

A Developer Diary

{about:"code learn and share"}

[Home](#)[Data Science](#)[Java](#)[JavaScript](#)[jBPM](#)[Tools](#)[Tips](#)[About](#)

November 8, 2015 By [Abhisek Jana](#) — [1 Comment \(Edit\)](#)

Get started with jBPM KIE and Drools Workbench – Part 3



In this part 3 of the **Get started with jBPM KIE and Drools Workbench** series we will complete our Admission Business Process and learn few more ways to create different types of process flow.

In our Admission Process, a Reviewer needs to review the GPA score and decide whether to approve the application or not. Instead of this to be User Task, we will now create a Guided Rule and perform the operation based on a cut-off GPA score.

Guided Rule:

Create a Guided Rule from **New Item -> Guided Rule**. Name it as **Check Eligibility** and click on Ok. You will now see the **WHEN** and **THEN** section. Click on the Plus **(+)** icon beside the WHEN. The following popup will be displayed.

Add a condition to the rule...



Position: ?

.....
Number ...
Student ...
.....
The following does not exist ...
The following exists ...
Any of the following are true ...
.....
From ...
From Accumulate ...
From Collect ...
From Entry Point ...
.....
Free form DRL

Select **Student** and Click on Ok. Then click on **There is a Student** and the following popup will be opened.

Modify constraints for Student



Modify constraints for Student

Add a restriction on a field

...

Multiple field constraint

...



Advanced options:

Add a new formula style expression

New formula

Expression editor

Expression editor

Variable name

student

Set

Enter the **Variable name** as **student** and click on **Set**. Click again on **There is a Student [student]** and from the **Add a restriction on a field** dropdown, select **gpa**. Then select dropdown beside gpa and choose **greater**

than . Then click on the pencil icon beside the dropdown, a popup will be opened,click on **Literal Value** button in the popup, which looks like:



There will be an input box now be available, enter **2** there.

Click on the Plus **(+)** icon beside the THEN. Then select **Modify student** and click on Ok.

Add a new action...

Position: ?

.....
Change field values of student...

.....
Delete student

.....
Modify student...

.....
Insert fact Number...

Insert fact Student...

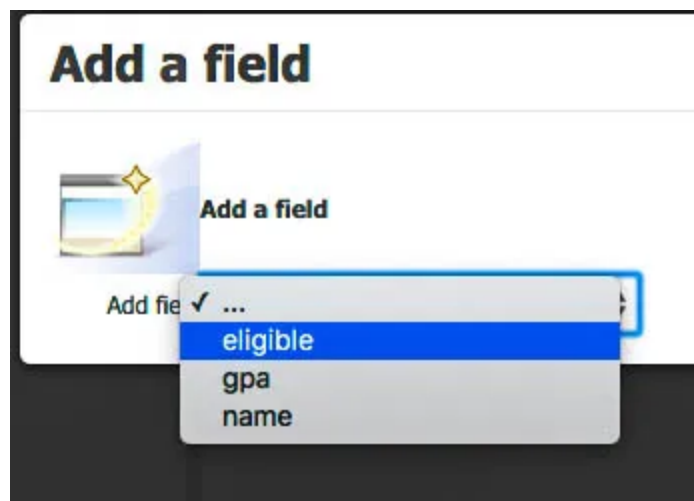
.....
Logically Insert fact Number...

Logically Insert fact Student...

.....
Add free form DRL

.....
Call method on student...

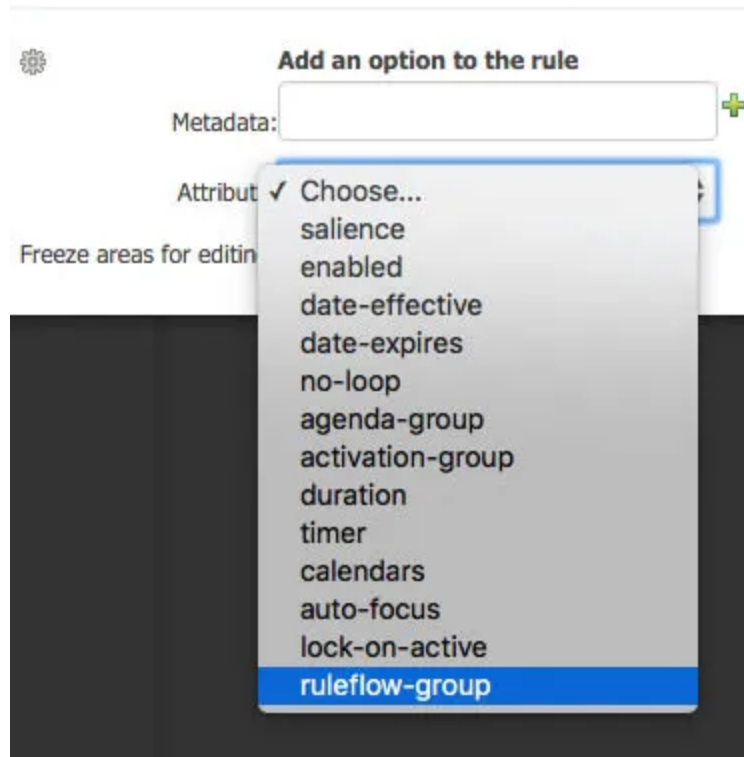
Now click on **Modify value of Student [student]** and select **eligible** from the **Add Field** Popup.



Again, click on the pencil icon and click on **Literal Value** button in the popup. This time you will have a drop down. Select true there.

We also need to add few options, click on **(show options...)**. It should automatically show **dialect** as **mvel**. Change it to **java**. Then click on the Plus **(+)** icon beside **(options)** and add 3 Attributes, **ruleflow-group** as **admission**, **no-loop** as **checked** and **lock-on-active** as **checked**.

Add an option to the rule



So in this rule, if the gpa is more then 2.0 then the eligible would be true.

Save the Guided Rule, it looks like this.

Check Eligibility.rdr1 - Gu... Save Delete Rename Copy Validate Latest Version

EXTENDS None selected

WHEN

1. gpa greater than 2

THEN

1. Modify value of Student [student] eligible true

(options)

Attributes:

dialect java

ruleflow-group admission

no-loop ☒

lock-on-active ☒

Update Business Rule Task:

Now select the **Review GPA** Task and change the Task Type to **Business Rule Task**. Open the properties panel and click on **RuleFlow Groups**.

Select the dropdown in the admission group, select **Check Eligibility.rdr1** and select that row then Click on Save.

5 admission

Check Eligibility.rdr1

Example Admission master

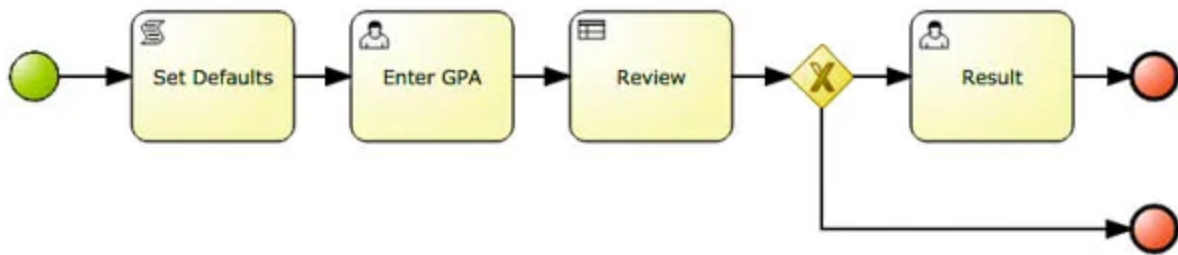
view

Now Save the Business Process.

We will add one **Script Task** before **Enter GPA** to set the default values. Add the following two lines in the script attribute.

```
student.setEligible(false);
student.setGpa(0.0f);
```

The final Business Process would look like this.



Now go ahead and execute it, it should run fine.

DRL File:

In case you are a developer and you want to use the DRL File to create the rule, click on New Item and select **DRL File**. Enter the name as **AdmissionDRL**.

Now enter the following into the editor.

```
package adeveloperdiary.admission;
```

```
rule "set isEligible"
ruleflow-group "admission"
lock-on-active true
when
    $p : Student(gpa > 2.0)
then
    $p.setEligible(true);
end
```

Here we have the same **ruleflow-group** as **admission**. We are performing the exact same validation here as we did in our Guided Rule.

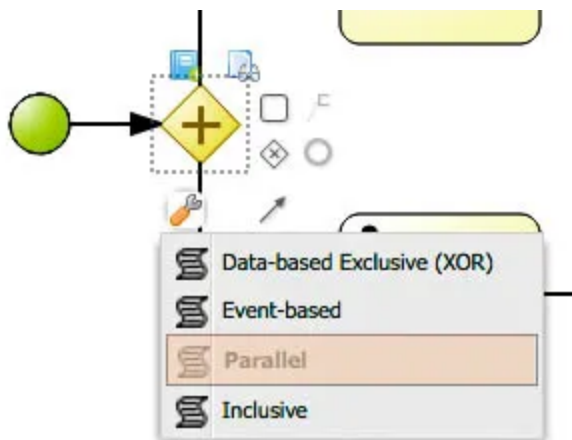
In order to execute this rule, you need to again select the **Review GPA** Task and open the properties panel then click on **RuleFlow Groups**.

Select the dropdown in the admission group, this time select **AdmissionDRL.dr1**, select that row then Click on Save.

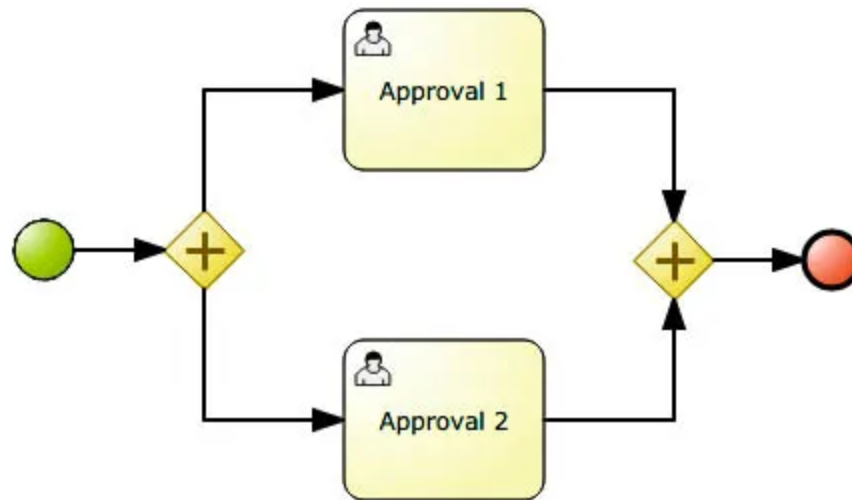
Deploy and execute the business process and it will work as it was.

Parallel Flow:

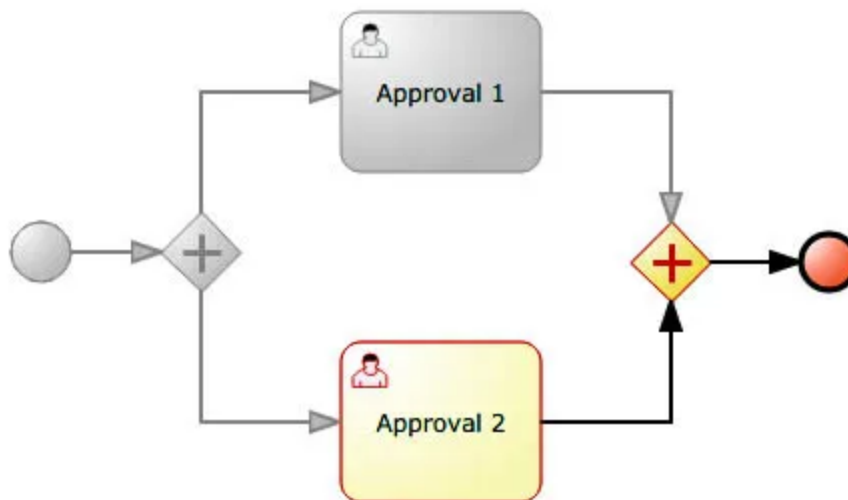
Let's look into parallel flows. For an example if we need approval from two users and we shall proceed only if we get approval from both of them then we can use the parallel option. Add a XOR and click on option and change it to **parallel**.



Here is the Business Process Flow.

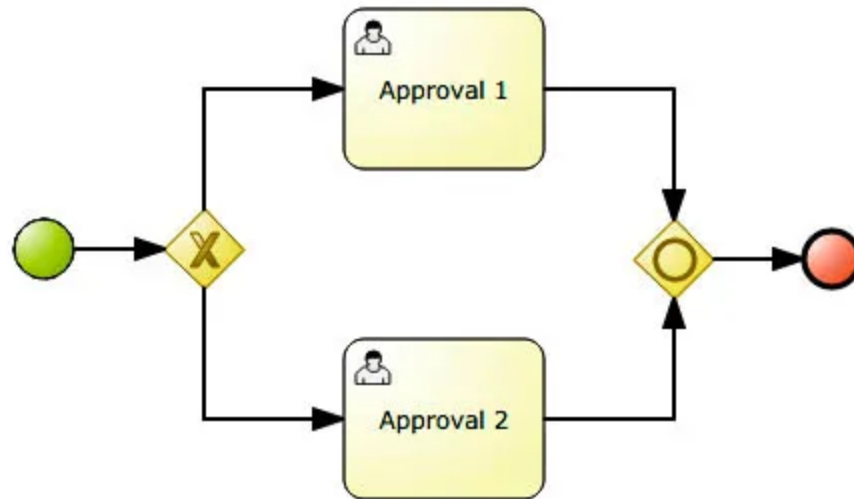


Once we deploy this and execute it, there will be two tasks assigned to both the users. If we complete one still the flow won't be completed. It will wait until the **Approval 2** User Task hasn't completed. Here is the Process Model:

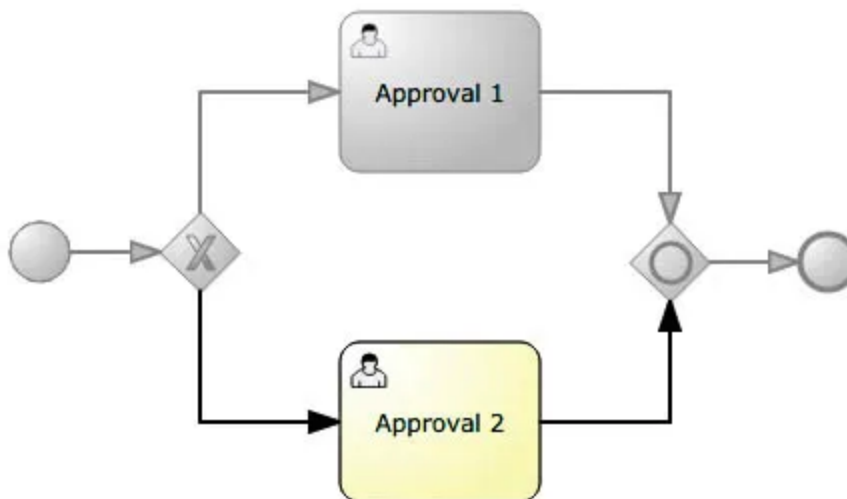


Inclusive Flow:

Now change the 1st Parallel to **XOR** (if-else) and 2nd Parallel to **Inclusive**. I have hard coded the logic in XOR, so that the top one is always true (return true/false; in the script for top and bottom arrow). Here is how this looks.



Deploy and execute this. Complete the task for **Approval 1**, you will notice the flow will end and it will not wait for the **Approval 2** to be completed. As long as one of the parallel flow has been completed it will execute the next step. Here is the Process Model:



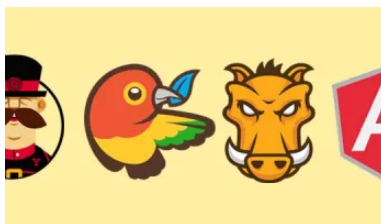
Hopefully the **Get started with jBPM KIE/Drools Workbench** series will help you to overcome the initial issues you may face with the toolset.

Related



Get started with jBPM KIE and Drools Workbench – Part 2

In "jBPM"



How to create an Angular project in Yeoman

In "Yeoman"



How to Learn and Master Angular easily – Part5

In "Angular 1.x"

Filed Under: [jBPM](#) | Tagged With: [BPM](#), [Business Process](#), [Drools](#), [JBOSS](#), [jBPM](#), [KIE](#), [KIE Workbench](#)

Subscribe to stay in loop

* indicates required

Email Address *

Subscribe

Comments



Soheli says

April 7, 2016 at 2:41 am

[\(Edit\)](#)

Could you please give some references on Artifact repository and whether can i upload a project?If yes, then what are the steps to deploy it and include it in project authoring.

[Reply](#)

Leave a Reply

Logged in as Abhisek Jana. [Edit your profile](#). [Log out?](#) Required fields are marked *

Comment *

Post Comment

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)

Copyright © 2024 A Developer Diary