Register | Login | Logout | Contact Us

Java-Success.com

Industrial strength Java/JEE Career Companion to open more doors

search here … Go

Home | Java FAQs | 600+ Java Q&As | Career | Tutorials | Member | Why?

Can u Debug? | Java 8 ready? | Top X | Productivity Tools | Judging Experience?

Home › member-paid › 08: Web.xml interview questions and answers

08: Web.xml interview questions and answers

Posted on May 14, 2015 by Arulkumaran Kumaraswamipillai

Since **servlet 3.0 specification**, the web.xml file in a web application is **optional**. The servlet 3.0 API introduced annotations to register servlets. Learn more at Servlet Interview Questions and Answers.

Q1. What is a web.xml file?
A1. The **$ROOT/WEB-INF/web.xml** file is the **Web Application Deployment Descriptor** of your application. This file is an XML document that defines everything about your application that a **web container** in the application server needs to know except the context path, which is configured via container specific deployment descriptor file. For example the “context path” for jboss web container is define as shown below in **jboss-web.xml** file

```
<!DOCTYPE jboss-web PUBLIC "-//JBoss//DTD Web Ap
<jboss-web>
```

600+ Full Stack Java/JEE Interview Q&As ♥Free ♦FAQs

open all | close all
Ice Breaker Interview
Core Java Interview (
JEE Interview Q&A (3
JEE Overview (2)
Web basics (8)
01: ♦ 12 Web ba
02: HTTP basics
03: Servlet interv
04: JSP overviev
05: Web patterns
06: ♦ MVC0, MV
07: When to use
08: Web.xml inte
WebService (11)
JPA (2)
JTA (1)
JDBC (4)
JMS (5)
JMX (3)
JNDI and LDAP (1)
Pressed for time? Jav

```
    <class-loading java2ClassLoadingCompliance="
        <loader-repository>
            org:archive=ecm-enabler-endpoint.war
            <loader-repository-config>java2Paren
        </loader-repository>
    </class-loading>
    <context-root>/my-enterprise-ws</context-roo
</jboss-web>
```

“/my-enterprise-ws” is the context root. So, you will access your application something like

```
http://localhost:8180/my-enterprise-ws/blah
```

Q2. Can you list some of the entries a typical web.xml file will have?
Q2.

1) Servlets: For a Java servlet to be accessible from a browser, you must tell the servlet container what servlets to deploy, and what URL’s to map the servlets to. This is done in the web.xml file of your Java web application.

```
<servlet>
   <servlet-name>logback-status</servlet-name>
   <servlet-class>ch.qos.logback.classic.ViewStat
</servlet>

<servlet-mapping>
    <servlet-name>logback-status</servlet-name>
    <url-pattern>/logback-status</url-pattern>
</servlet-mapping>
```

and then in a browser you can access the servlet as

```
http://localhost:8180/my-enterprise-ws/logback-st
```

2. Context parameters: You can also set some context parameters which can be read from all servlets in your application.

```
<context-param>
     <param-name>logbackConfigLocation</param-nam
     <param-value>classpath:caps-enterprise-servi
</context-param>
```

- SQL, XML, UML, JSC
- Hadoop & BigData Int
- Java Architecture Inte
- Scala Interview Q&As
- Spring, Hibernate, & I
- Testing & Profiling/Sa
- Other Interview Q&A 1
- ▶Free Java Interviev

16 Technical Key Areas

open all | close all

- Best Practice (6)
- Coding (26)
- Concurrency (6)
- Design Concepts (7)
- Design Patterns (11)
- Exception Handling (3
- Java Debugging (21)
- Judging Experience Ir
- Low Latency (7)
- Memory Managemen
- Performance (13)
- QoS (8)
- Scalability (4)
- SDLC (6)
- Security (13)
- Transaction Managen

80+ step by step Java Tutorials

open all | close all

- Setting up Tutorial (6)
- Tutorial - Diagnosis (2
- Akka Tutorial (9)
- Core Java Tutorials (2

and it will be referred in the logback framework/library as

```
String myContextParam =
        request.getSession()
               .getServletContext()
               .getInitParameter("logbackConfigLo
```

3) Servlet filters: A Servlet filter is an object that can intercept HTTP requests targeted at your web application. Servlet Filter is used for monitoring request and response from client to the servlet, or to modify the request and response, or to audit and log.

```
<filter>
    <filter-name>UrlRewriteFilter</filter-name>
    <filter-class>org.tuckey.web.filters.urlrewr
</filter>
<filter>
    <filter-name>Seam Filter</filter-name>
    <filter-class>org.jboss.seam.servlet.SeamFil
</filter>
<filter>
    <filter-name>Request Logging Filter</filter-
    <filter-class>com.myapp.filter.RequestLoggin
    <init-param>
        <param-name>excluded</param-name>
        <param-value>.js,css,javax.faces.resourc
    </init-param>
</filter>
```

4) Servlet listeners: Servlet Listener is used for listening to events in a web containers, such as when you create a session, or place an attribute in an session or if you passivate and activate in another container, to subscribe to these events you can configure listener in web.xml. With listeners you can track application-level, session-level, life-cycle changes, attribute changes etc.

```
<!-- Spring logback configuration listener -->
<listener>
      <listener-class> com.myapp.logging.LogbackC
</listener>
```

```
 package com.myapp.logging;

public class LogbackConfigListener implements Se

    @Override
```

⊞ Hadoop & Spark Tuto
⊞ JEE Tutorials (19)
⊞ Scala Tutorials (1)
⊞ Spring & HIbernate Tu
⊞ Tools Tutorials (19)
⊞ Other Tutorials (45)

100+ Java pre-interview coding tests

open all | close all
⊞ Can you write code? (
⊞ ♦ Complete the given
⊞ Converting from A to I
⊞ Designing your classe
⊞ Java Data Structures
⊞ Passing the unit tests
⊞ What is wrong with th
⊞ Writing Code Home A
⊞ Written Test Core Jav
⊞ Written Test JEE (1)

How good are your .....?

open all | close all
⊞ Career Making Know-
⊞ Job Hunting & Resum

```
    public void contextDestroyed(final ServletCo
        //some logic
    }

    @Override
    public void contextInitialized(final Servlet
       //some logic
    }
}
```

5) Resource mappings: The idea is that specifying resources in the web.xml has the advantage of separating the developer role from the deployer role. Let’s say you want to lookup java:comp/env/jdbc/dataSource/mydb, the container finds that web.xml has a <resource-ref&gy; element for jdbc/dataSource/mydb and it returns the object registered under the name of jdbc/dataSource/maindb.

```
<resource-ref>
   <description>MyApp Data Source</description>
   <res-ref-name>jdbc/dataSource/mydb</res-ref-na
   <res-type>javax.sql.DataSource</res-type>
   <jndi-name>jdbc/dataSource/maindb</jndi-name>
   <res-auth>Container</res-auth>
</resource-ref>
```

6) Security: The web.xml is the most important Java EE configuration piece allowing configuration of web container security.

BASIC authentication

```
<login-config>
  <auth-method>BASIC</auth-method>
  <realm-name>default</realm-name>
</login-config>
```

FORM based authentication

```
<login-config>
  <auth-method>FORM</auth-method>
  <form-login-config>
    <form-login-page>/login.html</form-login-page
    <form-error-page>/error.html</form-error-page
  </form-login-config>
</login-config>
```

Access control and constraints

```
<security-constraint>
   <display-name>excluded</display-name>
   <web-resource-collection>
      <web-resource-name>No Access</web-resource
      <url-pattern>/excluded/*</url-pattern>
      <url-pattern>/restricted/customer/excluded
      <url-pattern>/restricted/owner/excluded/*<
   </web-resource-collection>
   <web-resource-collection>
      <web-resource-name>No Access</web-resource
      <url-pattern>/restricted/*</url-pattern>
      <http-method>DELETE</http-method>
      <http-method>PUT</http-method>
      <http-method>HEAD</http-method>
      <http-method>OPTIONS</http-method>
      <http-method>TRACE</http-method>
      <http-method>GET</http-method>
      <http-method>POST</http-method>
   </web-resource-collection>
   <auth-constraint />
   <user-data-constraint>
      <transport-guarantee>NONE</transport-guara
   </user-data-constraint>
</security-constraint>
```

Q3. What are basic steps involved in bootstrapping Spring & Logback with the web container?

A3. Start with the **web.xml**

Step 1: web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.co
    xsi:schemaLocation="http://java.sun.com/xml/

    <display-name>Myapp Enterprise Service</disp

    <context-param>
        <param-name>logbackConfigLocation</param
        <param-value>classpath:myapp-enterprise-
    </context-param>

    <context-param>
        <param-name>logbackExposeWebAppRoot</par
        <param-value>false</param-value>
    </context-param>

    <!-- Spring logback configuration listener -
    <listener>
        <listener-class> com.myapp.logging.Logba
    </listener>

    <servlet>
        <servlet-name>myapp-enterprise-endpoint<
        <servlet-class>org.springframework.web.s
        <init-param>
            <param-name>contextClass</param-name
            <param-value>org.springframework.web
        </init-param>
        <init-param>
            <param-name>contextConfigLocation</p
            <param-value>com.myapp.myapp.configu
        </init-param>
        <init-param>
            <param-name>contextInitializerClasse
            <param-value>com.myapp.myapp.Enterpr
        </init-param>
        <load-on-startup>1</load-on-startup>
    </servlet>

    <servlet-mapping>
        <servlet-name>myapp-enterprise-endpoint<
        <url-pattern>/*</url-pattern>
    </servlet-mapping>

    <servlet>
        <servlet-name>logback-status</servlet-na
        <servlet-class>ch.qos.logback.classic.Vi
    </servlet>

    <servlet-mapping>
        <servlet-name>logback-status</servlet-na
        <url-pattern>/logback-status</url-patter
    </servlet-mapping>
</web-app>
```

Step 2: Define the Java classes wired up above like "**LogbackConfigListener**", "**MyAppEnterpriseEndpointConfiguration**", and **MyAppEnterpriseApplicationContextInitializer**.

```
//.....
public class LogbackConfigListener implements Se

    @Override
    public void contextDestroyed(final ServletCo
       //logic
    }

    @Override
    public void contextInitialized(final Servlet
        //logic
    }
}
```

```
//.....
@Configuration
@EnableWebMvc
@ComponentScan("com.myapp.endpoint")
public class MyAppEnterpriseEndpointConfiguratio

    @Override
    public void configureMessageConverters(final
        converters.add(new Jaxb2RootElementHttpM

        final MappingJackson2HttpMessageConverte
        jsonConverter.setObjectMapper(objectMapp
        converters.add(jsonConverter);
    }

    @Bean
    public ObjectMapper objectMapper() {
        final ObjectMapper objectMapper = new Ob
        objectMapper.registerModule(new JodaModu
        return objectMapper;
    }
}
```

```
//.....

public class MyAppEnterpriseApplicationContextIn
    ApplicationContextInitializer<ConfigurableAp

    public static final String DEFAULT_APPLICATI

    private static final Logger LOG = LoggerFact

    @Override
    public void initialize(final ConfigurableApp
        final ConfigurableEnvironment environmen
        final MutablePropertySources propertySou
        propertySources.addAfter(SYSTEM_ENVIRONM
    }

    protected PropertySource<?> getApplicationPr
        try {
            final Resource propertiesResource =
            final ResourcePropertySource propert
            LOG.info("Configured application pro
            return propertySource;
        } catch (final IOException ex) {
            throw new RuntimeException("Unable t
                + DEFAULT_APPLICATION_PROPERTIES
        }
    }
}
```

Popular Posts

♦ 11 Spring boot interview questions & answers

825 views

♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

766 views

18 Java scenarios based interview Questions and Answers

400 views

001A: ♦ 7+ Java integration styles & patterns interview questions & answers

388 views

01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

295 views

♦ 7 Java debugging interview questions & answers

293 views

01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

285 views

♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

279 views

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

239 views

001B: ♦ Java architecture & design concepts interview questions & answers

201 views

Bio | **Latest Posts**

Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

About Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

‹ ♥ Chrome Postman to test and debug RESTful Web Services

♥ Beginner array coding problems in Java ›

Posted in member-paid**,** Web basics

Empowers you to open more doors, and fast-track

Technical Know Hows

☀ Java generics in no time ☀ Top 6 tips to transforming your thinking from OOP to FP ☀ How does a HashMap internally work? What is a hashing function?
☀ 10+ Java String class interview Q&As ☀ Java auto un/boxing benefits & caveats ☀ Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect

Non-Technical Know Hows

☀ 6 Aspects that can motivate you to fast-track your career & go places ☀ Are you reinventing yourself as a Java developer? ☀ 8 tips to safeguard your Java career against offshoring ☀ My top 5 career mistakes

Prepare to succeed

☀ Turn readers of your Java CV go from “Blah blah” to “Wow”? ☀ How to prepare for Java job interviews? ☀ 16 Technical Key Areas ☀ How to choose from multiple Java job offers?

Select Category ▼

© Disclaimer

The contents in this Java-Success are copy righted. The author has the right to correct or enhance the current content without any prior notice.

These are general advice only, and one needs to take his/her own circumstances into consideration. The author will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. No guarantees are made regarding the accuracy or usefulness of content, though I do make an effort to be accurate. Links to external sites do not imply endorsement of the linked-to sites.

↑