EJB3 规范简化版

正式版

译者：卫建军

2008/1/6

译者序

Java 是当前 IT 领域中比较流行的技术之一。J2EE 是当前比较流行的企业级应用架构。本人一直致力于 J2EE 架构的学习和研究，但是总是对英文文档有不可言语的恐惧。我想很多 J2EE 爱好者和我有同样的感觉。这样就影响了我们深入学习 J2EE 原始规范的兴趣。但是 J2EE 原始的规范文档对我们深入理解 J2EE 有很大的帮助，因为它阐述了规范的来龙去脉，以及违反了规范会造成什么样的影响。了解了这些缘由和影响，会使我们对 J2EE 架构有更深层次的理解。这也是我翻译该规范的动力所在。

由于本人的英语水平有限，翻译中难免会出现错误和拗口之处，请大家多多指教。

这次主要翻译的规范有《EJB3 规范简化版》、《J2EE5.0 规范》、《EJB 核心规范》、《EJB3 持久化规范》和《JMS1.1 规范》。希望对大家有所帮助。

目录

1 注释符 4

1.1 特定类型 bean 的注释符 4

1.1.1 无状态会话 bean 4

1.1.2 有状态会话 bean 4

1.1.3 消息驱动 bean 6

1.2 指定 Local 或 Remote 接口的注释符 7

1.3 支持 EJB2.1 和早期客户端实图的注释符 7

1.4 TransactionManagement 8

1.5 务属性 8

1.6 拦截器和生命周期回调 9

1.7 Timeout 11

1.8 异常 11

1.9 安全和方法授权 11

1.9.1 安全角色引用 11

1.9.2 方法授权 12

1.9.3 PermitAll 12

1.9.4 DenyAll 12

1.9.5 RunAs 13

1.10 引用 EJB 13

1.11 引用资源 14

1 注释符

1.1 特定类型 bean 的注释符

1.1.1 无状态会话 bean

Stateless 用于标明企业 bean 是一个无状态会话 bean。Stateless 注释符用于 bean 的 class。

```
@Target(TYPE) @Retention(RUNTIME)
public @interface Stateless {
String name() default "";
String mappedName() default "";
String description() default "";
}
```

元素 name 缺省是 bean 类的简称（不带包名）。不管事是显式指定的还是缺省的名字，这个名字在 ejb-jar 中必须是唯一的。

元素 mappedName 是特定产品的名字，这个名字用于映射会话 bean。使用映射名字的应用是不可移植的。

1.1.2 有状态会话 bean

Stateful 注释符指明企业 bean 是有状态会话 bean。Stateful 注释符应用于 bean 的 class 上。

```
@Target(TYPE) @Retention(RUNTIME)
public @interface Stateful {
String name() default "";
String mappedName() default "";
String description() default "";
}
```

元素 name 缺省是 bean 类的简称（不带包名）。不管事是显式指定的还是缺省的名字，这个名字在 ejb-jar 中必须是唯一的。

元素 mappedName 是特定产品的名字，这个名字用于映射会话 bean。使用映射名字的应用是不可移植的。

1.1.2.1 有状态会话 bean 的 Init 注释符

Init 注释符用于指定 bean 类中与 EJB2.1 EJBHome 和/或 EJBLocalHome 的 create<METHOD>等价的方法，用于适配 EJB2.1 EJBHome 和/或 EJBLocalHome。Init 方法的返回类型必须是 void，而且它的参数类型必须和引用的 create<METHOD>方法的参数完全一致。

```
@Target(METHOD) @Retention(RUNTIME)
public @interface Init{
String value() default "";
}
```

当 Init 注释符用于适配有多个 create<METHOD>方法的有状态会话 bean 的 home 接口时，必须指定 value 元素。它指定了适配的 home 接口的对应的 create<MEHTOD>方法的名字。

只有当有状态会话 bean 有 RemoteHome 或 LocalHome 接口时才需要指定 Init 方法。如果有产生混淆的可能性，则必须指定适配的 create<METHOD>方法的名字。

1.1.2.2 有状态会话 bean 的 Remove 注释符

Remove 注释符用于指定有状态会话 bean 的清除方法。这个方法完成后容器会销毁这个 bean，在调用这个方法之前会调用 bean 的 PreDestroy 方法。retainIfException 元素可以防止这个方法由于应用异常而异常终止。

```
@Target(METHOD) @Retention(RUNTIME)
public @interface Remove{
```

boolean retainIfException() default false;

}

1.1.3 消息驱动 bean

MessageDriven 注释符指明一个企业 bean 是一个消息驱动 bean。这个注释符应用到 bean 的 class。

元素 name 缺省是 bean 类的简称（不带包名）。不管事是显式指定的还是缺省的名字，这个名字在 ejb-jar 中必须是唯一的。

元素 mappedName 是特定产品的名字，这个名字用于映射会话 bean。使用映射名字的应用是不可移植的。

元素 messageListenerInterface 指定 bean 的消息监听器接口。如果 bean 没有实现消息监听期接口或实现了除 java.io.Serializable，java.io.Externalizable 和 javax.ejb 包内的接口外的多个接口，则必须指定 messageListenerInterface。

```
@Target(TYPE) @Retention(RUNTIME)
public @interface MessageDriven {
String name() default "";
Class messageListenerInterface() default Object.class;
ActivationConfigProperty[] activationConfig() default {};
String mappedName() default "";
String description() default "";
}
@Target({}) @Retention(RUNTIME)
public @interface ActivationConfigProperty {
   String propertyName();
   String propertyValue();
}
```

1.2 指定 Local 或 Remote 接口的注释符

Remote 和 Local 注释符只用于会话 bean 和他们的接口。

Remote 注释符用于会话 bean 或远程业务接口。

Local 注释符用于会话 bean 或本地业务接口。

当 bean 实现了除了 java.io.Serializable，java.io.Externalizable 一家 javax.ejb 包的接口外多个接口时，才需要使用 Local。

只有在 bean 上使用注释符的时候才指定元素 value。只有当 bean 实现了除了 java.io.Serializable，java.io.Externalizable 和 javax.ejb 包内的接口外多个接口时才需要指定 value。

```
@Target(TYPE) @Retention(RUNTIME)
public @interface Remote {
Class[]value()default{};//listofremotebusinessinterfaces
}
@Target(TYPE) @Retention(RUNTIME)
public @interface Local {
Class[] value() default {}; // list of local business interfaces
}
```

1.3 支持 EJB2.1 和早期客户端实图的注释符

RemoteHome 和 LocalHome 只可以应用到会话 bean。

这些注释符只应用于 EJB3.0，用于适配 EJB2.1。也可以用于使用 EJB2.1 API 的 bean。

```
@Target(TYPE) @Retention(RUNTIME)
public @interface RemoteHome {
   Class value();      // home interface
}
@Target(TYPE) @Retention(RUNTIME)
```

```
public @interface LocalHome {
   Class value();      // local home interface
}
```

1.4 TransactionManagement

TransactionManagement 注释符指定会话 bean 或消息驱动 bean 的事务管理分隔类型。如果会话 bean 或消息驱动 bean 没有指定 TransactionManagement，那么使用容器管理的事务分隔。

```
@Target(TYPE) @Retention(RUNTIME)
public @interface TransactionManagement {
TransactionManagementType value()
default TransactionManagementType.CONTAINER;
}
```

枚举 TransactionManagementType 用于指定使用容器管理的事务管理还是 bean 管理的事务管理。

```
public enum TransactionManagementType {
   CONTAINER,
   BEAN
}
```

1.5 务属性

TransactionAttribute 注释符指定容器是否在事务上下文中调用业务方法。事务属性的语义在“EJB Core 协议和需求”的“支持事务”一章中描述。

TransactionAttribute 只能在使用容器管理的事务分隔时指定。这个注释符可以指定在 bean 的 class 和/或类的业务接口方法上。在 bean 的 class 上指定 TransactionAttribute 意味着事务属性应用到所有的业务接口方法。指定在方法上则意味着只应用到该方法上。如果两个地方都指定了 TransactionAttribute，且属

性值不同，那么方法上的 TransactionAttribute 覆盖类上的事务属性。

TransactionAttribute 注释符的值由枚举 TransactionAttributeType 指定。

如果没有指定 TransactionAttribute，并且 bean 使用容器管理事务分隔，那么使用 REQUIRED 事务属性。

```
public enum TransactionAttributeType {
    MANDATORY,
    REQUIRED,
    REQUIRES_NEW,
    SUPPORTS,
    NOT_SUPPORTED,
    NEVER
}
@Target({METHOD, TYPE}) @Retention(RUNTIME)
public @interface TransactionAttribute {
   TransactionAttributeType value()
default TransactionAttributeType.REQUIRED;
}
```

1.6 拦截器和生命周期回调

Javax.interceptor.Interceptors 注释符用于为 bean 指定一个或多个拦截器类。Interceptiors 注释符应用 bean 的 class 或 bean 的业务方法上。

```
package javax.interceptor;
@Target({TYPE, METHOD}) @Retention(RUNTIME)
public @interface Interceptors {
Class[] value();
}
```

Javax.interceptor.AroundInvoke 注释符用于指定拦截器方法。

```
package javax.interceptor;
```

@Target({METHOD}) @Retention(RUNTIME)

public @interface AroundInvoke {}

当在 bean 上使用 javax.interceptor.ExcludeDefaultInterceptors 注释符时，是指为 bean 的所有业务方法排除缺省的拦截器调用。当在业务方法上指定时，只是为这个方法排除缺省的拦截器。

package javax.interceptor;

@Target({TYPE, METHOD}) @Retention(RUNTIME)

public @interface ExcludeDefaultInterceptors {}

javax.interceptor.ExcludeClassInterceptors 注释符为方法排除类级别上的拦截器调用（但不是缺省拦截器）。

package javax.interceptor;

@Target({METHOD}) @Retention(RUNTIME)

public @interface ExcludeClassInterceptors {}

javax.annotation.PostConstruct ， javax.annotation.PreDestroy 和 javax.ejb.PostActivate 和 javax.ejb.PrePassivate 注释符指定生命周期回调方法。

package javax.annotation;

@Target({METHOD}) @Retention(RUNTIME)

public @interface PostConstruct {}

package javax.annotation;

@Target({METHOD}) @Retention(RUNTIME)

public @interface PreDestroy {}

package javax.ejb;

@Target({METHOD}) @Retention(RUNTIME)

public @interface PostActivate {}

package javax.ejb;

@Target({METHOD}) @Retention(RUNTIME)

public @interface PrePassivate {}

1.7 Timeout

Timeout 注释符用于指定企业 bean 的超时方法。

```
@Target({METHOD}) @Retention(RUNTIME)
public @interface Timeout {}
```

1.8 异常

ApplicationException 注释符用于指定一个应该被直接报告给客户端的应用异常（例如，解压）。ApplicationException 既可以是可被检查的也可以是不被检查的异常。rollback 元素用于指明容器是否必须在异常抛出时回滚事务。

```
@Target(TYPE) @Retention(RUNTIME)
public @interface ApplicationException {
boolean rollback() default false;
}
```

1.9 安全和方法授权

下面所述的与安全相关的注释符在 javax.annotation.security 包内。它们在【JSR-250:java 平台的共用注释符。http://jcp.org/en/jsr/detail?id=250】中详细描述，放在这里只是一个参考。

1.9.1 安全角色引用

DeclareRoles 注释符用于生命在企业 bean 的代码中引用的安全角色。

```
package javax.annotation.security;
@Target({TYPE}) @Retention(RUNTIME)
public @interface DeclareRoles {
String[] value();
}
```

1.9.2 方法授权

RolesAllowed 注释符指定安全角色可以调用的 bean 方法。RolesAllowed 注释符的值是一个安全角色名称的列表。

这个注释符可以指定在 bean 的类上，也可以指定属于业务接口方法的方法上。在 bean 的类上指定 RolesAllowed 注释符意味着适用于这个类上所有接口方法。如果在方法上指定则只使用于该方法。如果在类上和方法上都指定了 RolesAllowed，则方法上的值覆盖类上的值。如果在类上使用 PermitAll 注释符，并且在单个方法上也指定了 RolesAllowed，那么 RolesAllowed 的值覆盖类上的值。

```
package javax.annotation.security;
@Target({TYPE, METHOD}) @Retention(RUNTIME)
public @interface RolesAllowed {
   String[] value();
   }
```

1.9.3 PermitAll

PermitAll 注释符指明所有的角色都可以调用指定的方法——例如，指定的方法事“unchecked”。这个注释符可以指定在 bean 类上，也可以指定在 bean 的业务方法上。在类上指定 PermitAll 意味着应用到这个类的所有可应用的业务方法上。在方法上指定，则意味着只适用于该方法，而且会覆盖类一级的任何设置。

```
package javax.annotation.security;
@Target ({TYPE, METHOD}) @Retention(RUNTIME)
public @interface PermitAll {}
```

1.9.4 DenyAll

DenyAll 注释符指明任何角色都不允许调用指定的方法——例如，从执行中排除指定的方法。

```
package javax.annotation.security;
@Target (METHOD) @Retention(RUNTIME)
public @interface DenyAll {}
```

1.9.5 RunAs

RunAs 注释符用于指定 bean 的 run-as 属性。这个注释符应用于 bean 类。它的值事安全角色的名字。

```
package javax.annotation.security;
@Target(TYPE) @Retention(RUNTIME)
public @interface RunAs {
    String value();
  }
```

1.10 引用 EJB

EJB 注释符指定一个对 EJB 业务接口或 home 接口的引用。

元素 name 指的是在环境中用于查找的名字。元素 beanInterface 是引用的接口类型——业务接口或 home 接口。

元素 beanName 引用 Stateful 或 Stateless 注释符的 name 的值，不管是缺省的还是显式指定的（或者事 ejb-name 元素的值）。如果 ejb-jar 中的会话 bean 实现了同一个接口，那么 beanName 元素可以用来削除歧义。为了引用在同一个应用的其他 ejb-jar 文件，可以用 ejb-jar 包的名字+“#”+bean 名字的方式来引用。路径名相对于要引用其他包内 bean 的 bean 所在的 jar 文件。

元素 mappedName 是特定产品的名字，这个名字用于映射会话 bean。使用映射名字的应用是不可移植的。

```
@Target({TYPE, METHOD, FIELD}) @Retention(RUNTIME)
public @interface EJB {
String name() default "";
```

```
Class beanInterface() default Object.class;
String beanName() default "";
String mappedName() default "";
String description() default "";
}
@Target(TYPE) @Retention(RUNTIME)
public @interface EJBs {
   EJB[] value();
}
```

1.11 引用资源

Resource 注释符用于表达对外部资源的依赖。属性 name 指向在环境中用于查找的名称；type 是资源管理器连接工厂类型。authenticationType 指明容器或 bean 是否执行授权。shareable 元素指定自由管理器连接是否共享。mpapedName 元素是资源在特定产品下的映射名称。使用映射名称的应用是不可移植的。

```
package javax.annotation;
@Target({TYPE, METHOD, FIELD}) @Retention(RUNTIME)
public @interface Resource {
public enum AuthenticationType {
CONTAINER,
APPLICATION
}
String name() default "";
Class type() default Object.class;
AuthenticationType authenticationType()
default AuthenticationType.CONTAINER;
boolean shareable() default true;
String mappedName() default "";
```

```
String description() default "";
}
package javax.annotation;
@Target(TYPE) @Retention(RUNTIME)
public @interface Resources {
   Resource[] value();
}
```