What are the validation rules in Pega?

**Introduction**

Every application needs correct data to process any request.

Imagine you are visiting an employment portal and applying for a job. When you fill any application form, you may come across different fields with different validation. For example, in an email address field, if you don’t use ‘@’ and ‘.com’ you may get an error as “Enter valid email”.

The same way Pega application may use different forms to get input from the user. As a developer, we need to make sure the data provided by the user are valid. We use different validation rules to perform this task.

***What are the types of Validation?***

1. Server side validation – Implemented by java code. Involves a server trip for validation.
2. Client side validation – Implemented by javascript code. Validation happens in the browser.

We will see about these types in a separate lesson.

***What are the different Validation rules in Pega?***

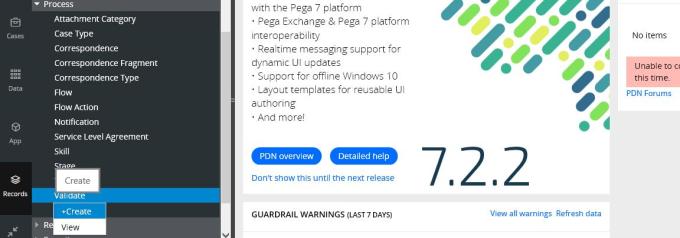
* + 1. Validate rule.
    2. Edit Validate rule.
    3. Edit input rule – formatting input.
    4. Constraints rule.
    5. Validation through Property definition and Control.
    6. Custom javascript validation.

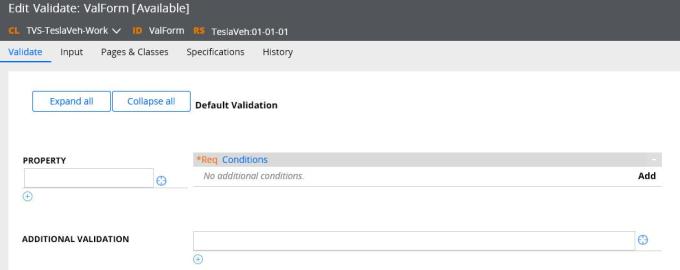
**1) What is a Validate rule?**

* Ensures that a valid data is entered in form or validates data received from other system.
* When a field fails validation, the rule adds an error message and prevents successful submission of form.
* We can add validation for any property rule and we can customize the error message.
* Mostly used in Flow actions and Activity rules.

***How to configure a Validate rule?***

* It is a server side validation. Mostly we will use in a flow action rule to validate the fields in form.
* It is part of process category.





There are 2 important tabs to configure in a Validate rule:

1. Validate
2. Input

**Validate tab**

Property – Enter a property. It can be single value or a page/pagelist.

Additional validation – We can refer another Validate rule.

Conditions – On clicking add, we get a popup to configure the conditions to validate.

1. *For single value property validation*,



In the popup, we can configure many criteria.

Required – Expects the field to be mandatory and throws error message as “This field may not be blank”, if empty.

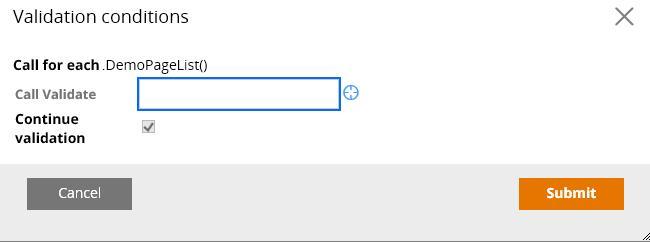
Enable conditions – True / False – Enables or disables the below conditions.

Add a function – Select all the required functions to validate the property. You can add multiple functions to validate a single property. Go through those functions.

Message – If the condition fails, then you can add a message.

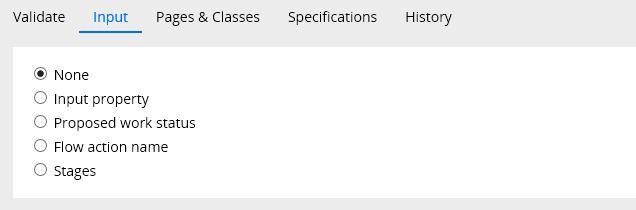
Continue Validation – True / False – If a condition fails, then you can stop the validation right at the moment by un-checking continue validation.

2. *For aggregate property*,



You need to use a validate rule to run on each page.

**Input tab**



There are 5 types of input values:

1. None – scenario explained above.
2. Input property.

Imagine a scenario where we have different validation conditions for male and female. We can handle that in single validate rule. Use input property as Gender.

Gender is a single value local list property.



Now switch to validate tab.

C:\Users\JANANDSUN\Desktop\image-7.jpg

Add the plus icon to add different blocks of validation conditions for different input values.



Here a new block is added for Gender = Male and separate validation conditions.

3. *Proposed work status*

We can have special validation for cases based on work status. Imagine different validation for a case when the status is “open”. Select proposed work status in input tab. In the validate tab, you can add separate block when input value is open.

4. *Flow action name* – Similar to above, we can have different validation conditions for different flow actions. Mainly used in screen flow, where there is no need to create separate validation rule for separate screen.

5. *Stages* – Similar to above, we can have different validation conditions for different stages.

***Where do we refer a Validate rule?***

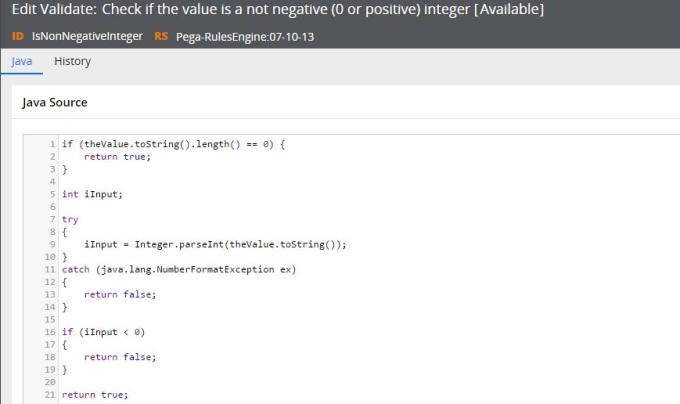
1. In flow action rule validate tab. The flow action uses the validate rule to validate all the fields before successfully submitting the form.
2. In activity using Obj-Validate method. Mainly used in validating fields while creating any case from background process or service.

**2) What is an Edit Validate rule?**

* Validates an input value using java code.
* Return Boolean value.

***How to configure an edit validate rule?***

* Edit validate rules comes under Data model category.
* Create a new edit validate rule.
* Edit validate rules do not belong to any class.
* Pega ships with a variety of OOTB edit validate rule. Check those before deciding to create a new one.



Edit validate rule single main tab – Java

* Enter the Java Boolean function.
* The code validate the automatically declared named ‘theValue’.

Validate rules don’t validate when Input value is blank.

***Where do we refer an Edit validate rule?***

1. In Property rule form advanced tab.
2. In activity using property validate method.
3. It can also be referenced in validation rule – run edit validate conditions.

**3) What is an Edit Input rule?**

* It is basically a conversion rule (not validation)  & cannot be included in validation rules.
* Converts the data from one format to different format.
* Uses java code for conversion.

***How to configure an Edit Input rule?***

* Edit Input rule comes under Data model category.
* Create a new edit input rule.

Imagine a scenario where you need to save the user input, only in capital letters. Use can use the below edit input to convert the format to upper case always.



***Where do we refer an Edit Input rule?***

* Property rule advanced tab can refer an edit input rule.

You can test an edit input rule. Add the uppercase edit input rule to any field on rule form. After submitting the form, verify the value in clipboard. You can see the data is formatted to upper case.

***Differences between edit validate and edit input rule:***

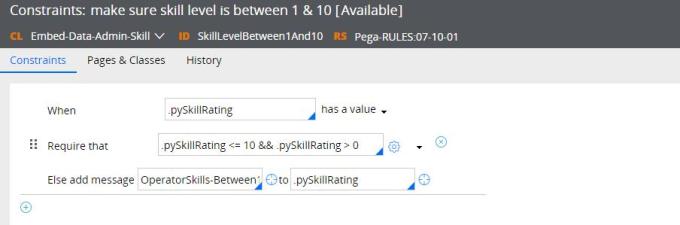
* Edit input is a conversion rule type, whereas Edit validate is a validation rule type.
* Edit Input formats the data even though the data is invalid.
* In a property rule, if both edit input and edit validate are specified, edit input occurs first followed by edit validate.

**4) What is a constraints rule?**

* Constraints is a declarative rule, which uses forward chaining.
* So whenever the property value is touch the validation fires by constraints rule.

***How to configure a constraints rule?***

* Constraints rule comes under decision category.
* Create a new Constraints rule.



There is a single main tab in a constraints rule – Constraints.

When – specify a when condition – this must evaluate to true.

Require that – Enter the validation function.You can use settings icon to include different functions available.

Else add message – If the ‘require that’ condition fails, then it adds a message.

Let’s look at the above example.  
Here, whenever pySkillRating value changes(non–null), it should be between 1-10. Else error message will be added.

***Where do we refer a Constraints rule?***

* Since it is a declarative rule, there is no need to explicitly call the rule. Pega engine takes care.

You can test a constraints rule. Use the pySkillRating property as one of the input field in a form. Enter the value as ‘100’. On submitting the form, an error message is thrown as configured in the Constraints rule.

Remember to check “Enable expression calculation” in flow action or harness form.

The feature is based on AJAX technology.

These four are the validation rules available in Pega

5. Validation through Property definition and Control.

6. Custom javascript validation.

We will discuss points 5 & 6 in ‘Server side vs Client client Validation’ ([http://myknowpega.com/2017/05/13/server-side-vali…-side-validation/](http://myknowpega.com/2017/05/13/server-side-validation-vs-client-side-validation/)) post.