



Java程序设计

第05-03讲 部署描述符文件

Java课程组



知识回顾/本讲先行知识



- 安全 (security)





本讲教学目标



- 熟悉部署描述符文件中各个元素的作用与使用方法





本讲内容

- 部署描述符文件的作用
- 常用元素的声明与配置
- JavaEE元素



部署描述符文件(web.xml)



- web.xml叫做部署描述符文件
 - 是在Servlet规范中定义的，是web应用的配置文件。
- 作用：
 - 描述了容器运行程序所需要的信息，提供站点的配置设定



部署描述符文件(web.xml)



- web.xml文件是XML文档
 - 必须以XML声明开头，指出XML版本及字符编码
 - 顶层(根)元素为<web-app>,元素名大小写敏感
 - web-App和WEB-APP都是不合法的，web-app必须用小写

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/java
ee http://java.sun.com/xml/ns/javaee/web-
app_3_0.xsd" id="WebApp_ID" version="3.0">
</web-app>
```



部署描述符文件(web.xml)



- web.xml文件是XML文档
 - 必须以XML声明开头，指出XML版本及字符编码
 - 顶层(根)元素为<web-app>,元素名大小写敏感
 - 各元素对出现在其他元素中的次序敏感
 - 可省略某些可选元素，但不能将其放在不正确的位置



web.xml定义:<web-app>元素



```
<?xml version="1.0" encoding="UTF-8"?>
<web-app
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/java
ee http://java.sun.com/xml/ns/javaee/web-
app_3_0.xsd" id="WebApp_ID" version="3.0">
</web-app>
```

- xsd文档的作用是定义XML文档的合法构建模块，类似dtd
- 学习XML Schema(.xsd)教程
 - <http://www.w3school.com.cn/schema/index.asp>
 - http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd



web.xml定义



| 元素 | 含义 |
|-----------------|---|
| icon | 指出IDE和GUI工具可用来代表Web应用的一个和两个图像文件的位置。 |
| display-name | display-name 元素提供GUI工具可能会用来标记这个特定应用的一个名称。 |
| description | 提供关于 Web 应用程序的描述性文本。 |
| context-param | 用来设定web应用的环境参数。 |
| filter | 过滤器元素将一个名字与一个实现javax.servlet.Filter接口的类相关联。 |
| filter-mapping | 一旦命名了一个过滤器,就利用filter-mapping元素把它与一个或多个servlet或JSP页面相关联。 |
| listener | 指出事件监听程序类 |
| servlet | servlet元素用于声明一个servlet |
| servlet-mapping | servlet-mapping定义了servlet和URL模式之间的映射 |



web.xml定义



| 元素 | 含义 |
|---------------------|---------------------------|
| session-config | 元素定义了该Web应用程序的会话参数 |
| mime-mapping | 定义了扩展名和MIME类型之间的映射 |
| welcome-file-list | 包含了一个有序的欢迎文件列表 |
| error-page | 指定错误页面 |
| jsp-config | 用来提供Web应用程序中的JSP文件的全局配置信息 |
| security-constraint | 用于关联安全约束和一个或多个Web资源集合 |
| login-config | 用于配置应该使用的认证方法 |
| security-role | 定义了一个安全角色 |



web.xml定义



| 元素 | 含义 |
|------------------|-------------------------------|
| resource-ref | 包含了部署组件对外部资源的引用声明 |
| resource-env-ref | 包含了部署组件和对部署组件环境中的资源有关的管理对象的引用 |
| distributable | 支持集群的服务器可安全的在多个服务器上分布Web应用 |
| env-entry | 声明了一个应用程序的环境入口 |
| ejb-ref | 声明了一个对企业bean的home引用 |
| ejb-local-ref | 声明了对企业bean的本地home引用 |





本讲内容

- 部署描述符的作用
- 常用元素的声明与配置
- JavaEE元素



icon



- 包含small-icon和large-icon元素，为大型和小型GIF或JPEG图标图片指定文件名
- 用于在GUI工具中表示父元素。
- 范例

```
<icon>  
  <small-icon>/images/small.gif</small-icon>  
  <large-icon>/images/large.gif</large-icon>  
</icon>
```



display-name



- 定义应用的名称。
- 范例

```
<display-name>myFirstApp</display-name>
```



description



- 对应用做出描述
- 范例

```
<description>测试应用V1.0</description>
```



context-param



- context-param 应用范围内初始化参数
 - `<param-name>` 参数名称 `</param-name>`
 - `<param-value>` 值 `</param-value>`
- 范例

```
<context-param>  
  <param-name>param_name</param-name>  
  <param-value> param_value</param-value>  
</context-param>
```



filter



- 声明了Web应用程序中的过滤器
 - filter-name元素是过滤器的逻辑名称。
 - filter-class是过滤器的完全限定类名。
 - init-param元素包含的名-值对作为此过滤器的初始化参数。

```
<filter>  
  <filter-name>authorizefilter</filter-name>  
  <filter-class>onest.web.filter.AuthorizeFilter</filter-class>  
</filter>
```



filter-mapping



- 容器使用filter-mapping决定哪个过滤器以什么样的顺序应用到请求。

```
<filter>
  <filter-name>authorizefilter</filter-name>
  <filter-class>onest.web.filter.AuthorizeFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>authorizefilter</filter-name>
  <url-pattern>/admin/* </url-pattern>
</filter-mapping>
```



listener



- 指定事件监听程序
 - listener-class声明应用程序中的一个类必须注册为Web应用程序监听器bean。它的值是监听器类的完全限定类名。

```
<listener>
  <listener-class>
    onest.dev.ServletcontextListenerDemo
  </listener-class>
</listener>
```



servlet



- servlet元素用于声明一个servlet
 - jsp-file元素命名JSP页面,包含到以 “/” 开头的Web应用程序中一个JSP文件的完全路径。
 - 所指的JSP文件可存放于WEB-INF目录
 - servlet-name元素包含了servlet的规范名称。
 - servlet-class包含了servlet的完全限定类名。
 - load-on-startup元素表示该servlet应该在Web应用程序启动时加载。



servlet



```
<servlet>
  <servlet-name>HelloServlet</servlet-name>
  <jsp-file>/hello.jsp</jsp-file>
  <load-on-startup>1</load-on-startup>
</servlet>
```

```
<servlet>
  <servlet-name>HelloServlet</servlet-name>
  <servlet-class>onest.HelloServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
```



servlet-mapping



- 定义了servlet和URL模式之间的映射。
 - servlet-name:定义Servlet的名称

```
<servlet>
  <load-on-startup>1</load-on-startup>
  <servlet-name>HelloServlet</servlet-name>
  <servlet-class>onest.HelloServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>HelloServlet</servlet-name>
  <url-pattern>/hello</url-pattern>
</servlet-mapping>
```



session-config



- 控制会话超时
- 如果某个会话在一定的时间内未被访问，服务器可以把它扔掉以节约内存。
- session-config和session-timeout元素给出一个适用于所有服务器的明确的超时期
- session-timeout元素的值的单位为分钟

```
<session-config>  
  <session-timeout> 3 </session-timeout>  
</session-config>
```



mime-mapping



- 关联文件与MIME类型
- 服务器一般都具有一种让Web站点管理员将文件扩展名与媒体相关联的方法。如：
 - 自动给予名为mom.jpg的文件一个image/jpeg的MIME类型

```
<mime-mapping>  
  <extension>foo</extension>  
  <mime-type>application/x-fubar</mime-type>  
</mime-mapping>
```



welcome-file-list



- 指定欢迎页
- 如果一个URL给出了一个目录名但未给出文件名，服务器应该使用指定的欢迎页面

```
<welcome-file-list>
  <welcome-file>index.html</welcome-file>
  <welcome-file>index.htm</welcome-file>
  <welcome-file>index.jsp</welcome-file>
  <welcome-file>default.html</welcome-file>
  <welcome-file>default.htm</welcome-file>
  <welcome-file>default.jsp</welcome-file>
</welcome-file-list>
```



error-page



- error-page元素来指定错误页面

```
<error-page>
```

```
  <error-code>500</error-code>
```

```
  <location>/500.jsp</location>
```

```
</error-page>
```

```
<error-page>
```

```
  <error-code>404</error-code>
```

```
  <location>/404.jsp</location>
```

```
</error-page>
```

```
<error-page>
```

```
  <exception-type>java.io.IOException</exception-type>
```

```
  <location>/error.jsp</location>
```

```
</error-page>
```



jsp-config



- jsp-config用来提供Web应用程序中的JSP文件的全局配置信息。
 - taglib元素可用来为Web应用程序中的JSP页面使用的标签库提供信息。使用<taglib>给TLD文件起一个别名

```
<jsp-config>
  <taglib>
    <taglib-uri>mytag</taglib-uri>
    <taglib-location>/WEB-INF/mytag.tld</taglib-
location>
  </taglib>
</jsp-config>
```





- 利用web.xml中相关元素为服务器内建功能提供安全性：
 - 指定验证的方法： `<login-config>`
 - 限制对web资源的访问：
`<security-constraint>`
 - 分配角色名： `<security-role>`



login-config



- 配置认证方法
 - auth-method为Web应用程序配置认证机制
 - 该元素的内容必须是BASIC、DIGEST、FORM、CLIENT-CERT、或提供商指定的认证模式
 - realm-name表示为Web应用程序选择用于认证模式的领域名。

```
<login-config>  
  <auth-method>BASIC</auth-method>  
</login-config>
```



security-constraint



- 使用<security-constraint>元素配置安全约束
 - <display-name>:可选, 指定安全约束

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>test2</web-resource-name>
    <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>tomcat1</role-name>
  </auth-constraint>
</security-constraint>
```



security-role



- 定义了一个安全角色。子元素role-name

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>test2</web-resource-name>
    <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>tomcat1</role-name>
  </auth-constraint>
</security-constraint>
<security-role>
  <role-name>tomcat1</role-name>
</security-role>
```



本讲内容

- 部署描述符的作用
- 常用元素的声明与配置
- JavaEE元素





- `<distributable/>`：支持集群的服务器可安全的在多个服务器上分布Web应用
- `<resource-env-ref>`：声明一个与某个资源有关的管理对象
- `<resource-ref>`：声明一个资源引用



distributable



- 在应用的web.xml加<distributable/> 即可

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
id="WebApp_ID" version="3.0">
<distributable/>
</web-app>
```



resource-env-ref



- resource-env-ref有两个子元素：
 - resource-env-ref-name定义资源名
 - 资源的名称 相对于java:comp/env

```
<resource-env-ref>
  <resource-env-ref-name>jdbc/mssql</resource-env-ref-
name>
  <resource-env-ref-type>javax.sql.DataSource</resource-
env-ref-type>
</resource-env-ref>
```



resource-ref



- resource-ref元素包括五个子元素，利用JNDI取得应用可利用资源。
 - description: 资源说明
 - res-ref-name: 资源名称

```
<resource-ref>  
  <description>JNDI JDBC DataSource</description>  
  <res-ref-name>jdbc/data</res-ref-name>  
  <res-type>javax.sql.DataSource</res-type>  
  <res-auth>Container</res-auth>  
</resource-ref>
```



小结



- 部署描述符的作用
- 常用元素的声明与配置
- JavaEE元素





本讲结束

- 谢谢大家

