Regular Expressions interview Q&A for Java developers

Posted on October 10, 2014 by Arulkumaran Kumaraswamipillai — No Comments ↓

Q1 What's the difference between a wildcard and a regular expression?

A1 A **wildcard** is a generic term referring to something that can be substituted for all possibilities. In computer terms, usually a simple "wildcard" is just a * that can match one or more characters, and possibly a **?** that can match any single character.

A **regular expression** (aka regex)certainly can do what a wild card can, and is a much more powerful pattern matcher that gives you the ability to restrict the type of characters.

To match somethiong like 123-456, 123-245, etc

A wildcard does

```
1 ???-???
```

A regular expression does

```
1 \d{3}-\d{3}
```

where "\" is an escape character, and "\d" means a digit. {3} means 3 times.

The Unix commandd shown below uses the wildcard character "?" in finding file names like job1.log, job2.log, job3.log, etc, and the grep command uses the regex to find contents within a file like "123-456".

```
1 find . -name "job?.log" -print  -exec grep   "[0-
```

The output of the above command is

```
1 ./job1.log
2 123-456
3 ./job2.log
4 123-245
5 ./job3.log
6
```

Q2 Why is regex very powerful? Where can we use them?
A2 Regular expressions are also known as regex, and they are very powerful and used widely in JavaScript, Unix scripting, development tools, monitoring tools like Tivoli, and programming languages. So, it really pays to have good knowledge of regexes. You can use online regex tools like regexpal.com or RegexBuddy to craft your regular expressions.

#1. Java APIs take regex as arguments. Splits it on "," with 0 or more leading and trailing spaces.

```
```

Other Interview Q&A f
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
Hadoop & Spark Tuto
JEE Tutorials (19)
Scala Tutorials (1)
Spring & HIbernate Tu
Tools Tutorials (19)
Other Tutorials (45)

```
1 String javaTechnologies = "Java , JEE, JDBC , Spr
2 String[] split = javaTechnologies.split("\\s*,\\s
```

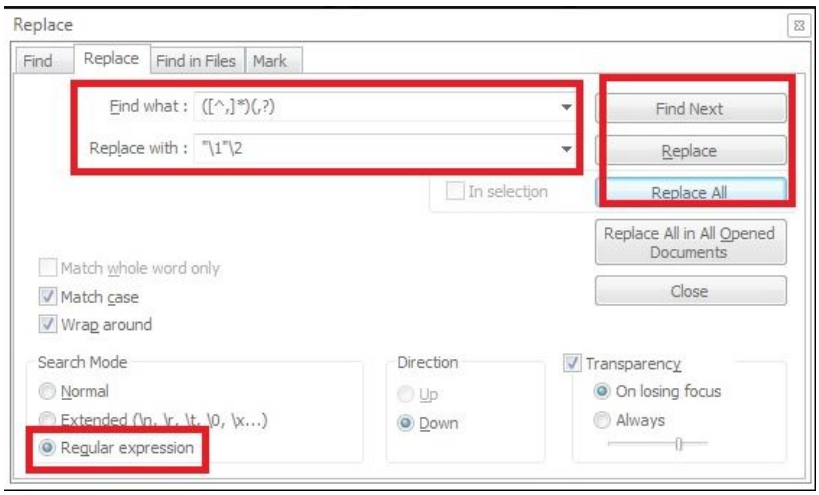
#2. Tools like Notepad++ provide regex based **find** and **find/replace**. The following comma separated text

```
1 BA555,04-May-2013,04-04-2013,12358,AUSTRALIAN BON
```

can be converted to quoted CSV text as shown below.

```
1 "BA555","04-May-2013","04-04-2013","12358","AUSTR
```

Now let's see how we can use notepad++ find/replace function to add quotes around each entry using its find/replace with regular expressions as shown below.



regex in Notepad++

As you can see

Find what regular expression is: **([^,]*)(,?)** , which means 0 or more characters but "," as first group stored in "\1" followed by 0 or 1 "," (i.e. optional ,), and grouped as \2.

Replace with regular expression is: **"\1"\2** where " is added then followed by \1, which is the value captured like BA555 and then followed by ", and followed by optional ",".

Similar approaches can be used for formatting other bulk records like removing spaces, adding quotes, replacing new line characters with tabs, replacing tabs with new lines, replacing commas with new lines, etc.

#3. Notepad++ with regex to construct SQL clause or any other data conversion.

Say you have data as shown below. This data could come from an excel spread sheet, word document, or copied from a confluence or wiki page.

```
1 id  type  rule_name              bean_name
2 633 ALL   asx100_rule            SECURITY_VALIDATIO
3 632 ALL   asx200_rule            SECURITY_VALIDATIO
4 634 ALL   ETF_rule               SECURITY_VALIDATIO
5 635 ALL   managed_fund_rule      SECURITY_VALIDATIO
```

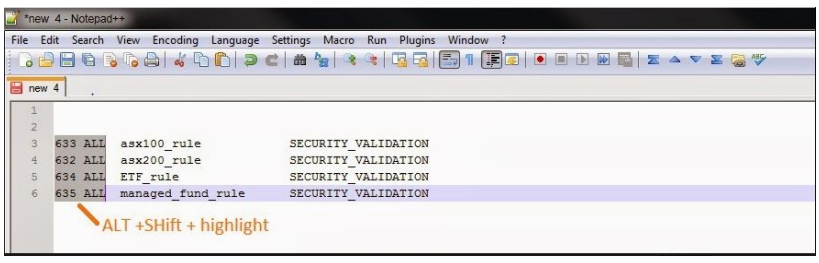
The SQL we need is:

```
1 SELECT * from rules_table
2   WHERE rule_name in ('asx100_rule', 'asx200_rule
```

This is how you go about converting it:

Step 1: Copy the data to Notepad++ and delete the header row by highlighting it and pressing the delete button.

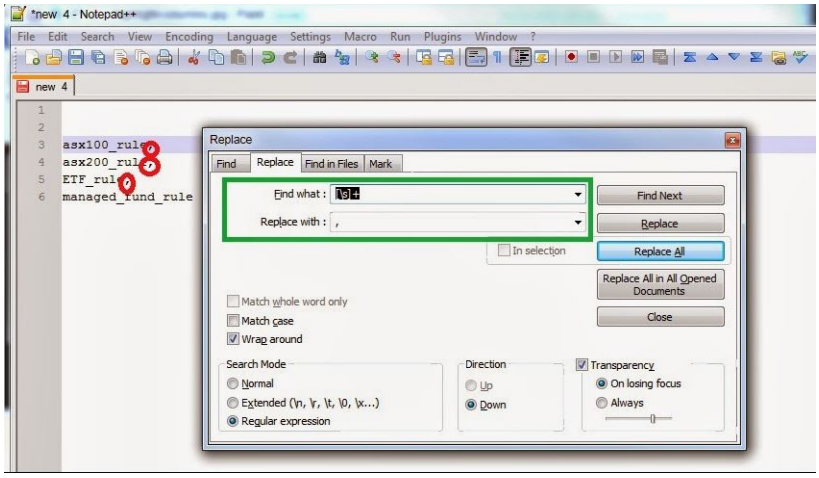
Step 2: You need to now remove all the columns except rule_name column. To do this place the cursor LHS of first two columns and press "**ALT + SHIFT**" keys together and highlight the columns you want to remove with the mouse. Do the same for the last column as well.



Notepad++ crop columns

Step 3: Next step is to remove any leading or trailing spaces. Use regex based find and replace command. Pressing **CTR+ F** will bring the Find dialog . You can also select it from the "Search" menu at the top.

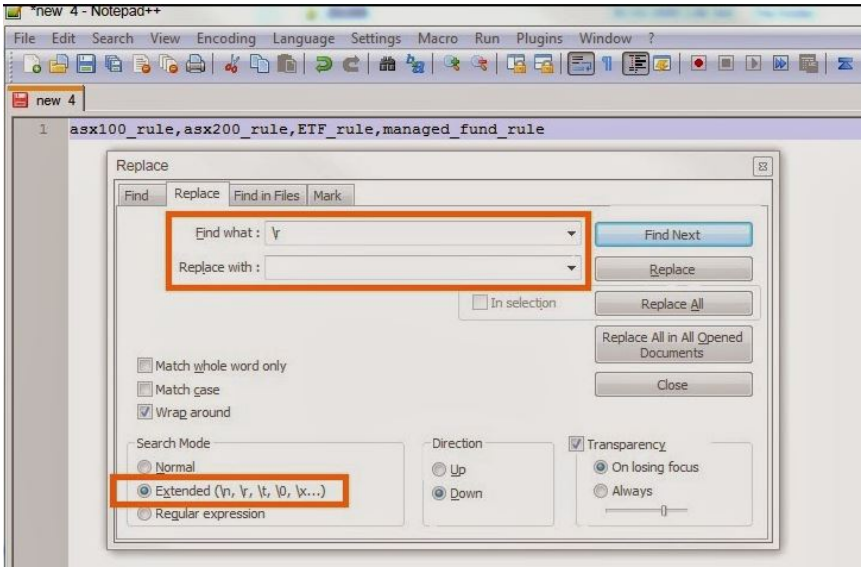
In the pop up find dialog, select the "replace" tab. Enter the find and replace value as shown below. Make sure the "Regular expression" option and "Wrap around" check box are ticked.



Notepad++ regex

```
1 Find What: [\s]+
2 Replace with: ,
```

Step 4: Remove the new line characters or carriage return by finding and replacing with the "Extended …" option turned on as shown below.




```
1 Find What: \r
2 Replace with:
```

Replace new line with nothing.

Step 5: You need to put a single quote (‘) around the entries for the SQL query. regex is again to the rescue.

```
1 Find What: ([^,]*)(,?)
2 Replace with: '\1'\2
```

The parentheses ‘( )’ are used to capture the values. and \1 and \2 represent both the captured values. The ‘ is add before \1 and \2. Where \1 is the value like “asx100_rule” and \2 is “,”. The * means 0 or many, and ? means 0 or 1.

You can now take the single line text and put it in your where clause. This is very handy when you have to work with larger data.

#4. Harnessing the power of **Unix scripts** and **regex**.

For example, if you have a pipe delimited text as shown below

```
1 |May 21 2014 12:00AM|
```

You can remove the last pipe with regex "**|$**", "-e" means expression,

```
1 cat myapp_last_run_date.dat | head -1 | sed -e 's
```

Remove the starting pipe with regex "**^|**". "g" means global change

```
1 cat myapp_last_run_date.dat | head -1 | sed -e 's
```

to break the commands into multiple lines, use "|"

```
1 cat myapp_last_run_date.dat | head -1 | \
2    sed -e 's/ *| */|/g' -e 's/\|$//' \
3        -e 's/^|//g'
4
```

#5. For validating user input in Java and JavaScript with regex. For example, to validate if the name entered is alphanumeric.

```
1  import org.apache.commons.lang.StringUtils;
2
3  public class AlphaNumericValidator
4  {
5      public boolean isAlphaNumeric(String input)
6      {
7          //fail fast
8          if (StringUtils.isEmpty(input))
9          {
10             throw new IllegalArgumentException("
11         }
12
13         boolean result = false;
14
15         if (input.matches("[a-zA-Z0-9]+")) // on
16         {
17             result = true;
18         }
19
20         return result;
21
22     }
23 }
```

#6. XSLT 2.0 makes text handling in XML documents a whole lot easier with its regular expression support.

#7. Regex can be used in **Maven pom.xml** files.

```
<project>
  [...]
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</group
        <artifactId>maven-surefire-plugin</artif
        <version>2.17</version>
        <configuration>
          <includes>
            <include>%regex[.*[Java|JEE].*Test.*
          </includes>
        </configuration>
      </plugin>
    </plugins>
  </build>
  [...]
</project>
```

There are a lot more examples.

Popular Posts

♦ 11 Spring boot interview questions & answers

828 views

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

768 views

18 Java scenarios based interview Questions and Answers

400 views

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

389 views

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

296 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

286 views

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

280 views

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

240 views

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

202 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**

‹ 10 Core Java Best Practices with an industry strength code sample

♥♦ Regular Expressions (aka Regex) by examples for Java developers ›

Posted in member-paid, Regex

Leave a Reply

Logged in as geethika. Log out?

Comment

Post Comment

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.

↑