

# Appendix

Version 5.0.17.RELEASE

# Chapter 1. XML Schemas

This part of the appendix lists XML schemas related to integration technologies.

## 1.1. The jee schema

The `jee` tags deal with Java EE (Java Enterprise Edition)-related configuration issues, such as looking up a JNDI object and defining EJB references.

To use the tags in the `jee` schema, you need to have the following preamble at the top of your Spring XML configuration file; the text in the following snippet references the correct schema so that the tags in the `jee` namespace are available to you.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       <em>xmlns:jee="http://www.springframework.org/schema/jee"</em>
       xsi:schemaLocation="
           http://www.springframework.org/schema/beans
           https://www.springframework.org/schema/beans/spring-beans.xsd
           <em>http://www.springframework.org/schema/jee
           https://www.springframework.org/schema/jee/spring-jee.xsd"</em>> <!-- bean definitions
       here -->

</beans>
```

### 1.1.1. <jee:jndi-lookup/> (simple)

Before...

```
<bean id="<strong>dataSource</strong>"
      class="org.springframework.jndi.JndiObjectFactoryBean">
    <property name="jndiName" value="jdbc/MyDataSource"/>
</bean>
<bean id="userDao" class="com.foo.JdbcUserDao">
    <!-- Spring will do the cast automatically (as usual) -->
    <property name="dataSource" ref="<strong>dataSource</strong>"/>
</bean>
```

After...

```

<jee:jndi-lookup id="dataSource" jndi-name="jdbc/MyDataSource"/>

<bean id="userDao" class="com.foo.JdbcUserDao">
  <!-- Spring will do the cast automatically (as usual) -->
  <property name="dataSource" ref="dataSource"/>
</bean>

```

### 1.1.2. <jee:jndi-lookup/> (with single JNDI environment setting)

Before...

```

<bean id="simple" class="org.springframework.jndi.JndiObjectFactoryBean">
  <property name="jndiName" value="jdbc/MyDataSource"/>
  <property name="jndiEnvironment">
    <props>
      <prop key="foo">bar</prop>
    </props>
  </property>
</bean>

```

After...

```

<jee:jndi-lookup id="simple" jndi-name="jdbc/MyDataSource">
  <jee:environment>foo=bar</jee:environment>
</jee:jndi-lookup>

```

### 1.1.3. <jee:jndi-lookup/> (with multiple JNDI environment settings)

Before...

```

<bean id="simple" class="org.springframework.jndi.JndiObjectFactoryBean">
  <property name="jndiName" value="jdbc/MyDataSource"/>
  <property name="jndiEnvironment">
    <props>
      <prop key="foo">bar</prop>
      <prop key="ping">pong</prop>
    </props>
  </property>
</bean>

```

After...

```

<jee:jndi-lookup id="simple" jndi-name="jdbc/MyDataSource">
  <!-- newline-separated, key-value pairs for the environment (standard Properties
format) -->
  <jee:environment>
    foo=bar
    ping=pong
  </jee:environment>
</jee:jndi-lookup>

```

#### 1.1.4. <jee:jndi-lookup/> (complex)

Before...

```

<bean id="simple" class="org.springframework.jndi.JndiObjectFactoryBean">
  <property name="jndiName" value="jdbc/MyDataSource"/>
  <property name="cache" value="true"/>
  <property name="resourceRef" value="true"/>
  <property name="lookupOnStartup" value="false"/>
  <property name="expectedType" value="com.myapp.DefaultFoo"/>
  <property name="proxyInterface" value="com.myapp.Foo"/>
</bean>

```

After...

```

<jee:jndi-lookup id="simple"
  jndi-name="jdbc/MyDataSource"
  cache="true"
  resource-ref="true"
  lookup-on-startup="false"
  expected-type="com.myapp.DefaultFoo"
  proxy-interface="com.myapp.Foo"/>

```

#### 1.1.5. <jee:local-slsb/> (simple)

The <jee:local-slsb/> tag configures a reference to an EJB Stateless SessionBean.

Before...

```

<bean id="simple"
  class="org.springframework.ejb.access.LocalStatelessSessionProxyFactoryBean">
  <property name="jndiName" value="ejb/RentalServiceBean"/>
  <property name="businessInterface" value="com.foo.service.RentalService"/>
</bean>

```

After...

```
<jee:local-slsb id="simpleSlsb" jndi-name="ejb/RentalServiceBean"
    business-interface="com.foo.service.RentalService"/>
```

### 1.1.6. <jee:local-slsb/> (complex)

```
<bean id="complexLocalEjb"
    class="org.springframework.ejb.access.LocalStatelessSessionProxyFactoryBean">
    <property name="jndiName" value="ejb/RentalServiceBean"/>
    <property name="businessInterface" value="com.foo.service.RentalService"/>
    <property name="cacheHome" value="true"/>
    <property name="lookupHomeOnStartup" value="true"/>
    <property name="resourceRef" value="true"/>
</bean>
```

After...

```
<jee:local-slsb id="complexLocalEjb"
    jndi-name="ejb/RentalServiceBean"
    business-interface="com.foo.service.RentalService"
    cache-home="true"
    lookup-home-on-startup="true"
    resource-ref="true">
```

### 1.1.7. <jee:remote-slsb/>

The <jee:remote-slsb/> tag configures a reference to a **remote** EJB Stateless SessionBean.

Before...

```
<bean id="complexRemoteEjb"
    class=
"org.springframework.ejb.access.SimpleRemoteStatelessSessionProxyFactoryBean">
    <property name="jndiName" value="ejb/MyRemoteBean"/>
    <property name="businessInterface" value="com.foo.service.RentalService"/>
    <property name="cacheHome" value="true"/>
    <property name="lookupHomeOnStartup" value="true"/>
    <property name="resourceRef" value="true"/>
    <property name="homeInterface" value="com.foo.service.RentalService"/>
    <property name="refreshHomeOnConnectFailure" value="true"/>
</bean>
```

After...

```
<jee:remote-slsb id="complexRemoteEjb"
    jndi-name="ejb/MyRemoteBean"
    business-interface="com.foo.service.RentalService"
    cache-home="true"
    lookup-home-on-startup="true"
    resource-ref="true"
    home-interface="com.foo.service.RentalService"
    refresh-home-on-connect-failure="true">
```

## 1.2. The `.jms` schema

The `.jms` tags deal with configuring JMS-related beans such as Spring's `MessageListenerContainers`. These tags are detailed in the section of the [JMS chapter](#) entitled [JMS namespace support](#). Please do consult that chapter for full details on this support and the `.jms` tags themselves.

In the interest of completeness, to use the tags in the `.jms` schema, you need to have the following preamble at the top of your Spring XML configuration file; the text in the following snippet references the correct schema so that the tags in the `.jms` namespace are available to you.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    <em>xmlns:jms="http://www.springframework.org/schema/jms"</em>
    xsi:schemaLocation="
        http://www.springframework.org/schema/beans
        https://www.springframework.org/schema/beans/spring-beans.xsd
        <em>http://www.springframework.org/schema/jms
        https://www.springframework.org/schema/jms/spring-jms.xsd"</em>> <!-- bean definitions
    here -->

</beans>
```

## 1.3. `<context:mbean-export/>`

This element is detailed in [Configuring annotation based MBean export](#).

## 1.4. The `cache` schema

The `cache` tags can be used to enable support for Spring's `@CacheEvict`, `@CachePut` and `@Caching` annotations. It also supports declarative XML-based caching. See [Enable caching annotations](#) and [Declarative XML-based caching](#) for details.

To use the tags in the `cache` schema, you need to have the following preamble at the top of your Spring XML configuration file; the text in the following snippet references the correct schema so that the tags in the `cache` namespace are available to you.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  <em>xmlns:cache="http://www.springframework.org/schema/cache"</em>
  xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    https://www.springframework.org/schema/beans/spring-beans.xsd
    <em>http://www.springframework.org/schema/cache
    https://www.springframework.org/schema/cache/spring-cache.xsd"</em>> <!-- bean
    definitions here -->

</beans>
```