Web Beans: Java Contexts and Dependency Injection

The new standard for dependency injection and contextual state management

Gavin King
JSR-299 specification lead
Red Hat Middleware LLC

Pete Muir
Web Beans (JSR-299 Reference Implementation) lead
Red Hat Middleware LLC
: Nicola Benaglia, Francesco Milesi
Spanish Translation: Gladys Guerrero
Red Hat Middleware LLC
Korean Translation: Eun-Ju Ki,
Red Hat Middleware LLC
Traditional Chinese Translation: Terry Chuang

Red Hat Middleware LLC

Note

JSR-299 has recently changed its name from "Web Beans" to "Java Contexts and Dependency Injection". The reference guide still refers to JSR-299 as "Web Beans" and the JSR-299 Reference Implementation as the "Web Beans RI". Other documentation, blogs, forum posts etc. may use the new nomenclature, including the new name for the JSR-299 Reference Implementation - "Web Beans".

You'll also find that some of the more recent functionality to be specified is missing (such as producer fields, realization, asynchronous events, XML mapping of EE resources).

I. ì»¨í##ì#¤í#¸ ê°#ì²´ ì#¬ì#©

Web Beans (JSR-299) ### ####### ## ## #### ## Java EE ### ## ### ### #####. Web Beans# JavaBeans # Enterprise Java Bean# #### ### Java ## ## ### ## ### ## ## # ## ## ### ######. #### Java EE ##### ### ## ####, Web Beans# ### ## #### #####:

- #### ### ##### #### ## ## ## ### ## ## ##
- ### ## *(dependency injection)*### ## ### (typesafe) ##
- ### ## ### ## ####
- ##### *(decorator)* ## ### ### ##### ## ##### ##### ##### ## ### ####, #### ## ### ## ## #### ##

API #### #####:

- ### ### ## ### #####?
- ### ## ###### ## ### ## # ####?
- ## ########?
- ### ## # ####?
- ## ##### #### ###?
- ## #### ## ## ## ### ### #### ###?
- ### ## ## ### ####, ### ### ## ### ### # # ####?
- ### ## ## ## #### ### ### ### # ####?

Web Bean# #### ## Web Beans# ## # ### ## #####. ## #### ## Web Bean ##### ## ### ##, #### ## ##, ### ## ### ### ### ####. # ### #### Web Bean# ### ##, ## ##, ### ## ### ####### ### ### ## ## ### ## ### # ####.

###, #### # ###### ### ### ### #### #### *(loose-coupling)* ### #####:

- ### ### ### ###### ### #### ## #####
- ##### #### #### ### ### ### ## #####
- ###### #### ### #### ###.

Web Beans# ### ## ### ## ### *(typesafe)* #### #####. Web Beans# ### ### ### ## ### #### ## ### ### #### #### ####. XML# #### ##### ## #### ####. ## Web Beans#

(binding annotations)### ### ### ## Java ## #### ## ## ### ### ### #### Web Beans # Web Beans# ###, ####, ##### # ### #### ## ####.

Web Beans #### ##### Java EE ### ## ### ## ## ### ### #####:

- ## JavaBeans
- ## EJB
- ## Servlets

Web Beans# ## ### #### ## Java EE #### ## ### ###### ## ### ## ### ## ### Web Beans# #### Web Beans #### ### ### ## ### Web Bean# ## #####.

Web Beans# Seam, Guice, Spring# #### ### ## Java ###### ## ### #### Web Beans# ##### ### ## ### ### #### ####: Seam ## # ## ###, ## # ## ###, Spring ## # XML-##, Guice ## # # # ######-###### ## ##

Web Beans# Java EE # ## ### EJB Lite# ### # ## Java SE ### ### JCP #####.

Web Beans#

Web Bean ### ### ######? ## ## ### Web Beans ### ## #### # ## ### ### #### ## # # ####! # ## ## ### #### ## #### ## #### Web Beans# ## ## # #### #### ####. ## #### #### #### Web Beans# #### ### ## # # ####.

1.1. # ## Web Bean

Java #### Web Bean## ### ## JavaBean# #####. ## ### ## EJB 3-### ## ## Web Bean###. ## ## ##### JavaBeans # EJB# Web Beans ### ## ### ### #### ### ## # ###, Web Beans### ## ### #### ## # ## ## #### # #### # ## ## Web Beans## ##, Web Beans XML ## ### ## ##, #### # ##### ##

Java #### ## ### #####. # ## #### #### ## #### ## #####:

```
public class SentenceParser {
    public List<String
> parse(String text) { ... }
}
```

########:

```
@Stateless
public class SentenceTranslator implements Translator {
    public String translate(String sentence) { ... }
}
```

`Translator`# ## ########:

```
@Local
public interface Translator {
    public String translate(String sentence);
}
```

####, ### ### #### Web Bean# #####:

```
public class TextTranslator {

    private SentenceParser sentenceParser;
```

```
    private Translator sentenceTranslator;

    @Initializer
    TextTranslator(SentenceParser sentenceParser, Translator sentenceTranslator) {
        this.sentenceParser = sentenceParser;
        this.sentenceTranslator = sentenceTranslator;
    }

    public String translate(String text) {
        StringBuilder sb = new StringBuilder();
        for (String sentence: sentenceParser.parse(text)) {
            sb.append(sentenceTranslator.translate(sentence));
        }
        return sb.toString();
    }

}
```

Web Bean, Servlet ## EJB# #### `TextTranslator` ##### ## # ####:

```
@Initializer
public setTextTranslator(TextTranslator textTranslator) {
    this.textTranslator = textTranslator;
}
```

Web Beans ### ### ## #### ##### ### # ####:

```
TextTranslator tt = manager.getInstanceByType(TextTranslator.class);
```

#######: `TextTranslator`# ## ## ## #### ## ####! ### ### Web Bean###? #### `@Initializer`## ###### #### ## ## ## ## ## ## #### ## ## #### Web Bean# # # ####.

######, `@Initializer` ###### ### ### #### #### ## ##### ## ####! `@Initializer`# ### ## Web Bean ### ### # ###, Web Bean# ###### # Web Bean ##### ### ## Web Bean ### ### ## #####. Web Bean #### ## ## Web Beans# ### ## ### ## ### #### ###.

Web Bean #### ### ## ### ##### ### Web Bean# ##### ## ##### ###. ### ## ### `Translator` ### ## ## # `SentenceTranslator` EJB# #### ## ## # Web Bean #### `UnsatisfiedDependencyException`# ### ###. ## ### `Translator`# ### # ## ##, Web Bean #### `AmbiguousDependencyException`# ### ###.

1.2. Web Bean# #####?

Web Bean# #####?

Web Bean# #### ### ## ## ###### ######. Web Bean# Java #### ## ##### ## Unified EL# ## ### # ####. Web Bean# #### #### #### # ####. Web Beans ##### #### Web Bean #### ## #### #####. #### Web Beans# ## ## *(stateful)* # #### #####. Web Bean ## ### ## Web Bean #### ## #####.

###. "#####"### ### #### #####? Web Beans# ## ## ## ###, #### #### ## # ######## #####. ### ## ## ## ## (#: ### ## #) ## singleton #### ## (#: ervlets ## singleton #)## ### Web Bean# ## ###### ## ### ## Web Bean# ## ###. #####-### ### ###### #### ## Web Bean# ##### ##### #####.

###, ### ## ### ### ### ## ## ## # ## ###, ###### ##### ## ### ##### ## ## #### #### ####. ##, Web Bean# ### ## ### #####:

- ### Web Bean ##### ## ##
- ## ###### ## Web Bean ##### ### ### ###

Web Beans ######## ## ### #### ##, Web Bean ### ### ## ##### ## # ####. ### ##### ##### ##### (#: ## ### ##) ## ### ## #### ### # ### (#: Web Bean# ## ### ##) ## ## ## ## #### # #### (Web Bean# ###### ### ##).

(#: ## ## Web Bean)# ### Web Bean# ##### ## ###. ### ## ###### #### ###### ## ##### ## ###.

Web Beans# ##### ##### ## ####! ###### #### Web Bean# ## ## ## ### #####, ## ## ### ##### # ### ####. Web Beans# #### #### ###### Web Bean ### ### ### ## ## ### #####. Web Beans# ### ## ### #### #### ####:

- # ### ## API# ## ## #####
- ## ### #### ## #####

Web Bean# ### API# #### ## Web Bean ### ### ### ## ## ## ## (## ##)# ## ## Web Bean## ### # ####. ##, *4.2#. "## ## "*## #### # ### Web Beans# ### Web Bean ### ###### ## ### ### #####.

Web Bean ###### Web Beans# # # ### #####. Servlets ## Message-Driven Beans# ## # ## ## ### #### ### ## # ### ### ## Web Beans# ### ### # ####.

spec# ###:

> Web Bean# ### ## ### #### ####:

- (## ## ##) API ## ##
- (## ## ##) ### ##### ## ##
- ##
- ## ##
- Web Bean ## (## ##)
- #### ### ## ##
- Web Bean ##

Web Bean ##### ### #### #####.

1.2.1. API ##, ### ## # ###

Web Beans# ### ### ## ## Web Beans# ### #####. ### ### ### Web Bean# ##### "##"# #####. #### ### ## ### #####:

- API ##
- ### ## ##

API# ### ## ### ## ########. (Web Bean# EJB ## ## ##, API ### `@Local` ######## ## bean-class ## #####.) ### ### #####-### #### #### ### API# ## ### ## #####.

`@BindingType`## ###### ### ## ###### ## #####. ## ##, ### ## ## ### `PaymentProcessor` API ## # `@CreditCard` ### ### ####:

```
@CreditCard PaymentProcessor paymentProcessor
```

##, ### ### ##`@Current`# #####.

##, Web Bean #### ### ##### (API# #### ## ### ### ##) Web Bean# #### ## #####.

Web Bean# `@CreditCard` ### ### ## `PaymentProcessor` API ### #####. ### ## ## ## ### ### # ####:

```
@CreditCard
public class CreditCardPaymentProcessor
   implements PaymentProcessor { ... }
```

Web Bean# ### ## ### ### #### ## ##, ## ### ### ### `@Current`## ### ### ### ####.

Web Beans# ##### #### ### ##### #### ## ### ##### ## ### Web Bean# ## ## ##### ### ### ## #### ## ####. ## ### ### *4#. ### ##* ## ######.

1.2.2. ##

Web Beans# #### ###. ## ### ## ## ##### #### #######, ## ## `@Mock`, `@Staging`, `@AustralianTaxLaw`# ####. ##### ##### # Web Beans# ###### #####. ## ### ## #### #### ## Web Beans ### ### ## #### ###.

Web Beans# #### ## ## `@Production`# #####, ### ## ## ### #### ### ### ####. ### ## Web Bean ### `@Production` ## ### ####.

####, `SentenceTranslator` Web Bean# "## ##"# ##### ###:

```
@Mock
public class MockSentenceTranslator implements Translator {
   public String translate(String sentence) {
      return "Lorem ipsum dolor sit amet";
   }
}
```

`MockSentenceTranslator` # ## ## Web Bean ##### `@Mock`# #### ## #### ## ### #### `@Mock` ## ### #### # ####.

4.2#. "## ## "## ### #### ### ### ## ### ## #####.

1.2.3.

Web Bean# ## ### ##### #### #####. Web Beans #### ### ## ####, ### ### #####. ###, ## ## ### #### #### ###, Web Bean #### #####. ### ##### ### ## #####.

##, # ####### ## ## Web Beans# ## # ####:

```
@SessionScoped
public class ShoppingCart { ... }
```

Web Bean# ##### ### #### ##### ## ###### #### ## ### ## #####.

#####, Web Beans# *dependent pseudo-scope*## ## ### ####. ### ### ## Web Beans# ###### ### ### ### #### ##### ### ##### ######.

5#. ## # #### ## ### ## ## #### ######.

1.2.4. Web Bean ## # Unified EL

Web Bean# ### ## # ###, Unified EL ##### ### # ####. Web Bean# ### #### ## ####:

```
@SessionScoped @Named("cart")
public class ShoppingCart { ... }
```

JSF ## JSP ##### ## Web Bean# ### # ####:

```
<h:dataTable value="#{cart.lineItems}" var="item">
   ....
</h:dataTable
>
```

Web Bean #### ### ##### ## ## ## ## #####:

```
@SessionScoped @Named
public class ShoppingCart { ... }
```

##, ## #### `shoppingCart`# ### # ### ### #### # #### #### ### ##.

1.2.5. #### ###

Web Beans# EJB beans # # ### ## Java #### ## EJB 3# ## ### #### ### #####. ## ###, Web Beans# ### ####, EJB beans # ## ## Web Beans# ### ## ### #####.

`@Interceptors` ###### ### ## #### #### ## ### # ####:

```
@SessionScoped
@Interceptors(TransactionInterceptor.class)
public class ShoppingCart { ... }
```

###, #### ### ### ## #### #### ## ## ## ## ####:

```
@SessionScoped @Transactional
public class ShoppingCart { ... }
```

7#. #### # 8#. ##### ## Web Beans #### # ###### ## ########.

1.3. ## ### ### Web Beans# # # ####?

JavaBeans, EJB # ## ## Java #### Web Beans# # # ### ########. ###, ### ## ### ### Web Beans###?

1.3.1. ## Web Beans

Web Beans ### ### ## ## ### Java #### ## Web Bean### ####:

- EJB, Servlet ## JPA #### ## EE #### ## ## ##### ## ##,
- #-### ## #### ## ##,
- ## #### ### ## ##,
- ## ## ## #### ## `@Initializer` ###### #### ## ##,

###, ## ## JavaBean# ## Web Bean# ###.

Web Bean# ## ### ## ##### #### ## ###### API ### ## Web Bean###. ### # ###### API ### ###.

1.3.2. ###### Web Beans

EJB 3 ### ## # singleton beans# EJB 3 ###### Web Beans## ####. Message driven beans# Web Beans# #### # ## ### ##### #### ## ## # ### ## ### ## # ##### ### #### Web Beans ### ### ## # ####.

Web Bean# ## ## ##### # ## ######## ###### Web Bean# API #####. EJB bean# bean ### ## ### ## ## ## bean ### # ## ## ## #### API #####.

`@Destructor`# ###### ## ### #### ###. Web Bean #### ## ### #### ## ## ## # ##### #### ## ### ### #####. ### ### ###### Web Bean# ### ##### ###.

```
@Stateful @SessionScoped
public class ShoppingCart {

   ...

   @Remove
   public void destroy() {}

}
```

Web Bean ## ###### Web Bean# #### ###? ### ## EJB# ## #### ## ###### #### ### # ## #### ###:

- ###-## #### ## # ##
- ### ##

- ## ## ## # # #### ## ### # ### ## # # #### ##
- ## # # ### ##
- ### # ### ##

Web Bean# #### ###. ### ## ## #### ## ##, ## Web Bean# # ### ####.

Web Beans (## ## ###### ## Web Bean ##)# ## ##### ## #####. ###, EJB 3.1# ## #### ### ### ## #####. #### ## # ###### ## Web Beans# EJB### ###.

Web Beans# ### #### ### #### ### EJB `@Stateless`/`@Stateful`/`@Singleton` ### ## ### ## #### ## ##### ### ### ## ### ### # #### ## ### ## #### ### ###.

#####, ###-## #### ##, ###-## ##, ###, ## ## ## ### ### #### #####.

Web Bean## ####, `@Stateless`, `@Stateful`, `@Singleton` ###### #### EJB# #####.

1.3.3. ###

Web Bean# ##### #### ## Web Bean #### ## #### #####. ### ### ##### Web Bean #### #### ## ## ####### #### ### ### #### ###. #:

```
@ApplicationScoped
public class Generator {

   private Random random = new Random( System.currentTimeMillis() );

   @Produces @Random int next() {
      return random.nextInt(100);
   }

}
```

Web Bean# ## #####.

```
@Random int randomNumber
```

##/#### ## ##### API ### ### #####. ## ### #### ## ## ###### API ### ###.

#####:

```
@Produces @RequestScoped Connection connect(User user) {
```

```
    return createConnection( user.getId(), user.getPassword() );
}
```

####:

```
void close(@Disposes Connection connection) {
    connection.close();
}
```

Web Bean #### ## #### #####.

6#. ### ## ## ### ### ## ## #### ####.

1.3.4. JMS

#####, JMS # ## ### Web Bean# # # ####. Web Beans# # # ### #### #### ## ### ## ### JMS ### #### ### ### ##### ##### ######. *13.4#. “JMS ##### ”*## JMS ###### ### ###.

JSF # ######

#####. JSF# #### ####### ### ###/###### ### ####. ##, #### ### ### ## # ### #### ## Web Bean# #####:

```
@Named @RequestScoped
public class Credentials {

    private String username;
    private String password;

    public String getUsername() { return username; }
    public void setUsername(String username) { this.username = username; }

    public String getPassword() { return password; }
    public void setPassword(String password) { this.password = password; }

}
```

Web Bean# ### ## JSF #### ### ##### ######:

```
<h:form>
    <h:panelGrid columns="2" rendered="#{!login.loggedIn}">
        <h:outputLabel for="username"
>Username:</h:outputLabel>
        <h:inputText id="username" value="#{credentials.username}"/>
        <h:outputLabel for="password"
>Password:</h:outputLabel>
        <h:inputText id="password" value="#{credentials.password}"/>
    </h:panelGrid>
    <h:commandButton value="Login" action="#{login.login}" rendered="#{!login.loggedIn}"/>
    <h:commandButton value="Logout" acion="#{login.logout}" rendered="#{login.loggedIn}"/>
</h:form
>
```

Web Bean ## ### ## ## ### #### ## Web Beans# `User` #### ######:

```
@SessionScoped @Named
public class Login {
```

```
    @Current Credentials credentials;
    @PersistenceContext EntityManager userDatabase;

    private User user;

    public void login() {

        List<User
> results = userDatabase.createQuery(
            "select u from User u where u.username=:username and u.password=:password")
            .setParameter("username", credentials.getUsername())
            .setParameter("password", credentials.getPassword())
            .getResultList();

        if ( !results.isEmpty() ) {
            user = results.get(0);
        }

    }

    public void logout() {
        user = null;
    }

    public boolean isLoggedIn() {
       return user!=null;
    }

    @Produces @LoggedIn User getCurrentUser() {
        return user;
    }

}
```

`@LoggedIn`# ### ########:

```
@Retention(RUNTIME)
@Target({TYPE, METHOD, FIELD})
@BindingType
public @interface LoggedIn {}
```

Web Bean# ## #### ## ### # ####:

```
public class DocumentEditor {

    @Current Document document;
    @LoggedIn User currentUser;
    @PersistenceContext EntityManager docDatabase;

    public void save() {
        document.setCreatedBy(currentUser);
        docDatabase.persist(document);
    }

}
```

Web Bean ###### ### ## ## ### ##### ###. ## ###### Web Beans ### ### ## ### ########.

Web Beans ##

Web Beans ## ## (Web Beans Reference Implementation)# *Seam* #### [http://seamframework.org/WebBeans]## #### ####. #### ### [http://seamframework.org/Download]## ## ### #### ###### # ####.

Web Beans RI# ## ### # ## ###### ## ## ####: war ### ## beans ## ## ## `webbeans-numberguess`# ## ear ### ###### beans# ## ## `webbeans-translator`# ####. ## #####, ## ### #####:

- ## Web Beans RI ###
- JBoss AS 5.0.0.GA, #
- Ant 1.7.0.

Currently, the Web Beans RI only runs on JBoss Application Server 5. You'll need to download JBoss AS 5.0.1.GA from *jboss.org* [http://www.jboss.org/jbossas/downloads/], and unzip it. For example:

```
$ cd /Applications
$ unzip ~/jboss-5.0.1.GA.zip
```

seamframework.org [http://seamframework.org/Download]## Web Beans RI# ###### ## ### ###. #:

```
$ cd ~/
$ unzip ~/webbeans-$VERSION.zip
```

####, JBoss# ### ## ## Web Beans## ### ###. `jboss-as/build.properties`# #### `jboss.home` ### #####. #:

```
jboss.home=/Applications/jboss-5.0.1.GA
```

##

A new deployer, `webbeans.deployer` is added to JBoss AS. This adds supports for Web Bean archives to JBoss AS, and allows the Web Beans RI to query the EJB3 container and discover which EJBs are installed in your application.

Web Beans is bundled with JBoss AS 5.1 and above.

To install Web Beans, you'll need Ant 1.7.0 installed, and the `ANT_HOME` environment variable set. For example:

```
$ unzip apache-ant-1.7.0.zip
$ export ANT_HOME=~/apache-ant-1.7.0
```

Then, you can install the update. The update script will use Maven to download Web Beans automatically.

```
$ cd webbeans-$VERSION/jboss-as
$ ant update
```

##, ### ### ### ### #####!

##

#####. ### ### ## ## ####:

- `ant restart` - exploded #### ### ######
- `ant explode` - ## ### ## #### ## exploded ### ########
- `ant deploy` - ### jar #### ### ######
- `ant undeploy` - #### ### ######
- `ant clean` - ### ####

numberguess ### #####:

```
$ cd examples/numberguess
ant deploy
```

Start JBoss AS:

```
$ /Application/jboss-5.0.0.GA/bin/run.sh
```

##

If you use Windows, use the `run.bat`script.

####### #### *http://localhost:8080/webbeans-numberguess*## ##### ####!

Web Beans RI## #### #### #### # ## ### ####. numberguess ### war ### ## beans ## #####; ## ### ear ### ###### beans# #### EJB ### ##### ####. ## #### ### #####:

```
$ cd examples/translator
ant deploy
```

####### #### *http://localhost:8080/webbeans-translator*# #####!

3.1. numberguess

numberguess ######## 1## 100### ### #### ## ### 10 ## #### ###. ## ### # ### ## ### ## ## #### #### ###.

numberguess ### ## Web Beans, ## ##, Facelet JSF #### #### ###, war# ##### #### ## #### #####.

`WEB-INF/`# ### ###, ## ## ### ## `WebContent`# #### ####. ##, `faces-config.xml`# ####, Facelets# ### JSF# #### ###:

```
<?xml version='1.0' encoding='UTF-8'?>
<faces-config version="1.2"
              xmlns="http://java.sun.com/xml/ns/javaee"
              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">

   <application>
      <view-handler
>com.sun.facelets.FaceletViewHandler</view-handler>
   </application>

</faces-config
>
```

`web-beans.xml` ### ##, ## Web Beans ######## ######## #####.

#####, `web.xml` ### ####:

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app version="2.5"
    xmlns="http://java.sun.com/xml/ns/javaee"
    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-app_2_5.xsd">

  <display-name
>Web Beans Numbergues example</display-name>

  <!-- JSF -->
                                                    (1)
  <servlet>
    <servlet-name
>Faces Servlet</servlet-name>
    <servlet-class
>javax.faces.webapp.FacesServlet</servlet-class>
                                                    (2)
    <load-on-startup
>1</load-on-startup>
  </servlet>

  <servlet-mapping>
                                                    (3)
    <servlet-name
>Faces Servlet</servlet-name>
    <url-pattern
>*.jsf</url-pattern>
  </servlet-mapping>
                                                    (4)
  <context-param>
    <param-name
>javax.faces.DEFAULT_SUFFIX</param-name>
                                                    (5)
    <param-value
>.xhtml</param-value>
  </context-param>

  <session-config>
    <session-timeout
>10</session-timeout>
  </session-config>
```

```
</web-app
>
```

1. Enable and load the JSF servlet
2. Configure requests to `.jsf` pages to be handled by JSF
3. Tell JSF that we will be giving our source files (facelets) an extension of `.jsf`
4. Configure a session timeout of 10 minutes

##

Whilst this demo is a JSF demo, you can use the Web Beans RI with any Servlet based web framework.

Let's take a look at the Facelet view:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
   xmlns:ui="http://java.sun.com/jsf/facelets"
   xmlns:h="http://java.sun.com/jsf/html"
   xmlns:f="http://java.sun.com/jsf/core"
   xmlns:s="http://jboss.com/products/seam/taglib">


 <ui:composition template="template.xhtml">                    (1)
   <ui:define name="content">
     <h1
>Guess a number...</h1>

     <h:form id="NumberGuessMain">                              (2)
       <div style="color: red">
         <h:messages id="messages" globalOnly="false"/>
           <h:outputText id="Higher" value="Higher!" rendered="#{game.number gt game.guess
 and game.guess ne 0}"/>
         <h:outputText id="Lower" value="Lower!" rendered="#{game.number lt game.guess and
 game.guess ne 0}"/>
       </div>


       <div>                                                    (3)
         I'm thinking of a number between #{game.smallest} and #{game.biggest}.
         You have #{game.remainingGuesses} guesses.
```

```
        </div>

        <div>
          Your guess:                                                    4
          <h:inputText id="inputGuess"
                       value="#{game.guess}"
                       required="true"
                       size="3"
                       disabled="#{game.number eq game.guess}">          5
            <f:validateLongRange maximum="#{game.biggest}"
                                 minimum="#{game.smallest}"/>
          </h:inputText>                                                 6
          <h:commandButton id="GuessButton"
                           value="Guess"
                           action="#{game.check}"
                           disabled="#{game.number eq game.guess}"/>
        </div>
        <div>
                    <h:commandButton id="RestartButton" value="Reset" action="#{game.reset}"
 immediate="true" />
        </div>
      </h:form>
    </ui:define>
  </ui:composition>
</html
>
```

1. Facelets is a templating language for JSF, here we are wrapping our page in a template which defines the header.
2. There are a number of messages which can be sent to the user, "Higher!", "Lower!" and "Correct!"
3. As the user guesses, the range of numbers they can guess gets smaller - this sentance changes to make sure they know what range to guess in.
4. This input field is bound to a Web Bean, using the value expression.
5. A range validator is used to make sure the user doesn't accidentally input a number outside of the range in which they can guess - if the validator wasn't here, the user might use up a guess on an out of range number.
6. And, of course, there must be a way for the user to send their guess to the server. Here we bind to an action method on the Web Bean.

4 ## #### #####, # ## # ## ### #####. ##, ### ### #### ## #### `@Random` ### ### ####:

```
@Target( { TYPE, METHOD, PARAMETER, FIELD })
@Retention(RUNTIME)
@Documented
@BindingType
public @interface Random {}
```

`@MaxNumber` ### ### ####:

```
@Target( { TYPE, METHOD, PARAMETER, FIELD })
@Retention(RUNTIME)
@Documented
@BindingType
public @interface MaxNumber {}
```

`Generator` #### ### ### ## ### ### #### # ### ####. ## ## ### ### ### ### ## ### #####:

```
@ApplicationScoped
public class Generator {

  private java.util.Random random = new java.util.Random( System.currentTimeMillis() );

  private int maxNumber = 100;

  java.util.Random getRandom()
  {
    return random;
  }

  @Produces @Random int next() {
    return getRandom().nextInt(maxNumber);
  }

  @Produces @MaxNumber int getMaxNumber()
  {
    return maxNumber;
  }

}
```

`Generator`# ###### ### ## ### #### ####; ### ## ## ## ### ## ## ###.

####### ## ### Web Bean# `Game` ## #####.

`@Named` ###### ##### JSF ##### EL# ## bean# ### # ### #### ####. ##### ### ### #### ### ### ### ### ######. ##, ## ###### ## #### ##### ##### ###, `FacesMessage`# #### #####.

```
package org.jboss.webbeans.examples.numberguess;


import javax.annotation.PostConstruct;
import javax.faces.application.FacesMessage;
import javax.faces.context.FacesContext;
import javax.webbeans.AnnotationLiteral;
import javax.webbeans.Current;
import javax.webbeans.Initializer;
import javax.webbeans.Named;
import javax.webbeans.SessionScoped;
import javax.webbeans.manager.Manager;

@Named
@SessionScoped
public class Game
{
   private int number;

   private int guess;
   private int smallest;
   private int biggest;
   private int remainingGuesses;

   @Current Manager manager;

   public Game()
   {
   }

   @Initializer
   Game(@MaxNumber int maxNumber)
   {
      this.biggest = maxNumber;
   }

   public int getNumber()
   {
      return number;
```

```
  }

  public int getGuess()
  {
    return guess;
  }

  public void setGuess(int guess)
  {
    this.guess = guess;
  }

  public int getSmallest()
  {
    return smallest;
  }

  public int getBiggest()
  {
    return biggest;
  }

  public int getRemainingGuesses()
  {
    return remainingGuesses;
  }

  public String check()
  {
    if (guess
>number)
    {
      biggest = guess - 1;
    }
    if (guess<number)
    {
      smallest = guess + 1;
    }
    if (guess == number)
    {
      FacesContext.getCurrentInstance().addMessage(null, new FacesMessage("Correct!"));
    }
    remainingGuesses--;
    return null;
```

```
  }

  @PostConstruct
  public void reset()
  {
    this.smallest = 0;
    this.guess = 0;
    this.remainingGuesses = 10;
    this.number = manager.getInstanceByType(Integer.class, new AnnotationLiteral<Random
>(){});
  }

}
```

3.2. ##

#####.

ear# ####, EJB# #### ###. ###, ## ### numberguess ### ## #####.

##

EJB 3.1 # Jave EE 6# EJB# war# ##### ##, ### ### ## #### ### ###!

##, `webbeans-translator-ear` ### ## ear aggregator# #####. Maven# #### `application.xml`# #####:

```
<plugin>
  <groupId
>org.apache.maven.plugins</groupId>
  <artifactId
>maven-ear-plugin</artifactId>
  <configuration>
    <modules>
      <webModule>
        <groupId
>org.jboss.webbeans.examples.translator</groupId>
        <artifactId
>webbeans-translator-war</artifactId>
        <contextRoot
>/webbeans-translator</contextRoot>
      </webModule>
```

```
    </modules>
  </configuration>
</plugin
>
```

url (*http://localhost:8080/webbeans-translator*)# #### #### ### #####.

##

Maven# #### ### ### ### ##, `META-INF/application.xml`# #####:

```
<?xml version="1.0" encoding="UTF-8"?>
<application xmlns="http://java.sun.com/xml/ns/javaee"
             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/application_5.xsd"
             version="5">
  <display-name
>webbeans-translator-ear</display-name>
  <description
>Ear Example for the reference implementation of JSR 299: Web Beans</
description>

  <module>
    <web>
      <web-uri
>webbeans-translator.war</web-uri>
      <context-root
>/webbeans-translator</context-root>
    </web>
  </module>
  <module>
    <ejb
>webbeans-translator.jar</ejb>
  </module>
</application
>
```

Next, lets look at the war. Just as in the numberguess example, we have a `faces-config.xml` (to enable Facelets) and a `web.xml` (to enable JSF) in `WebContent/WEB-INF`.

facelet# #####. numberguess ##### ## ##### #### #### (#### ## ## #### ###):

```
<h:form id="NumberGuessMain">

  <table>
    <tr align="center" style="font-weight: bold" >
      <td>
        Your text
      </td>
      <td>
        Translation
      </td>
    </tr>
    <tr>
      <td>
      <h:inputTextarea id="text" value="#{translator.text}" required="true" rows="5" cols="80" />
      </td>
      <td>
        <h:outputText value="#{translator.translatedText}" />
      </td>
    </tr>
  </table>
  <div>
    <h:commandButton id="button" value="Translate" action="#{translator.translate}"/>
  </div>

</h:form
>
```

###, #### ### ### #### ## ### #### ###.

#####, ejb ### `webbeans-translator-ejb`# #####. `src/main/resources/META-INF`## Web Beans# ## ## ### ##### #### ## #### # `web-beans.xml`# ####.

#######! ###### # ## ## beans# `SentenceParser` # `TextTranslator`# # ## ###### beans# `TranslatorControllerBean` # `SentenceTranslator`# ####. ## Web Bean# ## ## ## ## ## ## ####, ## ### ### ### ######.

`SentenceParser` # `TextTranslator`# beans# ####, `TextTranslator`# ### #### #####:

```
public class TextTranslator {
  private SentenceParser sentenceParser;
  private Translator sentenceTranslator;

  @Initializer
  TextTranslator(SentenceParser sentenceParser, Translator sentenceTranslator)
```

```
  {
    this.sentenceParser = sentenceParser;
    this.sentenceTranslator = sentenceTranslator;
```

`TextTranslator`# ### ##### (## #### ###### ##), ### #### ### ## ###, # #### ####!

#####, UI ## #### ##, ## #### ## #### #### #### ## ####. ## ## ##, ##, ## ## ##### #### #### ####.

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

  @Current TextTranslator translator;
```

bean# #### ## ## ### ## getter # setter# ####.

####, ## ### ## ### ###:

```
  @Remove
  public void remove()
  {

  }
```

Web Beans #### bean# ### # ## ### #### ###; ### ## ## #### #### ###.

That concludes our short tour of the Web Beans examples. For more on Web Beans , or to help out, please visit *http://www.seamframework.org/WebBeans/Development*.

##, ### ## ##, ## ##, ## ### ### ## ## #### ### #####.

##

Web Beans# ### ### ## # ## ## ##### #####:

##:

```
public class Checkout {

   private final ShoppingCart cart;

   @Initializer
   public Checkout(ShoppingCart cart) {
      this.cart = cart;
   }

}
```

##:

```
public class Checkout {

   private ShoppingCart cart;

   @Initializer
   void setShoppingCart(ShoppingCart cart) {
      this.cart = cart;
   }

}
```

##:

```
public class Checkout {

   private @Current ShoppingCart cart;

}
```

Web Bean ##### ## ######## ## ### ### #####.

- ##, Web Bean #### Web Bean #### #### Web Bean ##### #####.
- ####, Web Bean #### Web Bean# ### ## ### ## ######.
- # #, Web Bean #### Web Bean# ## ### ### #####.
- ##### Web Bean# `@PostConstruct` ### #####.

EJB# Web Bean #### ## EJB ##### ## ###### ## ### #### ### EJB beans# #### ####.

`@Current`# ##### ## #### ## ## # ### ##### ##### ###### ### ####. ### ### #### ### ### ### ##### ##, ## ### ### ### ### ## #### ###. ### ### ### #### ## ## ##, ## #### ## ###.

#####:

```
@Produces Checkout createCheckout(ShoppingCart cart) {
   return new Checkout(cart);
}
```

(*9#. ###* ## #####), ## ## # ### ## ### ## ## ### #####.

Web Beans ### ## ### ### ######## ### ####, Web Bean #### ## ### #### ## Web Bean# ### # ## ### ###. ### ##### #### #### ####, ## ## ####, ## ## ######. ## ### #### ### ##### ####, #### #### ## ### ##### #### Web Bean# #### #### ## ##, `UnsatisfiedDependencyException` ## `AmbiguousDependencyException`# ### ###.

Web Beans# #### ### API ### #### ## ####:

- ## ###### ### ###### #### ### ## ### #### ###,
- ###### #### ###### #### ## ## ### ### ## ###### ## #### ## ### ### ## #### ###.
- ### API ### ###### #### ## ## ## ## ### #### ### ## ### API ### ###### ###.

Web Beans #### ### Web Bean# ### ## #### ### ## #####.

4.1. ###

API ### #### ## ### Web Bean# ## ##, ## ### ### ###### #### ## Web Bean# ##### # ## ### # ####. ## ##, `PaymentProcessor`# # ## ### ## # ####:

```
@PayByCheque
public class ChequePaymentProcessor implements PaymentProcessor {
   public void process(Payment payment) { ... }
}
```

```
@PayByCreditCard
public class CreditCardPaymentProcessor implements PaymentProcessor {
    public void process(Payment payment) { ... }
}
```

`@PayByCheque` # `@PayByCreditCard`# ### ########:

```
@Retention(RUNTIME)
@Target({TYPE, METHOD, FIELD, PARAMETER})
@BindingType
public @interface PayByCheque {}
```

```
@Retention(RUNTIME)
@Target({TYPE, METHOD, FIELD, PARAMETER})
@BindingType
public @interface PayByCreditCard {}
```

Web Bean #### ### ###### #### ### ## Web Bean# ##### ## ## #####.

##:

```
@PayByCheque PaymentProcessor chequePaymentProcessor;
@PayByCreditCard PaymentProcessor creditCardPaymentProcessor;
```

##:

```
@Initializer
public void setPaymentProcessors(@PayByCheque PaymentProcessor
 chequePaymentProcessor,
                                 @PayByCreditCard PaymentProcessor creditCardPaymentProcessor) {
  this.chequePaymentProcessor = chequePaymentProcessor;
  this.creditCardPaymentProcessor = creditCardPaymentProcessor;
}
```

##:

```
@Initializer
```

```
public Checkout(@PayByCheque PaymentProcessor chequePaymentProcessor,
                @PayByCreditCard PaymentProcessor creditCardPaymentProcessor) {
  this.chequePaymentProcessor = chequePaymentProcessor;
  this.creditCardPaymentProcessor = creditCardPaymentProcessor;
}
```

4.1.1. ### ###

####:

```
@Retention(RUNTIME)
@Target({TYPE, METHOD, FIELD, PARAMETER})
@BindingType
public @interface PayBy {
   PaymentType value();
}
```

##, ## ## #####:

```
@PayBy(CHEQUE) PaymentProcessor chequePaymentProcessor;
@PayBy(CREDIT_CARD) PaymentProcessor creditCardPaymentProcessor;
```

Web Bean ##### `@NonBinding` ### ######## ### ##### ## ### ##### ## # ####.

4.1.2. ### ######

####:

```
@Asynchronous @PayByCheque PaymentProcessor paymentProcessor
```

##, ## ### ###### ## Web Bean# ### # ####.

4.1.3. ### ##### # ###

####:

```
@Produces
@Asynchronous @PayByCheque
PaymentProcessor createAsyncPaymentProcessor(@PayByCheque PaymentProcessor
 processor) {
```

```
    return new AsynchronousPaymentProcessor(processor);
}
```

4.1.4. ### ###

Web Beans# #### ### ### #### ## Web Bean## ## ## ### ## ### ### ### `@Current` ### ### #####.

`@Current`# #### ## ### #### ### # ## ####:

- ### ### ### ## ## #### ## #### ## ####
- ### ### ### ### ## ### ### ## Web Bean##

4.2. ##

Web Beans# ## ### ####. ### ## ### #### ## ### #### ##### ## Web Beans ### #####.

##, `@Mock`## ## ### ### # ###, ## #### #### ### ## ### ### # ##### ## Web Beans# #####:

```
@Retention(RUNTIME)
 @Target({TYPE, METHOD})
 @DeploymentType
 public @interface Mock {}
```

Web Bean# ### #####:

```
public class ExternalPaymentProcessor {

    public void process(Payment p) {
        ...
    }

}
```

Web Bean# ## ### #### #### ## ####, ## ### ## ## `@Production`# ## ###.

######. ### ## ### ### # ####:

```
@Mock
public class MockPaymentProcessor implements PaymentProcessor {
```

```
    @Override
    public void process(Payment p) {
        p.setSuccessful(true);
    }

}
```

Web Bean #### ## ### ## ### ### ## ### #####?

4.2.1. ## ##

Web Beans# # ## ### ## ### #####: `@Production` # `@Standard`. #### #### ##### ### ## ### ## Web Beans ## ######. `web-beans.xml`## ### #### ## #### #### ## ## ### ### # ####.

####, ### #### ### #, ## `@Mock` ### ##### ####:

```
<WebBeans>
    <Deploy>
        <Standard/>
        <Production/>
        <test:Mock/>
    </Deploy>
</WebBeans
>
```

Web Bean #### ### `@Production`, `@Standard`, `@Mock` ###### ## Web Beans# ## # #### ###.

`@Standard` ## ### Web Beans ### ## ### ## Web Beans### #####. ### Web Beans## ## ### # ### ## ##### # ####.

`@Production` ## ### ## ### #### ## Web Beans# ## ### ## #### ##### # ####.

4.2.2. ## ## ##

######, Web Bean #### ## ## # `ExternalPaymentProcessor` ## `MockPaymentProcessor` # # ### ## ### ###### ## ### ### ####. #### ### ## ### #### # ### ### ## #####:

```
@Current PaymentProcessor paymentProcessor
```

`PaymentProcessor` ### #### # ## Web Beans# ####. ## #### ### ###### ### ##-##### ####, #### ## ## ### ###### ### # ###, #### ### ### # ## ### ####!

####. ## ### ## ### `web-beans.xml`# #### ### ## #####. ### `@Mock`# `@Production` ## ### ##### ## ## ## ### ####.

Web Bean# ## ### ## ### ## (API ## # ### #####)# ### # ### #### ### # ##, ## Web Beans# ### ## ### #####. ### Web Bean# ## # ## ## ## ### ## ##, ### ## ## ## ### Web Bean# #####. ###, ## ### ## (##### ### ##)## ### # Web Bean #### `MockPaymentProcessor`# #### ###.

####. ### "##"# ##### classpath# ## #### ### ### #####, ### #### ###, #######, ## ### ## XML ## ### #### ### ###. Web Beans# XML# ## Web Bean ## # ### ##### ### ### ### ## ## #### ##, ## ### ## Web Beans ### ## XML ### ##### ###. ###, ### #### #### ## ## ##### Web Bean# ### ## ## ### # ####.

4.2.3. ## ##

#####, #### ## ### ####:

- ### # `@Mock` # `@Staging` ## ##
- ## ### Web Beans # `@AustralianTaxLaw`
- Web Beans## #### ### ##### # `@SeamFramework`, `@Guice`
- Web Beans ### ## ### ## Web Beans # `@Standard`

####...

4.3. ###### ###

API ### #### ## Web Beans# ## ### ### #, Web Bean #### ### ### Web Bean# #### ## ## ## ### ### ##### #### ###.

`UnsatisfiedDependencyException` ## `AmbiguousDependencyException`# ##### ####.

`UnsatisfiedDependencyException`# #####, API ### #### Web Bean# #### ## ### ### ### ### # ## API ### ## #### ## Web Bean# ## ### ##### ### ### ####.

`AmbiguousDependencyException`# #####, # ## API ## ## #### #### ## ### ### ##### ## # ### ## ### #### Web Bean #### ## #### #### ## ## ## ## ### ### # ####. `AmbiguousDependencyException`# # ## Web Beans# ### ### #### ### ## ### ## ## ### ### # ####.

Web Beans## ### ### ### # ##### # ### # ## # ####.

4.4. #####

Web Bean ###### Web Bean ##### #### ### #### ####.

Web Bean### ## ### #### ## ###### ### #### Web Bean# ### ###. ###### ### #### Web Bean# ## ## ## #### #####. ###, ### ### ## ### #### Web Bean# ## ##### #### ###!

Web Bean### ## ### #### ## ## ### #### Web Bean# ### ###. ### ## ##### #### ## ##### #### ## ### #### #####. ###, ###### ### #### Web Bean ##### ## ### #### Web Bean# ## ###### ####!

###, Web Bean# ### ## `@Dependent`# ## ## #, Web Bean #### ### ### ## ### ## ### Web Bean## ## ### ###. ### ##### #### ## ### ## Web Bean ##### ## ##### ### ###### #### # ### ####. ## ##### #### Web Beans# ### ## Web Beans# ##### #### ## ### #### ### ## ##### ## ##### ##### ###.

Java ### ### ##, ## Java ### Web Bean #### ## #### # ####. ###, ## ## ### #### # ## ## Web Bean #### `UnproxyableDependencyException`# ### ###.

Java ### Web Bean #### ## #### # ####:

- `final`### #### ### ## `final` ### ## ###,
- ## ## ## ### #### ## ## ###,
- ## # ## ##

`UnproxyableDependencyException`# #### ####. #### ### #### ## ### ## #### ####, ###### #####, ### Web Bean# ### `@Dependent`# #####.

4.5. ##### lookup# ## Web Bean

######## ### ## `Manager` ###### ##### ### # ####:

```
@Current Manager manager;
```

`Manager` ### ######## Web Bean ##### ### ## ## ### #####.

```
PaymentProcessor p = manager.getInstanceByType(PaymentProcessor.class);
```

`AnnotationLiteral` ### #### ######## ### # ###, ### ### Java## ##### ### ######## #####.

```
PaymentProcessor p = manager.getInstanceByType(PaymentProcessor.class,
                                               new AnnotationLiteral<CreditCard
>(){});
```

##, `AnnotationLiteral`# ### ###### ### # #### # ## named ## #### #### ###:

```
abstract class CreditCardBinding
   extends AnnotationLiteral<CreditCard
>
   implements CreditCard {}
```

```
PaymentProcessor p = manager.getInstanceByType(PaymentProcessor.class,
                                               new CreditCardBinding() {
                                                  public void value() { return paymentType; }
                                               } );
```

4.6. ## ## ##, @Resource, @EJB # @PersistenceContext

Enterprise Web Beans# EJB ### ## ### ## ## ## ### #####: @PostConstruct, @PreDestroy, @PrePassivate # @PostActivate.

Web Beans# @PostConstruct # @PreDestroy ## #### #####.

Web Beans# Java EE ### ##, EJB # JPA ### #### ## @Resource, @EJB # @PersistenceContext# ### ## #####. ## Web Beans# @PersistenceContext(type=EXTENDED)# ### #### ####.

@PostConstruct ### ## #### ### ## #####.

4.7. InjectionPoint

@Dependent ### ## Web Beans # ## ## ## ## #### ## ## #### ## ### ## ### ###. #:

- Logger # ## ##### #### ## ## #### #####.
- ## ### HTTP ## ## ### ## #### ## ## ## ## ## ### ######## #####.
- EL ### ## (expression evaluation) ### ### ## #### ### #### #####.

@Dependent ### ## Web Bean# InjectionPoint ##### #### ## ## ### ### ###### ######.

#####. ### ## ### ##### ######## #### #####:

```
Logger log = Logger.getLogger(MyClass.class.getName());
```

JDK Logger# #### ###:

```
class LogFactory {
```

```
  @Produces Logger createLogger(InjectionPoint injectionPoint) {
    return Logger.getLogger(injectionPoint.getMember().getDeclaringClass().getName());
  }

}
```

####:

```
@Current Logger log;
```

####? ##### ### # ## ### ####. HTTP ## ### #####, ### ### #### ###:

```
@BindingType
@Retention(RUNTIME)
@Target({TYPE, METHOD, FIELD, PARAMETER})
public @interface HttpParam {
  @NonBinding public String value();
}
```

####:

```
@HttpParam("username") String username;
@HttpParam("password") String password;
```

#####:

```
class HttpParams

  @Produces @HttpParam("")
  String getParamValue(ServletRequest request, InjectionPoint ip) {
    return request.getParameter(ip.getAnnotation(HttpParam.class).value());
  }

}
```

`HttpParam` ###### `value()` ###`@NonBinding`.# ######### Web Bean #### ## #### ##### ####.

Web Bean #### `InjectionPoint` ###### #### ### Web Bean# #####:

```
public interface InjectionPoint {
   public Object getInstance();
   public Bean<?> getBean();
   public Member getMember():
   public <T extends Annotation
> T getAnnotation(Class<T
> annotation);
   public Set<T extends Annotation
> getAnnotations();
}
```

####

#######. Web Bean# ### Web Bean ##### ##### #####. ## ### ## ###### ## Web Bean ##### #### # ## #####. Web Beans ### ## ### ### #####:

- ### ## Web Bean# ### ##### ### ##
- ### ## Web Bean# ## ##### ### ##
- ### ## ### ### ## Web Bean ##### #### #

##, Web Bean ## ### ## ##, `CurrentUser`, ### `HttpSession` ###### #### ## Web Beans# ### `CurrentUser` ##### #### ###. ### ##### #### `CurrentUser`# #### ### # #### #### ### ### #### #####.

5.1. ##

Web Beans# *extensible context model*# #### ###. ### ## ## ###### #### ### ### ### # ####:

```
@Retention(RUNTIME)
@Target({TYPE, METHOD})
@ScopeType
public @interface ClusterScoped {}
```

#####. ### ## ### #### ###, ### #### `Context` ### #### ###! `Context` ### ## #### ####, ##### ## ## #### ###.

Web Bean ### #### ## Web Bean ## #### ## ## ###### ### # ####:

```
@ClusterScoped
public class SecondLevelCache { ... }
```

##, Web Beans# ### ## # ### #### ###.

5.2. ###

Web Beans# # ## ### ### #####:

- `@RequestScoped`
- `@SessionScoped`
- `@ApplicationScoped`

- `@ConversationScoped`

Web Beans# #### # ####### ##:

- servlet ### ## ##, ## # ###### ### ######
- JSF ### ## ##### ### ######.

####:

- EJB ## ### ###
- EJB ## ###
- ### ## ## ### ###
- # ### ###

######## ## ##### ## ### ## Web Bean# ###### ##, #### `ContextNotActiveException`# Web Bean #### ## ### ###.

Java EE ##### #####, ### ### #####. ### # # ### ## ### ####.

5.3. #####

Web Beans ##### ### ### #### ### ### #### ## ## ## ### ####, ### ## ## ### ####. ###, ## ### ### ##### ### ### ## ### ####:

- ######## ## ##### ### ####
- JSF ######## ## ## # #### ## ### ### #####.

##, ## ### #####. ##### ##### ## #### #### ## ## ### ### #####. #### ### #### ### #### ## ##, ## ## ####### ## ###.

JSF ## # ##### ####. ### #### ###### ## #### #####. ###### ## ## ### ## ### #### # ##, ## ## ######## ##### ###.

5.3.1. ##### ##

Web Beans# JSF ######## ## ##### ## ### #### ## ### Web Bean# #####. ### Web Bean# ### ## ### # ####:

```
@Current Conversation conversation;
```

######, ###### #### `begin()` ### #####. ## ### #### ### ## ## ## ##### ##### ###### `end()`# #####.

###, ##### ## Web Bean# ## # ###### #####:

```
@ConversationScoped @Stateful
public class OrderBuilder {

   private Order order;
   private @Current Conversation conversation;
   private @PersistenceContext(type=EXTENDED) EntityManager em;

   @Produces public Order getOrder() {
      return order;
   }

   public Order createOrder() {
      order = new Order();
      conversation.begin();
      return order;
   }

   public void addLineItem(Product product, int quantity) {
      order.add( new LineItem(product, quantity) );
   }

   public void saveOrder(Order order) {
      em.persist(order);
      conversation.end();
   }

   @Remove
   public void destroy() {}

}
```

Web Bean# `Conversation` API# #### ## ### ### # ####. ### ## ## Web Beans# ## ### ### #### ## ### ####.

5.3.2. #####

JSF faces ## (JSF ## ##)# ## #### #####. ## ##, ### ## ###### ## ## non-faces ### ## #### #### ####.

non-faces ### ## ##### ### # ####. Web Beans ### ## #### ## `cid` ## ## ## ### ####. ###### ## #### `Conversation` #### ### # ###, `conversation`### Web Beans ### ####.

###, ### ## ### ###### #####:

```
<a href="/addProduct.jsp?cid=#{conversation.id}"
>Add Product</a
>
```

Web Bean #### ###### ## #### #### ## ###, ###### ## ###### #### ###. ## "###" ### ## ### ## ### #### ## #### POST-then-redirect ## ## #### ###. ### ## Web Bean #### ##### URL# ## ## ### #### #####.

5.3.3. ##### ##

Web Bean ###### #### #### ## ##### ## ## ## # ###### ### # ## ### ####. Web Bean ### ### ## ## ### #### ## ##### # ## Web Beans ### ## #### ####. ## ### ###### ### # #### ### ####.

`Conversation` ### ## ### #### ## ### #####. ## Web Bean ##### ### ##, ## ### ##### ## ####.

```
conversation.setTimeout(timeoutInMillis);
```

5.4. dependent pseudo-scope

###, Web Beans# *dependent pseudo-scope*## ### #### ###. ## ## ### #### ## Web Bean# ## ### #####.

##, Web Bean## `@Dependent` ## ### ####:

```
public class Calculator { ... }
```

Web Bean# ## ### ### Web Bean# ### #, ### Web Bean# ### ##### #### Web Bean# #### # ## #####. ### Web Beans# ##### ## Web Bean ##### ## ## ## ## ## #### ####. ## ## ## Web Bean ##### ## ### #####.

Web Bean ##### #####.

EJB bean# ## ## ## ### ## Web Bean## #### ###, Web Beans# Java ### ## EJB bean# ### ##### ## ### # ## ###.

5.4.1. `@New`

`@New` ### ###### ## #### ### Web Bean# ### ### #####. ### ## ## ### ##### #####:

```
@New Calculator calculator;
```

`@Dependent` ##, `@New` ### ##, `Calculator` API ##, `Calculator` ## ###, `@Standard` ## ### ## Web Bean# ##### #####.

`Calculator`# ## ## ### ## ## #### ### ## #####, #:

```
@ConversationScoped
public class Calculator { ... }
```

`Calculator` ##### ####:

```
public class PaymentCalc {

    @Current Calculator calculator;
    @New Calculator newCalculator;

}
```

`calculator` #### `Calculator`# ##### ## ##### #### ####. `newCalculator` #### `PaymentCalc` ### #### ## ### ## `Calculator`# ### ##### #### ####.

#####.

##

Web Bean #### ###### ## #### ### ### # ### # ## ### ### ### # ## ###. Web Beans# ## ### Web Beans #### #### ## ## ## ### ####. (*12#. XML# #### Web Beans ##* ## ### ### ## #######.)

spec# ###:

> Web Beans ### ### ### ## ### #####, ###
>
> - ### ### Web Beans# ##### # ### ####,
> - #### # ### ### ### ### # ### # ####
> - ### ## #### ### ### Web Bean #### ## #### ####

##, ### ### ### #####:

- Web Bean### JPA ### ##,
- Web Bean### JDK ### ##,
- ### ## #### ## ## ## ## #### #### ## Web Beans# ##,
- ### # ### API ## ##.

##, ### ### Web Beans# #### ### #### #### ###. ## ## ###, ## ### ## ## ### ### ### ######. ### ## #### #### Web Bean ### #####. ### ### ### ### ## ####:

```
@SessionScoped
public class Preferences {

   private PaymentStrategyType paymentStrategy;

   ...

   @Produces @Preferred
   public PaymentStrategy getPaymentStrategy() {
      switch (paymentStrategy) {
         case CREDIT_CARD: return new CreditCardPaymentStrategy();
         case CHEQUE: return new ChequePaymentStrategy();
         case PAYPAL: return new PayPalPaymentStrategy();
         default: return null;
      }
   }
```

```
}
```

##:

```
@Preferred PaymentStrategy paymentStrat;
```

######, #### Web Beans ## ### #### ### ### #####. ### ### Web Bean ##### ## #### ### ## ### ##### ## ##### #####.

6.1. ### ###

`@Dependent`# ## ####, # # Web Bean #### ### ## ## ### ### #### #### ## ### ### # ## #####. ###, ### ### ### ## `PaymentStrategy` ### ##### ### ## # ####.

#####, `@SessionScoped` ##### ### ### # ####.

```
@Produces @Preferred @SessionScoped
public PaymentStrategy getPaymentStrategy() {
    ...
}
```

####, ### `PaymentStrategy`# ## ##### ######. ### #### ### ### ## #### ####.

6.2. ### ####

####. `CreditCardPaymentStrategy` ### Java `new` #### #### ########. ######## ## ## ####### ### ### ### ### ## # ### ##### ## ####.

##, ### ### ### #### #### Web Bean ##### ### # ####:

```
@Produces @Preferred @SessionScoped
public PaymentStrategy getPaymentStrategy(CreditCardPaymentStrategy ccps,
                                          ChequePaymentStrategy cps,
                                          PayPalPaymentStrategy ppps) {
   switch (paymentStrategy) {
      case CREDIT_CARD: return ccps;
      case CHEQUE: return cps;
      case PAYPAL: return ppps;
      default: return null;
   }
}
```

CreditCardPaymentStrategy# ## ## Web Bean## ### ###? ## ## ### ### ## ## ## ##### ## ### "#####" ###. ## #####! ## ## ### ### #### # Web Bean #### ## #####, ### ## ### ## ### "## ##" ###. ### ### Web Bean #### ## #### #### ### ### Web Bean ##### ### ## ### ##### ####!

####. CreditCardPaymentStrategy ## ### ### # ###, Web Bean# ###### ### ## # ####. ## ## #### ### ### ### @Dependent ## @RequestScoped# #### ####.

@New ### ###### #### ####.

6.3. ### ### ## @New

#####:

```
@Produces @Preferred @SessionScoped
public PaymentStrategy getPaymentStrategy(@New CreditCardPaymentStrategy ccps,
                                          @New ChequePaymentStrategy cps,
                                          @New PayPalPaymentStrategy ppps) {
   switch (paymentStrategy) {
      case CREDIT_CARD: return ccps;
      case CHEQUE: return cps;
      case PAYPAL: return ppps;
      default: return null;
   }
}
```

CreditCardPaymentStrategy# ### ### ##### ####, ### #### ####, ### ### ## ### #, ##### ## ##### ######. ### ### ## #### Preferences ### ### # ## #### ####.

II. ë##ì#¨í##ê²# ì#°ê²°ë## (loosely-coupled) ë°©ì##ì## ì½#ë## ê°#ë°#

Web Beans# ### ## ### ### ## *(loose coupling)*###. ### ### ## ### ##### ## ### ### ## ### ####:

- ## ### ## ## #### ######
- ### ### ### #### ######
- #### #### ### Web Bean ## ### ## #####.

####. ###### ### API #### #### # ###, ## ### ##### ### # ####. ### #### ## ## ### ##### # ## ## #### ###.

####. #### # ### ##### ## ### ### # ####. ## ### ### #### #### ## ### ###### ## #### #### ## #######. Web Beans# ## ### ##### ### ### #### ## ### #####.

Web Beans# ### ## ### #### ## # ## ### ### ### #####:

- ##### #### #### ### ### ### ## #####
- ###### ## #### ### ## #### ## ### # ####.
- ### ### ### #### ## ### #### ## #####.

#####.

####

Web Beans# EJB 3.0# ## #### ### #####, # ## #### ### #####:

- Web Bean## ## beans ## ### ##### ## # ####.
- Web Beans# Web Beans# ##### ##### ## ## ### ##### ## ### #### ###.

EJB ### # ### #### ### #####:

- #### ## ####
- #### ## ####

Web Bean ###### ## Web Bean ## ### #####:

```
public class TransactionInterceptor {
    @AroundInvoke public Object manageTransaction(InvocationContext ctx) { ... }
}
```

#####:

```
public class DependencyInjectionInterceptor {
    @PostConstruct public void injectDependencies(InvocationContext ctx) { ... }
}
```

####.

7.1. ####

Web Beans# ### ##### # ### ##### ### #####. #### ### ## ## Web Beans# ### ## ## #### ## #### ### ###### ###:

```
@InterceptorBindingType
@Target({METHOD, TYPE})
@Retention(RUNTIME)
public @interface Transactional {}
```

`ShoppingCart`# #### #### ## ### # ####:

```
@Transactional
public class ShoppingCart { ... }
```

##, ### ## ## ##### ##### ### # ####:

```
public class ShoppingCart {
   @Transactional public void checkout() { ... }
}
```

7.2. ####

###. ## # ### ## EJB ##### #### ## `@Interceptor` # `@Transactional`# ####### ####.

```
@Transactional @Interceptor
public class TransactionInterceptor {
   @AroundInvoke public Object manageTransaction(InvocationContext ctx) { ... }
}
```

Web Beans ##### ## Web Beans##, ### ### #### ## ## ### ### ## # ####.

```
@ApplicationScoped @Transactional @Interceptor
public class TransactionInterceptor {

   @Resource Transaction transaction;

   @AroundInvoke public Object manageTransaction(InvocationContext ctx) { ... }

}
```

####.

7.3. ####

`web-beans.xml`## ##### ##### ###.

```
<Interceptors>
   <tx:TransactionInterceptor/>
```

```
</Interceptors
>
```

###?

XML #### # ## ### #####:

- ## ##### ## ##### ## #### ### ### # ### #####, #### ### #####
- ### #### #### ### ## ###### ###.

##, `TransactionInterceptor` ### ## ##### ##### ### # ####.

```
<Interceptors>
   <sx:SecurityInterceptor/>
   <tx:TransactionInterceptor/>
</Interceptors
>
```

####!

7.4. ### ####

`@Transactional` ###### ### ##### ### #####:

```
@InterceptorBindingType
@Target({METHOD, TYPE})
@Retention(RUNTIME)
public @interface Transactional {
   boolean requiresNew() default false;
}
```

Web Beans# `requiresNew` ## #### # ## ## #### `TransactionInterceptor`# `RequiresNewTransactionInterceptor` ### #### ###.

```
@Transactional(requiresNew=true) @Interceptor
public class RequiresNewTransactionInterceptor {
   @AroundInvoke public Object manageTransaction(InvocationContext ctx) { ... }
}
```

`RequiresNewTransactionInterceptor`# ### # ####:

```
@Transactional(requiresNew=true)
public class ShoppingCart { ... }
```

`requiresNew` ## #### ## #### ## #### `@NonBinding` ###### ### # ####:

```
@InterceptorBindingType
@Target({METHOD, TYPE})
@Retention(RUNTIME)
public @interface Secure {
   @NonBinding String[] rolesAllowed() default {};
}
```

7.5. #### #### ###

Web Bean# ##### ## #### ### ### ### #####. ## ##, ### ## ### #### ### Web Bean# `TransactionInterceptor` # `SecurityInterceptor`# #### # ####:

```
@Secure(rolesAllowed="admin") @Transactional
public class ShoppingCart { ... }
```

####:

```
@Transactional @Secure @Interceptor
public class TransactionalSecureInterceptor { ... }
```

`checkout()` ### #### # ####:

```
public class ShoppingCart {
   @Transactional @Secure public void checkout() { ... }
}
```

```
@Secure
public class ShoppingCart {
   @Transactional public void checkout() { ... }
}
```

```
@TransactionI
public class ShoppingCart {
   @Secure public void checkout() { ... }
}
```

```
@Transactional @Secure
public class ShoppingCart {
   public void checkout() { ... }
}
```

7.6. #### ### ##

Java ## ## ### ##### ### #####. ### ###### ### ### #### ## ### ### #### ## ###:

```
public @interface Action extends Transactional, Secure { ... }
```

Web Beans# ### ### Java# ### #### ### #### ### ### ## #### ### ### ###### # ####. #### #### #### #### # ### #### #### ## Web Bean# ## ####### ### #### #### ### # ####.

```
@Transactional @Secure
@InterceptorBindingType
@Target(TYPE)
@Retention(RUNTIME)
public @interface Action { ... }
```

`@Action`## ###### Web Bean# `TransactionInterceptor` # `SecurityInterceptor`# ######. (`TransactionalSecureInterceptor`# ## ## #### ######.)

7.7. `@Interceptors`

EJB ### ## ### `@Interceptors` ###### ###### # ## Web Beans ### #####. #:

```
@Interceptors({TransactionInterceptor.class, SecurityInterceptor.class})
public class ShoppingCart {
   public void checkout() { ... }
}
```

###, ### ### ### ## ### ####:

- ##### ### ##### ##### ####### #####
- ###### ### ## ###### # #####,
- ##### ### ######## # ## ### ### ### ##### ### ## ######.

Web Beans ## ##### ##### ### ## ######.

#####

#####. ##### ## Java ## ### ### # ####. ## #### ## # ### ## ### ### ##### #####. ###, ## ##### ### #### ### #### #### ####. ###, ##### #### ## ### #### ## #### #### ####.

########. ###### ## Java ###### #### ### ####, ###### ### ## #### #####. ## ## #### ### ##### ## ###### ### ### ####. ## ###### ##### ### ### ## ####. ###### ## ### ### #### ### ### ### # ####.

#####:

```
public interface Account {
   public BigDecimal getBalance();
   public User getOwner();
   public void withdraw(BigDecimal amount);
   public void deposit(BigDecimal amount);
}
```

Web Beans# `Account` ###### #####. ###, #### ## ##### ###, ## ### ##### ##, ### ##### ## ### #### ## ##### ###. ## ###### ### ## #####.

Web Bean## ### #### `@Decorator`# ########.

```
@Decorator
public abstract class LargeTransactionDecorator
      implements Account {

   @Decorates Account account;

   @PersistenceContext EntityManager em;

   public void withdraw(BigDecimal amount) {
      account.withdraw(amount);
      if ( amount.compareTo(LARGE_AMOUNT)
>0 ) {
         em.persist( new LoggedWithdrawl(amount) );
      }
   }

   public void deposit(BigDecimal amount);
      account.deposit(amount);
      if ( amount.compareTo(LARGE_AMOUNT)
```

```
>0 ) {
        em.persist( new LoggedDeposit(amount) );
    }
  }

}
```

Web Beans# ###, ###### ###### # # ####. ###### ###### ## ### ## ###### ### ### ## ## ##, # ### #### ### ####.

8.1. ##

####. ## ### ## # ### ### ## Web Beans# ###### #### ## #####. ## ## ### ###### ## ### ## ###### ## ## #### ###.

`Account`# #### ## Web Beans# ##### ###### #####.

```
@Decorates Account account;
```

####. # # ###### ### #### #### Web Beans# ##### ###.

```
@Decorates @Foreign Account account;
```

Web Bean## ######:

- API ### ## ## ## ##
- ## ### ## ### ## ### ##

###, ###### `InvocationContext.proceed()`# #### ## ### ### ####.

8.2. #####

`web-beans.xml`## ###### ##### ###.

```
<Decorators>
   <myapp:LargeTransactionDecorator/>
</Decorators
>
```

`<Interceptors>` ### ##### ## #### ## ### ### #####:

- ## ##### ## ###### ## #### ### ### # ### #####, #### ### #####
- ### ##### #### ### ## ###### ###.

#####.

###

Web Beans ### ## ### Web Beans# ## ## ## #### ## #### ###. ### #### #### ### # Web Bean #### ## ### #### #####. ### #### #### ###/### ### #### ####, # ## ## ### ####:

- ### #### ##### ## #### ## ###; #### ##### #### ## #####.
- #### "###" ### #### ### ## ### ### ####.
- #### ## ##### ## ## ##### ### ## ### ### ##### ### # ####.

9.1. ###

`@Observes`# #### Web Bean #####.

```
public void onAnyDocumentEvent(@Observes Document document) { ... }
```

####. ### ## ### ### #### ### #####. ### ### Web Beans ### ### ##### "###"# ### # ####. ### ### ### #### ### ##, ## ### ### ##### ####.

```
@BindingType
@Target({PARAMETER, FIELD})
@Retention(RUNTIME)
public @interface Updated { ... }
```

#####:

```
public void afterDocumentUpdate(@Observes @Updated Document document) { ... }
```

###. ### #### #### ## ##, ### ### #### ## ##### ### ## ###.

###, ## #### Web Beans ## ## ## ## #### ## #####:

```
public void afterDocumentUpdate(@Observes @Updated Document document, User user) { ... }
```

9.2. ###

####:

```
@Observable Event<Document
> documentEvent
```

`@Observable` ###### `@Dependent` ## # `@Standard` ## ##, Web Bean #### ## #### #### Web Bean# #####.

`Event` ###### `fire()` ### #### #### ####, ### ### #####:

```
documentEvent.fire(document);
```

Java #### ##### # # # ####. #### ### ## ### ## ## ### ### #####:

- ### ### ### # ## ### ## ### ####,
- ### ### ### #####.

Web Bean #### ## ### ### ####, ### ## ## ### ### ### #####. ### ### ## ### ### # ##, Web Bean #### ### ## ### #### ## ### `fire()` ### ## ## ### ###.

"###"# #####, ### #### ### ### ### `fire()` ### #### ###:

```
documentEvent.fire( document, new AnnotationLiteral<Updated
>(){} );
```

Java## #### #### ### ### `AnnotationLiteral`# ### ## #### ###### # ## ###.

#####:

- ### ### ### # ## ### ## ### ####,
- ## `fire()`# ### ### #### ## ## ### ### ## ### #### ####.

####:

```
@Observable @Updated Event<Document
> documentUpdatedEvent
```

`Event` ##### ## ### ## ##### ###### ### #### ####. #### ## ### ### #####:

- ### ### ### # ## ### ## ### ####,
- ## ### ## ## ### ###### ### ##### ## `fire()`# ### ### #### ## ## ### ### ## ### #### ####.

9.3. #### ####

#####. ######## `Observer` ###### #### `observe()` ### #### ### ### ## ##### ### # ####.

```
documentEvent.observe( new Observer<Document
>() { public void notify(Document doc) { ... } } );
```

`observe()` ### ### ### ## ##### #### ### # ####:

```
documentEvent.observe( new Observer<Document
>() { public void notify(Document doc) { ... } },
                                    new AnnotationLiteral<Updated
>(){} );
```

9.4. ### ###

####:

```
@BindingType
@Target({PARAMETER, FIELD})
@Retention(RUNTIME)
public @interface Role {
   RoleType value();
}
```

#####:

```
public void adminLoggedIn(@Observes @Role(ADMIN) LoggedIn event) { ... }
```

####:

```
@Observable @Role(ADMIN) Event<LoggedIn
```

```
> LoggedInEvent;}}
```

####, ### ### ## ## ## ### #### ## #### ### # ####. `AnnotationLiteral`# ### ###### #### #####:

```
abstract class RoleBinding
   extends AnnotationLiteral<Role
>
   implements Role {}
```

`fire()`# #####:

```
documentEvent.fire( document, new RoleBinding() { public void value() { return user.getRole();
 } } );
```

9.5. ## ###

####, #:

```
@Observable @Blog Event<Document
> blogEvent;
...
if (document.isBlog()) blogEvent.fire(document, new AnnotationLiteral<Updated
>(){});
```

####, ### ## ### ## ### ##### ###:

```
public void afterBlogUpdate(@Observes @Updated @Blog Document document) { ... }
```

```
public void afterDocumentUpdate(@Observes @Updated Document document) { ... }
```

```
public void onAnyBlogEvent(@Observes @Blog Document document) { ... }
```

```
public void onAnyDocumentEvent(@Observes Document document) { ... }}}
```

9.6. ####

####. ## ##, ### ## ### ### ##### `Category` ### ##### ###### #### ###### ###### ### ## ## ### ###### ###:

```
public void refreshCategoryTree(@AfterTransactionSuccess @Observes CategoryUpdateEvent
 event) { ... }
```

####:

- ##### ## ## ## ## ##### ##### #### ####, `@AfterTransactionSuccess` #### #####.
- ##### ## ## ## ## ##### ##### #### ## ####, `@AfterTransactionFailure` #### #####.
- ##### ## ## ## ## `@AfterTransactionCompletion` #### #####
- ##### ## ## ## ## `@BeforeTransactionCompletion` #### #####

Web Beans# ## ## ## ## #### ## #####. ## ## ### ###### ## #### #####.

JPA ## ## ### ##### #####:

```
@ApplicationScoped @Singleton
public class Catalog {

   @PersistenceContext EntityManager em;

   List<Product
> products;

   @Produces @Catalog
   List<Product
> getCatalog() {
      if (products==null) {
         products = em.createQuery("select p from Product p where p.deleted = false")
            .getResultList();
      }
      return products;
   }

}
```

`Product`# ## ## #####. ### ### ####, `Product` ##### ###### ###. ###, ### ## ### #### # ##### ##### ### # ## ##### ###!

`Product`# ## # #### Web Bean# #### ### # ####, #:

```
@Stateless
public class ProductManager {

   @PersistenceContext EntityManager em;
   @Observable Event<Product
> productEvent;

   public void delete(Product product) {
      em.delete(product);
      productEvent.fire(product, new AnnotationLiteral<Deleted
>(){});
   }

   public void persist(Product product) {
      em.persist(product);
      productEvent.fire(product, new AnnotationLiteral<Created
>(){});
   }

   ...

}
```

`Catalog`# ##### ##### ### ## #### #### # ####:

```
@ApplicationScoped @Singleton
public class Catalog {

   ...

   void addProduct(@AfterTransactionSuccess @Observes @Created Product product) {
      products.add(product);
   }

   void addProduct(@AfterTransactionSuccess @Observes @Deleted Product product) {
      products.remove(product);
   }

}
```

III. ê°#ì#¥ ê°#í## í##ì#´í## (strong typing) ì#¬ì#©

Web Beans# ### ## ### ## ### *(strong typing)*###. Web Bean# ###, ####, ###### ## ## # ### #### ## ### #### ## ### ##### ## ### # ## ## ### Java ## ### ## ####.

Web Beans #### ### ## #### ## # ####. # ### ###### # "### ## ##"### ### # ### ### #### ## ##### ### ### ### # ## #### ## #####!

IDE# ## ## ## ## ##, ##, ##### ### # ####. ### ## #### ### ####. ## ## ###### ## ##, ###, ##### ##### # ## ### ### ### ### ### ### # ####.

Web Beans# ### ### ## ###### #### ###. #:

- `@Asynchronous`,
- `@Mock`,
- `@Secure` ##
- `@Updated`,

##

- `asyncPaymentProcessor`,
- `mockPaymentProcessor`,
- `SecurityInterceptor` ##
- `DocumentUpdatedEvent`.

####. ## #### ## ## ### #### ### ##### ####, ### #### ###### ### # ####. ## #### ### #### #### ##### ### ##, ### ## # #### ## # # ## ###.

Web Beans# #### ### ### ### # ## # #####. #### ### ###### ##### #### ### ### ###. ## ##, #### ###, ## ## ## ## ### ### #### ## ## ### ## #### ######.

Web Beans XML ###### ### ######! XML # ##### #### Web Beans# XML #### ### ## XML# #### Java ## # ### #### ###. ###### Java ### ## # # ## ## # ## ### ### XML# ## # ## # # ## ###.

Web Beans# ## ## ### ## ### ### #####. ### ### ### ## ## ##### ## #### ## ## #### ### ##### ####. ### ### ### ### ### ### ## ## ##### ### ##, ### ### #### # ####.

(Stereotypes)

Web Beans ### ##:

> ## #####, #### ### #### Web Bean ### #### ### #####. ####### ##### #### ### ### #### ## ##### ### ## Web Beans# #### ###### #### ###.
>
> ####### ### ### ######:
>
> - ### ## ##
> - ### ## ##
> - Web Bean ##### ##,
> - ## ### Web Bean# ##### #### ## ## ##
> - #### ### ##### ##
>
> ####### ####### ## ## Web Beans# Web Bean ### ##### ### ### # # ####.
>
> Web Bean# 0 ## ## ### ####### ### ####.

####### Java ##### #####. ### ####### ## MVC ####### ## #### #####:

```
@Retention(RUNTIME)
@Target(TYPE)
@Stereotype
public @interface Action {}
```

Web Bean# ###### #### ####### #####.

```
@Action
public class LoginAction { ... }
```

10.1. ####### ## ### ## # ##

####### ## ####### ## Web Beans# ### ## ## ### ## ### ### # ####. ## ##, `@WebTier` ## ### #### # ######## ### ## Web Beans# #### ### ##, ### ## #### ## ### ## #### ### # ####:

```
@Retention(RUNTIME)
```

```
@Target(TYPE)
@RequestScoped
@WebTier
@Stereotype
public @interface Action {}
```

####:

```
@Dependent @Mock @Action
public class MockLoginAction { ... }
```

##, #### # ####.

10.2. ######## ## # ##

#####. Web Beans# ## ######## ## Web Beans# ## ## ### ##### ### # ## ###. #:

```
@Retention(RUNTIME)
@Target(TYPE)
@RequestScoped
@WebTier
@Stereotype(supportedScopes=RequestScoped.class)
public @interface Action {}
```

Web Beans ## ## ## ### ##### # ##, #### ## ### Web Bean #### ## #### ###.

Web Bean# #### ###### ##### #### ### # ####:

```
@Retention(RUNTIME)
@Target(TYPE)
@RequestScoped
@WebTier
@Stereotype(requiredTypes=AbstractAction.class)
public @interface Action {}
```

`AbstractAction` #### #### ## ##, #### ## ### Web Bean #### ## #### ###.

10.3. ###### # ####

######## ######## ## ## Web Beans# ## #### ## #### ### ### ### # ####.

```
@Retention(RUNTIME)
@Target(TYPE)
@RequestScoped
@Transactional(requiresNew=true)
@Secure
@WebTier
@Stereotype
public @interface Action {}
```

###!

10.4. ####### ## ### #####

Web Beans# Web Bean #### ## ##### Web Bean ### ### ### # ####. ### JSP ##### ### # ##, ### ### #### ### # ## ### # # ####. ### ### # `@Named` ###### #### ####:

```
@Retention(RUNTIME)
@Target(TYPE)
@RequestScoped
@Transactional(requiresNew=true)
@Secure
@Named
@WebTier
@Stereotype
public @interface Action {}
```

`LoginAction`# `loginAction`### ### ## ###.

10.5. ##

Web Beans ### ## ### ### ## ###### `@Interceptor` # `@Decorator`# #######.

Web Beans# ## ### ## ####### #####:

```
@Named
@RequestScoped
@Stereotype
@Target({TYPE, METHOD})
@Retention(RUNTIME)
public @interface Model {}
```

JSF# ## #### ## ####. JSF ## beans# #### ## Web Bean `@Model`# ####### ## ## JSF ##### #####.

###

Web Beans ### ## ### ### API ### #### ## ###. ## ##, ### ## ###### Web Bean# #### API `PaymentProcessor` ### #####:

```
@CreditCard @Stateless
public class CreditCardPaymentProcessor
      implements PaymentProcessor {
   ...
}
```

staging ####, ## Web Bean# #### `PaymentProcessor` ### #### ###:

```
@CreditCard @Stateless @Staging
public class StagingCreditCardPaymentProcessor
      implements PaymentProcessor {
   ...
}
```

`StagingCreditCardPaymentProcessor# #### ##### ### ## ### ### AsyncPaymentProcessor# ## #### ####. ### ## ## @Staging# ### ## ## @Production ## ## ## ### ## ### ###### ### ## ## ### #### ###`:

```
@CreditCard PaymentProcessor ccpp
```

`StagingCreditCardPaymentProcessor` ##### ### ###.

####:

- ## ## ### Web Bean# ####### Web Bean# ## API ### ### # ####.
- ## ## ### Web Bean# ####### Web Bean# ## ### ### ### # ####.
- ## ## ### Web Bean# ## ##### Web Bean# ### ### ## # ####.
- ####### Web Bean# ### ##, ## ## ## ### ### ### # ####.

##, ## ##### Web Bean# #### ## #### ### ###, ###### ## ### ### #### ### ####.

Web Beans# ##### ## ### ####, ##### ### ### ### ### ####. ### ### #### ## #### ####, ### ## #### ### #### ## ## ### #### ####.

11.1. ### ##

Web Beans # ###### Web Beans## ## #####. ### ### ##### ## ## ### Web Bean# ## ### ### ###:

- ##### Web Bean# #### ## #### ### ###.
- ##### Web Bean# ## Web Bean# ## ## Web Bean# ### ## ## #### # Web Bean# ###### Web Bean# ## ###### Web Bean# ### ###
- `@Specializes#` ######## ###.

```
@Stateless @Staging @Specializes
public class StagingCreditCardPaymentProcessor
      extends CreditCardPaymentProcessor {
   ...
}
```

Web Bean# ## ###### ###### ###.

11.2. ### ###

###:

- ###### ### ### `@Specializes#` ###### Web Bean# ## #### #####
- ###### Web Bean ### `@Specializes#` ###### Web Bean# ## #### #####
- ###### ## ### ### ##, ## ## # ### ### `@Specializes#` ###### Web Bean ##### #####.

`CreditCardPaymentProcessor#` `@CreditCard` ### ### `StagingCreditCardPaymentProcessor#` ## #####.

Web Bean #### ## ### ######:

- ###### ## API ### `@Specializes#` ###### Web Bean# API ##### (##### ###### ## ## ## ###### ###### ## ###### ###)
- `@Specializes#` ###### Web Bean# ## ### ###### ## #### ## ## ### ####
- ###### ##### #### Web Bean# ####.

##, Web Bean #### #### ## ### ### ###.

###, `@Specializes##` ###### Web Bean# #### #### #### ## ##### ###### ### #### ### ### # ####.

XML# #### Web Beans

Web Beans# ## #######. ### Web Bean# #### ## ###### ### # ## ### ## ####:

- ## #### ## ## ###### ##### ##,
- ### ## #### ## ### Web Beans# ## ##

##, Web Beans# # ## ### #####:

- ### ## ## ##
- XML# #### Web Bean ##

XML# #### Java ### ## ###### #####. ###, Web Beans# ### ## ### #### Java ### ##, ##, ### #### ## ###### #####. XML ## # ### ### ### ### # ## ### #### ##, Web Beans# XML ## #### ### ## ## ### #### ###.

XML #### ### ### #### ## ## XML #### ### # ### ####. #### Java #### XML #### #### #### ## ### ## # # ####. ## ### ## ### ## ## ### ### ## ### ### ### # ####.

12.1. Web Bean ###

Java #### ##, Web Beans# ## XML namespace# #####. namespace# Java ### ### `urn:java:#` ###### #####. `com.mydomain.myapp` #### ##, XML namespace# `urn:java:com.mydomain.myapp`###.

Java ### #### #### namespace# ## XML ### #### #####. ## ### Java ### ### ###. ## ## # ### ### namespace# ## ## ### ## #####. ### ######## ## ##, ### ### ### ## #####.

##, ### XML ### ## `<util:Date/>` ### `java.util.Date` #### #####:

```
<WebBeans xmlns="urn:java:javax.webbeans"
      xmlns:util="urn:java:java.util">

   <util:Date/>

</WebBeans
>
```

`Date`# ## Web Bean## #### # ## #####! `Date`# ##### ## ## Web Bean# ## ### # ####:

```
@Current Date date
```

12.2. Web Bean #####

Web Bean ### #### ## ### #### ##, ## ##, #### ### ### ### # ####:

```
<myapp:ShoppingCart>
   <SessionScoped/>
   <myfwk:Transactional requiresNew="true"/>
   <myfwk:Secure/>
</myapp:ShoppingCart
>
```

#####:

```
<util:Date>
   <Named
>currentTime</Named>
</util:Date>

<util:Date>
   <SessionScoped/>
   <myapp:Login/>
   <Named
>loginTime</Named>
</util:Date>

<util:Date>
   <ApplicationScoped/>
   <myapp:SystemStart/>
   <Named
>systemStartTime</Named>
</util:Date
>
```

`@Login` # `@SystemStart`# ### ##### #####.

```
@Current Date currentTime;
@Login Date loginTime;
@SystemStart Date systemStartTime;
```

Web Bean# ## ### ### ### # ####:

```
<myapp:AsynchronousChequePaymentProcessor>
   <myapp:PayByCheque/>
   <myapp:Asynchronous/>
</myapp:AsynchronousChequePaymentProcessor
>
```

Web Beans###, ## ## ## Web Bean## ### # ####:

```
<myfwk:TransactionInterceptor>
   <Interceptor/>
   <myfwk:Transactional/>
</myfwk:TransactionInterceptor
>
```

12.3. Web Bean ##

##!

12.4. ### Web Beans

Web Beans# ## #### Web Bean# ### # ## ###. #:

```
<myapp:System>
   <ApplicationScoped/>
   <myapp:admin>
      <myapp:Name>
         <myapp:firstname
>Gavin</myapp:firstname>
         <myapp:lastname
>King</myapp:lastname>
         <myapp:email
>gavin@hibernate.org</myapp:email>
      </myapp:Name>
   </myapp:admin>
</myapp:System
>
```

`<Name>` ### ### ### ### ## `@Dependent` ### ## Web Bean # `Name` #### #####. Web Bean# ### #### ## #### #### ### ## ## #### ### # ####.

Web Beans XML ### Java ### ## #### ##### #### ###. ## ### ###### #### ####, ### ######!

12.5. ###

Java ### ### ## ### #### # ## #### ## ### XML ## ### ### ##, #### #### ###. #### ## ## ### ## Web Beans# ### ### ####.

```
<WebBeans xmlns="urn:java:javax.webbeans"
        xmlns:myapp="urn:java:com.mydomain.myapp"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="urn:java:javax.webbeans http://java.sun.com/jee/web-beans-1.0.xsd
                          urn:java:com.mydomain.myapp http://mydomain.com/xsd/myapp-1.2.xsd">

   <myapp:System>
      ...
   </myapp:System>

</WebBeans
>
```

XML ### ### ## #####. ### Web Beans RI ##### #### ## #### XML #### #### #### ### #### ###.

IV. Web Beans ë°#
Java EE ì##ì½#ì##ì#¤í##

Web Beans# # ## ### ## #####. Web Beans# ## ### ### ##### ####, ###### #### ## ### ## #### ###. ## Java EE ###### #### #### ### ###### ## ## ###### #### ## ## # ### ### ### ###.

EJB# JSF #### ## ##### ## Web Beans# ### EJB # JSF# ####### #######. ## ## ### #####. Web Beans# #### #### ## ##, ## ## # #####, ### ## ## ### ## ## ### ##### ## ### ### ### #####. Java EE #### Java ###### ### #### ## ### #### # ###, Web Beans# Java EE ## ## ## #### #### ### ## ### ## ## ### # ## ###.

Web Beans# #### ######## Java EE #### ### ##### #### ### ## #######. ## Web Beans# ### ### #### ## #### SPI ### #### #### ########. ### SPI ### ## ### ### ## # # ###, ## ### ## ### ### ### ### ## ##### ####. ## #### ### ## ### ### # ## ### ### ### ##### ### # ####.

Java EE

Web Beans# Java EE #### ## #####. Web Beans# Java EE ### # JPA ### ##### ######. ## JSF # JSP ##### Unified EL ##### ### # ####. ## Web Beans# ## Servlets # Message-Driven Beans# ## ## ### ### # # ####.

13.1. Java EE #### Web Bean##

Web Beans# `@Resource`, `@EJB`, `@PersistenceContext`# #### Java EE ### ### ### ## # ####. ## ## ## ## ## #######:

```
@Transactional @Interceptor
public class TransactionInterceptor {

   @Resource Transaction transaction;

   @AroundInvoke public Object manageTransaction(InvocationContext ctx) { ... }

}
```

```
@SessionScoped
public class Login {

   @Current Credentials credentials;
   @PersistenceContext EntityManager userDatabase;

   ...

}
```

Java EE `@PostConstruct` # `@PreDestroy` ### ## ###### Web Beans# ## #####. ## ### ### # `@PostConstruct` ### #####.

####: `@PersistenceContext(type=EXTENDED)`# ### Web Beans# ## #### ####.

13.2. Servlet## Web Bean

Java EE 6# ## Servlet## Web Bean# ##### ####. Web Beans ## ## ### ## ### #### Web Bean# #####.

```
public class Login extends HttpServlet {

    @Current Credentials credentials;
    @Current Login login;

    @Override
    public void service(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        credentials.setUsername( request.getAttribute("username") ):
        credentials.setPassword( request.getAttribute("password") ):
        login.login();
        if ( login.isLoggedIn() ) {
            response.sendRedirect("/home.jsp");
        }
        else {
            response.sendRedirect("/loginError.jsp");
        }
    }

}
```

Web Beans ##### #### Servlet## ## ## # ### HTTP ### ### `Credentials` # `Login` ##### ### #### #####.

13.3. Message-Driven Bean## Web Bean

Web Beans ### ## EJB# ####, Web Bean ### ## ## ## ## #### (## ##, JNDI lookup# ## ###### `@EJB`# #### ##### ##) #####. ##, Message-Driven Beans## Web Beans ### ### # ###, ## ### # ## ### ## Web Beans# #### ####.

Message-Driven Beans# ## Web Beans #### #### ### # ####.

```
@Transactional @MessageDriven
public class ProcessOrder implements MessageListener {

    @Current Inventory inventory;
    @PersistenceContext EntityManager em;

    public void onMessage(Message message) {
        ...
    }
```

```
}
```

###, Web Beans #### #### ## ## ######. ### #### Message-Driven Bean## ### # ### # ## ## ## ## ## ##### ### ##### ####. `@RequestScoped` # `@ApplicationScoped` Web Beans ## ### # ####.

Web Beans# #### #### #### ####.

13.4. JMS

JMS# #### #### #### ## ## ### # ####. ## ### ## ### #### ## #####. ## ## `Queue`, `QueueConnectionFactory`, `QueueConnection`, `QueueSession` `QueueSender`# ####. ### ## `Topic`, `TopicConnectionFactory`, `TopicConnection`, `TopicSession`, `TopicPublisher`# ####. ### ### ### ##### ## ## # ### ### ## ####.

Web Beans# ### ## ## #####. ### #### # ### `web-beans.xml`# ## # ## ### #### ### ### ## # ## #### #### ####.

```
<Queue>
   <destination
>java:comp/env/jms/OrderQueue</destination>
   <connectionFactory
>java:comp/env/jms/QueueConnectionFactory</connectionFactory>
   <myapp:OrderProcessor/>
</Queue
>
```

```
<Topic>
   <destination
>java:comp/env/jms/StockPrices</destination>
   <connectionFactory
>java:comp/env/jms/TopicConnectionFactory</connectionFactory>
   <myapp:StockPrices/>
</Topic
>
```

`Queue`, `QueueConnection`, `QueueSession`, `QueueSender`# ### # ##, ### ## `Topic`, `TopicConnection`, `TopicSession` `TopicPublisher`# ### # ####.

```
@OrderProcessor QueueSender orderSender;
@OrderProcessor QueueSession orderSession;
```

```
public void sendMessage() {
   MapMessage msg = orderSession.createMapMessage();
   ...
   orderSender.send(msg);
}
```

```
@StockPrices TopicPublisher pricePublisher;
@StockPrices TopicSession priceSession;

public void sendMessage(String price) {
   pricePublisher.send( priceSession.createTextMessage(price) );
}
```

JMS ### ##### Web Bean #### ## #### #####.

13.5. ### #

Web Beans# ## ## ##### #### ####. JAR, EJB-JAR ## WAR# ## Web Beans # ###### classpath# ## ## ### ##### # ####. ### Web Beans# ## ### ###### `META-INF` ## `WEB-INF` ##### ## `web-beans.xml`## ### ## ### ###. ### ## ## # # ####. `web-beans.xml` ### ## ##### ### Web Beans# ######## ### # ####.

Java SE ### ##, ### ### EJB Lite ##### ## ### EJB# ### # ## ### Web Beans# ### # ####. # ##### `web-beans.xml` ### ## ### ###.

Web Beans

Web Beans# ## #### ##, ##, ###### ## #### ## ## ####. ###, Web Beans# Web Beans## ### ### ## #### ### ## SPI ### #####. ## ##, ### ## ### ### Web Beans #### ## #######.

- #### #### ## #### ##
- Spring, Seam, GWT, Wicket# ## ### ####### ##
- Web Beans ##### ### #### ## ### ##

Web Beans ### ### `Manager` #####.

14.1. `Manager`

`Manager` ###### Web Beans, ####, #####, ### # ##### ####### #### ### # ## ###.

```
public interface Manager
{

  public <T
> Set<Bean<T
>
> resolveByType(Class<T
> type, Annotation... bindings);

  public <T
> Set<Bean<T
>
> resolveByType(TypeLiteral<T
> apiType,
      Annotation... bindings);

  public <T
> T getInstanceByType(Class<T
> type, Annotation... bindings);

  public <T
> T getInstanceByType(TypeLiteral<T
> type,
      Annotation... bindings);

  public Set<Bean<?>
> resolveByName(String name);
```

```
   public Object getInstanceByName(String name);

   public <T
> T getInstance(Bean<T
> bean);

   public void fireEvent(Object event, Annotation... bindings);

   public Context getContext(Class<? extends Annotation
> scopeType);

   public Manager addContext(Context context);

   public Manager addBean(Bean<?> bean);

   public Manager addInterceptor(Interceptor interceptor);

   public Manager addDecorator(Decorator decorator);

   public <T
> Manager addObserver(Observer<T
> observer, Class<T
> eventType,
         Annotation... bindings);

   public <T
> Manager addObserver(Observer<T
> observer, TypeLiteral<T
> eventType,
         Annotation... bindings);

   public <T
> Manager removeObserver(Observer<T
> observer, Class<T
> eventType,
         Annotation... bindings);

   public <T
> Manager removeObserver(Observer<T
> observer,
         TypeLiteral<T
> eventType, Annotation... bindings);

   public <T
```

```
> Set<Observer<T
>
> resolveObservers(T event, Annotation... bindings);

  public List<Interceptor
> resolveInterceptors(InterceptionType type,
        Annotation... interceptorBindings);

  public List<Decorator
> resolveDecorators(Set<Class<?>
> types,
        Annotation... bindings);

}
```

Manager ##### ## # ####:

```
@Current Manager manager
```

14.2. Bean

Bean ## #### ##### Web Beans# #####. ######## ## ## Web Bean# #### Manager ### ### Bean ##### ####.

```
public abstract class Bean<T> {

   private final Manager manager;

   protected Bean(Manager manager) {
      this.manager=manager;
   }

   protected Manager getManager() {
      return manager;
   }

   public abstract Set<Class> getTypes();
   public abstract Set<Annotation> getBindingTypes();
   public abstract Class<? extends Annotation> getScopeType();
   public abstract Class<? extends Annotation> getDeploymentType();
   public abstract String getName();
```

```
   public abstract boolean isSerializable();
   public abstract boolean isNullable();

   public abstract T create();
   public abstract void destroy(T instance);

}
```

Web Beans ## (## Web Beans # ###### Web Beans, ### ## # JMS #####)# ## ### # #### ### ### Web Beans# #### ## `Bean` #### #### `Manager.addBean()`# #### ##### ### # ####. ## ##, `Bean` #### #### Web Beans# ### ## ###### ## #### ### #### # # ####.

Web Beans ### ## #### # ## `Bean` ## #### ####: `Interceptor` # `Decorator`.

14.3. `Context`

`Context` ###### Web Beans# ### ### ### ##### ### ### ### ### ## #####.

```
public interface Context {

   public Class<? extends Annotation> getScopeType();

   public <T> T get(Bean<T> bean, boolean create);

   boolean isActive();

}
```

##, Web Beans# #### #### ### ##### ## Wicket# #### ####### ##### ### ## ### #### ## `Context`# ### # ####.

##

Web Beans# ### ##### ### ## ###### ## ### ### ## ####.

Web Beans ### ### Web Beans ### ## ### #####. ## # 100 #### # # ### # # ### ### ## # ####. ## ### ### ### ##### ### ###, `http://jcp.org/en/jsr/detail?id=299`## ## # ####.

Web Beans Reference ### `http://seamframework.org/WebBeans`## #### ####. RI ### # Web Beans ### `http://in.relation.to`# ## #### ####. ### ### ##### #### ### ### #### #### ###.

A. Web Beans RI# ## ####

Web Beans RI# JBoss AS 5### #####; RI# ## ## EE ## (## ## Glassfish# ## ## ###### ##)## ##, ## servlet ##### ## (#: Tomcat), ## ### EJB3.1 ### #### ## ####. Appendix## ### ### ## #### #######.

##

SE #### Web Beans# ### # ### ###, ### #### # ## ## ### ## # ## ### ## ###. ## Web Beans RI# ## ## ### #### ## ####, Web Beans RI #### ## ## ##### ###.

A.1. Web Beans RI SPI

Web Beans SPI# `webbeans-ri-spi` ### #### ### `webbeans-ri-spi.jar`# ##### ####. ## SPI# ## ####, ### ### ####### ##, ## ### #####.

SPI# ## ## ###### ##### ### #### `Forwarding` #### #####.

A.1.1. Web Bean

```
public interface WebBeanDiscovery {
  /**
   * Gets list of all classes in classpath archives with web-beans.xml files
   *
   * @return An iterable over the classes
   */
  public Iterable<Class<?>
> discoverWebBeanClasses();

  /**
   * Gets a list of all web-beans.xml files in the app classpath
   *
   * @return An iterable over the web-beans.xml files
   */
  public Iterable<URL
> discoverWebBeansXml();

}
```

Web Bean ### # `web-bean.xml` ## ### ## # # #### (##### JSR-299 ## 11.1## #### #### ### ## #### ####).

A.1.2. EJB

Web Beans RI# ##### EJB3 bean ### ##### EJB3 ###### ##### `ejb-jar.xml`# ### ### ####. ######## ## ### EJB# ## EJBDescriptor# #### ###:

```
public interface EjbDiscovery
{
  public static final String PROPERTY_NAME = EjbDiscovery.class.getName();

  /**
   * Gets a descriptor for each EJB in the application
   *
   * @return The bean class to descriptor map
   */
  public Iterable<EjbDescriptor<?>
> discoverEjbs();

}
```

```
public interface EjbDescriptor<T
> {

  /**
   * Gets the EJB type
   *
   * @return The EJB Bean class
   */
  public Class<T
> getType();

  /**
   * Gets the local business interfaces of the EJB
   *
   * @return An iterator over the local business interfaces
   */
  public Iterable<BusinessInterfaceDescriptor<?>
> getLocalBusinessInterfaces();

  /**
   * Gets the remote business interfaces of the EJB
   *
   * @return An iterator over the remote business interfaces
```

```
   */
   public Iterable<BusinessInterfaceDescriptor<?>
> getRemoteBusinessInterfaces();

   /**
    * Get the remove methods of the EJB
    *
    * @return An iterator over the remove methods
    */
   public Iterable<Method
> getRemoveMethods();

   /**
    * Indicates if the bean is stateless
    *
    * @return True if stateless, false otherwise
    */
   public boolean isStateless();

   /**
    * Indicates if the bean is a EJB 3.1 Singleton
    *
    * @return True if the bean is a singleton, false otherwise
    */
   public boolean isSingleton();

   /**
    * Indicates if the EJB is stateful
    *
    * @return True if the bean is stateful, false otherwise
    */
   public boolean isStateful();

   /**
    * Indicates if the EJB is and MDB
    *
    * @return True if the bean is an MDB, false otherwise
    */
   public boolean isMessageDriven();

   /**
    * Gets the EJB name
    *
    * @return The name
```

```
   */
  public String getEjbName();


}
```

`EjbDescriptor`# ## # # ### ## EJB ### #### #### ### ###### #### ###. ### # ## ###### ###, ## #### ###### #### `BusinessInterfaceDescriptor`# ####. (EJB ##### ##### #### ##### ### # jndi ### ###)

A.1.3. `@EJB`, `@PersistenceContext` and `@Resource` resolution

The resolution of `@EJB`, `@PersistenceContext` and `@Resource` is delegated to the container. You must provide an implementation of `org.jboss.webbeans.ejb.spi.EjbResolver` which provides these operations. Web Beans passes in the `javax.inject.manager.InjectionPoint` the resolution is for, as well as the `NamingContext` in use for each resolution request.

A.1.4. Transaction Services

The Web Beans RI must delegate JTA activities to the container. The SPI provides a couple hooks to easily achieve this with the `TransactionServices` interface.

```
public interface TransactionServices
{
  /**
   * Possible status conditions for a transaction. This can be used by SPI
   * providers to keep track for which status an observer is used.
   */
  public static enum Status
  {
    ALL, SUCCESS, FAILURE
  }

  /**
   * Registers a synchronization object with the currently executing
   * transaction.
   *
   * @see javax.transaction.Synchronization
   * @param synchronizedObserver
   */
  public void registerSynchronization(Synchronization synchronizedObserver);

  /**
   * Queries the status of the current execution to see if a transaction is
```

```
    * currently active.
    *
    * @return true if a transaction is active
    */
   public boolean isTransactionActive();
}
```

The enumeration `Status` is a convenience for implementors to be able to keep track of whether a synchronization is supposed to notify an observer only when the transaction is successful, or after a failure, or regardless of the status of the transaction.

Any `javax.transaction.Synchronization` implementation may be passed to the `registerSynchronization()` method and the SPI implementation should immediately register the synchronization with the JTA transaction manager used for the EJBs.

To make it easier to determine whether or not a transaction is currently active for the requesting thread, the `isTransactionActive()` method can be used. The SPI implementation should query the same JTA transaction manager used for the EJBs.

A.1.5. The application context

Web Beans expects the Application Server or other container to provide the storage for each application's context. The `org.jboss.webbeans.context.api.BeanStore` should be implemented to provide an application scoped storage. You may find `org.jboss.webbeans.context.api.helpers.ConcurrentHashMapBeanStore` useful.

A.1.6. Bootstrap and shutdown

The `org.jboss.webbeans.bootstrap.api.Bootstrap` interface defines the bootstrap for Web Beans. To boot Web Beans, you must obtain an instance of `org.jboss.webbeans.bootstrap.WebBeansBootstrap` (which implements `Boostrap`), tell it about the SPIs in use, and then request the container start.

The bootstrap is split into phases, bootstrap initialization and bootstrap. Initialization will create a manager, and add the standard (specification defined) contexts. Bootstrap will discover EJBs, classes and XML; add beans defined using annotations; add beans defined using XML; and validate all beans.

To initialize the bootstrap you call `Bootstrap.initialize()`. Before calling `initialize()` you must have called `Bootstrap.setEjbResolver()`. If you are not using the built in `DefaultNamingContext` or the built in `DefaultResourceLoader` you must set these before calling `initialize()`.

Having called `initialize()`, the `Manager` can be obtained by calling `Bootstrap.getManager()`.

To boot the container you call `Bootstrap.boot()`. Before calling `boot()` you must have called `Bootstrap.setWebBeanDiscovery()`, `Bootstrap.setEjbDiscovery()` and `Bootstrap.setApplicationContext()`.

To shutdown the container you call `Bootstrap.shutdown()`. This allows the container to perform any cleanup operations needed.

A.1.7. JNDI

The Web Beans RI implements JNDI binding and lookup according to standards, however you may want to alter the binding and lookup (for example in an environment where JNDI isn't available). To do this, implement `org.jboss.webbeans.resources.spi.NamingContext`:

```
public interface NamingContext extends Serializable {

   /**
    * Typed JNDI lookup
    *
    * @param <T
> The type
    * @param name The JNDI name
    * @param expectedType The expected type
    * @return The object
    */
   public <T
> T lookup(String name, Class<? extends T
> expectedType);

   /**
    * Binds an item to JNDI
    *
    * @param name The key to bind under
    * @param value The item to bind
    */
   public void bind(String name, Object value);

}
```

A.1.8. ###

Web Beans RI# ### classpath## ### # #### #### ###. ##### RI# #### ## ##### ### classloader## #####, ## ##### #### ## # # ####. ### ##, `org.jboss.webbeans.spi.ResourceLoader`# ### # ####:

```
public interface ResourceLoader {

  /**
   * Creates a class from a given FQCN
   *
   * @param name The name of the clsas
   * @return The class
   */
  public Class<?> classForName(String name);

  /**
   * Gets a resource as a URL by name
   *
   * @param name The name of the resource
   * @return An URL to the resource
   */
  public URL getResource(String name);

  /**
   * Gets resources as URLs by name
   *
   * @param name The name of the resource
   * @return An iterable reference to the URLS
   */
  public Iterable<URL
> getResources(String name);

}
```

A.1.9. Servlet injection

Java EE / Servlet does not provide any hooks which can be used to provide injection into Servlets, so Web Beans provides an API to allow the container to request JSR-299 injection for a Servlet.

To be compliant with JSR-299, the container should request servlet injection for each newly instantiated servlet after the constructor returns and before the servlet is placed into service.

To perform injection on a servlet call `WebBeansManager.injectServlet()`. The manager can be obtained from `Bootstrap.getManager()`.

A.2. #####

API# ## ##### ### ### #### ## ## Web Beans RI# ##### #### ## # ## ## ### ####

Classloader ##

Web Beans RI# ### ####### ### #### #### ### ##, ### Web Beans ####### ## ##### ### #### ## ### ### ## ### ##### ###.

Servlet listener and filters

Web Beans# Servlet #### ### ##, Servlet# #### ### Web Beans ####### ## #### ## ### ### ## `org.jboss.webbeans.servlet.WebBeansListener`# Servlet #### #### ###.

If you are integrating the Web Beans into a JSF environment you must register `org.jboss.webbeans.servlet.ConversationPropagationFilter` as a Servlet listener, either automatically, or through user configuration, for each Web Beans application which uses JSF. This filter can be registered for all Servlet deployment safely.

Session Bean ####

Web Beans# EJB #### ##### # ## ###### beans# #### ### Web Beans ###### ### ## EJB ####### ## EJB ###### `org.jboss.webbeans.ejb.SessionBeanInterceptor`# #### ## ### ### ## #### ###.

##

You must register the `SessionBeanInterceptor` as the inner most interceptor in the stack for all EJBs.

`webbeans-ri.jar`

Web Beans# ###### ### #### #### ##### # ##, `webbeans-ri.jar`# ###### ## classloader# #### ###. ## ## classloader## ### # ####.