

Useful links:

Suspicious -

<https://dzone.com/articles/a-complete-tutorial-on-the-drools-business-rule-en>

Good -

https://subscription.packtpub.com/book/application_development/9781782165545/1/ch01/vl1sec06/top-5-features-you-need-to-know-about

Advanced -

https://docs.jboss.org/drools/release/5.5.0.Final/droolsjbpm-introduction-docs/html_single/

update(\$f) - обновляет объект в базе знаний, это форсит перепроверить те правила где этот объект фигурирует, в свое время помогло решить задачу с currency field.

modify(\$f) { ... } - modifies object, therefore it can be modified directly set..(), get...()

salience <0, negative value> works

import function com.fxclub.fxbank.api.v3.withdraw_api_ws.entity.RequisiteFormCondition.or;
- static method.

REX_EXP_KEY	REX_EXP_VALUE	EXAMPLES	DESCRIPTION
VALUABLE_STRING	^(?!s*\$).+	Dsfdl;df fdsd	Any string, but not spaces, \n, \t and so on.
PERCENT_VALUE	(^100([\.,]0{1,2})?\$(^([1-9]([0-9])? 0)([\.,][0-9]{1,2})?\$\$)	100 0,23 23,34	Number between 0 and 100 with two digits after comma dot.
ZERO_VALUE	^0((\.,)(0)+)?\$	0,000000 0	Zero value with any number of digits after comma dot.
POSITIVE_NUMBER_ACCURACY_N	=> N = 0 / ^\d+?\$/, N > 0 ^\d+((\.,)\d{1," + N + "})?\$\$	0,23232323 12.3232 12 434	Any Number with any number of digits after comma dot. Or without digits and dot comma.

=

```

end

rule "if bankInfo.countryIso3 empty currency field empty and disabled"
no-loop
when
    $f: RequisiteFormField(code == "currency")
then

```

Ya ETO CHYT NE ZAPYSHIL !!!!

Using `update` makes the rules engine aware that a fact has been modified. Therefore rules which depend upon that fact must be re-evaluated. This does have the effect that a rule may fire in an endless cycle. For instance, given the following DRL, you will see that the "Infinite loop" rule will activate continuously:

```

declare AreWeThereYet
    answer : boolean
end

rule "Are we there yet?"
when
    not AreWeThereYet()
then
    insert(new AreWeThereYet());
end

rule "Infinite loop"
no-loop
when
    $question: AreWeThereYet()
then
    $question.setAnswer(false);
    update($question);
end

```

This is because the rules engine has been instructed by `update($question)`, that `$question` has changed and that it needs to re-evaluate it.

There are ways of preventing this though. Just put `no-loop` on the line between the rule name and `when` to prevent rules from re-activating as a result of their own consequences. Another rule attribute which can control this is `lock-on-active`.

Long story short:

- `no-loop`: avoid the re-activation of a rule caused by the RHS of that SAME rule.
- `lock-on-active`: avoid the re-activation of a rule NO MATTER what the cause is.

Long story long -

<https://ilesteban.wordpress.com/2012/11/16/about-drools-and-infinite-execution-loops/>