Faces Config Structure - Managed Beans

The "managed-bean" element represents a JavaBean, of a particular class, that will be dynamically instantiated at runtime.

General

Item	Description
Name	Stands for `<managed-bean-name>`. Represents the attribute name under which a managed bean will be searched for, as well as stored (unless the "managed-bean-scope" value is *none*).
Class	Stands for `<managed-bean-class>`. Represents the fully qualified class name of the Java class that will be used to instantiate a new instance if creation of the specified managed bean is requested.
Scope	Stands for `<managed-bean-scope>`. Represents the scope into which a newly created instance of the specified managed bean will be stored (unless the value is *none*).
Display Name	`<display-name>`: a short name describing the entity associated with the managed bean.
Description	`<description>`: Contains textual description of the managed bean.
Small icon / Large Icon	`<small-icon>` / `<large-icon>`. Contains the resource path to an (16x16)/(32x32) icon image. The image may be in either GIF or JPG format.

Properties bean

A *Properties Bean* can contain three kinds of properties: a *Map Entries Property*, a *List Entries Property*, and a *Valuable Property*. Each property is initialized with the corresponding element, however "map-entries" and "list-entries" elements serve also to initialize corresponding managed beans.

Valuable property

Item	Description
Name	Stands for `<property-name>`. Represents the JavaBeans property name under which the corresponding value may be stored.
Class	Stands for `<property-class>`. Represents the Java type of the value associated with this property name.
Display Name	`<display-name>`: a short name describing the entity associated with the property.
Description	`<description>`: contains textual description of the property.
Small icon / Large Icon	`<small-icon>` / `<large-icon>`. Contains the resource path to an (16x16)/(32x32) icon image. The image may be in either GIF or JPG format.
Value/Null Value	Stands for `<value>` / `<null-value>`. The *value* element is the String representation of a literal value to which a scalar managed property will be

set, or a value reference expression ("#{...}") that will be used to calculate the required value. It will be converted as specified for the actual property type. The "null-value" element indicates that the managed property in which we are nested will be explicitly set to null if our managed bean is automatically created. This is different from omitting the managed property element entirely, which will cause no property setter to be called for this property.

Map entries property

The "map-entries" element represents a set of key-entry pairs that will be added to the computed value of a managed property of type java.util.Map.

Item	Description
Key Class	Stands for `<key-class>`. Defines the Java type to which each *key* element in a set of "map-entry" elements will be converted to.
Value Class	Stands for `<value-class>`. Defines the Java type to which each *value* element's value will be converted to, prior to adding it to a "map-entries" map for a managed property that is a java.util.Map.
Key	Stands for `<key>`. String representation of a map key that will be stored in a managed property of type java.util.Map.
Value/Null Value	Stands for `<value>` / `<null-value>`. The *value* element is the String representation of a literal value to which a scalar managed property will be set, or a value reference expression ("#{...}") that will be used to calculate the required value. It will be converted as specified for the actual property type. The "null-value" element indicates that the managed property in which we are nested will be explicitly set to null if our managed bean is automatically created. This is different from omitting the managed property element entirely, which will cause no property setter to be called for this property.

List entries property

The "list-entries" element represents a set of initialization elements for a managed property that is a java.util.List or an array.

Item	Description
Value Class	Stands for `<value-class>`. Defines the Java type to which each *value* element's value will be converted to, prior to adding it to the "list-entries" list for a managed property that is a java.util.List. If omitted, "java.lang.String" is assumed.
Value/Null Value	Stands for `<value>` / `<null-value>`. The *value* element is the String representation of a literal value to which a scalar managed property will be set, or a value reference expression ("#{...}") that will be used to calculate the required value. It will be converted as specified for the actual property type. The "null-value" element indicates that the managed property in which we are nested will be explicitly set to null if our managed bean is automatically created. This is different from omitting the managed property element entirely, which will cause no property setter to be called for this property.

Map entries bean \ list entries bean

For the options description see Map Entries Property \ List Entries Property, because the element that is used to initialize both map entries property and map entries bean (list entries property and list entries bean) is the same.

See Also

Concepts:

- JSF

Procedures:

- Enabling JSF Support
- Creating and Editing Faces Configuration

Web Resources:

- Developer Community