



Reguläre Ausdrücke (Regular Expressions)

Zugehörige Informationen		
Abfragecode und Ausdrücke, Ausdrücke in Prüfreden, Ausdrücke in Skripten, Reguläre Ausdrücke	Allgemeine Bedienungshinweise	
Nummernkreise (VG), Variablendefinition in internen Abfragen, Textformulare		

Reguläre Ausdrücke

Reguläre Ausdrücke können in ASYS in Prüfreden und Abfragen als Kriterien eingesetzt werden.

In ASYS wird das RegEx-API (package 'java.util.regex') des Java Runtime Environment verwendet. Die Syntax entspricht weitgehend dem Perl5-Standard.

Für weitere Informationen zur Syntax und den Abweichungen vom Perl5-Funktionsumfang siehe <http://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html>.

Das Thema 'Reguläre Ausdrücke' wurde im ASYS-Workshop für Fachadministrator am 03./04.03.2008 in Würzburg behandelt. Im abschließenden Skript zum Workshop sind in Kapitel 3.3 eine Reihe von Beispielen und Erklärungen enthalten. Dieser Teil des Skripts ist nachfolgend wiedergegeben:

3.3 Verbesserungen der freien Abfragen

Wenn freie Abfragen mit der passenden ID des zugehörigen Aufgabenbereichs als erstem Result-Parameter versehen werden, kann die Maske des Aufgabenbereichs aus der Trefferliste der Abfrage mit dem passenden Datensatz direkt geöffnet werden.

Die durch die Anwender einzugebenden Parameter einer freien Abfrage können mit ergänzenden Eigenschaften durch den Administrator versehen werden (s. Kap. 5.2.7 des Admin-Handbuchs)¹⁾.

RegExp (Regular Expression = regulärer Ausdruck) ist eine Zeichenkette, mit der ein Muster für den durch den Anwender einzugebenden Parameterwert beschrieben wird. Das Muster wird im Prinzip Zeichen für Zeichen beschrieben. Nachfolgend einige grundlegende Bestandteile, aus denen sich reguläre Ausdrücke zusammensetzen:

Code	Bedeutung
.	Ein beliebiges Zeichen Ein Punkt als Zeichen wird mit \. ausgedrückt!
H	Genau das Zeichen H (aber nicht h!)

Code	Bedeutung
[abcd]	Zeichenklasse: eines der Zeichen a, b, c oder d (kann auch als [a-d] geschrieben werden)
[^abcd]	Negation, hier also irgendein Zeichen, außer a, b, c oder d
^ding	Sucht nach der Zeichenkette ‚ding‘ am Zeilenanfang - Achtung: keine eckigen Klammern hier!
ding\$	Sucht nach der Zeichenkette ‚ding‘ am Zeilenende (und ^\$ sucht nach Leerzeilen ohne Leerzeichen)
	Logisches Oder
()	Logische Gruppierung, z.B. um die Alternativen für den Oder-Operator abzugrenzen
?	Der davor stehende Ausdruck ist optional (0 oder 1 mal vorkommend)
+	Der davor stehende Ausdruck ist mindestens einmal vorhanden (1 bis n mal vorkommend)
*	Der davor stehende Ausdruck ist beliebig oft vorhanden (0 bis n mal vorkommend)
[ABC]{3}	Genaue Anzahl: der davor stehende Ausdruck muss genau 3 mal vorkommen
{1,5}	Intervall: Der davor stehende Ausdruck muss mindestens einmal vorkommen, darf aber höchstens fünfmal vorkommen
\w	Ein ‚Wortzeichen‘, entspricht [a-zA-Z0-9_]
\W	Alle ‚Nicht-Wortzeichen‘, entspricht [^a-zA-Z0-9_]
\d	Eine Ziffer (Digit), entspricht [0-9]
\D	Alle ‚Nicht-Ziffern‘, entspricht [^0-9]
\s	Ein Leerzeichen (Space), also Leerzeichen, Tab, Zeilenumbruch (Carriage Return), Formfeed und weitere
\t	Tab
\n	Newline - neue Zeile, die horizontale Position der Schreibmarke wird ggf. nicht verändert. ²⁾
\r	Carriage Return - Wagenrücklauf, eine neue Zeile wird ggf. nicht angehängt. ³⁾
\p{Lu}	Vordefinierte Zeichenklassen aus dem Unicode-Zeichenraum mit Namen, hier Lu=Großbuchstaben Weitere Klassen: Ll=Kleinbuchstaben Sm=mathematische Symbole, z.B. + / - * = ≥ ± ≠ Sc=Währungszeichen, z.B. \$ € £ ¥ ... Nl=römische Ziffern Uvm.

Aufgabe 13: Was bedeuten die folgenden regulären Ausdrücke? Schreiben Sie ein gültiges Beispiel auf. Wo setzen wir das wohl in ASYS/der BMU-Schnittstelle ein?

- **[1]\d{13}**

11234567890123

Begleitscheinnummer beginnend mit einer 1 und anschließend 13 Ziffern; für
Übernahmescheinnummern lautet der reguläre Ausdruck [2]\d{13}.

- **[RD][01]\d**

R08

Code für ein R- oder D-Verfahren beginnend mit R oder D, dann 0 oder 1 und anschließend eine Ziffer, aber Achtung: erlaubt ist hier auch D19!

- **[ABCDEFGHIKLMNPRS]{8}**

I664S00A1

Eine behördliche Nummer, beginnend mit einem Landeskenner und anschließend acht beliebige Zeichen.

- **(E|S)N[ABCDEFGHIKLMNPRS]{9}**

ENF123Ä56_8@

Eine Nachweisnummer, beginnend mit E oder S, danach ein N, anschließend ein Landeskenner und dahinter neun beliebige Zeichen.

- **`\d\d\d\d(\.w(w|\.w(w))?)?`**

08.11 oder 08.11.b oder 08.11.bb oder 08.11.b.bb

Eine Ziffer nach Anhang der 4. BIMSChV beginnend mit zwei Ziffern, einem Punkt und zwei weiteren Ziffern. Damit kann die Angabe enden. Wenn Sie weiter geht, folgen ein Punkt und ein Buchstabe. Auch hier kann die Angabe enden. Wenn Sie weitergeht, folgen entweder ein Buchstabe oder ein Punkt und zwei Buchstaben.

- **`[\p{Lu}]{2}`**

DE

Zwei Großbuchstaben für den ISO-Ländercode.

- **`[012]\d[01]\d[012349]\d`**

160402

Ein Abfallschlüssel, beginnend mit einer 0, 1 oder 2 und einer Ziffer gefolgt von einer 0 oder 1 und einer Ziffer und abschließend einer 0, 1, 2, 3, 4 oder 9 gefolgt von einer Ziffer.

Mit diesem Ausdruck werden aber nur grobe Fehler im Abfallschlüssel verhindert, es können weiterhin ungültige Abfallschlüssel eingetragen werden.

Aufgabe 14: Formulieren Sie einen regulären Ausdruck für folgende Zeichenketten.

- Freistellungsnummern aus Rheinland-Pfalz

`FRG.{9}`

FR für Freistellung, G für den Landeskenner Rheinland-Pfalz und neun beliebige Zeichen.

- Meyer, Maier, Mair, Meierhuber

`M[ae][iy]e?r(huber)?` oder etwas allgemeiner **`M[ae][iy]e?r.*`**

Ein M, anschließend a oder e, gefolgt von i oder y. Dann kann ein e folgen (muss aber nicht) und abschließend kann ein huber folgen (muss aber nicht) oder es folgen beliebig viele Zeichen.

- I447E1259, I318T003A, I613S1A44

`I\d{3}[EST][\dA-Z]{4}`

Ein I, anschließend drei Ziffern gefolgt von E, S oder T. Danach viermal eine Ziffer oder ein Großbuchstabe.

- Wie lässt sich ein Ausdruck für die R/D-Verfahren definieren, so dass nur die tatsächlich erlaubten Codes vorkommen können (D01 bis D15 und R01 bis R13)?

`((DR)((0[1-9])|1[0-3]))|D1[45]`

Ein D oder R gefolgt von einer 0 und einer Ziffer von 1 bis 9 oder gefolgt von einer 1 und einer Ziffer von 0 bis 3 oder als erste Zeichen D1 gefolgt von einer 4 oder 5.

- Ein gültiges Datum der Form dd-mm-jjjj (wobei wir Schaltjahre außen vor lassen wollen)

`((([01-9])|([12]\d)-(0[1-9]|1[012]))|(30-(0[13456789]|1[012]))|(31-(0[13578]|1[02])))-\d{4}`

Dieser Ausdruck muss in seine Bestandteile zerlegt werden:

`((0[1-9])|([12]\d)-(0[1-9]|1[012]))`

Eine 0 gefolgt von einer Ziffer 1 bis 9 oder eine 1 oder 2 gefolgt von einer Ziffer (also die Zahlen 01 bis 29), anschließend ein - und danach eine 0 gefolgt von einer Ziffer 1 bis 9 oder eine 1 gefolgt von einer 0, 1 oder 2 (also die Zahlen 01 bis 12). Alle Monate haben die Tage 01 bis 29 (der Februar zwar nur in Schaltjahren, aber das wollten wir ja außen vor lassen).

`(30-(0[13456789]|1[012]))`

Die Zeichenfolge 30- gefolgt von einer 0 und einer der Ziffern 1,3,4,5,6,7,8 oder 9 oder gefolgt von einer 1 und einer der Ziffern 0, 1 oder 2 (also alle Monate außer 02). Alle diese Monate haben einen 30. Tag.

`(31-(0[13578]|1[02]))`

Die Zeichenfolge 31- gefolgt von einer 0 und einer der Ziffern 1,3,5,7 oder 8 oder gefolgt von

einer 1 und einer der Ziffern 0 oder 2 (also die Monate 01, 03, 05, 07, 08, 10 und 12). Alle diese Monate haben einen 31. Tag.

Diese drei Bestandteile werden mittels Oder verknüpft und ergeben somit alle Kombinationen aus Tageszahl und Monatszahl. Abgeschlossen wird dieser Ausdruck durch ein - und daran anschließend vier Ziffern für die Jahreszahl.

Für Format siehe z.B. <http://www.apostate.com/vb-format-syntax>⁴⁾

Für RegExp siehe z.B. <http://networking.ringofsaturn.com/Web/regex.php>



Achtung: Sie können in den Variablen ‚Default‘, ‚Type‘, ‚Format‘ und ‚RegExp‘ Eintragungen vornehmen, die in sich widersprüchlich sind, oder sich sogar gegenseitig ausschließen (z.B. wenn der Default-Wert nicht auf den Typ oder den regulären Ausdruck passt).

(Hinweis: Für den Inhalt dieser Seite ist die IKA verantwortlich)

Weitere Informationen zu diesem Thema															
—															
landesspezifische Zusatzinformationen:	SH	HH	NI	HB	NW	HE	RP	BW	BY	SL	BE	MV	ST	BB	TH

¹⁾

Dieser Verweis ist mittlerweile überholt und nicht mehr gültig, da das hier vorliegenden Wiki das Handbuch abgelöst hat.

²⁾ ³⁾

Diese beiden Zeichen sind in der Windows-Welt begrifflich noch sehr an Schreibmaschinen angelehnt: Mit Newline wird nur die Walze mit dem Papier um eine Zeile fortbewegt. Mit Carriage Return wird die Position des Schreibkopfes an den Anfang der Zeile bewegt, die Walze mit dem Papier bleibt unverändert. Der Vorschubhebel einer Schreibmaschine sorgte für den Rücktransport des Schreibkopfes - bzw. der Walze mit dem Papier - zum Beginn der Zeile und schob das Papier mittels Walzendrehung eine Zeile nach oben: Carriage Return & Newline -> \r\n. Andere Betriebssysteme - z.B. Unix, Linux und ihr Abkömmlinge - weichen z.T. hiervon ab.

⁴⁾

War Thema im Workshop, wird hier aber nicht behandelt!

From:

<https://hilfe.gadsys.de/asyshilfe/> - **ASYS-Onlinehilfe**

Permanent link:

<https://hilfe.gadsys.de/asyshilfe/doku.php?id=adm6:thm:regexp>

Last update: **2018/03/23 14:35**

