄されません。

[begin]ボタンを押下すると Conversations クラスの begin メソッドが実行され、Conversation クラスの begin メソッドを実行し、対話を「long-running」の状態にします。

[end]ボタン押下すると Conversations クラスの end メソッドが実行され、Conversation クラスの end メソッドを実行し、対話を「transient」の状態にします。

#### リスト 6-2 org.jboss.webbeans.examples.conversations.Data

```
@ConversationScoped
@Named
public class Data implements Serializable
```

```
{  
...  
}
```

`org.jboss.webbeans.examples.conversations.Data` クラスは、`javax.context.ConversationScoped` アノテーションにより対話スコープのコンポーネントとなります。

[start]ボタンが押下されてから[end]ボタンが押下されるまで、Data クラスのインスタンスは生存します。

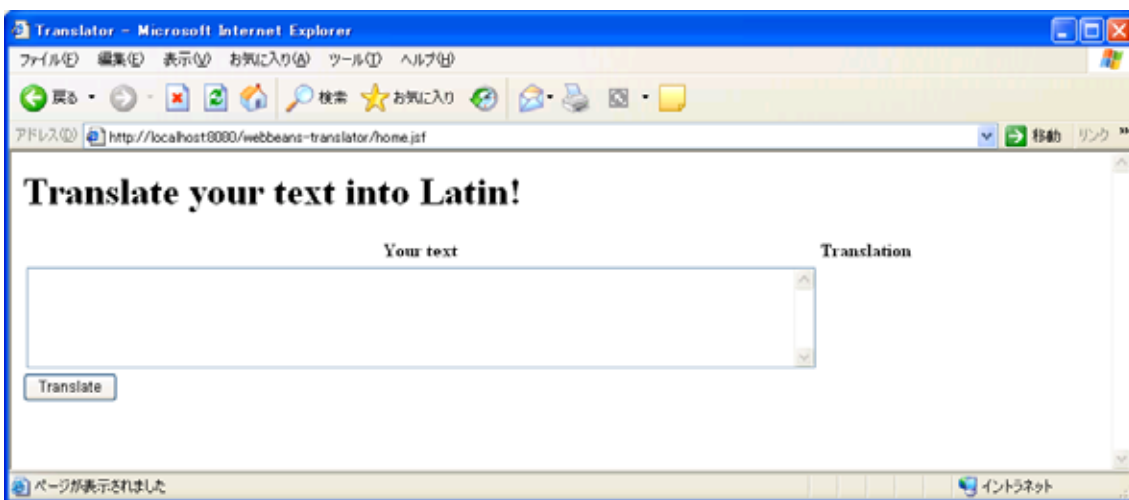
## 7.JSF における EJB コンポーネントの使用

EJB コンポーネントを JSF マネージド Bean として使用することができます。

### 7.1. 使用方法

参照実装の translator サンプルで EJB コンポーネントを JSF マネージド Bean として使用する方法を説明します。

図 7-1 translator サンプル



リスト 7-1 org.jboss.webbeans.examples.translator.TranslatorControllerBean

```
...
@Stateful
@RequestScoped
@Named("translator")
public class TranslatorControllerBean implements TranslatorController
{
...
}
```

org.jboss.webbeans.examples.translator.TranslatorControllerBean クラスは、javax.ejb.Stateful アノテーションが設定されているので、EJB のステートフル・セッション Bean です。

そして、`javax.context.RequestScoped` アノテーションによりリクエストスコープのコンポーネントです。また、`javax.annotation.Named` アノテーションにより「`translator`」という名前が付けられています。

リスト 7-2 home.xhtml

```
...  
<h:commandButton id="button" value="Translate" action="#{translator.translate}"/>  
...
```

home.xhtml にて、統合式言語により「`translator`」という名前を指定して `TranslatorControllerBean` クラスを使用しています。



## 8.まとめ

Web Beans の主な特徴であるタイプセーフな依存性注入、統合式言語、対話スコープ、JSF における EJB コンポーネントを JSF マネージド Bean として使用方法についてサンプルを使用して説明しました。その他の特徴として、イベント通知、インターセプタおよびデコレータ等があります。

仕様が確定するにはもうしばらく掛かりそうですが、参照実装、もしくは、JBoss Seam を使用することによって Web Beans のメリットを体験することができます。

Web Beans によって Java に標準の依存性注入と業務に合った対話コンテキストが導入されることはとても意義があり、Java EE 5 のテーマであった Ease of Development (開発しやすさ)の流れに沿ったものでより開発しやすくなります。現在、多くの Web アプリケーションが採用している構成である「Struts – DI コンテナ (Spring, Seasar2) – ORM (Hibernate、iBatis)」から Web Beans も含まれる Java EE 6 による「JSF – Web Beans – EJB (JPA)」という構成への移行が進むのではないかと思います。

## 9.参考資料

- 『JSR 299: Web Beans』  
<http://jcp.org/en/jsr/detail?id=299>
- 『Web Beans Overview』  
<http://www.seamframework.org/WebBeans>
- 『Web Beans (JSR-299): スペックリード Gavin King 氏との質疑応答』  
<http://www.infoq.com/jp/news/2009/01/webbeansqa>

開発部
-----

大森 洋行
-------