Java-Success.com

Industrial strength Java Career Companion

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?

Spring loading property files & properties

Posted on October 2, 2015 by Arulkumaran Kumaraswamipillai

conf/db.properties

```
# jdbc.X
jdbc.driverClassName=org.hsqldb.jdbcDriver
jdbc.url=jdbc:hsqldb:hsql://localhost/mytestdb
jdbc.user=SA
jdbc.pass=
# hibernate.X
hibernate.dialect=org.hibernate.dialect.HSQLDialect
hibernate.show_sql=false
hibernate.hbm2ddl.auto=
```

#1. Register a Properties File in Application Context XML

The "**<property-placeholder>**" tag.

```
1  <context:property-placeholder location="classpath
```

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

open all | close all

OR

```
1  <bean class="org.springframework.beans.factory.co
2    <property name="locations">
3      <list>
4        <value>classpath:conf/db.properties</value>
5      </list>
6    </property>
7    <property name="ignoreUnresolvablePlaceholders"
8  </bean>
```

#2. Register a Properties File via Java configuration

The **@PropertySource** annotation.

```
1  @Configuration
2  @PropertySource("classpath:conf/db.properties")
3  public class MyAppConfig {
4
5    @Bean
6    public static PropertySourcesPlaceholderConfig
7      return new PropertySourcesPlaceholderConfig
8    }
9  }
```

OR

```
1  @Bean
2  public static PropertySourcesPlaceholderConfigure
3    PropertySourcesPlaceholderConfigurer propSource
4    Resource[] resources = new ClassPathResource[ ]
5    propSourcePlaceHolder.setLocations( resources )
6    propSourcePlaceHolder.setIgnoreUnresolvablePlac
7    return propSourcePlaceHolder;
8  }
```

#3. Injecting the properties via annotations

```
1
2
3    @Value("${jdbc.url:}")
4    private String jdbcUrl;
5
6    @Value("${jdbc.driverClassName:}")
7    private String jdbcDriverClassName;
8
9
```

#4. Accessing via the Environment API

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 I
- ⊞ Low Latency (7)
- ⊞ Memory Management
- ⊞ Performance (13)
- ⊞ QoS (8)
- ⊞ Scalability (4)
- ⊞ SDLC (6)
- ⊞ Security (13)
- ⊞ Transaction Managen

80+ step by step Java Tutorials

```
1  @Configuration
2  @Import(MyAppCommonConfiguration.class)
3  @EnableTransactionManagement
4  @ComponentScan("com.myapp")
5  @PropertySource("classpath:conf/db.properties")
6  @EnableJpaRepositories(
7      entityManagerFactoryRef = "entityManagerFact
8      basePackages = { "com.myapp.domain.repositor
9      transactionManagerRef = "transactionManager"
10 @EnableJpaAuditing
11 public class MyAppServiceConfiguration {
12
13     private static final Logger LOG = LoggerFact
14
15     @Inject
16     private Environment environment;
17
18     @Bean
19     DataSource dataSource() {
20         final JndiDataSourceLookup jndiLookup =
21         final String dataSourceJndi = environmen
22         return jndiLookup.getDataSource(dataSour
23     }
24
25     /**
26      * Creates the persistence properties bean.
27      */
28     @Bean
29     public Properties myDbDomainProperties() {
30         final Properties properties = new Proper
31         properties.put(DIALECT, environment.getR
32         properties.put(ORDER_UPDATES, environmen
33         properties.put(USE_SECOND_LEVEL_CACHE, e
34         properties.put(USE_QUERY_CACHE, environm
35         properties.put(USE_STRUCTURED_CACHE, env
36         return properties;
37     }
38
39     @Bean
40     public List<String> domainPackagesToScan() {
41         return ImmutableList.of("com.myapp.domai
42     }
43 }
```

100+ Java pre-interview coding tests

How good are your .....?

#5. Using the custom ApplicationContextInitializer or WebApplicationInitializer

Spring 3.1 introduced a very convenient hook for customizing the application context with ApplicationContextInitializer. It can be used to set active profiles and register custom property sources. Spring loading properties files with **ApplicationContextInitializer** or with **WebApplicationInitializer**.

The "**WebApplicationInitializer**" is applicable in a Servlet 3.0+ compliant servlet container and provides a hook to pro-

grammatically configure the servlet context.

Popular Member Posts

♦ 11 Spring boot interview questions & answers

904 views

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

816 views

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

427 views

18 Java scenarios based interview Questions and Answers

408 views

♦ 7 Java debugging interview questions & answers

323 views

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

311 views

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

303 views

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

301 views

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

251 views

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

209 views

0
Like
Share

Tweet

submit

reddit

0

G+1

Share

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**

‹   Spring loading properties files with ApplicationContextInitializer

Spring Controller Unit Testing   ›

Posted in Spring

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.