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?

Reloading configuration files in Java without stopping the server

Posted on December 18, 2015 by Arulkumaran Kumaraswamipillai

| 1 |
| Like |
| Share |

| 2 |
| G+1 |

Q1. Why & when do you want to use properties files for the project configuration as opposed to Java constants?
A1. It is a best practice to externalize application related properties that are likely to change. If you hard code the properties that are likely to change, then you need to recompile, package, and re-deploy the application when a property changes. If you put it in a configuration file like XML or name/value pair in .properties/.yml file, then when a property changes, you need to STOP the server, copy the modified configuration file to its location, and then START the server.

9 tips to earn more | What can u do to go places? | **945+** paid members. LinkedIn Group. **Reviews**

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

open all | close all

⊞ Ice Breaker Interview
⊟ Core Java Interview Q
  ⊞ Java Overview (4)
  ⊞ Data types (6)
  ⊞ constructors-metho
  ⊞ Reserved Key Wor
  ⊞ Classes (3)
  ⊞ Objects (8)
  ⊞ OOP (10)
  ⊞ GC (2)
  ⊞ Generics (5)
  ⊞ FP (8)
  ⊟ IO (7)

Q2. Why do you want to auto reload configuration files(s) when any changes happen?

A2. This prevents any server down-time as the server does not have to be stopped and started. The modified file can be copied to its location and the changes can be reloaded by the application.

Q3. How do you auto reload a configuration file in Java?

A3. The Apache commons-configuration library simplifies this task.

Step 1: Add commons configuration library dependency in your pom.xml file.

```
1
2      <properties>
3          <commons-configuration>1.10</commons-con
4      </properties>
5      <dependencies>
6          <dependency>
7              <groupId>commons-configuration</grou
8              <artifactId>commons-configuration</a
9              <version>${commons-configuration}</v
10         </dependency>
11     </dependencies>
12
```

Step 2: Sample Java/Spring code that use the library classes to reload the properties file. The key class is the "FileChangedReloadingStrategy".

```
1
2
3  package com.test.impl;
4
5  import java.io.File;
6  import java.util.List;
7
8  import javax.annotation.PostConstruct;
9
10 import org.apache.commons.configuration.Configur
11 import org.apache.commons.configuration.Properti
12 import org.apache.commons.configuration.reloadin
13 import org.slf4j.Logger;
14 import org.slf4j.LoggerFactory;
15 import org.springframework.beans.factory.annotat
16 import org.springframework.stereotype.Service;
17
18 //...
19
20 @Service
```

As a Java Architect

[Java architecture & design concepts interview Q&As with diagrams](#) | [What should be a typical Java EE architecture?](#)

```
21  public class WorkFlowManagerImpl implements Work
22  {
23      private static final Logger LOG = LoggerFact
24
25      private PropertiesConfiguration config;
26
27      @Value("${lookup-polling-seconds:120}")
28      private long lookupPropertiesPollingInterval
29
30      @Value("${lookup-file-path:120}")
31      private string lookupPropertiesFilePath;
32
33      @PostConstruct
34      public void configureObservableProperties()
35      {
36          File lookupPropertiesFile = new File(loo
37          config = new PropertiesConfiguration(loo
38          FileChangedReloadingStrategy strategy =
39          strategy.setRefreshDelay(lookupPropertie
40          config.setReloadingStrategy(strategy);
41      }
42  }
43
44
```

Q4. Is there any other alternative approaches to auto load configuration files in Java?

A4. Yes. The "WatchService" that was introduced in Java 7. It has two methods take() and poll(). The take() method wait for next change to happen. This is a blocking method. The poll() method does not block, and its task is to check for event changes. This approach is handy for applications requiring customised logic for auto reload. The Apache configuration library is much more easier to implement as explained earlier for standard auto load of configuration on any property change.

Popular Posts

♦ 11 Spring boot interview questions & answers

885 views

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

839 views

18 Java scenarios based interview Questions and Answers

454 views

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

410 views

Senior Java developers must have a good handle on

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

♦ 7 Java debugging interview questions & answers

315 views

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

310 views

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

302 views

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

286 views

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

265 views

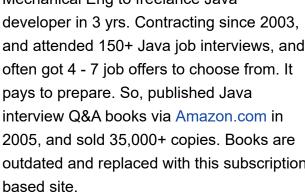8 Git Source control system interview questions & answers

221 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.

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

100+ Preparing for pre-interview Java written home assignments & coding tests

open all | close all

⊞ Can you write code?
⊞ ♦ Complete the given
⊞ Converting from A to
⊞ 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...to go places?

open all | close all

⊞ Career Making Know-
⊞ Job Hunting & Resum

35,000+ copies. Books are outdated and replaced with
this subscription based site.

‹   ♥ Overloaded methods Vs Generic methods in Java with JD-GUI &

javap to look under the covers

JUnit with Hamcrest   ›

Posted in IO

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.