Utilisation d’expressions régulière dans Spectre

Description

Il y a trois fonctions permettant l'utilisation d'expressions régulières dans Spectre :

- ***regexp()***
- ***regexp_subst()***
- ***regexp_value()***

Les expressions régulières sont des fonctions de recherche avec correspondance de modèles.

Plutôt que de chercher une chaine basée sur un mot, une phrase ou une autre chaine de caractère, les expressions régulières recherchent des modèles spécifiques, fournis par l'utilisateur.

Ne soyez pas effrayé(e) par les expressions régulières, elles peuvent être intimidantes au début, mais elles ne sont pas aussi compliquées qu'elles en ont l'air.

Retourner un booléen

Voici un exemple de syntaxe basé sur une expression régulière :

```
Non.[a-zA-Z\\s']*père
```

Explications / détail :

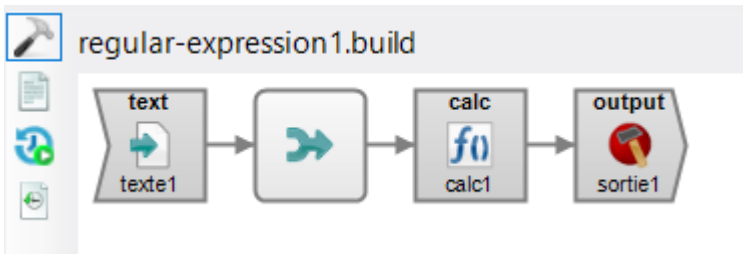
[a-ZA-Z] --> n'importe quelle lettre, en majuscule ou en minuscule.

[\\s] --> un jeton représentant un caractère d’espace (le slash supplémentaire est pour l'échappement).

['] --> un jeton représentant l'apostrophe.

***** --> un modificateur pour le modèle précédent, indiquant une correspondance autant de fois que possible jusqu’à ce que le prochain modèle soit trouvé.

Voici ci-dessous un script ***.build*** :



Le contenu de l'entrée ***texte1*** est le suivant :

(ligne)	Colonne1
1	Non. Je ne suis pas ton père
2	Oui. Je suis ton père
3	Oui. Je suis ta mère
4	Non. Je ne suis pas ta mère
5	Non. Dark Vador n'est pas ton père
6	Non. R2-D2 n'est pas ton père

Le contenu de l'objet ***calc1*** est le suivant :

Expression for Resultat recherche

```
1  regexp(value("Colonne1"),"Non.[a-zA-Z\\s']*père")
```

Un test sur l'objet ***calc1*** donne ceci :

Test results

(ligne)	Colonne1	Resultat recherche
1	Non. Je ne suis pas ton père	True
2	Oui. Je suis ton père	False
3	Oui. Je suis ta mère	False
4	Non. Je ne suis pas ta mère	False
5	Non. Dark Vador n'est pas ton père	True
6	Non. R2-D2 n'est pas ton père	False

Retourner une valeur

Voici un exemple de syntaxe basé sur une expression régulière :

`[0-9]{4}-[0-9]{2}-[0-9]{2}`

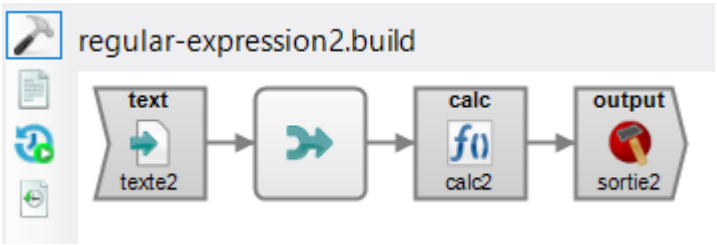
Explications / détail :

[0-9]{4} --> recherche un chiffre exactement 4 fois, ou dit différemment recherche un nombre avec exactement 4 chiffres.

- --> recherche le tiret.

[0-9]{2} --> recherche un chiffre exactement 2 fois, ou dit différemment recherche un nombre avec exactement 2 chiffres.

Voici ci-dessous un script ***.build*** :



Le contenu de l'entrée ***texte2*** est le suivant :

Test results

(ligne)	Colonne1
1	Transaction numéro 1243-63-45
2	Transaction numéro 1256*63-45
3	Num transaction 4583-32-26 donné par l'utilisateur
4	Transaction 3659-52-36
5	numéro non fourni
6	Transaction num 1236-256-33

Le contenu de l'objet ***calc2*** est le suivant :

Expression for Resultat recherche

```
1  regexp_value(value("Colonne1"),"[0-9]{4}-[0-9]{2}-[0-9]{2}")
```

Un test sur l'objet ***calc2*** donne ceci :

Test results

(ligne)	Colonne1	Resultat recherche
1	Transaction numéro 1243-63-45	1243-63-45
2	Transaction numéro 1256*63-45	
3	Num transaction 4583-32-26 donné par l'utilisateur	4583-32-26
4	Transaction 3659-52-36	3659-52-36
5	numéro non fourni	
6	Transaction num 1236-256-33	

Retourner une substitution

Voici un exemple de syntaxe basé sur une expression régulière :

```
[(]([0-9]*)([.]?[0-9]*)[)]
```

[(] ou ***[)]*** --> recherche la présence d'une parenthèse ouvrante ou fermante.

([0-9]*) --> recherche n'importe quel nombre composé de plusieurs chiffres. Les parenthèses autour du modèle créent ce que l'on appelle un groupe de capture.

([.]?[0-9]*) -->recherche la présence optionnelle d'un point (séparateur de décimales) puis recherche n'importe quel nombre composé de plusieurs chiffres. Les parenthèses autour du modèle créent ce que l'on appelle un groupe de capture.

Voici ci-dessous un script ***.build*** :

regular-expression3.build

Le contenu de l'entrée ***texte3*** est le suivant :

Test results

(ligne)	Vendeur	CA
1	Olivier	235.00
2	Jacques	(236.96)
3	Michèle	1225.36
4	Carole	(3658.30)
5	Robert	233
6	Hyacinthe	(256)

Le contenu de l'objet ***calc3*** est le suivant :

Expression for Resultat recherche

```
1 regexp_subst(value("CA"),"[(]([0-9]*)([.]?[0-9]*)[)]","-$1$2")
```

Le symbole ***$*** représente un groupe de capture.

Un test sur l'objet ***calc3*** donne ceci :

Test results

(ligne)	Vendeur	CA	Resultat recherche
1	Olivier	235.00	235.00
2	Jacques	(236.96)	-236.96
3	Michèle	1225.36	1225.36
4	Carole	(3658.30)	-3658.30
5	Robert	233	233
6	Hyacinthe	(256)	-256

Tags

1. calcul
2. script
3. Spectre