

## BPM - An Overview

Business process management is a discipline in operations management in which people use various methods to discover, model, analyze, measure, improve, optimize, and automate business processes. BPM focuses on improving corporate performance by managing business processes.

## Alfresco Process Services

Alfresco Process Services (powered by Activiti) is an enterprise Business Process Management (BPM) solution targeted at business people and developers. At its core is a high performance open-source business process engine based on Activiti with the flexibility and scalability to handle a wide variety of critical processes. Alfresco Process Services provides a powerful suite of end user tools and integrates with a range of enterprise systems, including Alfresco Content Services, Box and Google Drive.

Alfresco Process Services enables: Users to access all information, tasks, and documents to quickly and efficiently complete key business processes. Collaboration with users working as a team to complete tasks.

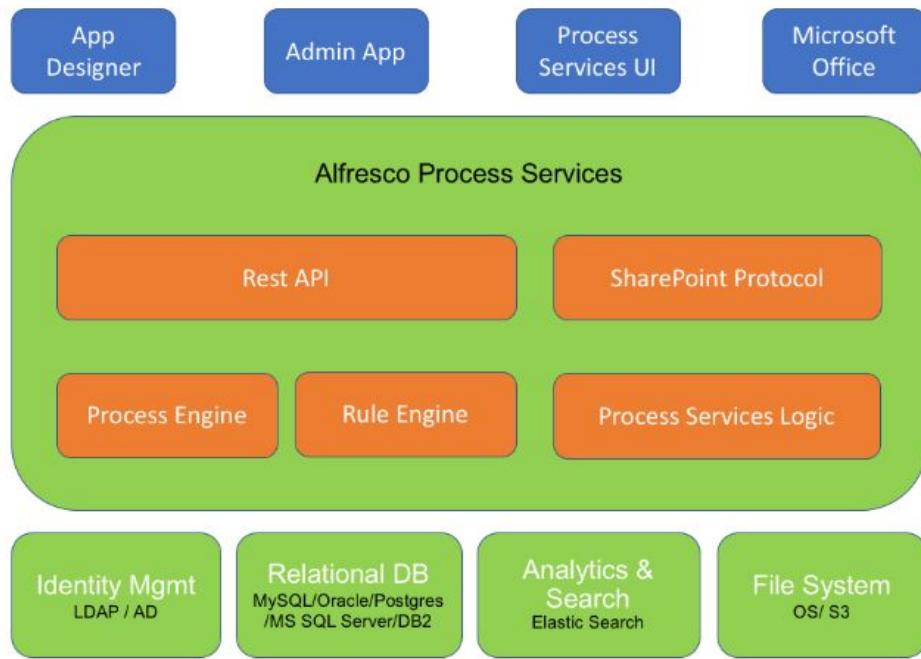
## Getting Started with Alfresco Process Services

With Alfresco Process Services it's easy to create, publish, and use process models and apps. Before you begin, make sure that you've following the instructions in Installing Alfresco Process Services.

Most BPM solutions are too rigid to allow users to adjust for non-routine situations, which results in slowing the flow of business. Alfresco Process Services enables:

- Users to access all information, tasks, and documents to quickly and efficiently complete key business processes
- Collaboration with users working as a team to complete tasks. Users can allocate tasks and subtasks to individuals, allowing users to address non-routine tasks
- Group and individual tasks inboxes allow organizations to model processes to support a wide range of business scenarios

## APS Architecture



## Using Alfresco Process Services

The Alfresco Process Services Landing Page continues to provide a user interface for managing your tasks but with the additional features for process design and profile management.

The Landing Page is your user interface to Process Services. Each tile gives you tools for distinct sets of tasks.

The Landing Page is the starting point from which you can use:

[\*\*App Designer\*\*](#) - Design your process

[\*\*My Tasks\*\*](#) - View your task inbox or queue

[\*\*Profile management / Identity management\*\*](#) - Manage user and group capabilities

[\*\*Analytics\*\*](#) - Generate reports on process performance

**Profile management** will appear for you only if you are a user. This is where you manage your personal information. If you have administrator capabilities, then **Profile management** will be displayed as **Identity management**. Use this tile to access your profile page as well as to manage user, group, and capability management pages for your tenant or the

whole system.

Your landing page is dynamic, and new tiles will appear when you create new process apps in the App Designer and deploy them in the Task App. If you are an administrator, your landing page is slightly different. Instead of the Profile management tile, you'll find a more powerful set of tools called the Identity management tile.

## BPMN Editor

With the BPMN editor you can create process definitions using the capabilities of BPMN 2.0. You build your process by dragging and dropping from a palette of grouped components to a canvas on which your process diagram is built.

The BPMN editor is structured into several areas:

**Palette** - On the left side of BPMN editor is the palette, which consists of collapse-able groups of BPMN objects.

**Canvas** - On the right side of BPMN editor is the canvas, where the BPMN objects can be added to create a process model.

**Properties sheet** - Below the canvas is the properties sheet, which shows the properties of the selected BPMN object on the canvas, or if no BPMN object is selected, the properties of the process itself. You can click on any of the properties to modify its value. The property sheet is collapse-able to allow you more screen space to view your process diagram.

**Toolbar** - The toolbar is displayed on the top with a set of grouped command icons.

There are two ways of adding BPMN objects to your process:

- Use the controls that appear when you click on a current object icon. Using this method will create a valid connector between the current event icon and the new event icon.
- Drag and drop an object icon from the palette. In this case you add flows to the current event icons in the process yourself by picking the icons from the palette.

The following object groups are shown in a collapsible list in the palette. The groups consist of all the objects available in the BPMN 2.0 specification, and additional Alfresco Process Services extensions such as the Publish to Alfresco task, Publish to Box, Publish to Google Drive.

**Start events** - A start event indicates where a process starts.

**Activities** - An activity describes a single item of work to be performed in a process.

**Structural components** - You use structural components to group multiple components in a sub process to reuse in a parent process definition, and to embed and call other process definitions from inside your own process.

**Gateways** - You use gateways to control the flow of execution in your process.

**Boundary events** - You use boundary events to handle an event associated with an activity.

**Intermediate catching events** - An intermediate catching event is a step in the process where the process needs to wait for a specific trigger

**Intermediate throwing events** An intermediate throw event is used to explicitly throw an event of a certain type.

**End events** You use an end event to signify the end of a process or sub-process, or the end of a path in a process or sub-process.

**Swimlanes** You use swimlanes to display activities in your process divided by business function or participant group.

## Sample Example

### Step 1: Create your first BPMN process model

This is the first of three simple steps in creating a process app. In this step, you are going to design a simple expense approval process that includes three BPMN elements:

- A **start event** to trigger the process by submitting a new expense
- A **user task** to approve or reject the request
- An **end event** to end the process

#### 1. Start Event:

Start events define where a Process or Subprocess starts. The process engine supports different types of start events:

##### 1.1 None Start Event or Start Event -

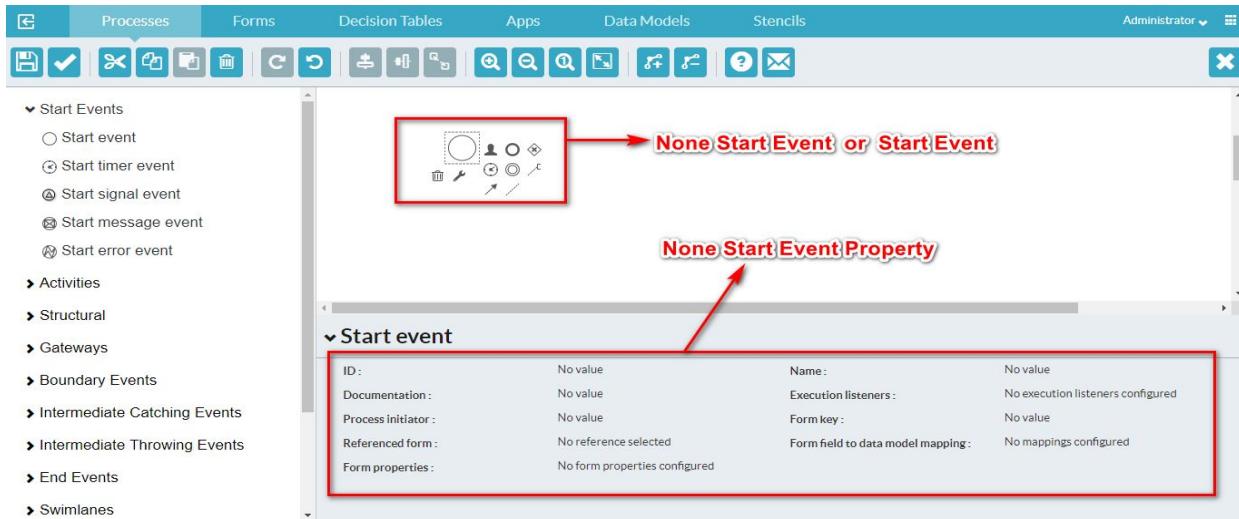
A start event with an unspecified trigger.

Undefined trigger to start of process or Entry point to a Sub Process.

A none start event can have a start form. If so, the start form will be displayed when selecting the process definition from the processes list. Note that a process instance is not started until the start form is submitted. A none start event without a form will simply have a button displayed to start the process instance.

It is visualized as a circle with no icon.





S.no	Property	Description
1	Id	A unique identifier for this element.
2	Name	A name for this element.
3	Documentation	A description of this element.
4	Execution listeners	Execution listeners configured for this element instance. An execution listener is a piece of logic that is not shown in the diagram and can be used for technical purposes.
5	Process Initiator	The process variable in which the user ID of the initiator of this instance should be stored.
6	Form key	A key that provides a reference to a form. This property is available for compatibility with Activiti , but should not be used directly when using Forms. Use the <i>Referenced form</i> property instead.
7	Referenced form	A form reference.
8	Form properties	A form definition with a list of form properties. Form properties are the way forms are defined in the community version of Alfresco Process Services. Configuring them has no impact on the rendered form in the Alfresco Process Services, the <i>Referenced form</i> property should be used instead.

## Example



In this BPMN diagram example, the process shows that we first get our eggs, then scramble them and eat them for breakfast. We don't indicate why the process was started. It could be because we were hungry, because we're on a dietary schedule, or because our mom told us we had to eat our breakfast. We don't know, or care. Trying to figure out what triggers the egg scrambling process doesn't help us understand the process any better – so we use the none start event.

### 1.2 Start Timer Event -

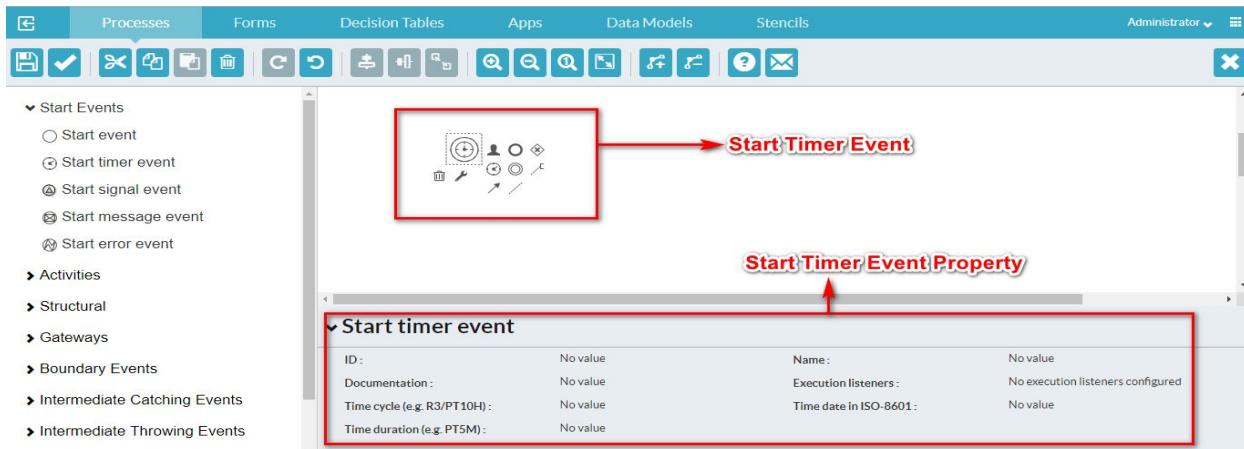
A timer start event initiates a process instance at specific time. You can use it both for processes which must start only once and for processes that must start in repeated time intervals.

Note that a process instance started by a timer start event can't have a start form, as it is started by the system. Similarly, it does not have a process initiator like a none start event. As such when assigning tasks later on in the process definition, keep in **mind that the assignment 'assigned to process initiator' will not work.**

**A subprocess can't have a timer start event.**

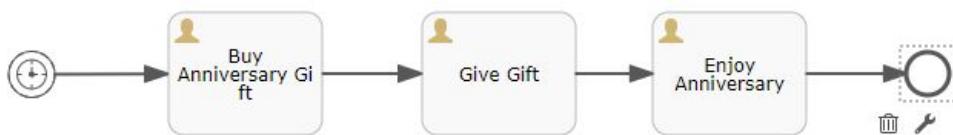
It is visualized as a circle with a clock icon





S.No	Property	Description
1	<b>Id</b>	A unique identifier for this instance.
2	<b>Name</b>	A name for this element.
3	<b>Documentation</b>	A description of this element.
4	<b>Execution listeners</b>	Execution listeners configured for this instance. An execution listeners is a piece of logic that is not shown in the diagram and can be used for technical purposes.
5	<b>Time Cycle</b>	<a href="http://en.wikipedia.org/wiki/ISO_8601">A timer cycle defined in http://en.wikipedia.org/wiki/ISO_8601 format, for example: R3/PT10H.</a>
6	<b>Time Date in ISO-8601</b>	<a href="http://en.wikipedia.org/wiki/ISO_8601_date">A point in time defined as a http://en.wikipedia.org/wiki/ISO_8601 date, for example: 2015-04-12T20:20:32Z.</a>
7	<b>Time Duration</b>	<a href="http://en.wikipedia.org/wiki/ISO_8601duration">A period of time defined as a http://en.wikipedia.org/wiki/ISO_8601duration, for example: PT5M.</a>

### Example



The timer event is used in BPMN diagrams to indicate that the process begins at a set time or date. It can also be used for processes that repeat on a schedule (hourly, daily, etc).

In our example, on the actor's wedding anniversary, he buys a gift, gives it to his spouse, and then enjoys their anniversary. This process is triggered by the timing of the anniversary.

### 1.3 Start Signal Event -

A signal start event starts a process instance using a named signal. The signal is fired from a process instance using the intermediary signal throw event (or programmatically through the java or REST API). In both cases, a process instance for any process definitions that have a signal start event with the same name are started. You can select a synchronous or asynchronous start of the process instances.

A signal start event is visualized as a circle with a triangle inside. The triangle is white inside.



The screenshot shows the Camunda BPMN editor interface. The top navigation bar includes 'Processes', 'Forms', 'Decision Tables', 'Apps', 'Data Models', and 'Stencils'. Below the toolbar, there are various icons for creating and managing models. On the left, a sidebar lists event types: Start Events (Start event, Start timer event, Start signal event, Start message event, Start error event), Activities, Structural, Gateways, Boundary Events, Intermediate Catching Events, Intermediate Throwing Events, and End Events. The 'Start signal event' icon is selected and highlighted with a red box. A tooltip 'Start Signal Event' points to the icon. A larger red box highlights the 'Start signal event' properties panel below, which shows fields for ID, Documentation, Name, and Execution listeners. The 'Name' field is currently empty, and the 'Execution listeners' section indicates 'No execution listeners configured'.

S.No	Property	Description
1	Id	A unique identifier for this element.
2	Name	A name for this element.
3	Documentation	A description of this element.

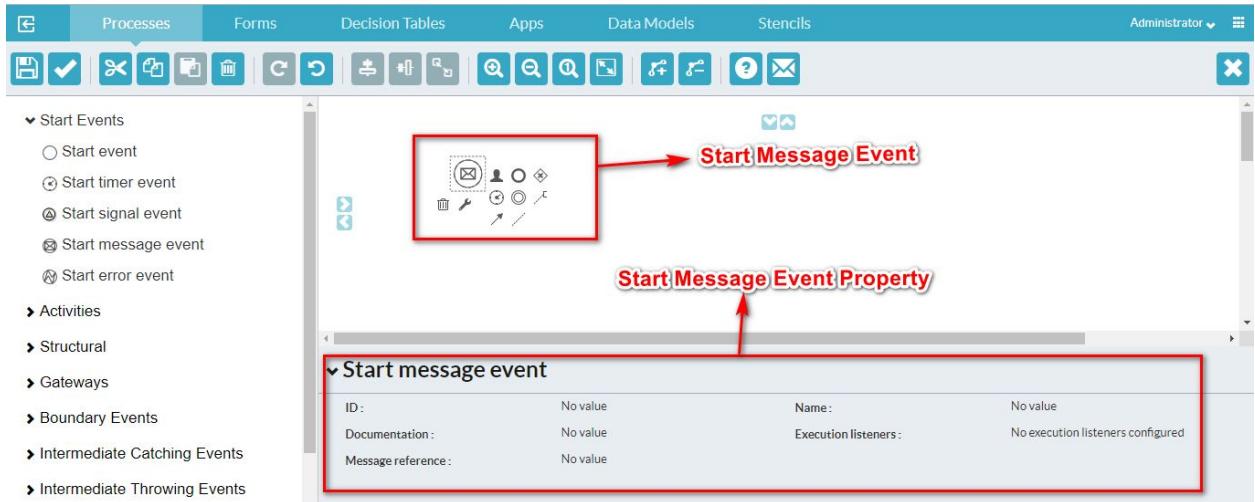
4	Execution listeners	Execution listeners configured for this instance. An execution listeners is a piece of logic that is not shown in the diagram and can be used for technical purposes.
5	Signal reference	The name of the signal that initiates this event. Note that signal references are configured on the root level of the process instance and then linked to the signal start event via this property. To configure it, deselect any other element and click the <i>Signal definitions</i> property.

#### 1.4 Start Message Event -

A message start event starts a process instance using a named message. It is mainly used for starting process instances from external systems. It is depicted as a circle with an envelope icon inside. The envelope is white inside.

When you deploy a process definition with one or more message start events, consider the following points:

- The name of the message start event must be unique across the whole process definition. Alfresco Process Services will throw an exception on deployment of a process definition with two or more message start events that reference the same message or with two or more message start events that reference messages with the same name.
- The name of the message start event must be unique across all deployed process definitions. Alfresco Process Services will throw an exception on deployment of a process definition with one or more message start events that reference a message with the same name as a message start event already deployed in a different process definition.
- When a new version of a process definition is deployed, the message subscriptions of the previous version are canceled. This is also true for message events that are not present in the new version.



S.No	Property	Description
1	Id	A unique identifier for this element.
2	Name	A name for this element.
3	Documentation	A description of this element.
4	Execution listeners	Execution listeners configured for this instance. An execution listener is a piece of logic that is not shown in the diagram and can be used for technical purposes.
5	Message reference	The name of the message that initiates this event. Note that messages are configured on the root level of the process instance and then linked to the message start event via this property. To configure it, deselect any other element and click the 'Message definitions' property.

### Example



When a process is initiated by an outside actor sending us a message (telling us to start the event), we use the message start event.

This BPMN diagram example shows a process of getting water, giving it to a child, and returning to bed. The process is initiated by the receipt of a message. We become aware that the child is complaining of thirst and unable to sleep. When we receive that message, we initiate the process of getting water for the child.

### 1.5 Start Error Event -

An error start event triggers an event Sub-Process. An error start event can't be used for starting a process instance.

It is visualized as a circle with lightning icon inside. The icon is white inside.



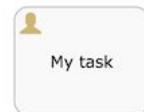
S.No	Property	Description
1	Id	A unique identifier for this element.
2	Name	A name for this element.
3	Documentation	A description of this element.
4	Execution listeners	Execution listeners configured for this instance. An execution listeners is a piece of logic that is not shown in the diagram and can be used for technical purposes.
5	Error reference	The name of the error that initiates this event. This reference needs to match the error identifier thrown by the event that throws the particular error.

## 2. Activities:

An activity describes a single item of work to be performed in a process. An activity is always visualized as a rectangle with rounded corners.

### 2.1 User Task

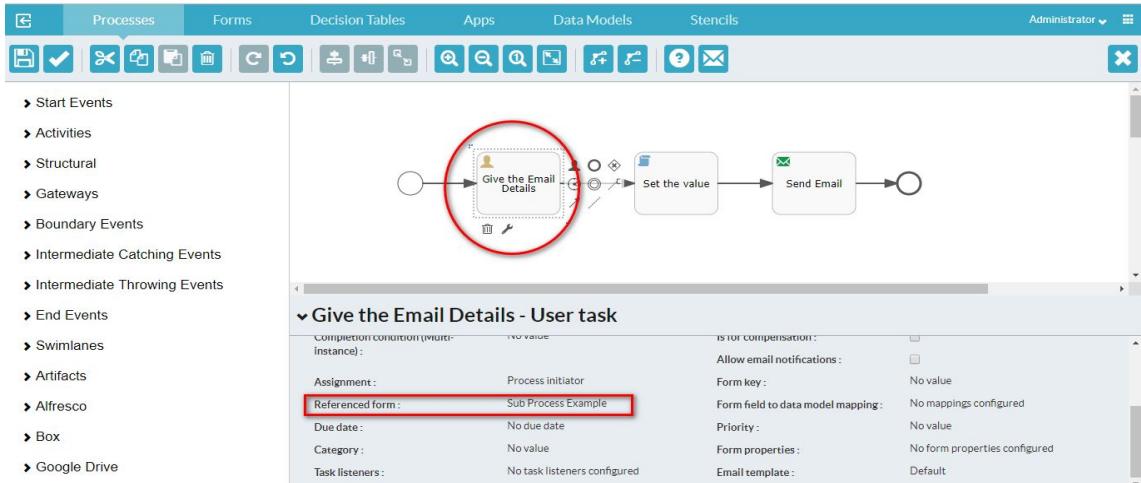
A user task enables you to model work to be done by a human actor. When process execution arrives at a user task in the process definition, it creates a new task in the task list of the assignee or assignees defined in the task. A user task is depicted as a rounded rectangle with a user icon on the top-left corner.



#### Example:

#### Create process and change the property for Variable

The screenshot shows the Activiti BPMN2 Workbench interface. The top navigation bar includes 'Processes', 'Forms', 'Decision Tables', 'Apps', 'Data Models', and 'Stencils'. Below the navigation is a toolbar with various icons for process management. On the left, a sidebar lists process elements: Start Events, Activities, Structural, Gateways, Boundary Events, Intermediate Catching Events, Intermediate Throwing Events, End Events, Swimlanes, Artifacts, Alfresco, Box, and Google Drive. The main workspace displays a process diagram with three tasks: 'Give the Email Details' (user task), 'Set the value' (script task), and 'Send Email' (mail task). The 'User and Mail and Script Task Example' properties panel is open, showing configuration details for the process. The 'Variables' section is highlighted with a red box, indicating '3 variables configured'. Below this, a modal dialog titled 'Change value for "Variables"' is shown, allowing the configuration of individual variables like 'mailbody', 'subject', and 'emailid'. The 'Save' button is visible at the bottom right of the dialog.



The screenshot shows the Activiti Form editor interface. The "Forms" tab is selected. A form definition for "Sub Process Example" is displayed, featuring two fields: "Subject" and "Mail Body".

## Mapping the variables

The screenshot shows the "Change value for 'Referenced form'" dialog. It displays a table mapping source process variables to target process variables:

Source process variable	Form variable	Target process variable
	mailbody	mailbody
	subject	subject

Below the table, there are sections for "Mapping type" (In/Out mapping, Input mapping, Output mapping), "Form field or variable" (Mail Body - mailbody - multi-line-text), and "Value type" (New Variable, Form field, Variable). The "Source process variable" column is highlighted with a red box.

This example is continued in Script task and then mail task.

## 2.2 Service Task

Use a service task to invoke an external Java class or execute an expression. A service task is visualized as a rounded rectangle with a cog icon inside.



### Example

Required:

- I. Identity management--->Organization-->Create group and Add user that group
- II. Get the group id for below image

The screenshot shows a web application interface for managing groups. At the top, there are tabs for 'Tenants', 'Users', 'Capabilities', 'Organization' (which is selected), and 'Personal'. On the right, there's an 'Administrator' dropdown. Below the tabs, there's a 'Create group' button and a list of existing groups: 'Admin Team', 'Demo Library Approver', and 'Demo Library User'. The 'Demo Library User' group has its 'Group Id' field highlighted with a red box and a red arrow pointing to it. The main area shows 'Subgroups (0) + add subgroup' and 'No subgroups found'. Under 'Group members', there's a '+ add user' button and a note 'Group manager + select user' with 'No manager selected'. At the bottom, the developer tools Network tab is open, showing a list of network requests:

Name	Status	Type	Initiator	Size	Time	Waterfall
3003	200	xhr	angular.js:9827	1.4 KB	18 ms	[Waterfall]
users?page=0&pageSize=50	200	xhr	angular.js:9827	592 B	18 ms	[Waterfall]
select-people-popover.html?1559803611398	200	xhr	angular.js:9827	7.4 KB	9 ms	[Waterfall]
select-people-popover.html?1559803611400	200	xhr	angular.js:9827	7.4 KB	166 ms	[Waterfall]

- III. Webservice Ref. APS API given link.  
<http://localhost:8080/activiti-app/api-explorer.html>

**groups : Retrieve and manage user groups**

Show/Hide | List Operations | Expand Operations

Query groups

GET /enterprise/groups/{groupId}/users

List members of a group

Response Class (Status 200)

OK

Model Example Value

```
{
  "data": [
    {
      "company": "string",
      "email": "string",
      "externalId": "string",
      "firstName": "string",
      "id": 0,
      "lastName": "string",
      "pictureId": 0
    }
  ]
}
```

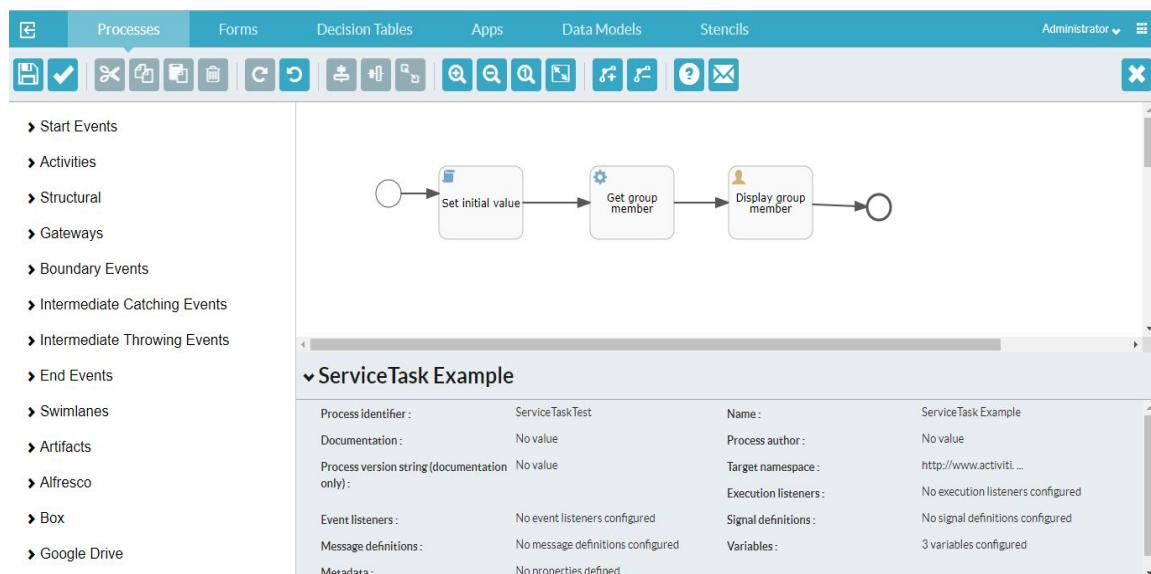
Response Content Type application/json ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type
groupId	3003	groupId	path	long

Try it out! Hide Response

### Process:



### Configure Global variables for the Process:

Screenshot of the Alfresco BPMN2 Modeler interface showing a process definition and a configuration dialog.

**Process Definition:**

```

graph LR
    Start(( )) --> SetInitialValue[Set initial value]
    SetInitialValue --> GetGroupMember[Get group member]
    GetGroupMember --> DisplayGroupMember[Display group member]
    DisplayGroupMember --> End(( ))
  
```

**ServiceTask Example Configuration:**

- Process identifier: ServiceTaskTest
- Documentation: No value
- Process version string (documentation only): No value
- Event listeners: No event listeners configured
- Message definitions: No message definitions configured
- Metadata: No properties defined
- Name: ServiceTask Example
- Process author: No value
- Target namespace: http://www.activiti...
- Execution listeners: No execution listeners configured
- Signal definitions: No signal definitions configured
- Variables:** 3 variables configured

A red arrow points from the text "Configured Global variable" to the "Variables" section of the configuration table.

**Change value for "Variables" Dialog:**

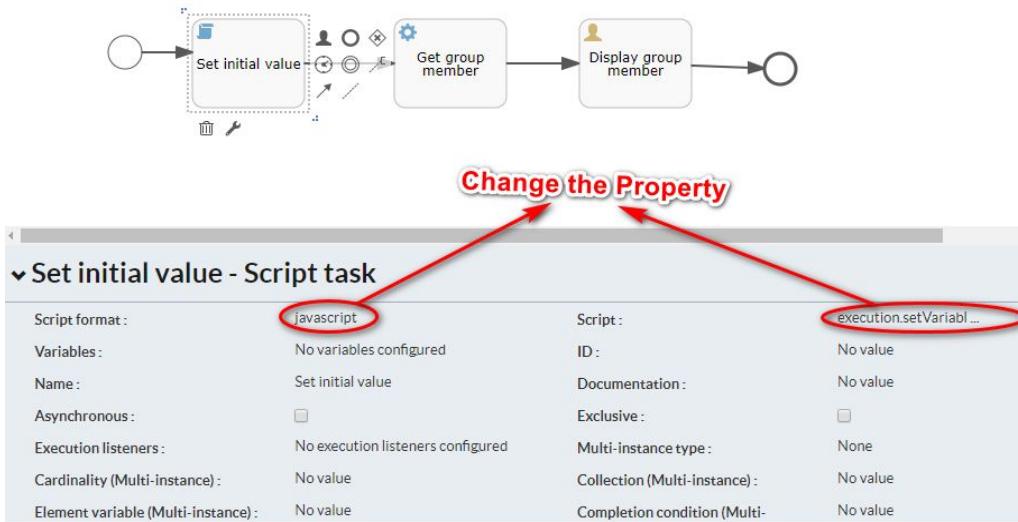
Variable name	Variable type
message	string
groupDetails	string
groupId	string

Buttons: +, -, Cancel, Save

### Global variable Table

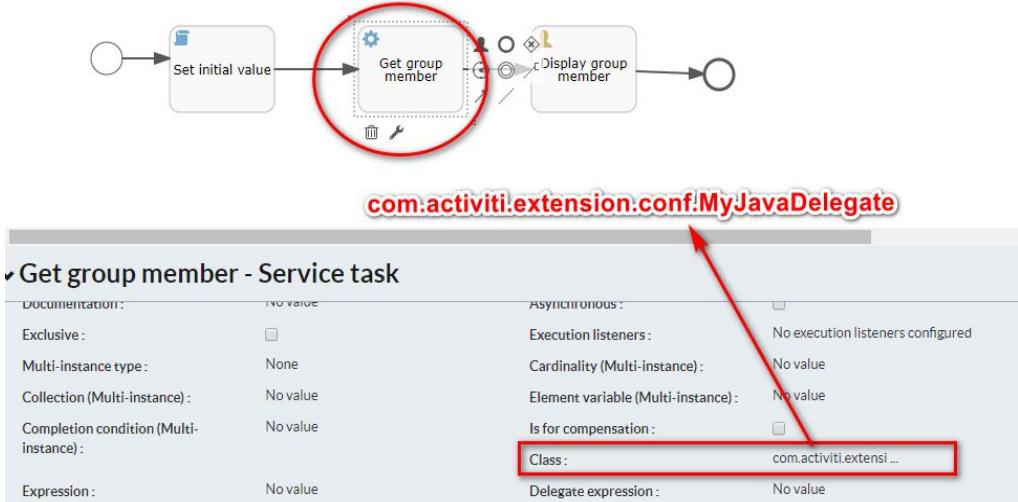
S.No	Variable name	Variable Type
1	message	string
2	groupDetails	string
3	groupId	string

## Set Initial value:

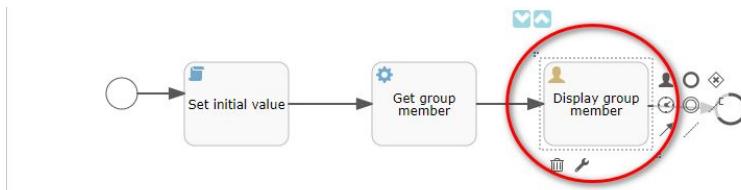


S.No	Property	Value
1	Script format	javascript
2	Script	execution.setVariable("groupId", 3003);

## Get Group Members:



## Display Group Members:



**Display group member - User task**

Completion condition (multi-instance):	Two tasks	Is for compensation:	<input type="checkbox"/>
Assignment:	Process initiator	Allow email notifications:	<input type="checkbox"/>
Referenced form:	Display Group Details	Form key:	No value
Due date:	No due date	Form field to data model mapping:	No mappings configured
Category:	No value	Priority:	No value
Task listeners:	No task listeners configured	Form properties:	No form properties configured
		Email template:	Default

**Forms** (highlighted with a red circle)

**Display Group Details**

Version 49 Last updated by Administrator, Today at 12:34 PM

- Design
- Tabs
- Outcomes
- Style
- Javascript
- Properties
- Variables

Show value of "message"      Show value of "groupDetails"

**Edit display value field 'Message'**

General      Visibility      Style

Label: Message

Form field      Variable

message - message - string

Colspan: 1

**Close**

Edit display value field 'groupDetails'

General      Visibility      Style

Label:  
groupDetails|

Form field	Variable
------------	----------

groupDetails - groupDetails - string

Colspan:  
1

**Close**

Processes      Forms      Decision Tables      **Apps**      Data Models      Stencils      Service...      Administrator

App definition details: ServiceTask Example

PREVIEW

Icon  
\* Change icon...

Theme  
Change theme...

Publish app directly to dashboard for the following users/groups

+ Add another person    + Add another group

No publish identities defined

Models included in the app definition

ServiceTask Example

**Edit included models**

v61

```

graph LR
    Start(( )) --> A[ ]
    A --> B[ ]
    B --> C[ ]
    C --> End(( ))
  
```

### Backend Code:

**Make the above java code to jar file and put on the folder path:**

C:\Program Files (x86)\alfresco\process-services-1.9.0.3\tomcat\webapps\activiti-app\WEB-INF\lib

### Output:

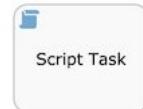
The screenshot shows the Activiti App interface with the following details:

- URL:** localhost:8080/activiti-app/workflow/#/apps/14014/tasks?taskId=482511
- Header:** Apps, APS, ACS REST API, APS REST API, ServiceTask Exam..., Administrator
- Left Sidebar:** Tasks, Processes, Reports, START, NEW FILTER, INVOLVED TASKS, MY TASKS, QUEUED TASKS, COMPLETED TASKS.
- Task Details:**
  - Title:** Display group member
  - Created:** an hour ago by Administrator
  - Assignee:** Administrator
  - Due:** No due date
  - Part of process:** ServiceTask Example - June 6th 2019
  - Ended:** an hour ago
  - Duration:** 17 seconds
  - Status:** No people involved, No groups involved, No content items, No comments, No checklist
  - Buttons:** SHOW DETAILS, Audit Log, COMPLETE
- Message:** Service Task Response
- groupDetails:**

```
[{"size":2,"total":2,"start":0,"data":[{"id":2002,"firstName":"Vasan","lastName":"M","email":"vasanmalai@gmail.com"}, {"id":1001,"firstName":"Malai","lastName":"D","email":"malaivasan.d@gmail.com"}]]
```

## 2.3 Script Task

A script task defines a JavaScript script or other script language (JSR-223 compatible language) that is executed when a process instance executes this step. A script task is visualized as a rounded rectangle with a paper icon inside.



### Example:

For script task, change the properties in both the files.

1. C:\Program Files (x86)\alfresco\process-services-1.9.0.3\tomcat\lib\activiti-app.properties
2. C:\Program Files (x86)\alfresco\process-services-1.9.0.3\tomcat\webapps\activiti-app\WEB-INF\classes\META-INF\activiti-app\activiti-app.properties

```
validator.editor.bpmn.disable.executionlistener=false
validator.editor.bpmn.disable.cameltask=false
validator.editor.bpmn.disable.muletask=false
validator.editor.bpmn.disable.mailtask=false
validator.editor.bpmn.disable.scripttask=false
validator.editor.bpmn.disable.scripttask.groovy=false
validator.editor.bpmn.disable.manualtask=false
validator.editor.bpmn.disable.businessruletask=false
validator.editor.bpmn.disable.servicetask=false
validator.editor.bpmn.disable.intermediatethrowevent=false
validator.editor.bpmn.disable.startevent.timer=false
```

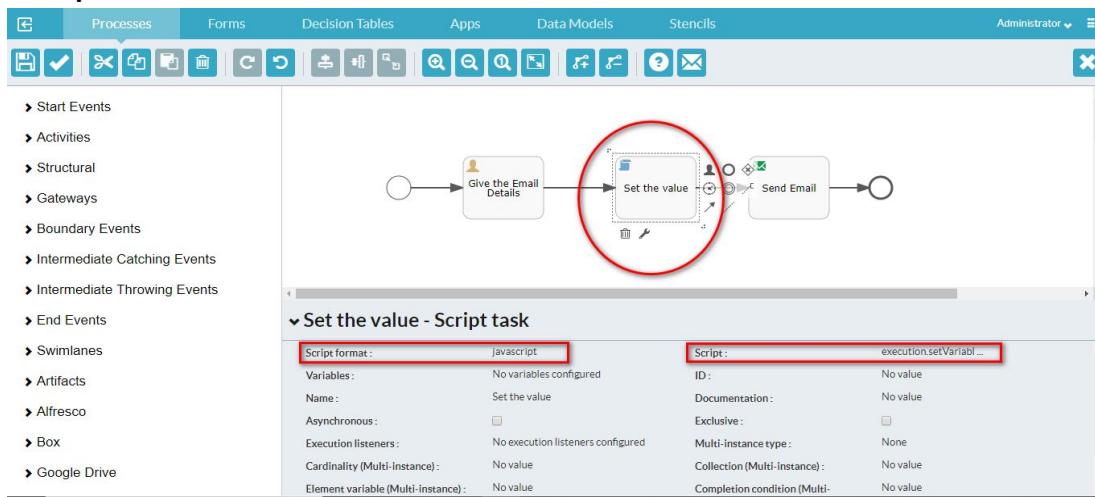
```

validator.editor.bpmn.disable.startevent.signal=false
validator.editor.bpmn.disable.startevent.message=false
validator.editor.bpmn.disable.startevent.error=false
validator.editor.bpmn.disable.startevent.timecycle=false
validator.editor.bpmn.disable.loopback=false
validator.editor.bpmn.limit.servicetask.only-class=false
validator.editor.bpmn.limit.multiinstance.loop=false
validator.editor.bpmn.limit.usertask.assignment.only-idm=false
editor.form.javascript.disable=false
javascript.secure-scripting.enabled=false
## only classes in activiti/javascript-whitelist-classes.conf will be allowed
javascript.secure-scripting.enable-class-whitelisting=true
## max stack depth to avoid stackoverflow exceptions
javascript.secure-scripting.max-stack-depth=100
## max execution time in milliseconds
javascript.secure-scripting.max-execution-time=3000
## max memory in bytes (eg. 5242880 = 5MB)
javascript.secure-scripting.max-memory=5242880
## number instructions to execute before checking max memory & max execution time
javascript.secure-scripting.instructions-before-callback=100

```

Uncomment all the properties in  
process-services-1.9.0.3\tomcat\webapps\activiti-app\WEB-INF\classes\activiti\whitelisted-scripts.conf

### Example

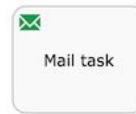


### Change the property for

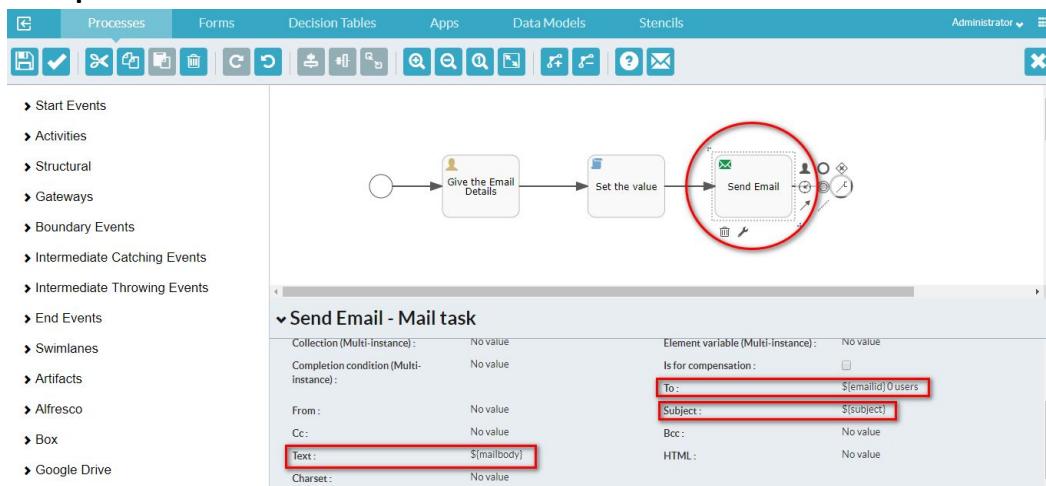
S.No	Property	Value
1	Script Format	javascript
2	Script	execution.setVariable("emailid", 'malaivasan.d@muraai.com');

## 2.4 Mail Task

You can enhance your business process with this automatic mail service task that sends emails to one or more recipients. The task supports normal email features such as cc lists, bcc lists, and HTML content. The mail task is depicted as a rounded rectangle with an envelope icon in the top-left corner.



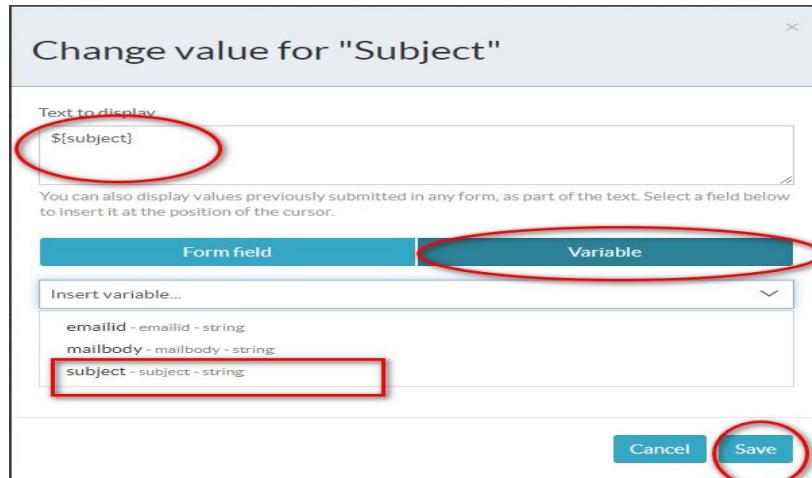
### Example



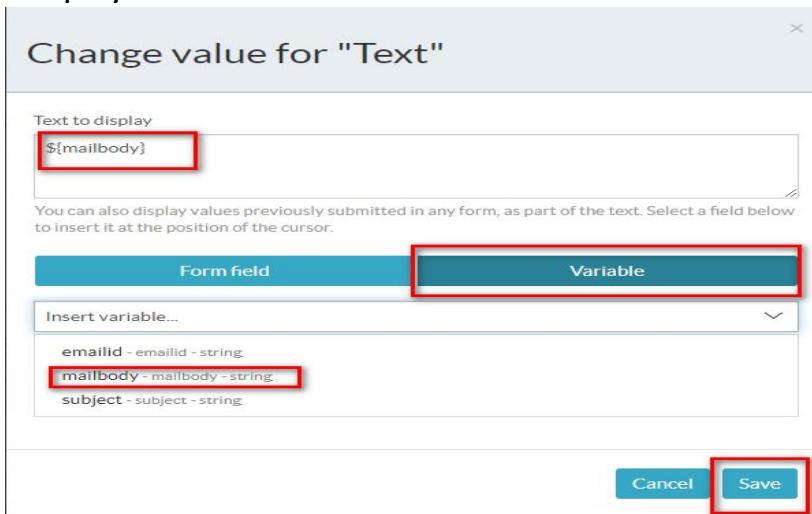
### Change the To Property:

This screenshot shows the 'Mail Recipients' dialog box. At the top, there are tabs for 'Identity store' and 'Fixed values'. The 'Fixed values' tab is selected and highlighted with a red oval. In the 'Expression' section, there is a text input field containing \${emailid}, which is also highlighted with a red box. At the bottom right of the dialog, there are 'Cancel' and 'Save' buttons, with the 'Save' button being highlighted with a red oval.

### Change the Subject Property:



### Change the Text Property:

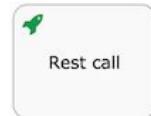


### Output

The screenshot shows a task card titled 'Give the Email Details'. The card displays two input fields: 'Subject' (containing 'Hi') and 'Mail Body' (containing 'Welcome'). At the bottom right of the card are two buttons: 'SAVE' and 'COMPLETE', with 'COMPLETE' circled in red. The left sidebar of the application shows navigation tabs for 'Tasks', 'Processes', and 'Reports', along with sections for 'INVOLVED TASKS', 'MY TASKS', 'QUEUED TASKS', and 'COMPLETED TASKS'. The top right corner shows the user 'Administrator' and a 'User and Mail and...' link.

## 2.5 Rest Call Task

The rest call task is used to communicate with a REST endpoint. The endpoint can be defined in the process definition, or it can be defined company-wide by an administrator. In the latter case, a logical name is all that is needed. A rest call task is visualized as a rounded rectangle with a rocket icon the top-left corner.



### Configure Endpoints

Go to Identity management -->Tenants-->Endpoints

The screenshot shows the Activiti Identity Management interface. The top navigation bar has tabs: Tenants (circled in red), Users, Capabilities, Organization, Personal, and Administrator. The main content area is for the 'alfresco.com' tenant, which is active. On the left sidebar, the 'Endpoints' item is highlighted with a red box. In the main panel, under 'Endpoints', there is a table with one row: Name: APS, Host: localhost, Path: /, Basic Auth Config: aps\_admin. Under 'Basic Auths', there is a table with one row: Name: aps\_admin, Username: admin@app.activiti.com. A modal window titled 'Edit Basic Auth configuration' is open at the bottom, showing fields for Name (aps\_admin), Username (admin@app.activiti.com), and Password (redacted). An arrow points from the word 'Admin' to the password field. The 'Save' button in the modal is also highlighted with a red box.

Edit endpoint

Name:	APS
Protocol:	HTTP
Host:	localhost
Port:	8080
Path:	/
Basic Auth configuration:	aps_admin

**Request Headers**

**Cancel** **Save**

## Example

### Create Process and property variable

Processes

Forms Decision Tables Apps Data Models Stencils Administrator

Start Events Activities Structural Gateways Boundary Events Intermediate Catching Events Intermediate Throwing Events End Events Swimlanes Artifacts Alfresco Box Google Drive

**Rest call task Example**

Process identifier:	RestcalltaskExample	Name:	Rest call task Exam ...
Documentation:	No value	Process author:	No value
Process version string (documentation only):	No value	Target namespace:	http://www.activiti..._
Event listeners:	No event listeners configured	Execution listeners:	No execution listeners configured
Message definitions:	No message definitions configured	Signal definitions:	No signal definitions configured
Metadata:	No properties defined	Variables:	3 variables configured

### Change value for "Variables"

Variable name	Variable type
groupId	string
message	string
groupDetails	string

Variable name \*  
groupId

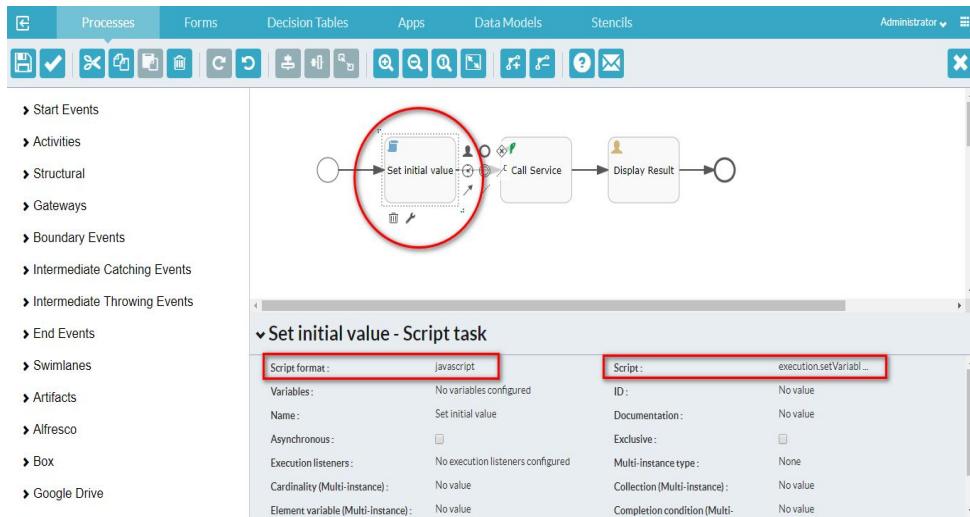
Variable type  
string

Map variable with data model entity?

**+** **-**

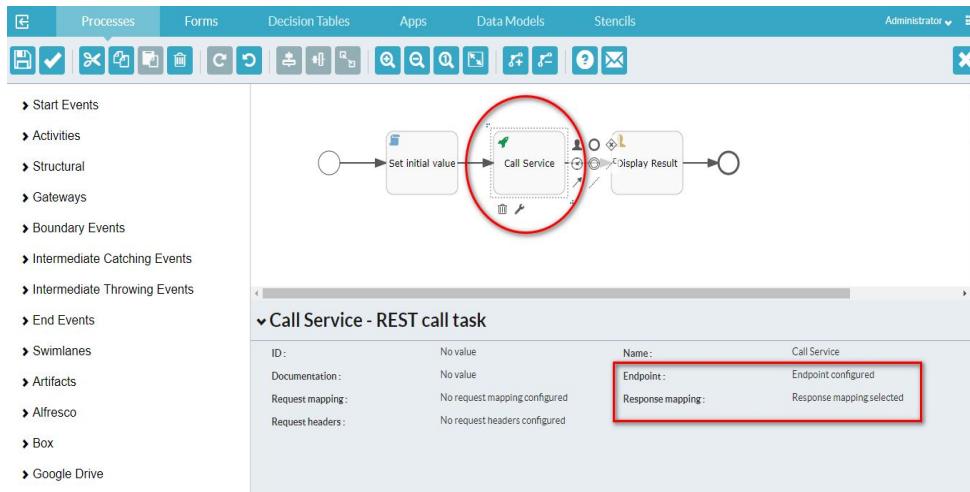
**Cancel** **Save**

## Set initial value



S.No	Property	Value
1	Script format	javascript
2	Script	execution.setVariable("groupId", 3003); execution.setVariable("message", 'Welcome');

## Call Service



## Change the property for Endpoint

### Change value for "Endpoint"

http://localhost:8080/activiti-app/api/enterprise/groups/\${groupId}/users Test

**HTTP method**  
GET

**Base endpoint**  
APS (http://localhost:8080/)

**Rest URL**  
To add URL parameters, use the Request configuration tab.  
activiti-app/api/enterprise/groups/\${groupId}/users

**Form field** | **Variable**

You can also display variables as part of the REST URL. Select a field below to insert it at the position of the cursor.

**Insert variable...**

groupDetails - groupDetails - string  
groupId - groupId - string  
message - message - string

Save

### Change the property for Response mapping

### Change value for "Response mapping"

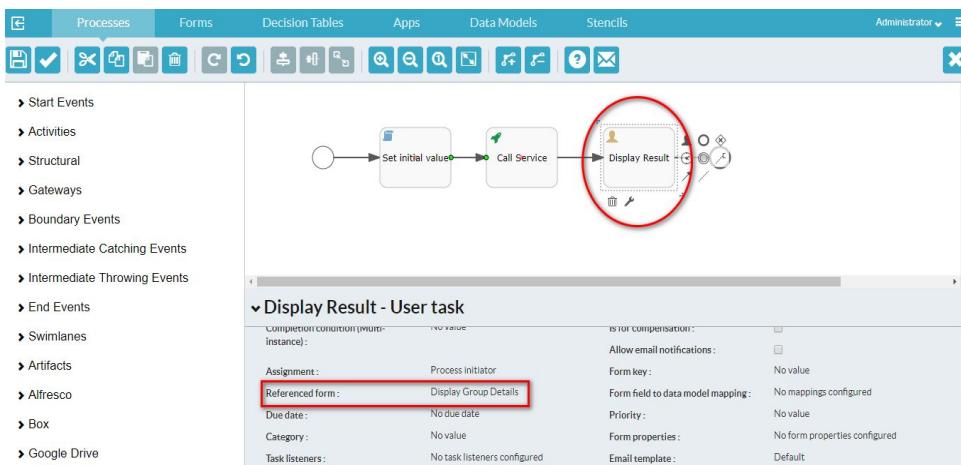
<b>JSON property</b>	<b>Variable</b>
data	groupDetails

+ -

<b>Property name:</b>	data
<b>Variable type:</b>	string
<b>Variable name:</b>	groupDetails

Save

### Display Result



## Display Group Details Form

The screenshot shows the Appian Designer interface with the 'Forms' tab selected. A modal window titled 'Display Group Details' is open, showing two input fields: 'Show value of "message"' and 'Show value of "groupDetails"'. The 'Variables' tab is highlighted in red.

### Edit display value field 'message'

The screenshot shows the 'Edit display value field 'message'' dialog. The 'Variable' tab is selected and highlighted in red. The 'message' variable is selected in the dropdown.

[Close](#)

### Edit display value field 'groupDetails'

The screenshot shows the 'Edit display value field 'groupDetails'' dialog. The 'Variable' tab is selected and highlighted in red. The 'groupDetails' variable is selected in the dropdown.

[Close](#)

## Output

## 2.6 Decision Task

You use a decision task to select a decision table while designing your process model. A decision table enables you to define a set of business rules that will be applied when it's executed. See the [LINKHERE](#) section for more information. A decision task is depicted as a rounded rectangle with a table icon the top-left corner.



### Example

For Decision task, change the property in both the files.

1. C:\Program Files (x86)\alfresco\process-services-1.9.0.3\tomcat\lib\activiti-app.properties
2. C:\Program Files (x86)\alfresco\process-services-1.9.0.3\tomcat\webapps\activiti-app\WEB-INF\classes\META-INF\activiti-app\activiti-app.properties

**validator.editor.dmn.expression=true**

### Create Process And Add the property for variables

Process variable	Description
Process author	No value
Target namespace	http://www.activiti...
Execution listeners	No execution listeners configured
Signal definitions	No signal definitions configured
Variables	7 variables configured

### Change value for "Variables"

Variable name	Variable type
total	integer
avg	integer
name	string
m1	string
m2	string
m3	string

Variable name \*

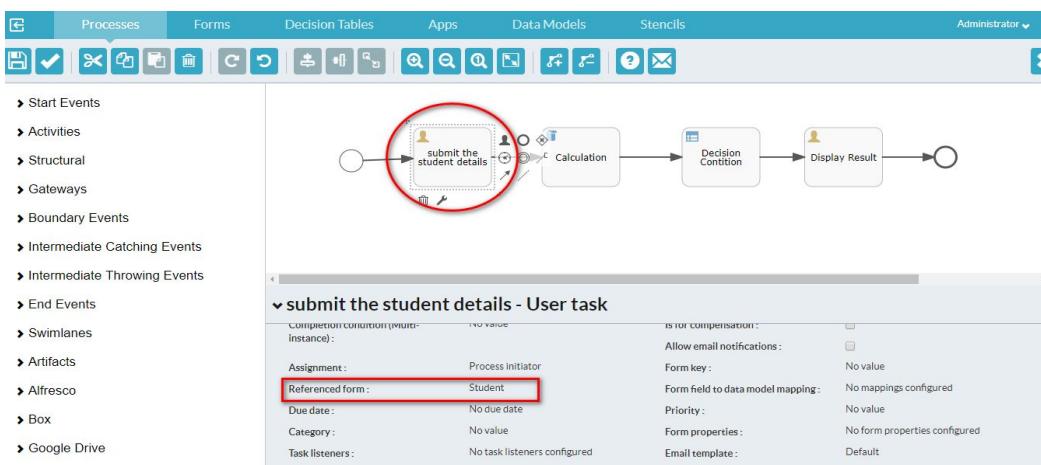
Variable type

integer

Map variable with data model entity?

**Cancel** **Save**

### Submit the student details



### Student Form

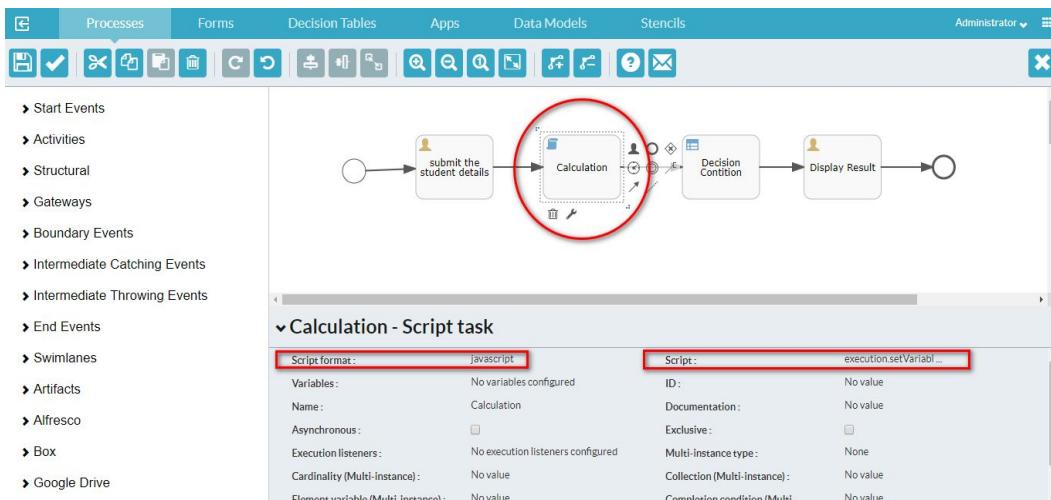
The screenshot shows a form editor for a 'Student' form. The left sidebar lists various input types:

- Text
- Multi-line text
- Number
- Checkbox
- Date
- Date and time
- Dropdown
- Typeahead
- Amount
- Radio buttons
- People
- Group of people
- Dynamic table
- Hyperlink
- Header

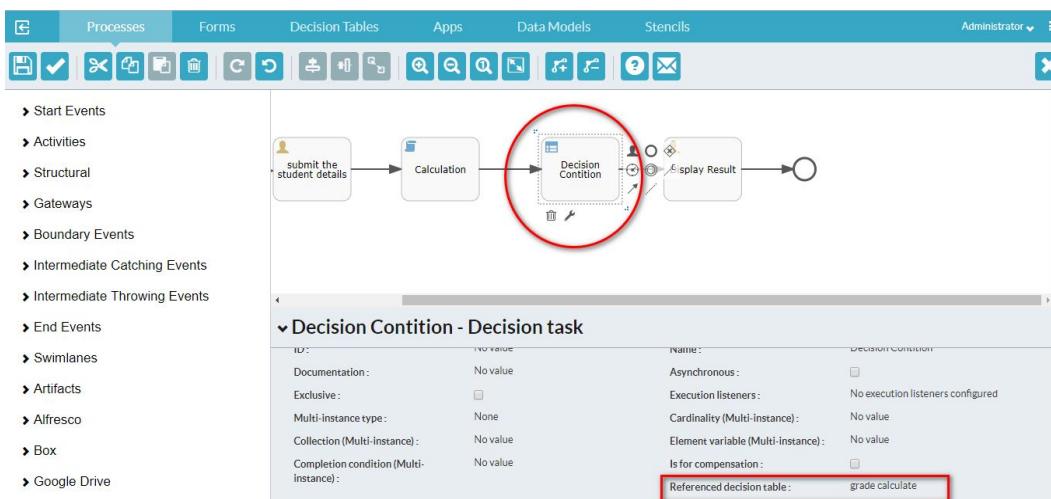
The main area shows the 'Student' form design with the following structure:

Name	Mark 1
Text	123
Mark 2	Mark 3
123	123

## Marks Calculation



S.No	Property	Value
1	Script Format	javascript
2	Script	<pre>execution.setVariable("total", execution.getVariable("mark1") + execution.getVariable("mark2") + execution.getVariable("mark3")); execution.setVariable("avg", execution.getVariable("total") / 3);</pre>



## Grade Calculation

Processes   Forms   **Decision Tables**   Apps   Data Models   Stencils   Administrator ▾

Hit Policy: First (Single pass)   **Add input**   **Add output**

	Average [avg]	Grade [grade]
1	> 90	"S+ Grade"
2	> 80	"S Grade"
3	>= 50	"B Grade"
4	< 50	"F Grade"

Remove rule   Move up   Move down   Add rule

### Edit input column

Select input variable as input for the column

Column label:

Average

Column type:

Existing

Create new

Variable type:

Form field

Variable

Variable:

avg - avg - integer

Cancel

**Save**

### Edit output column

Select an existing output variable or create a new one

Column label:

Grade

Column type:

Existing

Create new

Variable type:

Form field

Variable

Variable:

grade - grade - string

Cancel

**Save**

## Add Rules

Edit rule expression for column "Average"

Operator:	Greater than
Variable type:	Number
Number:	90
Calculation:	Method:

Cancel OK

Edit rule expression for column "Grade"

Variable type:	String
String:	S+ Grade

Cancel OK

## Display Result

Processes Forms Decision Tables Apps Data Models Stencils Administrator

Start Events Activities Structural Gateways Boundary Events Intermediate Catching Events Intermediate Throwing Events End Events Swimlanes Artifacts Alfresco Box Google Drive

Display Result - User task

Completion condition (when instance):	Value
Assignment:	Process initiator
Referenced form:	Student Display
Due date:	No due date
Category:	No value
Task listeners:	No task listeners configured
Is for compensation:	<input type="checkbox"/>
Allow email notifications:	<input type="checkbox"/>
Form key:	No value
Form field to data model mapping:	No mappings configured
Priority:	No value
Form properties:	No form properties configured
Email template:	Default

## Student Display Form

Processes Forms Decision Tables Apps Data Models Stencils Administrator Default form

**Student Display**  
Version 3 Last updated by Administrator, 05/31/2019

Design Tabs Outcomes Style Javascript Properties Variables

Show value of "name"  
Show value of "m1"  
Show value of "m2"  
Show value of "m3"  
Show value of "total"  
Show value of "avg"  
Show value of "grade"

**Edit display value field 'Grade'**

General Visibility Style

Label: Grade

Form field Variable

grade - grade - string

Colspan: 1

**Close**

## Output

Tasks Processes Reports + START Decision Example1 Administrator

**Nameless task**  
Assigned: Administrator Due: No due date Part of process: Decision Example1 - June 7th 2019  
No people involved No groups involved No content items No comments No checklist SHOW DETAILS

Name	Mark 1
Dharumaiyan	95
Mark 2	Mark 3
89	97

**OK**

**Nameless task**  
Assigned: Administrator Due: No due date Part of process: Decision Example1 - June 7th 2019  
No people involved No groups involved No content items No comments No checklist SHOW DETAILS

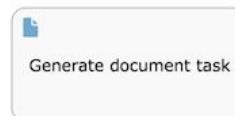
name	m1
Dharumaiyan	95
m2	m3
89	97
total	avg
201	93.66666666666667
Grade	S= Grade

**OK**

For more examples, refer this [link](#).

## 2.7 Generate Document Task

The generate document task generates a document (Word or PDF) and stores the reference to the document as a process variable. The document is based on a (Word) template that describes how the document needs to be rendered, using process variables and various constructs (such as if-clauses and loops). See the Developer Guide on how to use the generate document task. A generate document task appears as a rounded rectangle with a document icon on the top-left corner.



### Example

Create Document Template and upload the document to  
**Identity management-->Tenants-->Document templates**

A screenshot of the Alfresco tenant management interface. The top navigation bar shows tabs for Tenants (circled in red), Users, Capabilities, Organization, and Personal. The right side of the bar shows "Administrator". The main content area is titled "alfresco.com" and "Domain: default-tenant Active: ✓". It shows a search bar and a list of "Found 2 matching template(s), showing 1 to 2". The list includes "College ID.docx" uploaded on 05/31/2019 and "Test Document Generate Task.docx" uploaded on 05/30/2019. Each item has a context menu with options like "Upload new version", "Download", and "Delete".

For more document template related information, refer this [link](#)

### Create Process And Add Variables Property

A screenshot of the Activiti process editor. The top navigation bar shows tabs for Processes (circled in red), Forms, Decision Tables, Apps, Data Models, and Stencils. The right side shows "Administrator". The left sidebar lists process elements: Start Events, Activities, Structural, Gateways, Boundary Events, Intermediate Catching Events, Intermediate Throwing Events, End Events, Swimlanes, Artifacts, Alfresco, Box, and Google Drive. The main workspace shows a process flow: a start event leads to a "Fill the Student Details" activity, which then leads to a "Value Initialization" activity, which finally leads to a "pdf generate" activity. Below the workspace is a panel titled "Test Doc Generate" with the following details:

Process identifier :	TestDocGenerate	Name :	Test Doc Generate
Documentation :	No value	Process author :	No value
Process version string (documentation only):	No value	Target namespace :	http://www.activiti...
Event listeners :	No event listeners configured	Execution listeners :	No execution listeners configured
Message definitions :	No message definitions configured	Signal definitions :	No signal definitions configured
Metadata :	No properties defined	Variables :	14 variables configured

The "Variables" row is highlighted with a red box.

Change value for "Variables"

Variable name	Variable type
name	string
course	string
rollno	string
batch	string
year	string
email	string

Variable name \*

Variable type

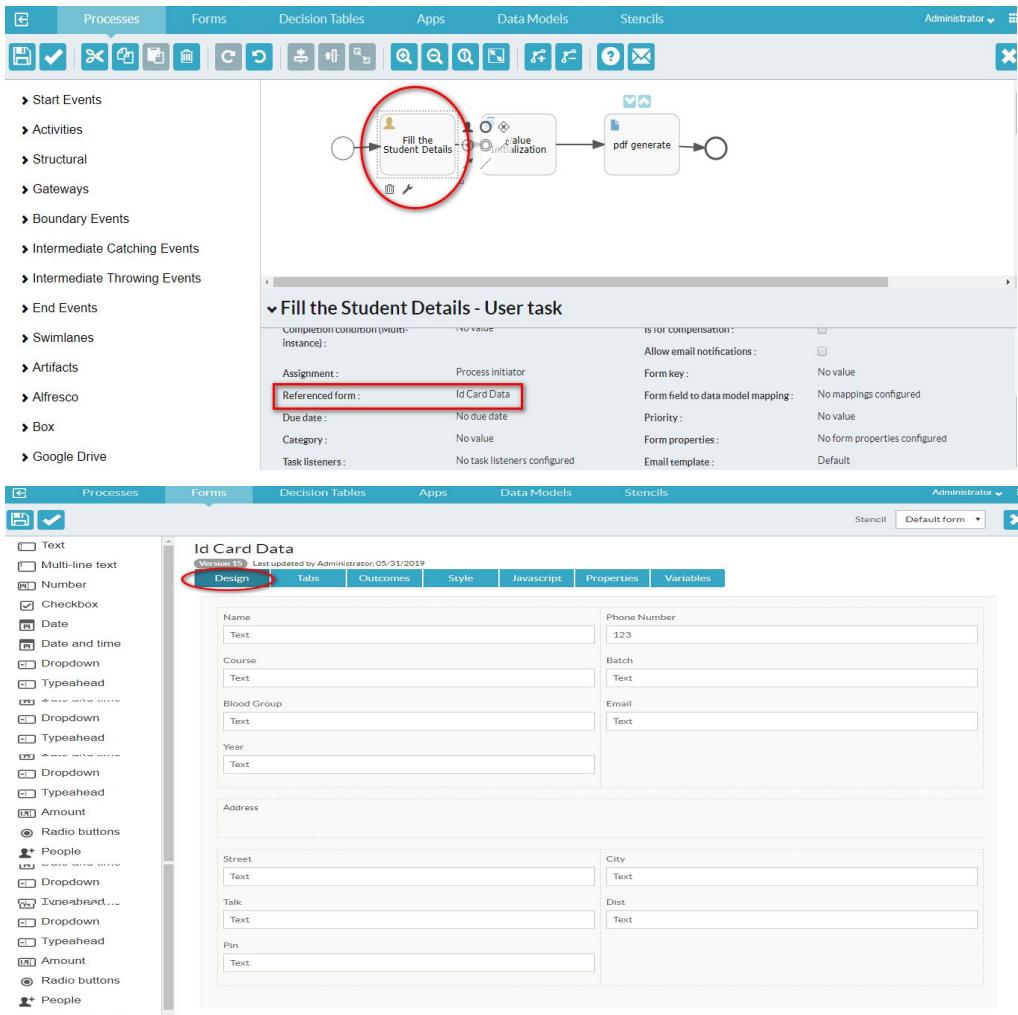
Map variable with data model entity?

**+** **-**

**Cancel** **Save**

S.No	Variable name	Variable type
1	name	string
2	course	string
3	rollno	string
4	batch	string
5	year	string
6	email	string
7	blood	string
8	phone	string
9	street	string
10	post	string
11	talk	string
12	dist	string
13	pin	string

**Fill student Details**



## Map Variables

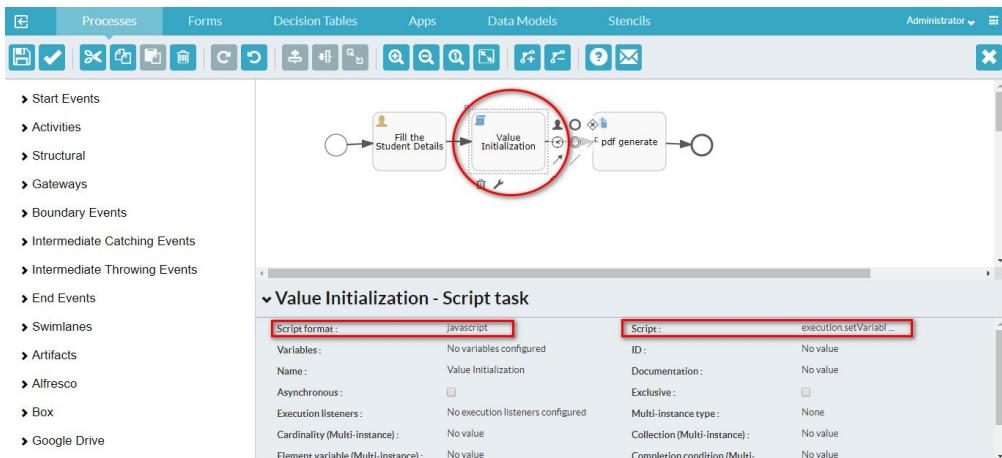
The screenshot shows the "Change value for 'Referenced form'" dialog. It has a table for mapping variables:

Source process variable	Form variable	Target process variable
	batch	batch
	bloodgroup	blood
	post	post
	course	course
	dist	dist
	email	email

Below the table, there are tabs for Mapping type: In/Out mapping, Input mapping, and Output mapping. The "Output mapping" tab is selected. A "Form field or variable" input field contains "Dist - dist - text". Under Value type, there are tabs for New Variable, Form field, and Variable. The "Variable" tab is selected. A "Target process variable" input field contains "dist - dist". At the bottom, there are buttons for < Select Form, Cancel, and Save, with the "Save" button highlighted by a red box.

S.No	Source process variable	Form variable	Target process variable
1		name	name
2		course	course
3		rollno	rollno
4		batch	batch
5		year	year
6		email	email
7		blood	blood
8		phone	phone
9		street	street
10		post	post
11		talk	talk
12		dist	dist
13		pin	pin

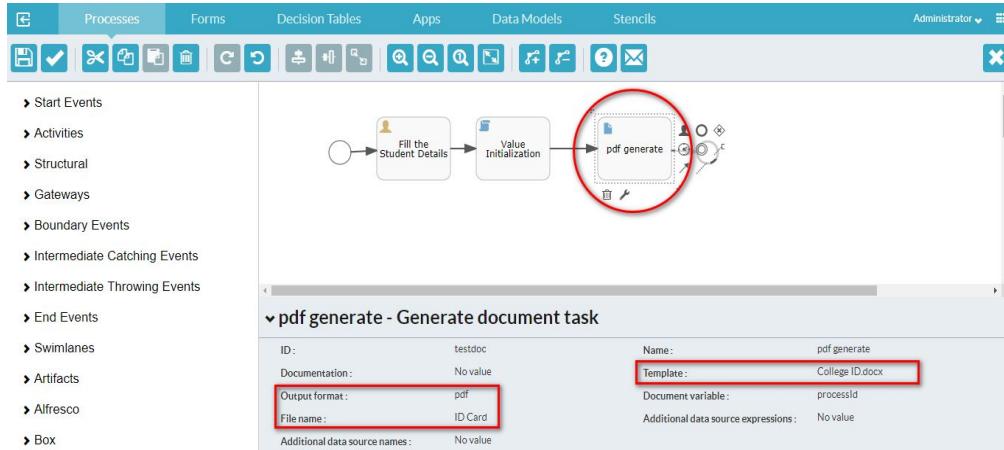
## Value Initialization



S.No	Property	Value
1	Script format	javascript

2	Script	execution.setVariable("rollno","R-"+execution.getProcessInstanceId());
---	--------	--

## Generate pdf



S.No	Property	Value
1	Output format	pdf
2	File Name	ID Card
3	Template	College ID.docx

## Ouput

The screenshot shows the Camunda BPM Task list interface. On the left, there are sections for INVOLVED TASKS (MY TASKS, QUEUED TASKS, COMPLETED TASKS) and COMPLETED TASKS. In the center, a specific task titled 'Nameless task' is selected. The task details form is displayed, showing the following information:

**Nameless task**

Assignee: Administrator Due: No due date Part of process: Test Doc Generate - May 31st 2019 Ended: 7 days ago Duration: 1 minutes

No people involved | No groups involved | No content items | No comments | No checklist | SHOW DETAILS | Audit Log

**Fields:**

- Name: Dharumaiyan
- Phone Number: 9512360487
- Course: BE
- Batch: II
- Blood Group: O+ve
- Email: dharumaiyan@gmail.com
- Year: 2005-2009
- Address:
  - Street: 2/25 New Street
  - City: Thirucherai City
  - Talk: Kumbakonam TK
  - Dist: Thanjavur Dist
  - Pin: 612605



## 2.8 Business Rule Task

A Business rule task executes one or more rules. It is mainly there for compatibility with Alfresco Process Services community. Alfresco recommends that you use Decision tables. See [LINKHERE](#) for more information. A business rule is depicted as a rounded rectangle with a table icon in the top-left corner.



## 2.9 Store Entity Task

Use the Store entity task to update data models or entities with process values such as variables or form fields. The updated entities can then be mapped to variables and used while creating processes.



## 2.10 Manual Task

A Manual Task defines a task that is external to Alfresco Process Services. You use it to model work done which the Process Engine does not know of. A manual task is handled as a pass-through activity, the Process Engine automatically continues the process from the instant process execution arrives at a manual task activity.

## 2.11 Receive Task

A Receive Task waits for the arrival of an external trigger. This trigger is sent programmatically (via Java or REST API). For process to process triggering, use the signal events. A receive task is visualized as a rounded rectangle with an envelope icon in the

top-left corner.

## 2.12 Camel Task

You use the Camel task to send messages to, and receive messages from Apache Camel. You can find more information on Apache Camel [here](#). Note that Camel is by default not installed and would need to be added by the system admin. A camel task is visualized as a rounded rectangle with a camel icon in the top-left corner.

## 2.13 Mule Task

Use the Mule task to send messages to the Mule ESB (Enterprise Service Bus). You can find more information on Mule ESB [here](#). Note that Mule is by default not installed and would need to be added by the system admin. A mule task is visualized as a rounded rectangle with the Mule logo in the top-left corner.

# 3. Structural Components:

You use structural components to group multiple components in a sub process to reuse in a parent process definition, and to embed and call other process definitions from inside your own process.

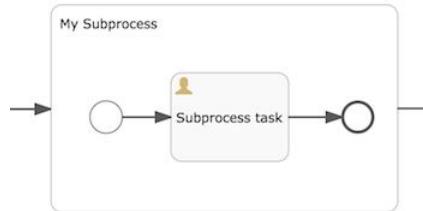
## 3.1 Sub-process

A sub process is a single activity that contains activities, gateways, and events which form a process. A sub process is completely embedded inside a parent process. You can use a sub process to create a new scope for events. Events that are thrown during execution of the sub process, can be caught by [Boundary events](#) on the boundary of the sub process, creating a scope for that event limited to just the sub process.

Sub-processes must have the following characteristics:

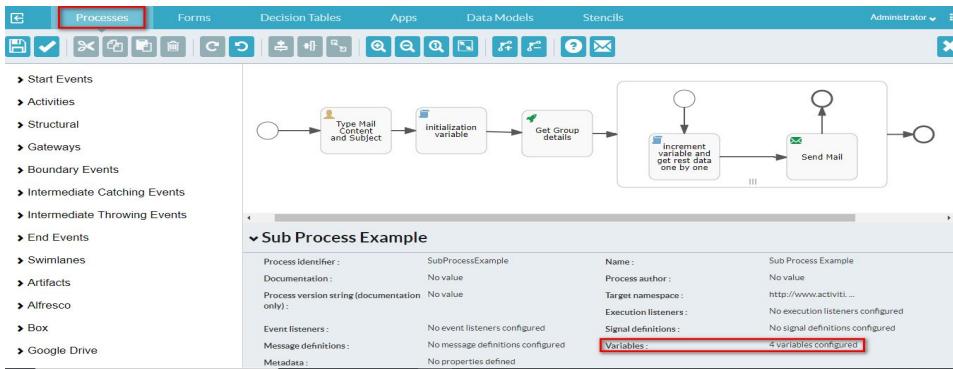
- A sub process has exactly one none start event. No other start event types are permitted. A sub process must have at least one end event.
- Sequence flow cannot cross sub process boundaries.

A sub-process is visualized as a rounded rectangle:



## Example

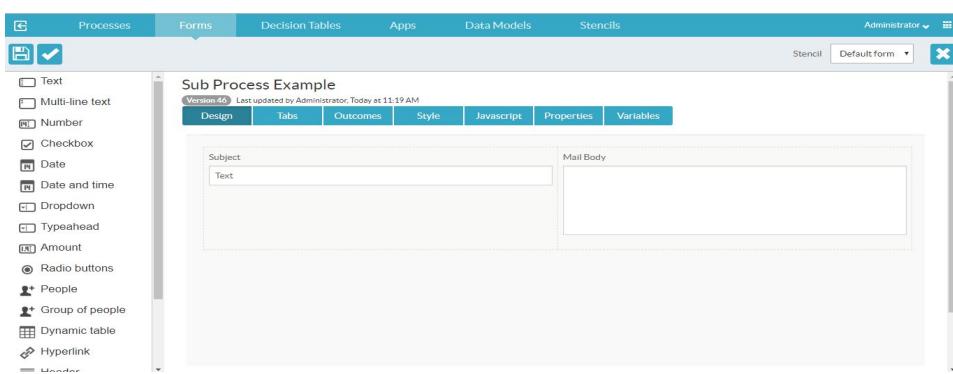
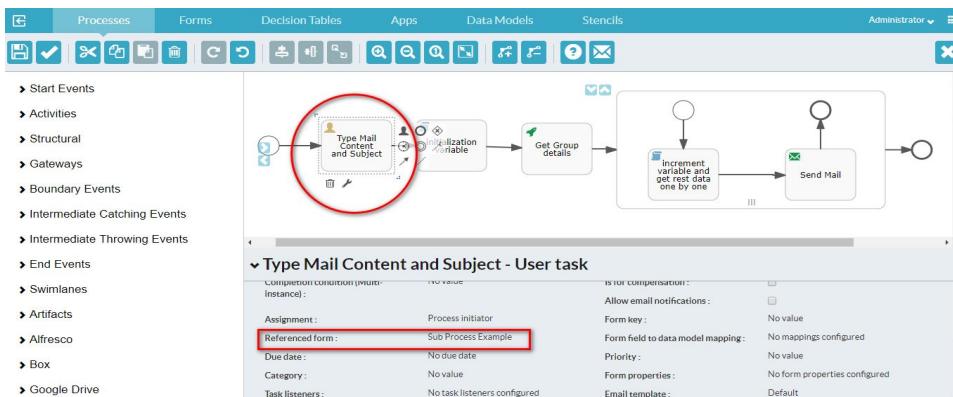
### Create Process and Add Variables Property



### Change value for variables

S.No	Variables name	Variable type
1	mailbody	string
2	groupId	integer
3	innerIndex	integer
4	subject	string

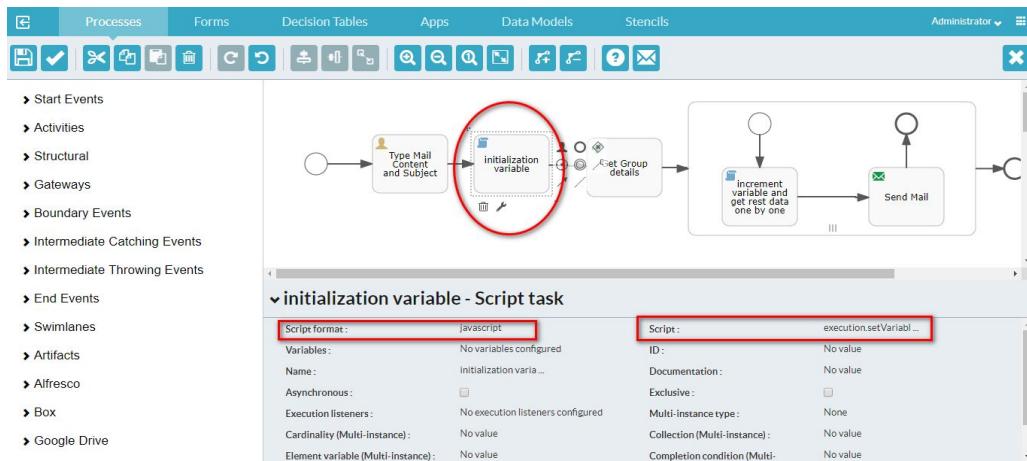
### Type Mail Content and Subject



## Map Variables

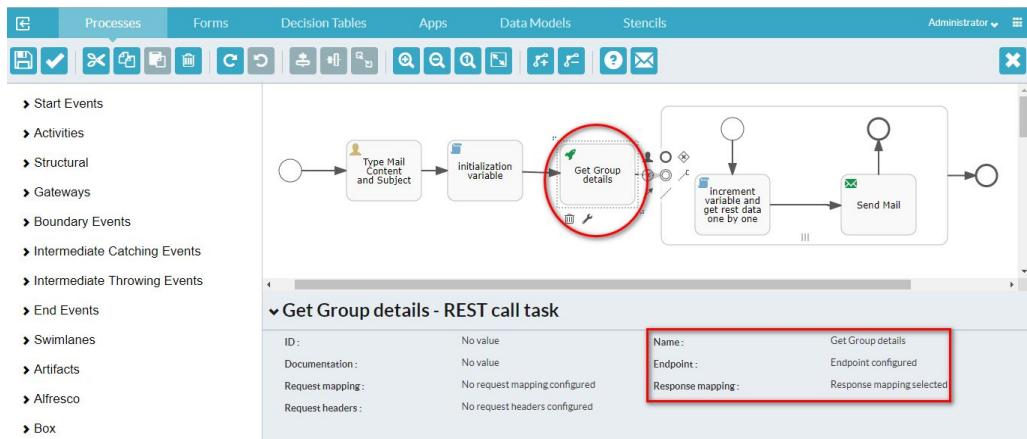
S.No	Source process variable	Form variable	Target process variable
1		mailbody	mailbody
2		subject	subject

## Initialization variable



S.No	Property	value
1	Script format	javascript
2	Script	execution.setVariable("groupId", 3003); execution.setVariable("innerIndex", 0);

## Get Group Details



## Change the Endpoint Property

Change value for "Endpoint"

http://localhost:8080/activiti-app/api/enterprise/groups/\${groupId}/users Test

HTTP method	GET
Base endpoint	APS (http://localhost:8080/)
Rest URL	activiti-app/api/enterprise/groups/\${groupId}/users

To add URL parameters, use the Request configuration tab.

activiti-app/api/enterprise/groups/\${groupId}/users

Form field Variable

You can also display variables as part of the REST URL. Select a field below to insert it at the position of the cursor.

Insert variable...

groupId - groupId - integer  
innerIndex - innerIndex - integer  
mailbody - mailbody - string  
subject - subject - string

Cancel Save

## Change Property Response mapping

Change value for "Response mapping"

JSON property	Variable
data	groupDetails
total	teamsize

Property name: data  
Variable type: string  
Variable name: groupDetails

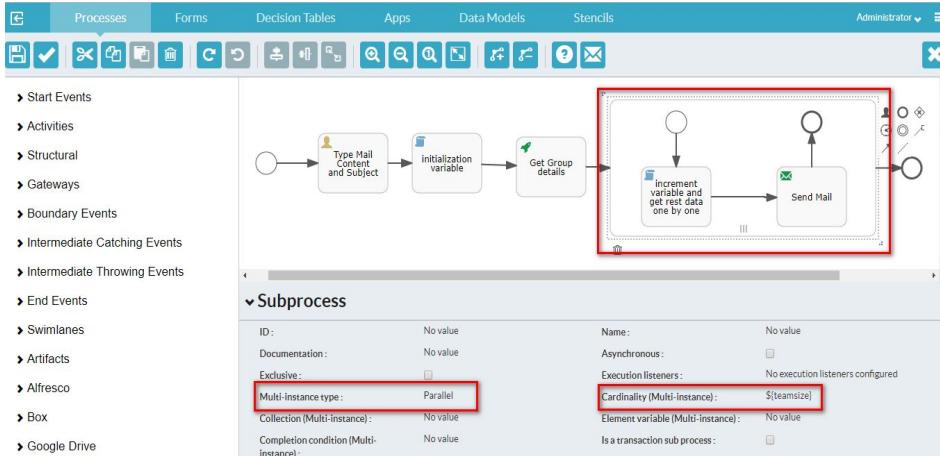
Cancel Save

Change value for "Response mapping"

JSON property	Variable
data	groupDetails
total	teamsize

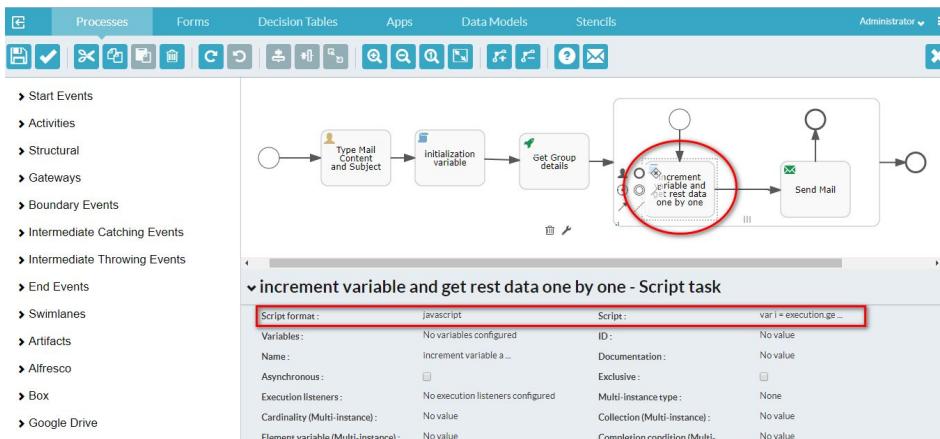
Property name: total  
Variable type: string  
Variable name: teamsize

Cancel Save



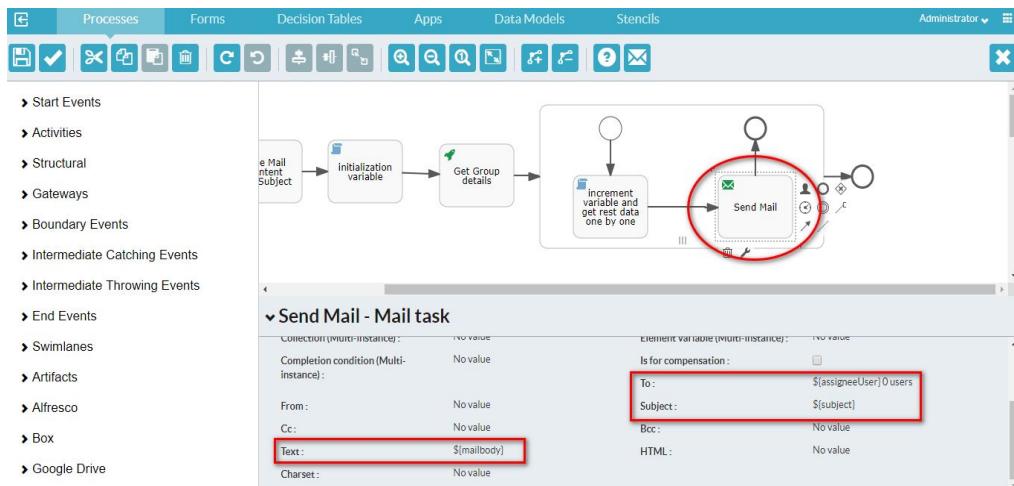
S.No	Property	Value
1	Multi-Instance type	Parallel
2	Cardinality (Multi-instance)	<b> \${teamszize} </b>

### Increment variable and get rest data one by one

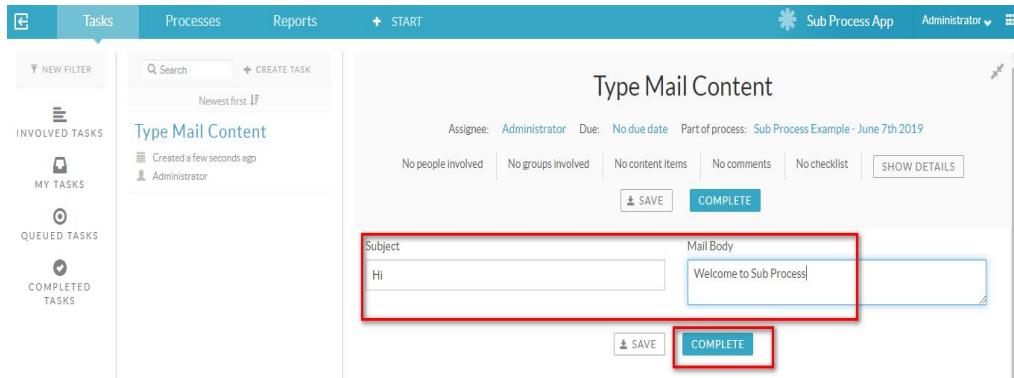


S.No	Property	value
1	Script format	javascript
2	Script	<pre>var i = execution.getVariable("innerIndex"); var temp = execution.getVariable("groupDetails"); var data = JSON.parse(temp); execution.setVariable("assigneeUser", data[i].email); i++; execution.setVariable("innerIndex", i);</pre>

## Send Mail



## Output



After press 'COMPLETE' , it will send mail to all the users in User group.

### 3.2 Collapsed sub-process

You use a collapsed sub-process to add an existing process from your available process definitions as a sub-process to the process definition you are currently editing. Note that during process instance execution, there is no difference between a collapsed or embedded sub-process. They both share the full process instance context.

When you drag a collapsed sub-process from the palette to your canvas, and click on the Referenced Subprocess property, you are presented with a visual list of the process definitions you have access to. You can choose from the list, and the chosen process will be added to the current process definition. Note the process chosen must have exactly one none start event, and no other start event type, and it must have at least one end event.



### 3.3 Event sub-process

An event sub-process is a sub-process that is triggered by an event. You can use an event sub-process in your main process, or in any sub-process. Your event sub-process does not have any incoming or outgoing sequence flows. An event sub-process is triggered by an event, so there can be no incoming sequence flow. You can find more information [here](#).

### 3.4 Call Activity

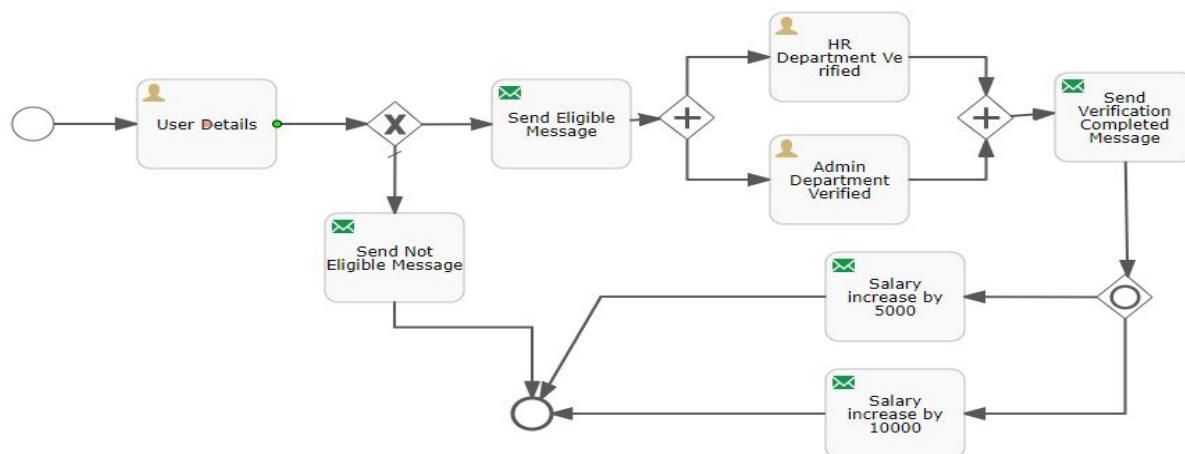
A call activity is used to execute another process definition as part of the current process instance. The main difference between a sub-process and a call activity is that the call activity does not share context with the process instance. Process variables are explicitly mapped between the process instance and the call activity.

## 4. Gateways:

You use gateways to control the flow of execution in your process. As the term gateway suggests, it is a gating mechanism that either allows or prevents passage of a token through the gateway. As tokens arrive at a gateway, they can be merged together on input and/or split apart on output from the gateway. A gateway is displayed as a diamond, with an icon inside. The icon depicts the type of gateway.

### Example

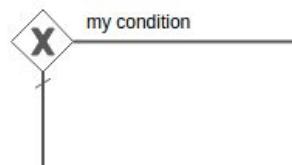
Here, we are taking the Employee Appraisal System as an example.



## 4.1 Exclusive Gateway

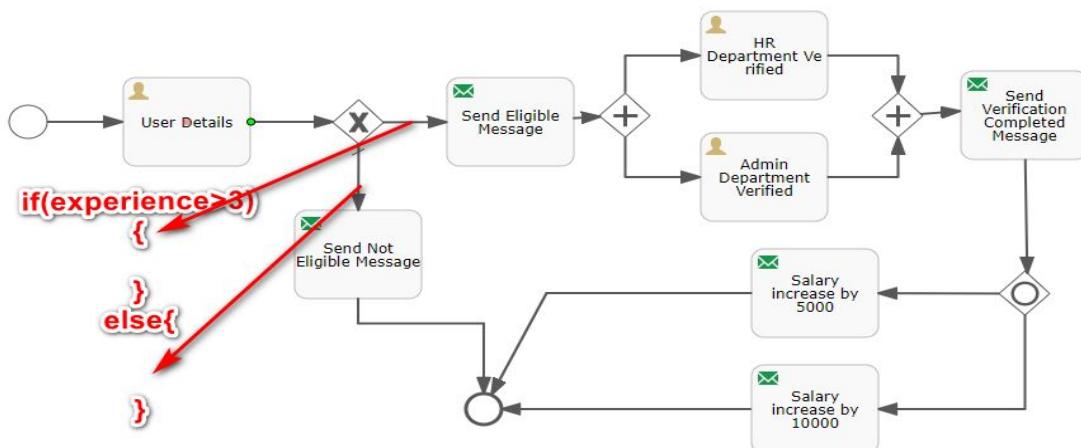
You use an exclusive gateway to model a decision in your process. When execution arrives at an exclusive gateway, the outgoing sequence flows are evaluated in the order in which they are defined. The first sequence flow whose condition evaluates to true, or which does not have a condition set, is selected and the process continues. Note that if no sequence flow is selected, an exception will be thrown.

An exclusive gateway is visualized as a diamond shape with an X inside.



### Example

Initially, an employee has to enter his details( name, salary, experience ) in User Details task. This data will go to an Exclusive Gateway.

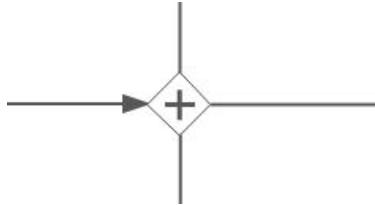


Depending on condition, it will either send eligible message or not eligible message to an employee.

## 4.2 Parallel Gateway

You use a parallel gateway to model concurrency in a process. It allows you to fork multiple outgoing paths of execution or join multiple incoming paths of execution. In a fork, all outgoing sequence flows are followed in parallel, which creates one concurrent execution for each sequence flow. In a join, all concurrent executions arriving at the parallel gateway wait at the gateway until an execution has arrived for every incoming sequence flow. Then the process continues past the joining gateway. Note that the gateway simply waits until the required number of executions has been reached and does not check if the executions are coming from different incoming sequence flow.

**Unlike other gateways, the parallel gateway does not evaluate conditions. Any conditions defined on the sequence flow connected with the parallel gateway are ignored.**



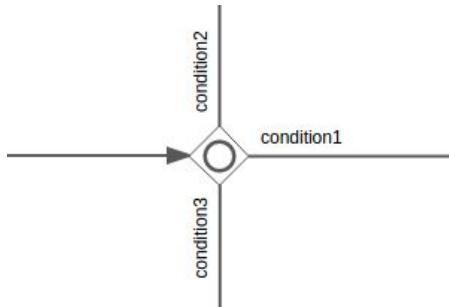
### **Example**

If it sent eligible message to an Employee, then through the parallel gateway, it will send Employees data to HR and Admin Department parallelly. After verification from both HR and Admin departments, again through the parallel gateway, it will send the Verification Completion message to an Employee.

### **4.3 Inclusive Gateway**

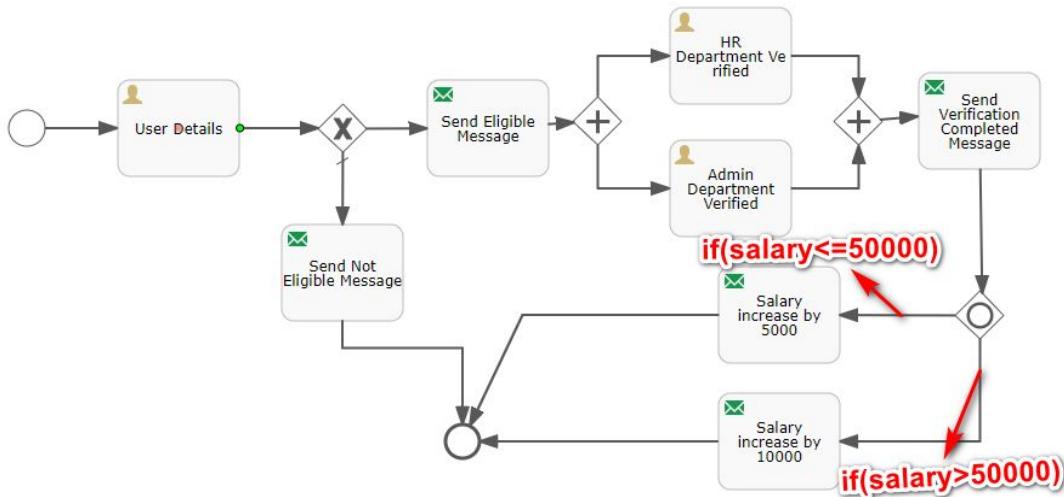
You use an inclusive to join and fork multiple sequence flows based on conditions. Like an exclusive gateway you can define conditions on outgoing sequence flows and the inclusive gateway will evaluate them, but an inclusive gateway can take more than one sequence flow, like the parallel gateway. All outgoing sequence flow conditions are evaluated. Every sequence flow with a condition that evaluates to true, is followed in parallel, creating one concurrent execution for each sequence flow.

Note that an inclusive gateway can have both fork and join behavior, in which case there are multiple incoming and outgoing sequence flows for the same inclusive gateway. The gateway will join all incoming sequence flows that have a process token, before splitting into multiple concurrent paths of executions for the outgoing sequence flows that have a condition that evaluates to true.



### **Example**

After Verification confirmation message sent to an Employee, the process is going to an Inclusive gateway.



Again depending on conditions, Appraisal will be given to an Employee.

**In Inclusive gateway, more than one outcomes might be there depending upon conditions.**

#### 4.4 Event based Gateway

You use an event gateway to route process flow based on events. Each outgoing sequence flow of the event gateway must be connected to an intermediate catching event. When process execution reaches an event gateway execution is suspended, and for each outgoing sequence flow, an event subscription is created. The flow for the event that occurs first, will be followed.

Outgoing sequence flows connect to an event gateway are never "executed", but they do allow the process engine to determine which events an execution arriving at an event-based gateway needs to subscribe to. The following restrictions apply to event gateways:

- The gateway must have two or more outgoing sequence flows.
- An event-based gateway can only be followed by intermediate catching events. Receive tasks after an event gateway are not supported by Alfresco Process Services.
- An intermediate catching event connected to an event gateway must have a single incoming sequence flow.

## 4. Boundary Events:

You use boundary events to handle an event associated with an activity. A boundary event is always attached to an activity.

While the activity the boundary event is attached to is active (meaning the process instance execution is currently executing it right there), the boundary event is listening for a certain type of trigger. When the event is caught, the activity is either interrupted and the sequence flow going out of the event is followed (interrupting behavior) or a new execution is created from the boundary event (non-interrupting behavior).

### 4.1 Boundary Error Event

A boundary error event catches an error that is thrown within the boundaries of the activity the event is based on and continues process execution from the event. A boundary error event is always interrupting. A boundary timer event is visualized as a circle with a lightning icon inside.



#### Example

##### Create Process and Add Variables

The screenshot shows the Activiti BPMN2.0 Modeler interface. The top navigation bar has 'Processes' selected. The main workspace displays a process diagram with various activities like 'Type Mail Content and Subject', 'Initialization variable', 'Get Group details', 'Get Loop Size', 'Get Group Member one by one', 'Waiting for Approval', 'Loop Increment', and boundary events marked with a lightning bolt symbol. Below the diagram, a section titled 'Boundary Error Event Example' shows the configuration for a boundary error event. The 'Variables' field is highlighted with a red box, showing '6 variables configured'.

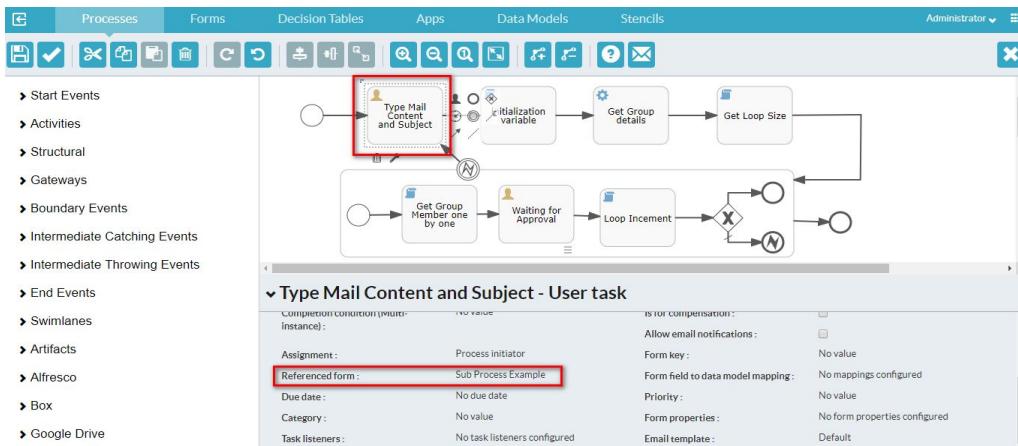
Process identifier:	BoundaryErrorEventEx...	Name:	Boundary Error Event ...
Documentation:	No value	Process author:	No value
Process version string (documentation only):	No value	Target namespace:	http://www.activiti... Execution listeners:
Event listeners:	No event listeners configured	Signal definitions:	No signal definitions configured
Message definitions:	No message definitions configured	Variables:	6 variables configured
Metadata:	No properties defined		

#### Change the Variables Property

S.No	Variable name	Variable type
1	subject	string
2	mailbody	string
3	index	string

4	groupId	string
5	groupDetails	string
6	groupSize	string

## Type Mail Content and Subject



## Map Variables

Change value for "Referenced form"

Source process variable	Form variable	Target process variable
mailbody	mailbody	mailbody
subject	subject	subject

Mapping type: In/Out mapping

Form field or variable: Mail Body - mailbody - multi-line-text

Value type: New Variable

Target process variable: mailbody - mailbody

Buttons: < Select Form, Cancel, Save

## Initialization variable

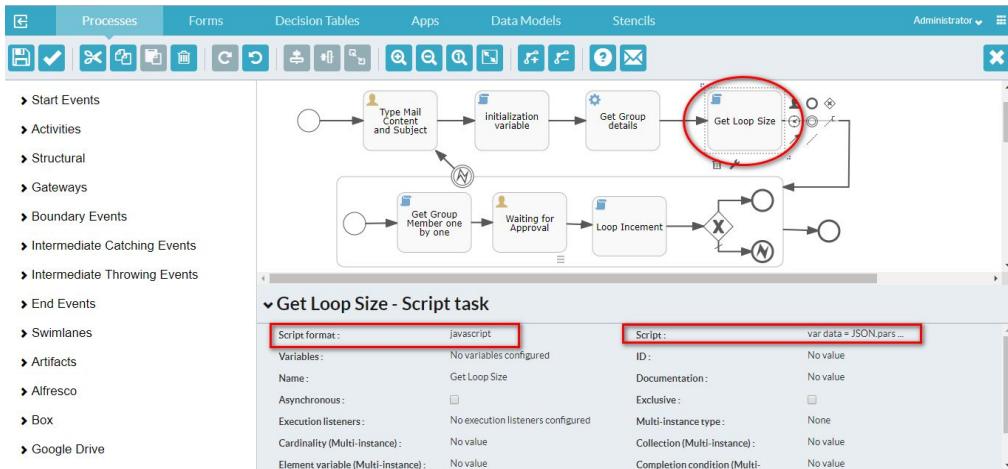
The screenshot shows the Activiti BPMN editor interface. On the left, there is a sidebar with various process-related options like Start Events, Activities, Structural, etc. The main area displays a BPMN process diagram. A specific task node labeled "initialization variable" is highlighted with a red circle. Below the diagram, a detailed view of the "initialization variable - Script task" is shown in a modal window. The "Script format" is set to "javascript" and the "Script" field contains the following code:

S.No	Property	Value
1	Script format	javascript
2	Script	<code>execution.setVariable("groupId", 3003); execution.setVariable("index", 0);</code>

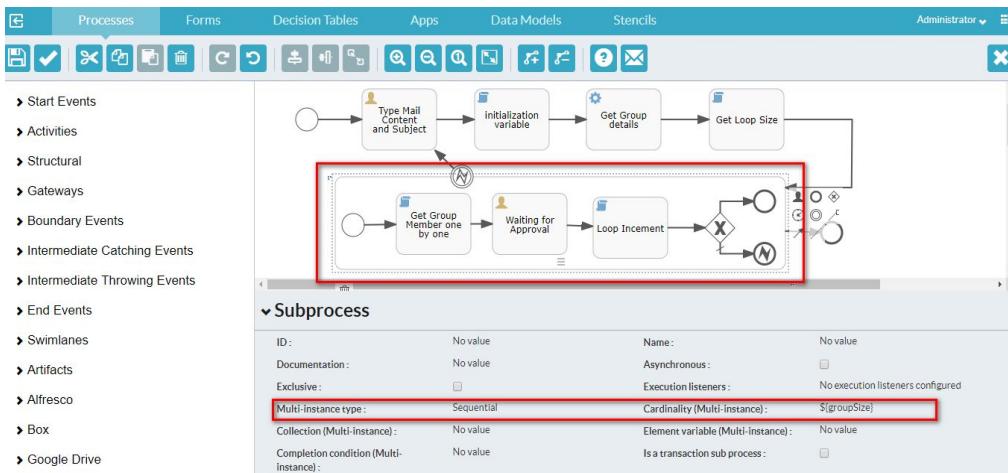
## Get Group details

The screenshot shows the Activiti BPMN editor interface. Similar to the previous section, it includes a sidebar with process-related options and a main process diagram area. In the diagram, a task node labeled "Get Group details" is circled in red. Below the diagram, a detailed view of the "Get Group details - Service task" is shown in a modal window. The "Class" field is highlighted with a red box and contains the value "com.activiti.extension.conf.MyJavaDelegate".

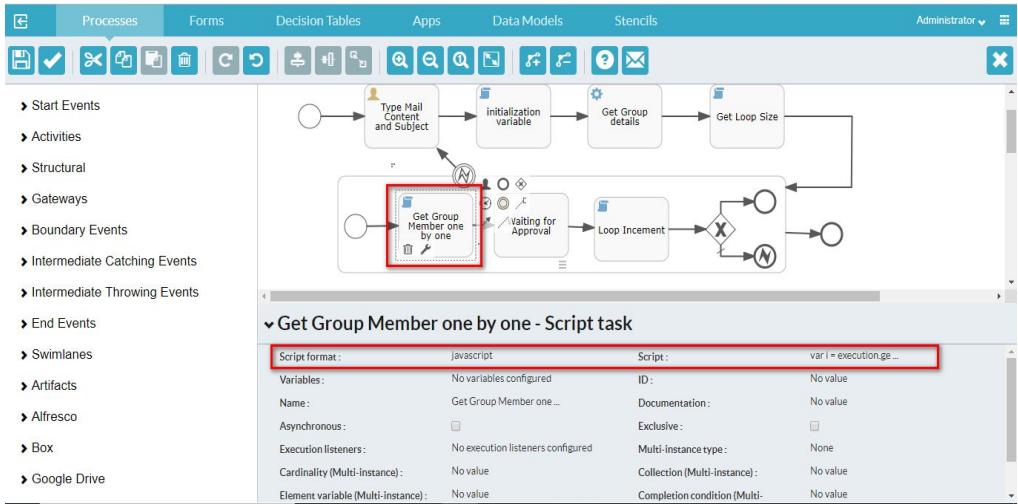
## Get Loop Size



S.No	Property	Value
1.	Script format	javascript
2.	Script	<pre>var data = JSON.parse(execution.getVariable("groupDetails")); ); execution.setVariable("groupSize", data.total);</pre>

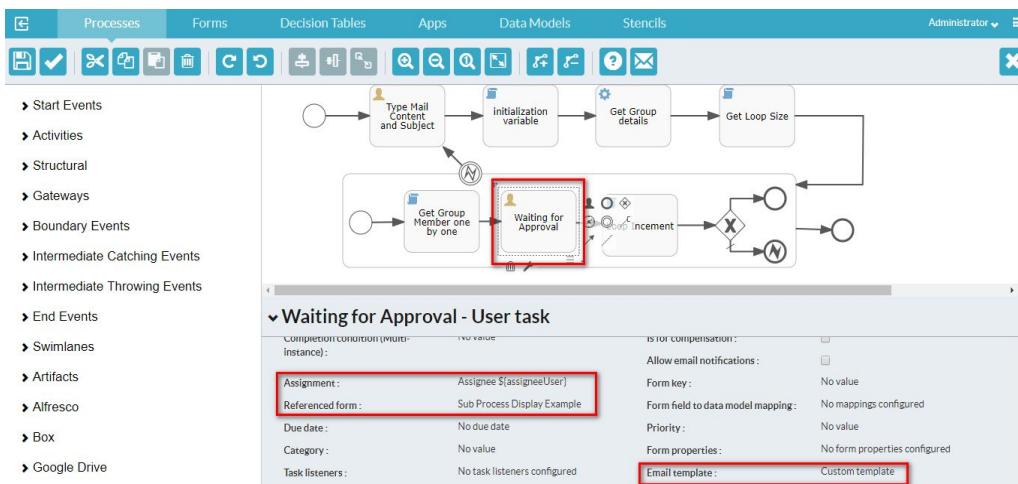


## Get Group Member one by one



S.No	Property	Value
1	Script format	javascript
2	Script	<pre>var i = execution.getVariable("index"); var temp = execution.getVariable("groupDetails"); var groupData = JSON.parse(temp); execution.setVariable("assigneeUser", groupData.data[i].id); execution.setVariable("subject", execution.getVariable("subject")); execution.setVariable("mailbody", execution.getVariable("mailbody"));</pre>

## Waiting for Approval



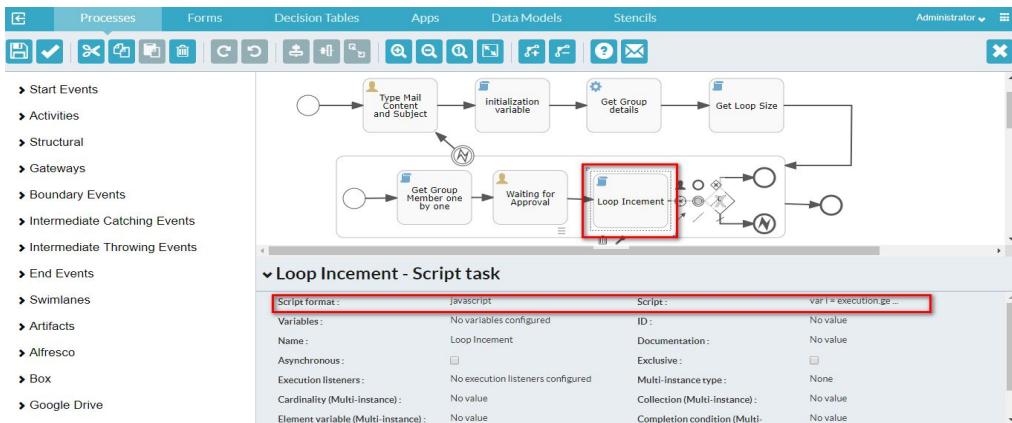
Screenshots showing the 'Forms' tab selected in the top navigation bar. The left sidebar lists various form field types: Text, Multi-line text, Number, Checkbox, Date, Date and time, Dropdown, Typeahead, Amount, Radio buttons, People, Group of people, Dynamic table, Hyperlink, and Header.

The main content area displays a form titled 'Sub Process Display Example' (Version 10). It includes tabs for Design, Tabs, Outcomes, Style, Javascript, Properties, and Variables. The Design tab shows two input fields: 'Subject \${subject}' and 'Body \${mailbody}'.

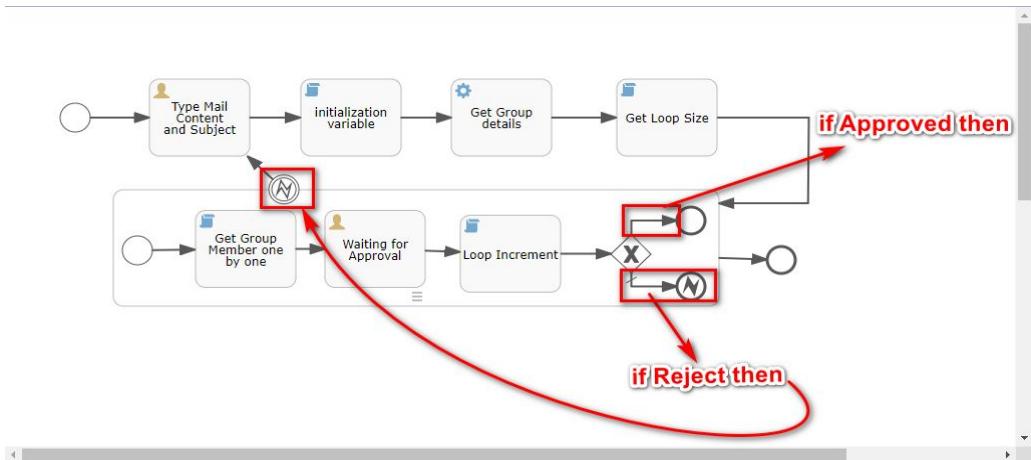
Screenshots showing the 'Forms' tab selected in the top navigation bar. The left sidebar lists various form field types.

The main content area displays a form titled 'Sub Process Display Example' (Version 10). It includes tabs for Design, Tabs, Outcomes, Style, Javascript, Properties, and Variables. The Outcomes tab contains instructions about defining outcomes and a list of possible outcomes: 'Reject' and 'Approved'. The 'Approved' entry is highlighted with a red box.

## Loop Increment

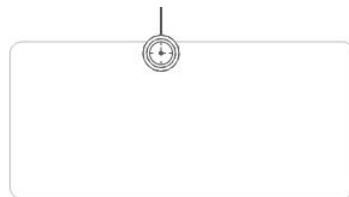


S.No	Property	value
1	Script format	javascript
2	Script	<pre>var i = execution.getVariable("index"); i++; execution.setVariable("index",i);</pre>



## 4.2 Boundary Timer Event

A boundary timer event puts a timer on the activity it is defined on. When the timer fires, the sequence flow going out the boundary event is followed. A boundary timer event is visualized as a circle with a clock icon inside.



### Example

ID :	No value	Name :	No value
Documentation :	No value	Time cycle (e.g. R3/PT10H) :	R3/PT20S
Time date in ISO-8601 :	No value	Time duration (e.g. PT5M) :	PT10S
Cancel activity :	<input type="checkbox"/>		

## 4.3 Boundary Signal Event

A boundary signal event listens to a signal being fired (from within the process instance or system-wide) while the activity upon which the event is defined is active. A boundary signal event is visualized as a circle with a triangle icon inside.



## Example

### Create the process

**Boundary signal event Example**

Process identifier:	BoundarysignaleventE...	Name:	Boundary signal even...
Documentation:	No value	Process author:	No value
Process version string (documentation only):	No value	Target namespace:	http://www.activiti...
Event listeners:	1 event listeners	Execution listeners:	No execution listeners configured
Message definitions:	No message definitions configured	Signal definitions:	1 signal definitions
Metadata:	No properties defined	Variables:	No variables configured

### Change the Signal definitions Property

ID	Name	Scope
test	signaltest	global

ID: test  
Name: signaltest  
Scope: Global

**Change value for "Signal definitions"**

**Cancel** **Save**

### Change Event listeners Property

Change value for "Event listeners"

Event	Implementation
TASK_COMPLETED	<code>\$(TrainingListener)</code>

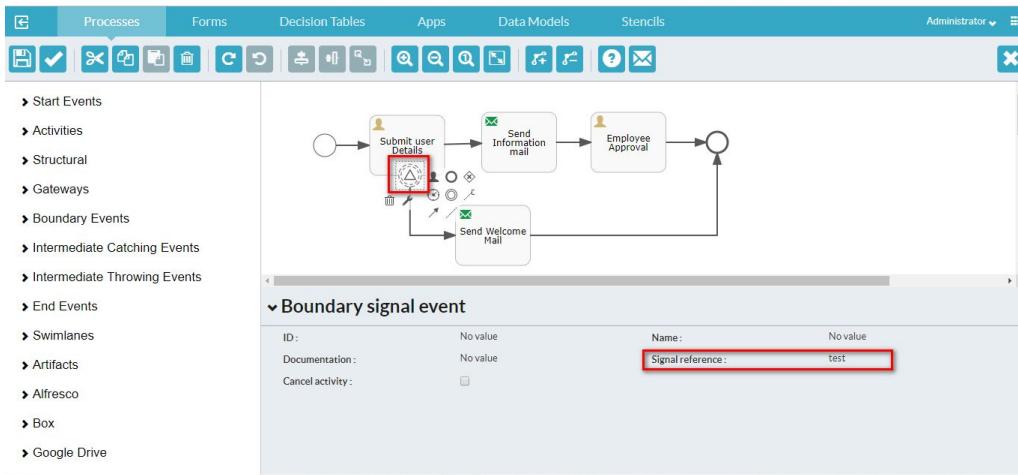
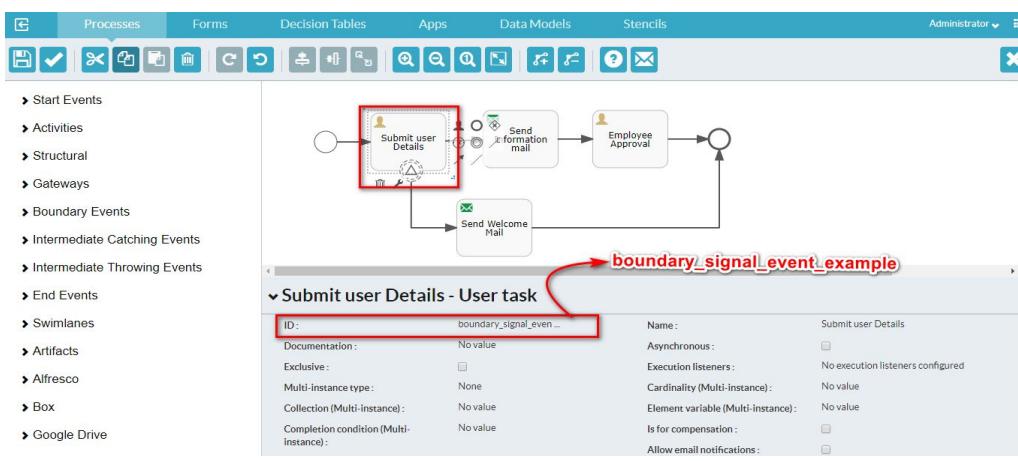
Events

Class

Delegate expression

Entity type

Rethrow event?



## Backend Code

Make the above java code to jar file and put on the folder path  
**C:\..\process-services-1.9.0.3\tomcat\webapps\activiti-app\WEB-INF\lib**

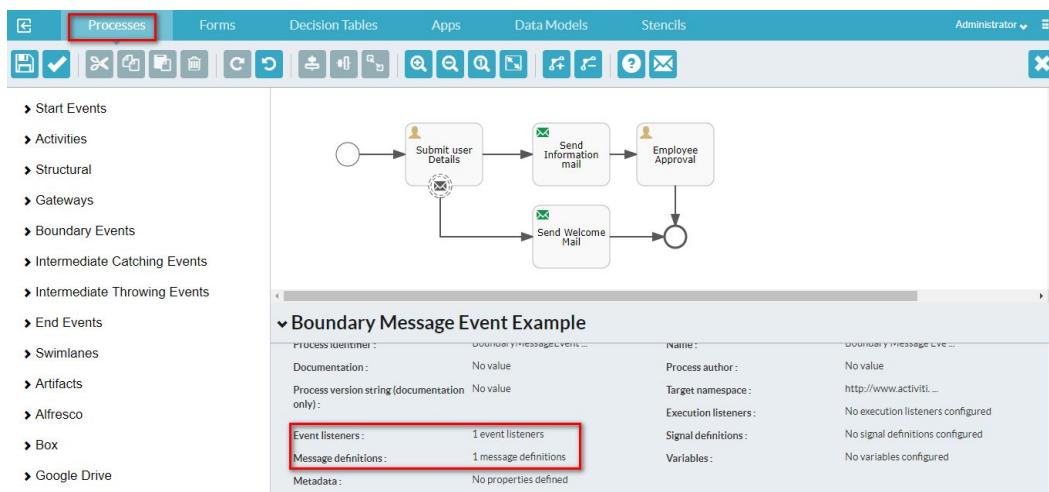
#### 4.4 Boundary Message Event

A boundary message event listens to a message being received while the activity upon which the event is defined is active. A boundary message event is visualized as a circle with an envelope icon inside.



#### Example

##### Create Process



#### Change Event listeners Property

The screenshot shows the 'Change value for "Event listeners"' dialog. It has two tabs: 'Event' and 'Implementation'. The 'Event' tab shows 'TASK\_COMPLETED' selected. The 'Implementation' tab shows the expression `\${TrainingListener}`. Below these tabs are buttons for sorting and adding items. At the bottom, there are sections for 'Events', 'Class', 'Delegate expression', 'Entity type', and checkboxes for 'Rethrow event?' and 'Rethrow event?'. The 'Events' and 'Delegate expression' sections are highlighted with red boxes.

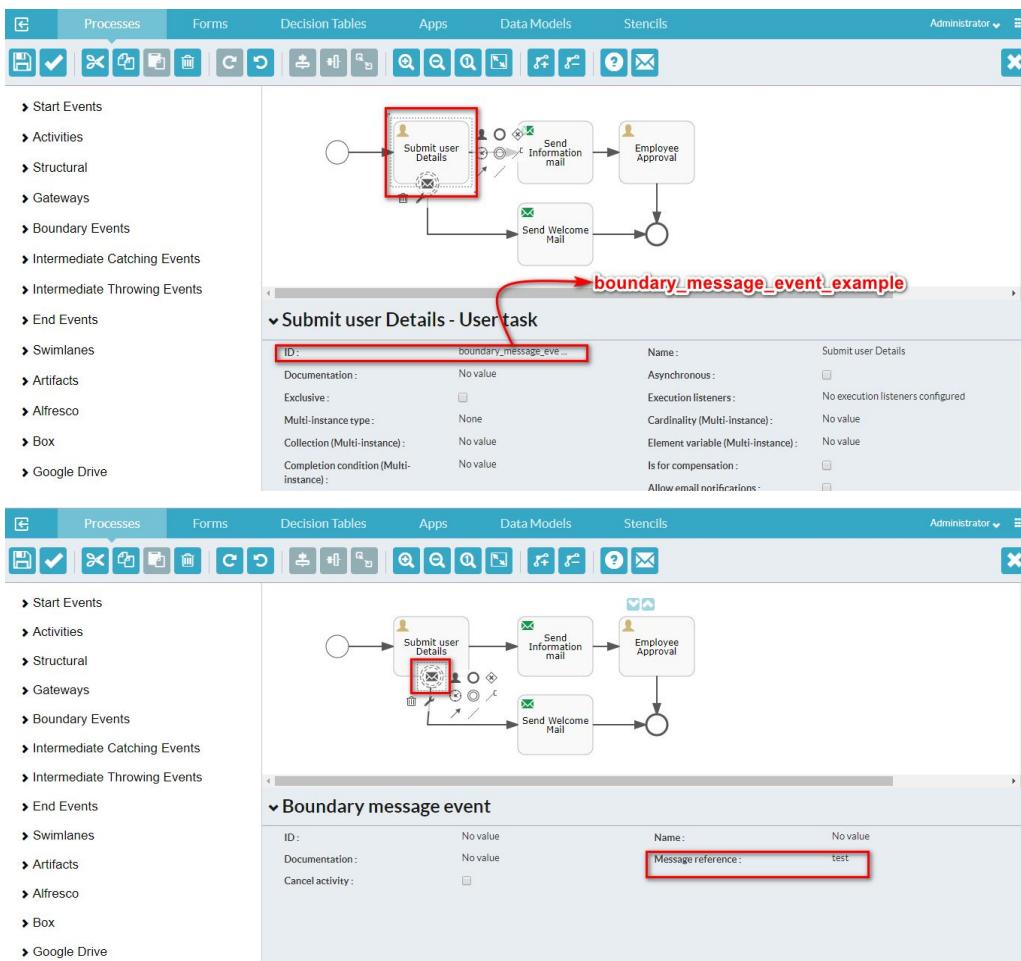
## Change the Message definitions Property

Change value for "Message definitions"

ID	Name
test	messageTrigger

+
-

Save
Cancel



## Backend Code

## 5. Intermediate catching events:

An intermediate catching event is a step in the process where the process needs to wait for a specific trigger. Conceptually, the intermediate catch events are close to the boundary events, with the exception they don't define a scope (the activity) for when the event is active. An intermediate catch event is active as long as the trigger hasn't happened. A boundary event on the other hand can be destroyed if the activity completed. All the supported intermediate catch events are configured similar to their boundary event counterparts.



1            2            3

1. Intermediate Timer catching event
2. Intermediate Signal catching event
3. Intermediate Message catching event

## 6. Intermediate throwing events:

An intermediate throw event is used to explicitly throw an event of a certain type. Currently, two types are supported:

- The **none intermediate throwing event**. No event is thrown. This is mainly used as a marker in the process definition (for example to attach execution listeners that are used to indicate somehow that some state in the process has been reached).
- The **signal intermediate throwing event**. Throws a signal event that will be caught by boundary signal events or intermediate signal catch events listening to that particular signal event.

An intermediate event is displayed as two concentric circles which may contain an icon. If present, the icon shows the type of intermediate event. A throwing none event contains no icon.

## Intermediate signal throwing event & Intermediate signal catching event Example

### Create Process

Screenshot of the Activiti BPMN 2.0 Designer interface showing the creation of a process.

The process diagram illustrates a workflow for book development:

```

graph TD
    Start(( )) --> OR1{ }
    OR1 --> Concept[Develop Book Concept]
    Concept --> ConceptReady1(( ))
    Concept --> WriteText[Write Book Text]
    ConceptReady1 --> OR2{ }
    OR2 --> Cover[Develop Book Cover]
    Cover --> ConceptReady2(( ))
    ConceptReady2 --> OR3{ }
    OR3 --> Ready[Book Ready]
    WriteText --> OR3
  
```

The process includes intermediate signal events:

- Intermediate Signal Events:**
  - Concept Ready:** Triggered by the completion of the "Develop Book Concept" activity.
  - Book Ready:** Triggered by the completion of the "Book Ready" activity.
- Intermediate Throwing Events:**
  - Concept Ready:** Triggered by the completion of the "Develop Book Concept" activity.
  - Book Ready:** Triggered by the completion of the "Book Ready" activity.

**Process Details:**

Process identifier:	Intermediate_Signal_...	Name:	Intermediate_Signal_...
Documentation:	No value	Process author:	No value
Process version string (documentation only):	No value	Target namespace:	http://www.activiti...
Event listeners:	No event listeners configured	Execution listeners:	No execution listeners configured
Message definitions:	No message definitions configured	Variables:	No variables configured
Metadata:	No properties defined		

**Signal Definitions:**

Change value for "Signal definitions"

ID: book\_concept\_signal  
Name: bookconcept  
Scope: processinstance

+ -

**Save**

### Intermediate signal throwing event

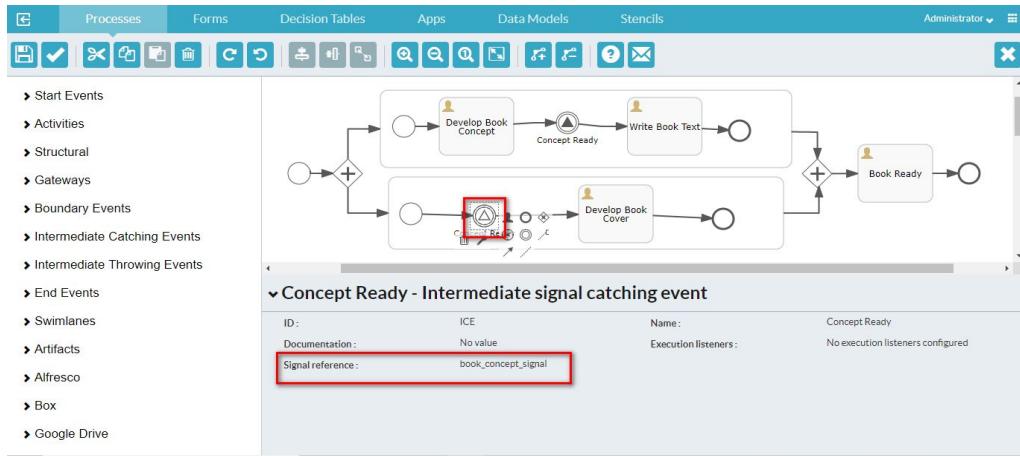
Screenshot of the Activiti BPMN 2.0 Designer interface showing the configuration of an intermediate signal throwing event.

The process diagram shows the "Concept Ready" intermediate signal throwing event highlighted with a red box.

**Intermediate Signal Event Configuration:**

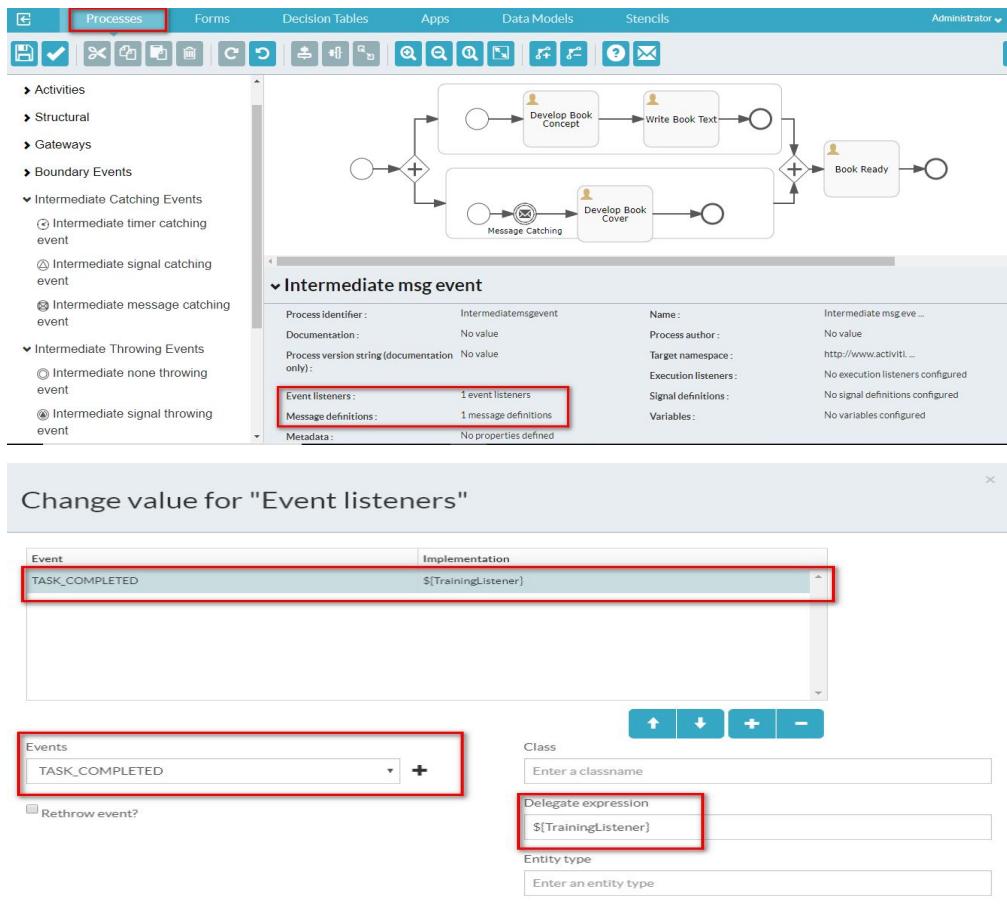
ID:	ITE	Name:	Concept Ready
Documentation:	No value	Execution listeners:	No execution listeners configured
Signal reference:	book_concept_signal		

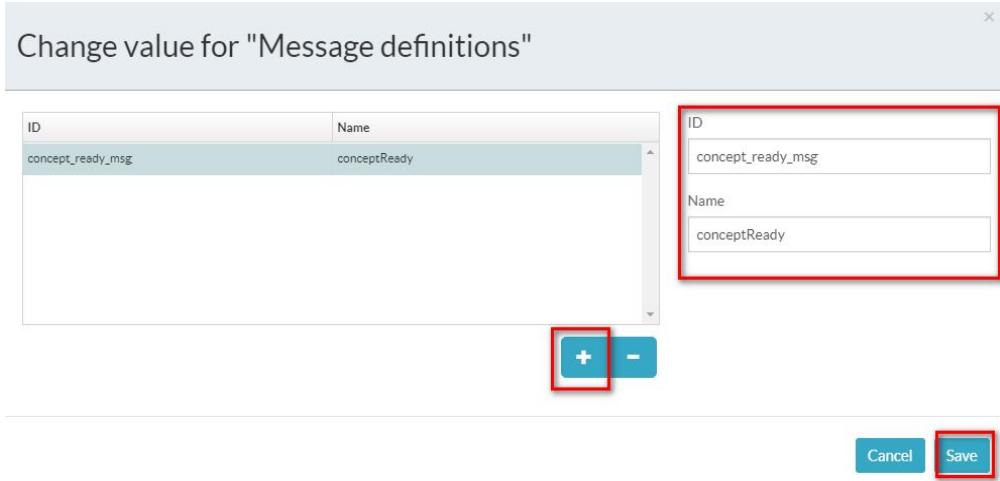
## Intermediate signal catching event



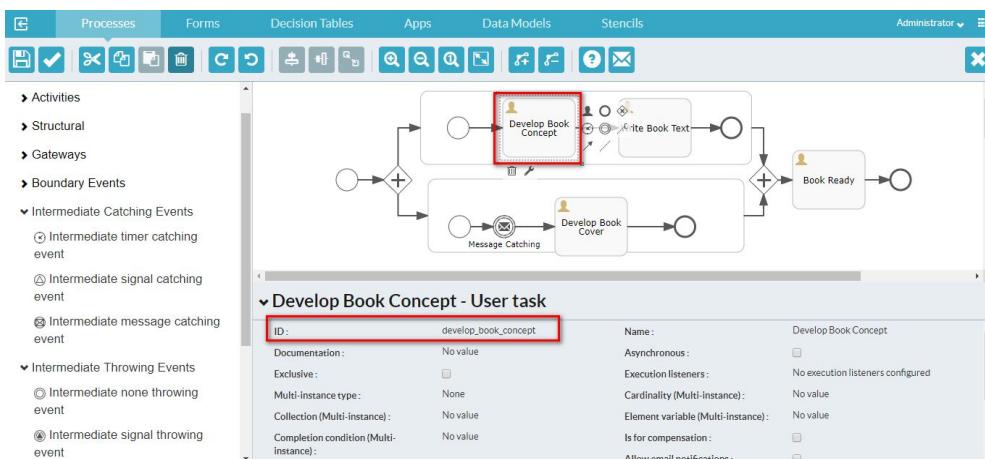
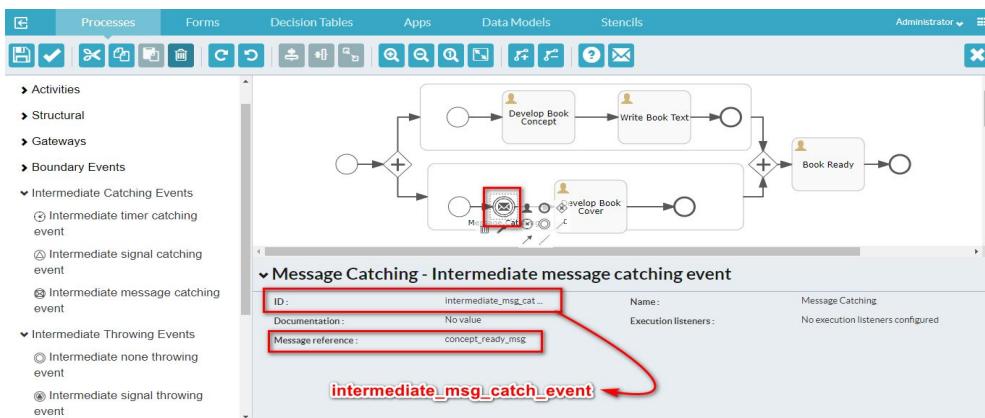
## Intermediate Message catching event Example

### Create the process





## Intermediate messages catching event



## Backend Code

## **7. End Events:**

You use an end event to signify the end of a process or sub-process, or the end of a path in a process or sub-process. In a subprocess or process instance, only when all executions have reached an end event will the subprocess be continued or the whole process instance ended. An end event is displayed as thick black circle which may contain an icon. If present, the icon shows the type of end event. A none end event has no icon.

### **7.1 None End Event**

A none end event ends the current path of execution.



### **7.2 Error End Event**

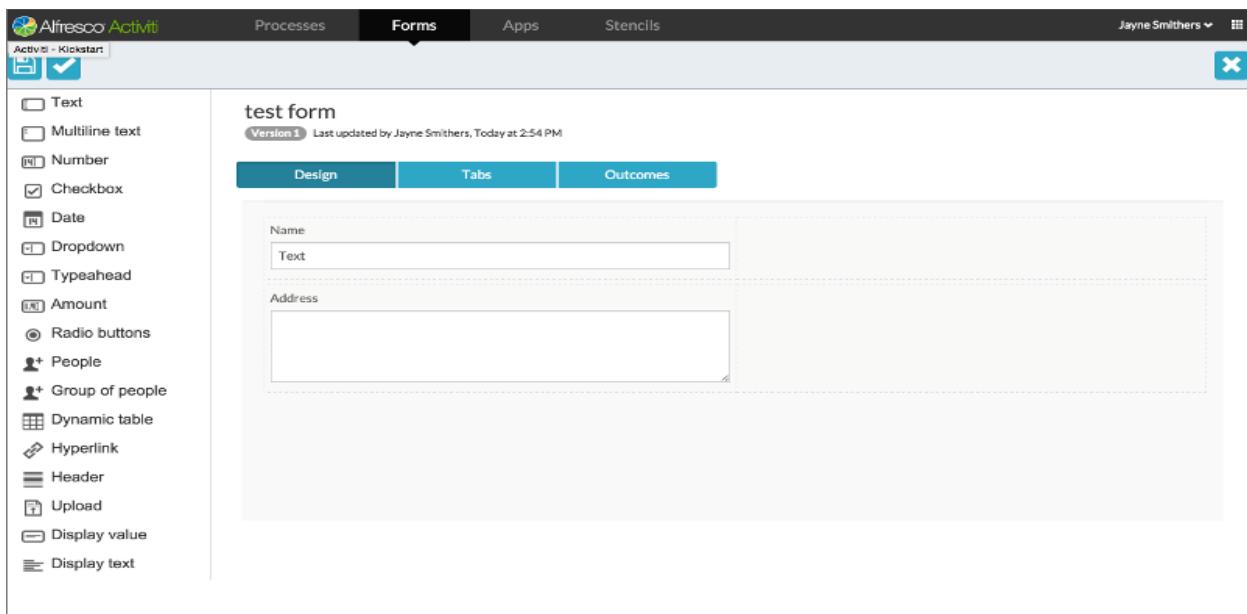
You use the end error event to throw an error and end the current path of execution. The error can be caught by an intermediate boundary error event that matches the error. If no matching boundary error event is found, an exception will be thrown.



## Form Editor

The form editor provides a powerful drag and drop interface to let you design forms from a rich set of controls. You can define form outcomes and create forms with multiple tabs. Individual controls and whole tabs can be made visible depending on the value of other form fields and process variables. You can design your form with groups of controls in varying numbers of columns.

### Example



### Form Controls

The form controls for each field determine how the field is displayed and handled. For more information about form control, refer this [link](#).

## Step Editor

The Step Editor guides you through creating a business process through a sequence of simple steps. The processes you create using the step editor do not exploit the full power of BPMN 2.0 like those created by the BPMN editor, but you can use it to design both simple and quite complex process models, without knowledge of BPMN 2.0.

For more information about form control, refer this [link](#).

