

PUBLIC

# **Build a Workflow from Scratch with SAP Cloud Platform Workflow Management**

## **DEV163**

Exercise 1 | Workflow  
Archana Shukla / SAP

## TABLE OF CONTENTS

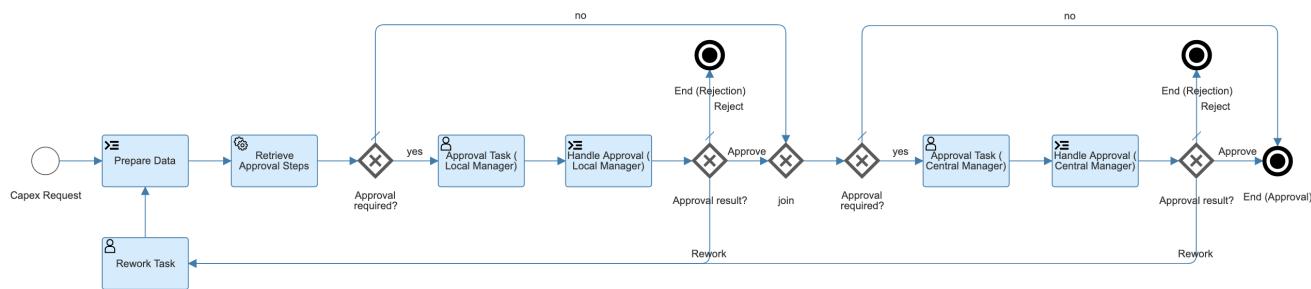
OVERVIEW .....	3
MODEL WORKFLOW USING BUSINESS APPLICATION STUDIO.....	4
BUILD AND DEPLOY WORKFLOW .....	45
EXECUTE AND MONITOR WORKFLOW.....	47
SUMMARY .....	53

## OVERVIEW

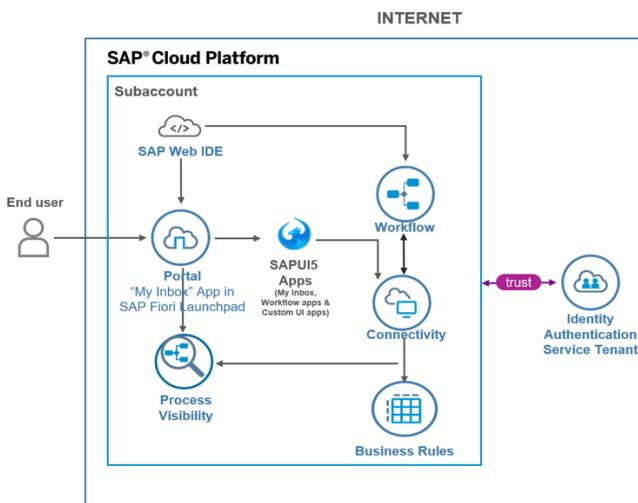
Estimated time: 90 minutes

Capital Expenditure process starts when any company receives a funding request for capital expenditures. These requests are then thoroughly reviewed, evaluated and finally approved or rejected based on the available budgets for the current period. Approval is not straight forward. There are many rules that are involved to determine how many levels of approvals will be there or who will approve it based on the investment details.

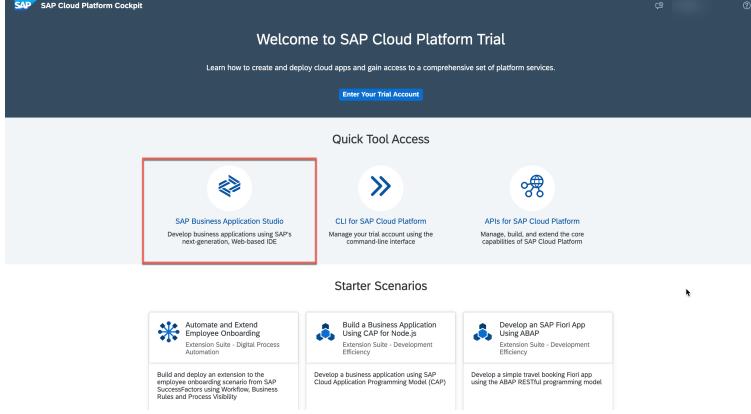
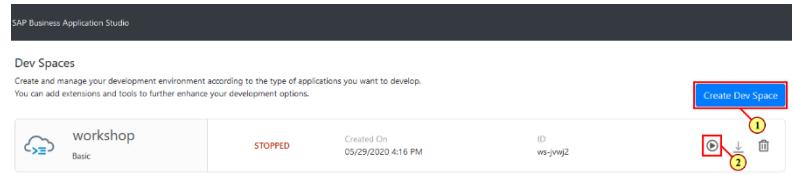
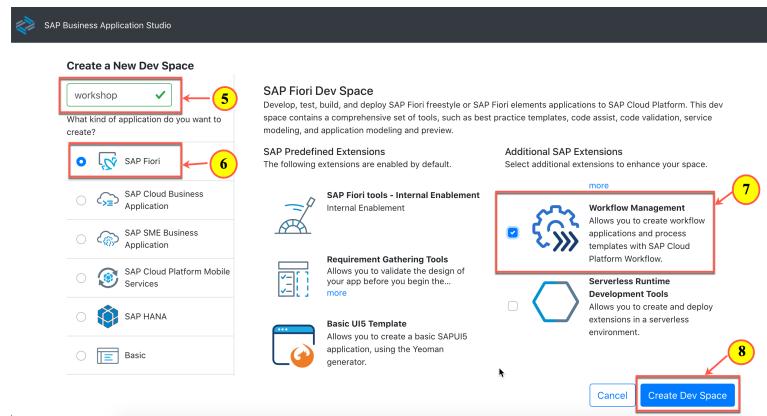
There are many challenges with capital expenditure process and companies are always looking for managing investment decisions in a transparent way. Automating capital expenditure approval process enables customer to accelerate investment decisions and to bring the agility into every step and decision such that the corporate policies are followed while taking the investment decisions.

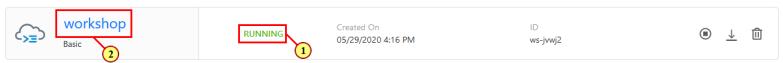
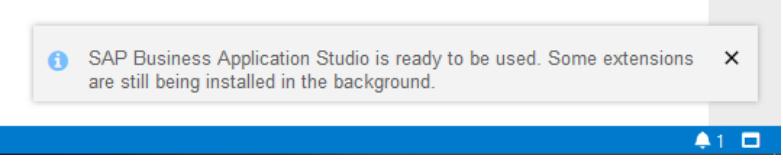
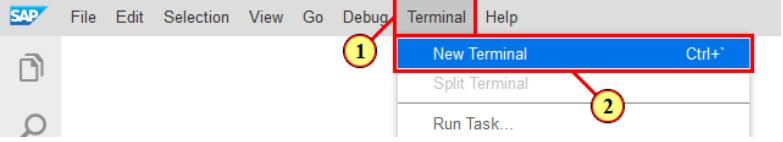
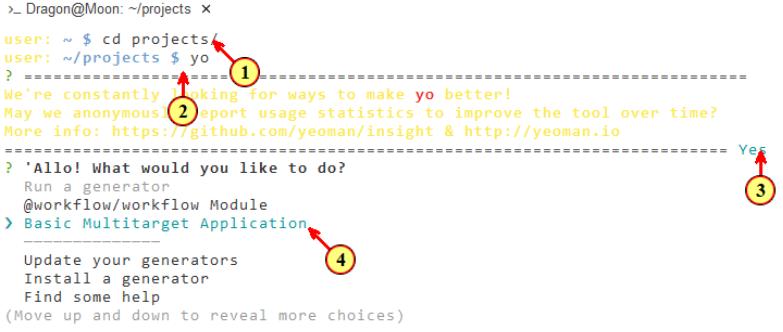
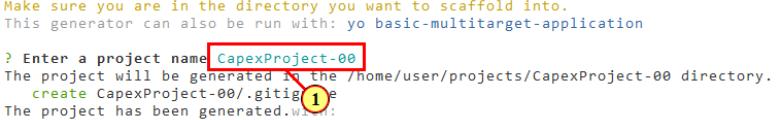


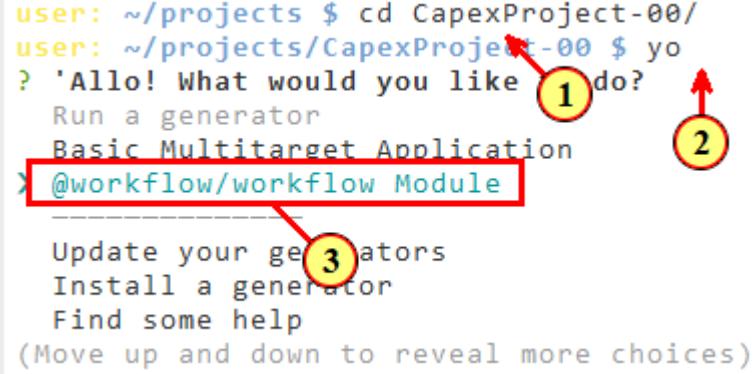
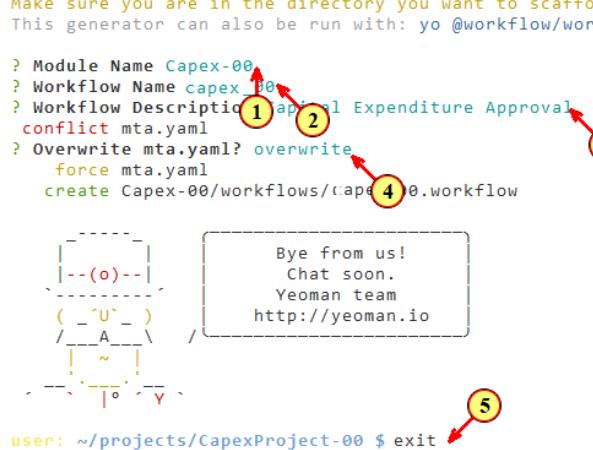
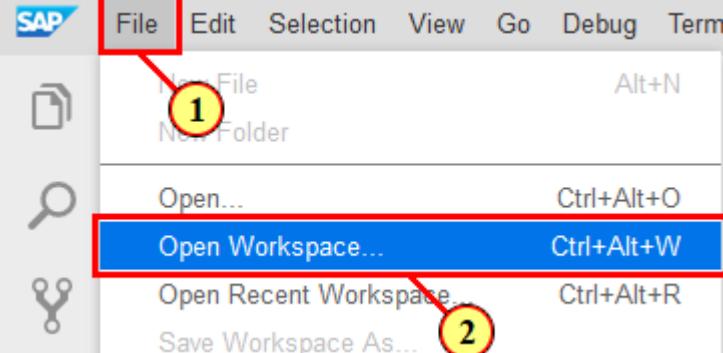
The below solution diagram explains how customer can automate CAPEX scenario using SAP Cloud Platform.

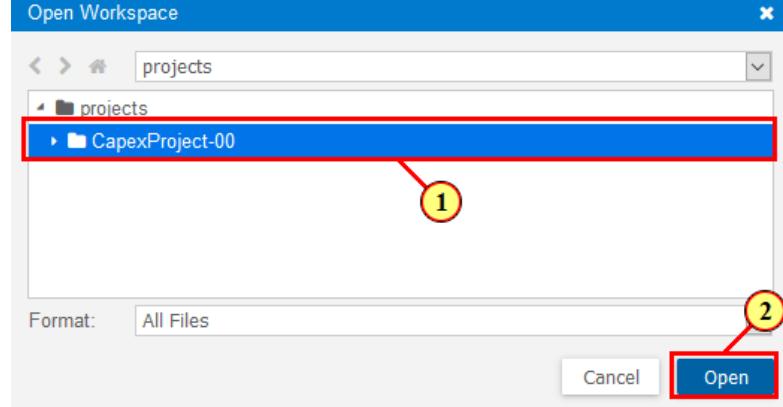
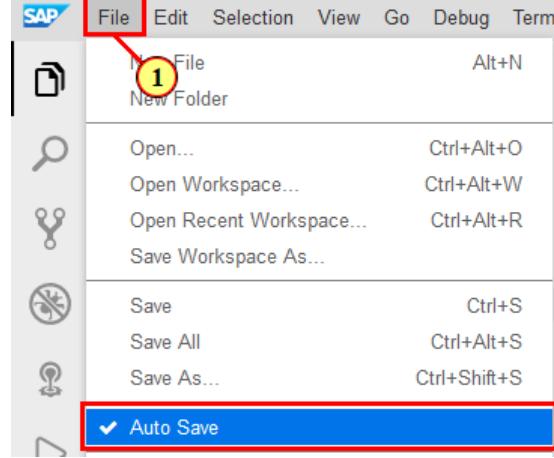
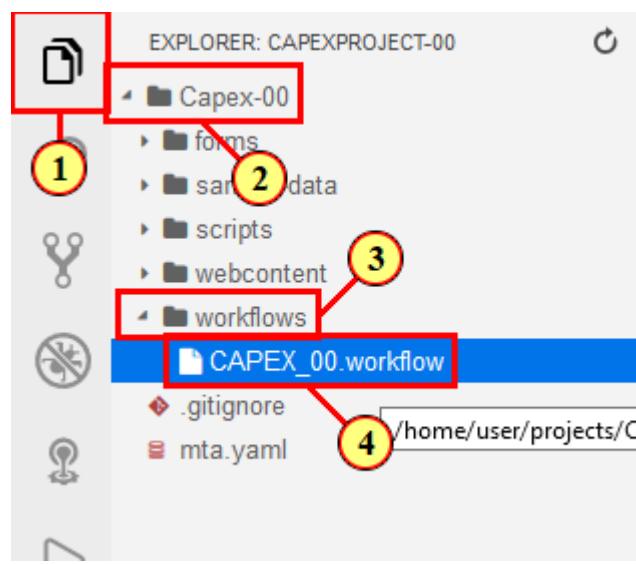


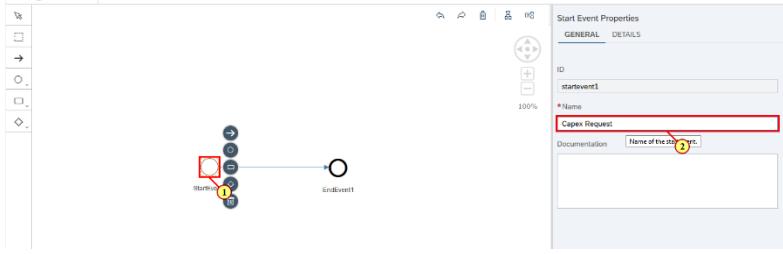
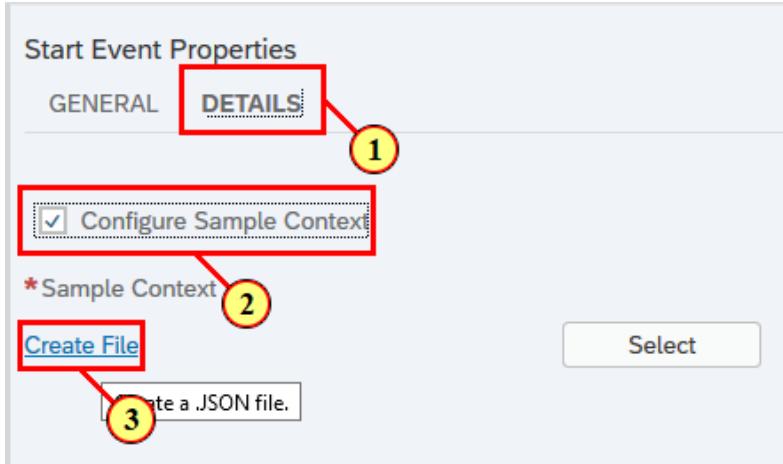
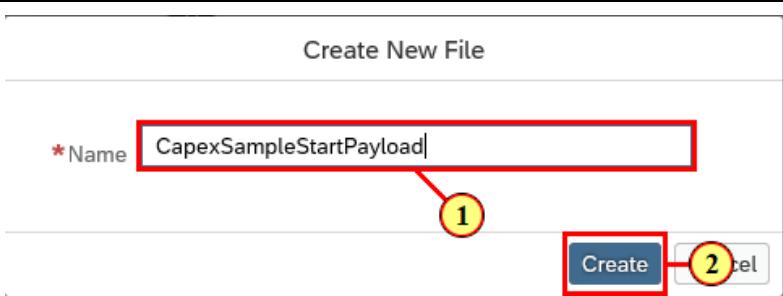
## MODEL WORKFLOW USING BUSINESS APPLICATION STUDIO

Explanation	Screenshot
<ol style="list-style-type: none"> <li>Logon to <b>SAP Cloud Platform Trial Home Page</b> with your trial username and password.  <a href="https://cockpit.hanatrial.ondemand.com/cockpit/#/home/trial">https://cockpit.hanatrial.ondemand.com/cockpit/#/home/trial</a></li> <li>Choose <b>SAP Business Application Studio</b>.</li> </ol>	
<p>Each SAP Business Application Studio Development Space is a virtual machine hosting your project files, securely separated from other spaces.</p> <ol style="list-style-type: none"> <li>If you have no dev space, click the <b>Create Dev Space</b> and follow instructions in step 5.</li> <li>Else select <b>PLAY</b> to start the space if you see the status as <b>STOPPED</b>.</li> </ol> <p>Note: Idle dev space will be shut down after some time. This is indicated by a yellow bar in the bottom of the screen with the text "Offline". In this case, return to this Dev Space selection screen from your bookmark and restart it.</p>	
<ol style="list-style-type: none"> <li>Enter <b>workshop</b> as the new <b>Dev Space name</b>.</li> <li>Ensure <b>Application Type</b> is selected as <b>SAP Fiori</b>.</li> <li>Select the extension <b>Workflow Management</b>. Note: This extension will enable the workflow modelling features.</li> <li>Select <b>Create Dev Space</b>.</li> </ol>	

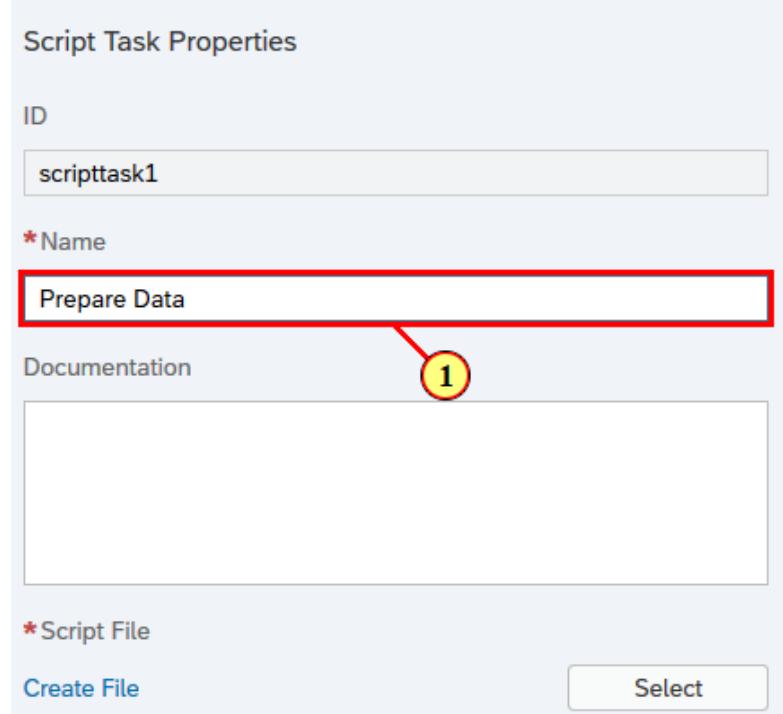
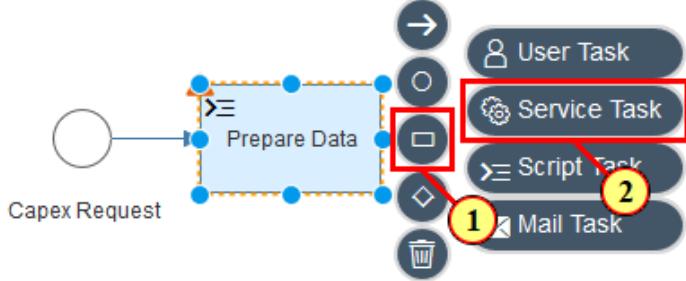
Explanation	Screenshot
<p>9. Wait until the virtual machine providing this dev space is in status <b>RUNNING</b>.</p> <p>10. Open the <b>SAP Business Application Studio</b> by clicking on the dev space name.</p>	
<p>Note: SAP Business Application Studio is a web application that loads the features of the extensions. There might be a short delay for all features to be operational when first accessed.</p>	
<p>11. From the menu, select <b>Terminal</b>.</p> <p>12. Select <b>New Terminal</b> to open a Linux Terminal window.</p>	
<p>13. Enter <b>cd projects/</b> in the <b>Terminal</b> window to go into the folder where you are going to create the workflow module. Note: This project folder is already created.</p> <p>14. Enter <b>yo</b> in the terminal window to start Yeoman.</p> <p>Note: Yeoman is a tool for quickly setting up the basic structure of projects.</p> <p>15. If asked, choose whether you want to enable telemetric for Yeoman.</p> <p>16. Select <b>Basic Multitarget Application</b> by using the up/down arrow keys and press <b>Enter</b>.</p>	
<p>17. Enter <b>CapexProject-00</b> or any name of your choice as <b>Project name</b>.</p>	

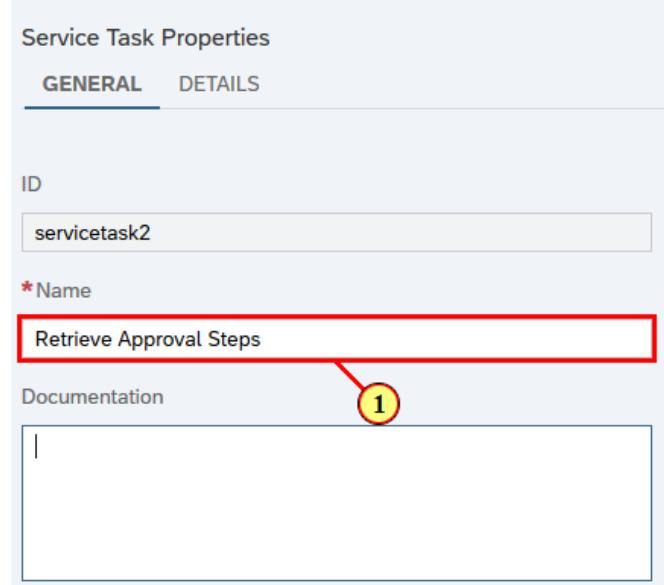
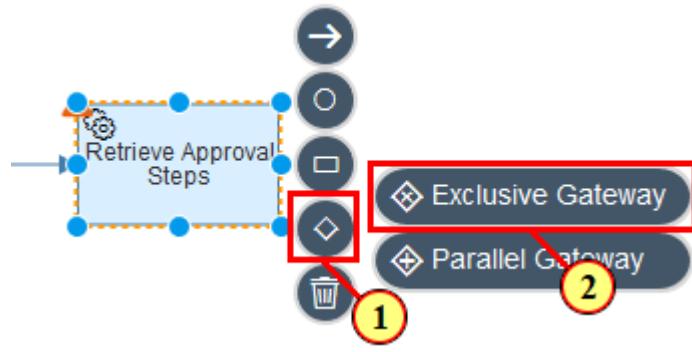
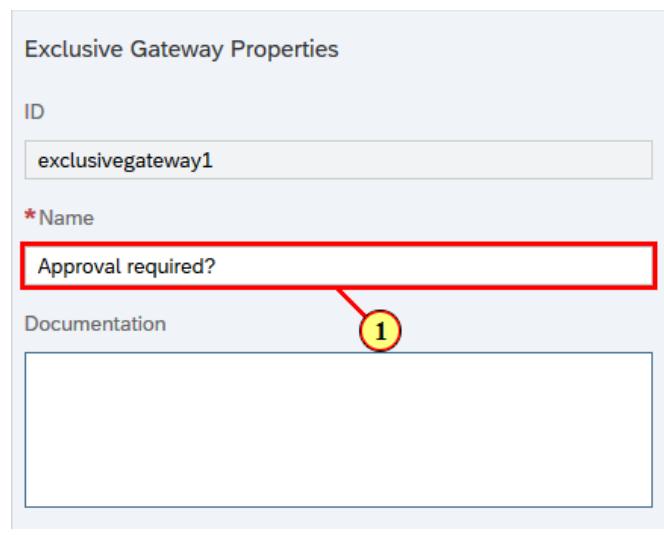
Explanation	Screenshot
<p>18. Change into the newly created directory by entering <b>cd CapexProject-00</b>. You can also use <b>TAB</b> for auto-completion, i.e. entering <b>cd Cap&lt;TAB&gt;</b> will be sufficient.</p> <p>19. Enter <b>yo</b> in the window to start Yeoman again.</p> <p>20. Select <b>Workflow Module</b> by using the up/down arrow keys and press <b>Enter</b>.</p>	 <pre>user: ~/projects \$ cd CapexProject-00/ user: ~/projects/CapexProject-00 \$ yo ? 'Allo! What would you like to do?   Run a generator   Basic Multitarget Application ❯ @workflow/workflow Module Update your generators Install a generator Find some help (Move up and down to reveal more choices)</pre>
<p>21. Enter <b>Capex-00</b> as <b>Module Name</b>.</p> <p>22. Enter <b>Capex_00</b> as <b>Workflow Name</b>.</p> <p>23. Enter <b>Capital Expenditure Approval</b> as <b>Workflow Description</b>.</p> <p>24. Yeoman will warn you that there already is an <b>mta.yaml</b> file. Type <b>Y</b> to overwrite it.</p> <p>25. You can close the <b>Terminal</b> by typing <b>exit</b> or clicking on the X in the window header.</p>	 <pre>Make sure you are in the directory you want to scaffold into. This generator can also be run with: yo @workflow/workflow-module  ? Module Name Capex-00 ? Workflow Name capex_00 ? Workflow Description Capital Expenditure Approval conflict mta.yaml ? Overwrite mta.yaml? overwrite   force mta.yaml create Capex-00/workflows/capex_00.workflow  [...] Bye from us!   Chat soon.   Yeoman team   http://yeoman.io  user: ~/projects/CapexProject-00 \$ exit</pre>
<p>26. From the menu, select <b>File</b>.</p> <p>27. Select <b>Open Workspace....</b></p>	 <p>SAP</p> <p>File Edit Selection View Go Debug Term</p> <p>New File New Folder Alt+N</p> <p>Open... Ctrl+Alt+O</p> <p>Open Workspace... Ctrl+Alt+W</p> <p>Open Recent Workspace Ctrl+Alt+R</p> <p>Save Workspace As... 2</p>

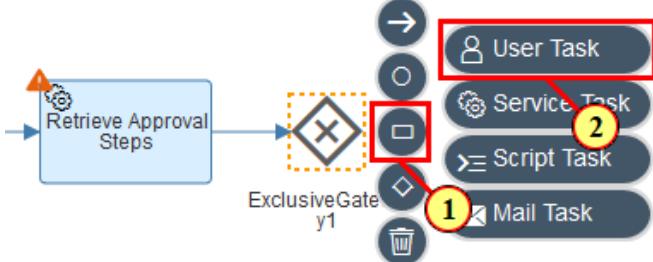
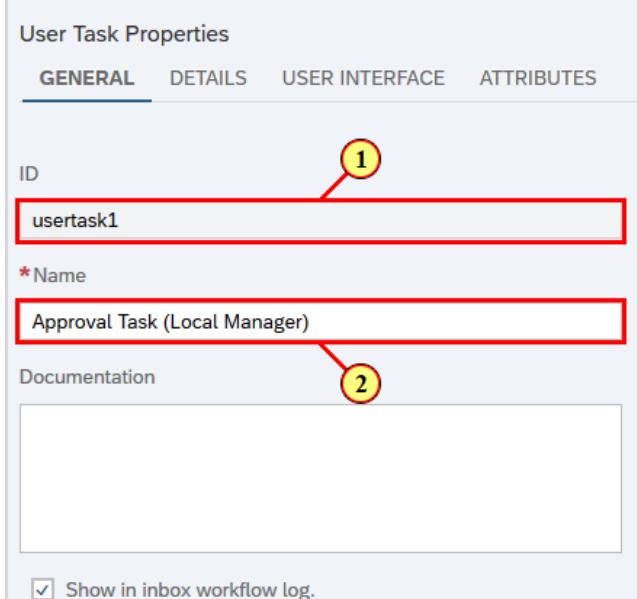
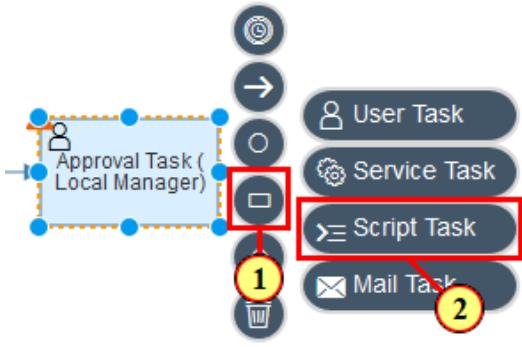
Explanation	Screenshot
<p>28. Select the directory containing your module <b>CapexProject-00</b>.</p> <p>29. Select <b>Open</b>.</p>	
<p>30. From the menu, select <b>File</b>.           Note that there is an <b>Auto Save</b> functionality (enabled by default). If you choose to disable it, remember to manually save your files.</p>	
<p>31. If it is not already open, select the <b>Explorer</b>  from the sidebar.</p> <p>32. Open the <b>Capex-00</b> module by selecting it.</p> <p>33. Open the <b>workflows</b> directory by selecting it.</p> <p>34. Open the <b>CAPEX_00.workflow</b> by double-clicking on it.</p> <p>Note: If the graphical workflow editor is not opened, or a textual (code) editor is opened instead, the workflow plugin did not load correctly. Refresh your browser window and try again after SAP Business Application Studio has fully loaded.</p>	

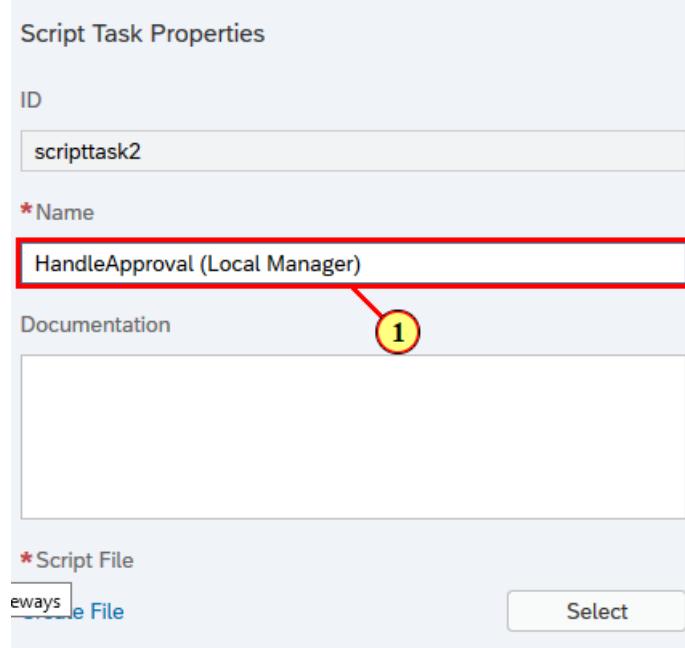
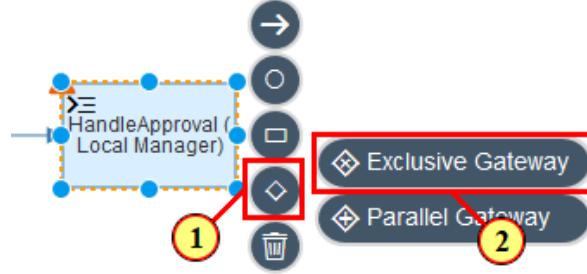
Explanation	Screenshot
<p>A new workflow with a default Start and End event is created.</p> <ul style="list-style-type: none"> <li>○ On the left side there is a toolbar for adding new flow elements.</li> <li>○ On the right side is the properties pane for the currently selected element.</li> </ul> <p>35. Click on the <b>Start Event</b>  to select it. This will show the quick action icons and the property pane will show the details of this element.</p> <p>36. Enter <b>Capex Request</b> as the Name of the start event.</p> <p>This will later be shown in the workflow <i>Execution Log</i> as well as in Process Visibility.</p>	
<p>37. Click on the tab <b>Details</b> of the start event.</p> <p>38. Select <b>Configure Sample Context</b>. When starting a new workflow, an initial context (JSON data) can be given. With this option, an example context can be configured.</p> <p>39. Click <b>Create File</b>.</p>	
<p>1. In the popup, enter <b>CapexSampleStartPayload</b> as File Name.</p> <p>2. Click <b>Create</b>.</p>	

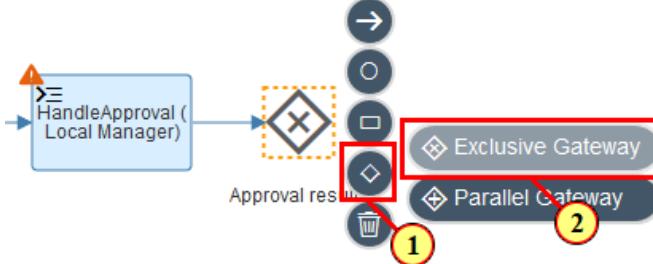
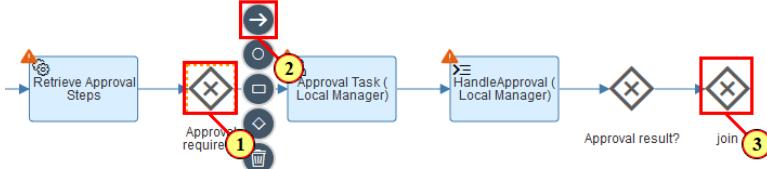
Explanation	Screenshot
<p>A new file with the name <b>CapexSampleStartPayload.json</b> is created in the project folder below <b>sample-data</b>. It is created with some default data as example.</p>	 <pre>* CAPEX_00.workflow x CapexSampleStartPayload.json x 1 {  2   "product": "Hamlet (Paperback)", 3   "inStock": true, 4   "inventory": 20000, 5   "price": 7.49, 6   "publishingDate": "1600-04-23T18:25:43.511Z", 7   "author": { "name": "William Shakespeare" }, 8   "publishers": [ "Simon &amp; Brown", "SparkNotes", "Dover Publications" ] 9 }</pre>
<ol style="list-style-type: none"> <li>3. Remove the example content and Copy-Paste the example context given in the file <b>CapexSampleStartPayload.json</b> into the editor.</li> <li>4. Set the RequestId to CAPEX_REQ_XX_001. Replace XX with the number assigned to you by the trainer.</li> <li>5. Insert your name, email address and user id into the respective fields. Your <b>UserId</b> is <b>your trial user email ID</b>.</li> <li>6. Close the file by clicking on  in the header bar. If shows a round circle instead this indicates that there are changes, which will be saved by closing.</li> </ol>	 <pre>* CAPEX_00.workflow x CapexSampleStartPayload.json x 1 {  2   "RequestId": "CAPEX_REQ_00_001", <span style="border: 1px solid red;">1</span> 3   "title": "Build mobile apps", 4   "Requester": { 5     "Name": "&lt;your name&gt;", 6     "Email": "&lt;your email address&gt;", 7     "UserId": "&lt;your userId&gt;", 8     "Comment": "Please Approve" <span style="border: 1px solid red;">2</span> 9   }, 10  "Investment": { 11    "TotalCost": 10000, 12    "Type": "Software", 13    "CAPEX": 10000, 14    "ROI": 5, 15    "Country": "Germany", 16    "Currency": "EUR", 17    "Description": "Provide a fresh experience for our customers by providing new apps for our services" 18  } 19 }</pre>
<ol style="list-style-type: none"> <li>7. Click the <b>Start event</b>  to select it.</li> <li>8. Click on <b>Add Task</b> .</li> <li>9. Select the <b>Script Task</b> .</li> </ol>	

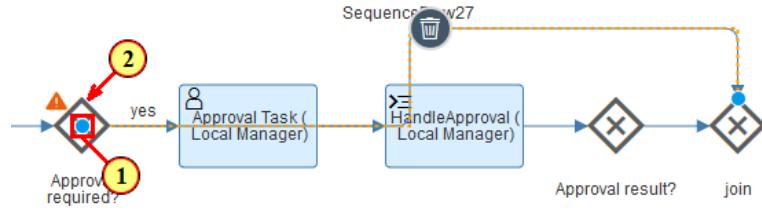
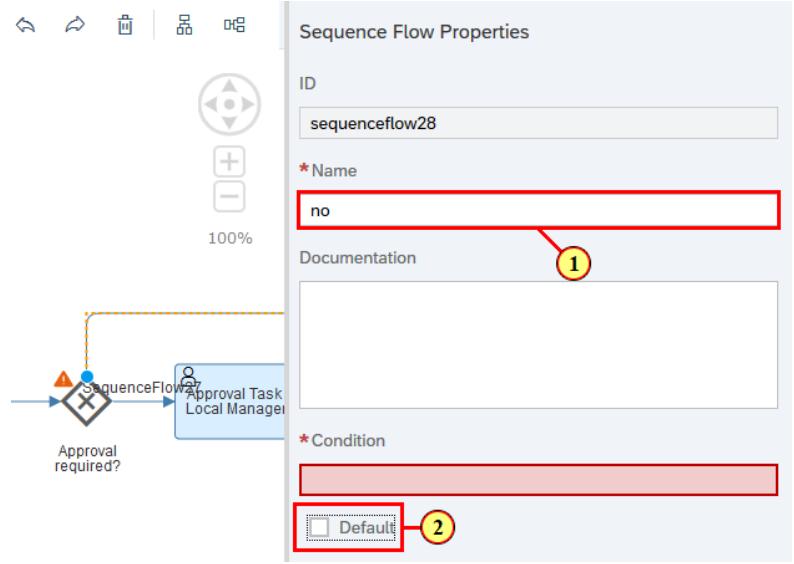
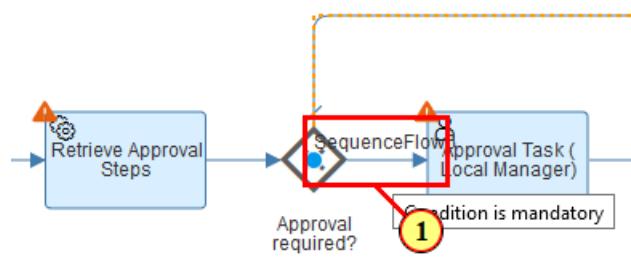
Explanation	Screenshot
<p>10. In the Script Task Properties, enter <b>Prepare Data</b> as <b>Task name</b>.</p> <p>Note: We will configure the task later.</p>	 <p>The screenshot shows the 'Script Task Properties' dialog box. The 'Name' field contains 'scripttask1'. The 'Task name' field is highlighted with a red border and contains 'Prepare Data'. A yellow circle with the number '1' is positioned next to the 'Documentation' input field.</p>
<p>11. On the selected Script Task 'Prepare Data', click <b>Add Task</b>  in the speed buttons.</p> <p>12. Select <b>Service Task</b> .</p>	 <p>The screenshot shows a workflow diagram with a start event labeled 'Capex Request' leading to a script task named 'Prepare Data'. A context menu is open over the 'Prepare Data' task, listing options: 'User Task', 'Service Task', 'Script Task', and 'Mail Task'. The 'Service Task' option is highlighted with a red box and a yellow circle with the number '1'. Other options are shown with yellow circles and numbers '2' and '3' respectively.</p>

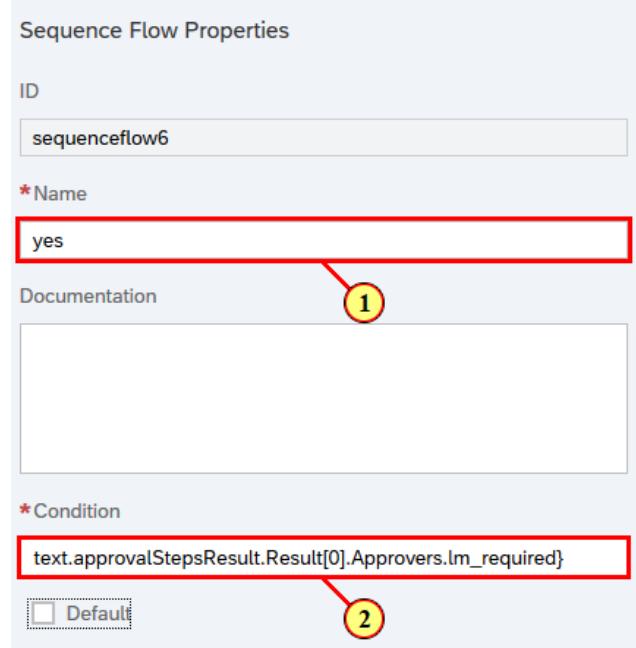
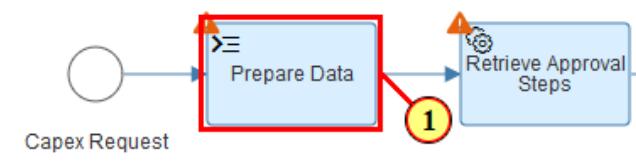
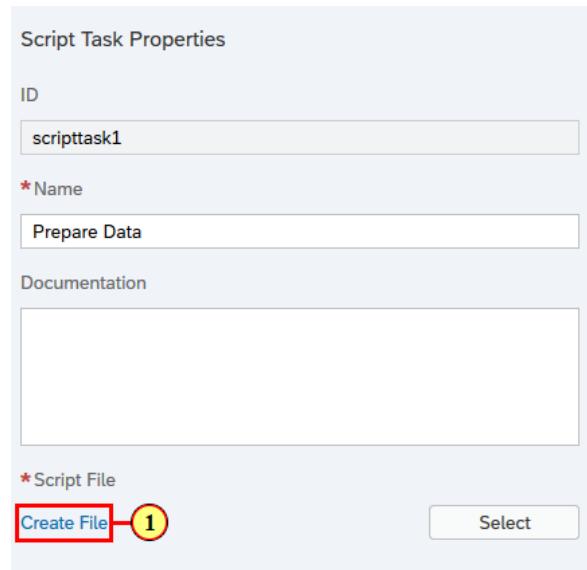
Explanation	Screenshot
<p>13. Enter <b>Retrieve Approval Steps</b> as <b>Name</b>.</p> <p>Note: We will configure the task later.</p>	 <p>The screenshot shows the 'Service Task Properties' dialog with the 'GENERAL' tab selected. The 'Name' field contains 'servicetask2' and is highlighted with a red rectangle. A yellow circle with the number '1' points to the 'Documentation' text area below it.</p>
<p>14. On the selected 'Retrieve Approval Steps' service task, click <b>Add Gateway</b> .</p> <p>15. Select <b>Exclusive Gateway</b> .</p>	 <p>The screenshot shows a workflow diagram. A blue rounded rectangle represents a service task named 'Retrieve Approval Steps'. To its right are several gateway icons: a sequence connector (arrow), an inclusive gateway (circle with 'O'), an exclusive gateway (diamond with a diagonal line), a parallel gateway (diamond with a plus sign), and a delete connector (trash can). The 'Exclusive Gateway' icon is highlighted with a red rectangle and a yellow circle with the number '1' points to it. A yellow circle with the number '2' points to the 'Parallel Gateway' icon.</p>
<p>16. Enter <b>Approval required?</b> as gateway <b>Name</b>.</p> <p>Note: We will configure the gateway later, once we added more outgoing branches.</p>	 <p>The screenshot shows the 'Exclusive Gateway Properties' dialog. The 'Name' field contains 'exclusivegateway1' and is highlighted with a red rectangle. A yellow circle with the number '1' points to the 'Documentation' text area below it.</p>

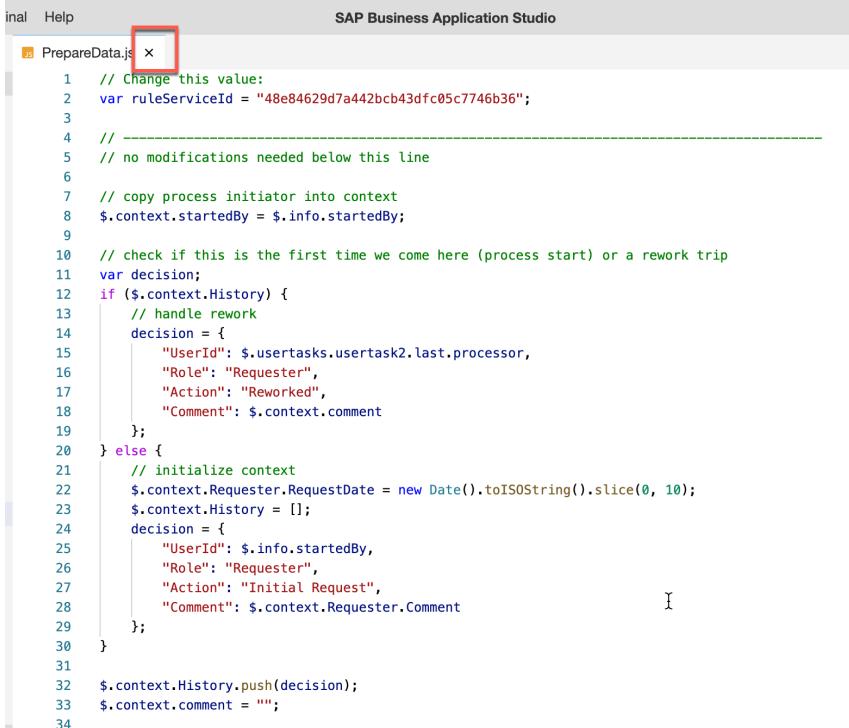
Explanation	Screenshot												
<p>17. Select <b>Add Task</b> .</p> <p>18. Select <b>User Task</b> .</p>	 <pre> graph LR     A[Retrieve Approval Steps] --&gt; B{ExclusiveGate y1}     B --&gt; C((User Task))     B --&gt; D((Service Task))     B --&gt; E((Script Task))     B --&gt; F((Mail Task))     </pre>												
<p>Note that the <b>Task ID</b> is <b>usertask1</b>. This will be needed later.</p> <p>19. Enter <b>Approval Task (Local Manager)</b> as <b>Task name</b>.</p> <p>Note: IDs are automatically given to objects. If an object is deleted and re-created, it will get a new ID.</p>	 <table border="1"> <thead> <tr> <th colspan="2">User Task Properties</th> </tr> <tr> <th colspan="2">GENERAL</th> </tr> </thead> <tbody> <tr> <td>ID</td> <td>1 usertask1</td> </tr> <tr> <td>*Name</td> <td>2 Approval Task (Local Manager)</td> </tr> <tr> <td>Documentation</td> <td>(empty box)</td> </tr> <tr> <td colspan="2"><input checked="" type="checkbox"/> Show in inbox workflow log.</td> </tr> </tbody> </table>	User Task Properties		GENERAL		ID	1 usertask1	*Name	2 Approval Task (Local Manager)	Documentation	(empty box)	<input checked="" type="checkbox"/> Show in inbox workflow log.	
User Task Properties													
GENERAL													
ID	1 usertask1												
*Name	2 Approval Task (Local Manager)												
Documentation	(empty box)												
<input checked="" type="checkbox"/> Show in inbox workflow log.													
<p>20. Select <b>Add Task</b> .</p> <p>21. Select <b>Script Task</b> .</p>	 <pre> graph LR     A[Approval Task (Local Manager)] --- B((User Task))     B --- C((Service Task))     B --- D((Script Task))     B --- F((Mail Task))     </pre>												

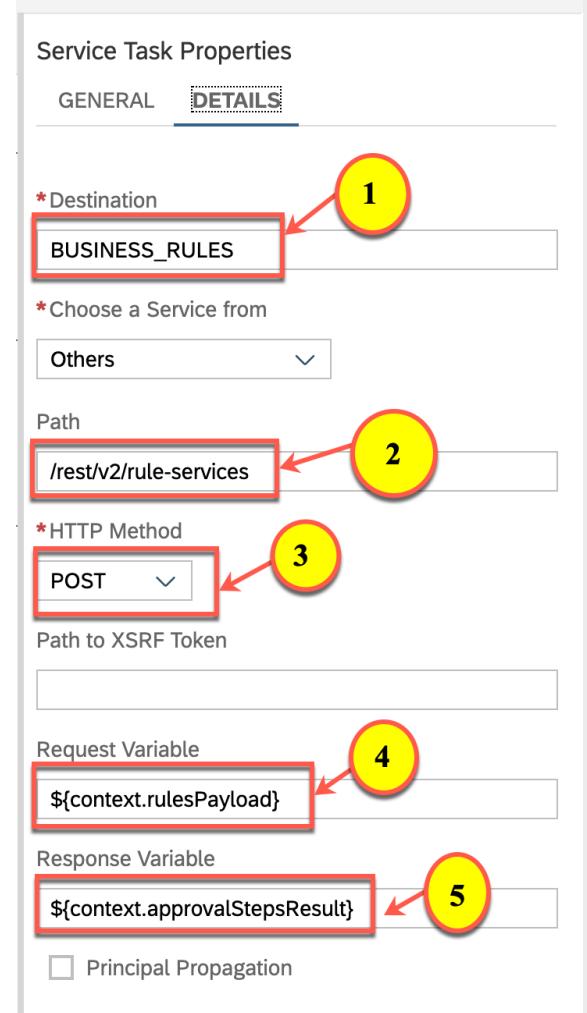
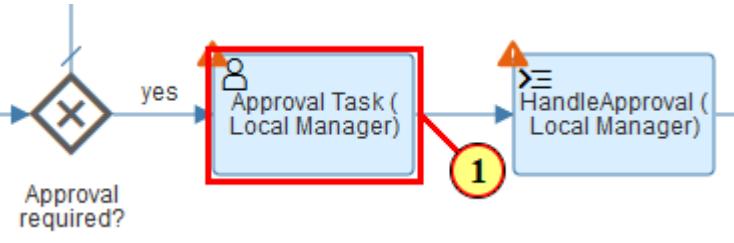
Explanation	Screenshot
<p>22. Enter <b>Handle Approval (Local Manager)</b> as <b>Name</b>.</p> <p>Note: We will configure the task later.</p>	 <p>The screenshot shows the 'Script Task Properties' dialog. The 'Name' field contains 'HandleApproval (Local Manager)' and is highlighted with a red box. A yellow circle with the number '1' points to the 'Documentation' field below it.</p>
<p>23. Click <b>Add Gateway</b> .</p> <p>24. Select <b>Exclusive Gateway</b> .</p>	 <p>The screenshot shows a workflow canvas with a script task labeled 'HandleApproval (Local Manager)'. To its right are icons for various gateways: Exclusive Gateway (highlighted with a red box and yellow circle '1'), Parallel Gateway (yellow circle '2'), and others. A yellow circle '1' also points to the 'Exclusive Gateway' icon.</p>
<p>25. Enter <b>Approval result?</b> as <b>Name</b>.</p>	 <p>The screenshot shows the 'Exclusive Gateway Properties' dialog. The 'Name' field contains 'Approval result?' and is highlighted with a red box. A yellow circle '1' points to the 'Documentation' field below it.</p>

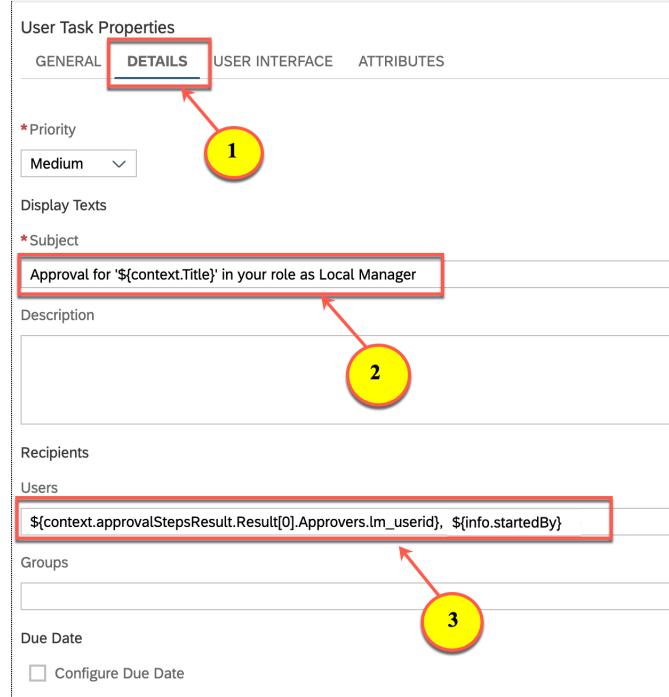
Explanation	Screenshot
<p>26. Select <b>Add Gateway</b> .</p> <p>27. Select <b>Exclusive Gateway</b> .</p>	
<p>28. Enter <b>join</b> as gateway <b>Name</b>.</p> <p>Note: This gateway will be joining the branches from a previous gateway together. In most cases, such joining gateway are optional and implicit, when sequence flows point to the same element. However, when the next element is a gateway as well, this needs to be made explicit.</p>	<p>Exclusive Gateway Properties</p> <p>ID: exclusivegateway3</p> <p>*Name: <b>join</b> </p> <p>Documentation: </p>
<p>29. Select the <b>Exclusive Gateway</b>  with the name '<b>Approval required?</b>'</p> <p>30. Click on the <b>Sequence Flow Connector</b>  and start dragging it</p> <p>31. Drop the <b>Sequence Flow Connector</b> onto the top corner of the <b>Join gateway</b>.</p> <p>Note: Sequence Flow Connectors have Connection Dots at each end. Those can be placed either in the middle or on certain positions on the edge of Flow Elements. The Layout Assistant takes this into account when drawing the connector. Additionally, the lines can be dragged with the mouse.</p>	

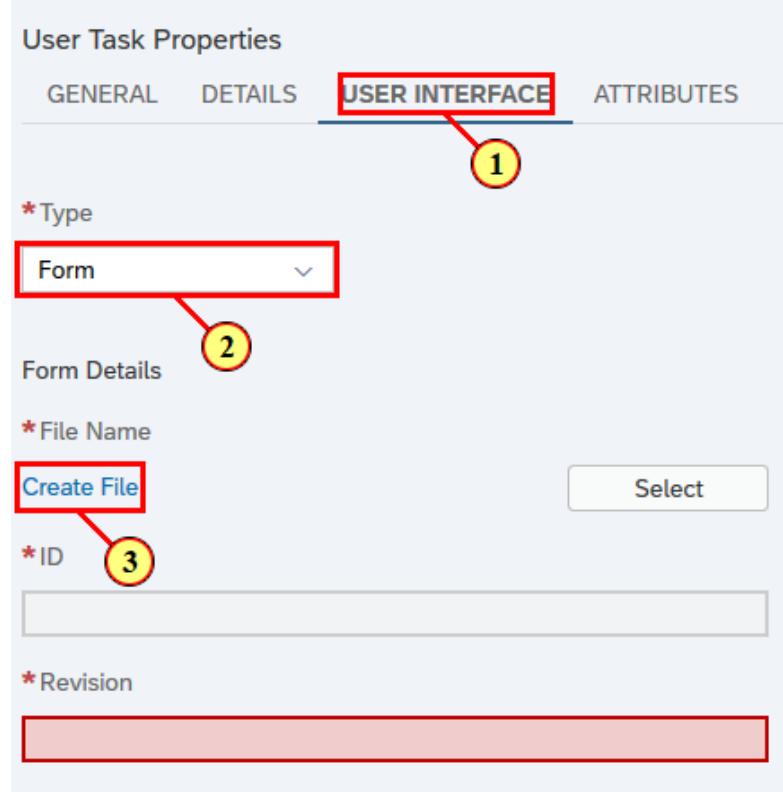
Explanation	Screenshot
<p>32. Drag the <b>Connection Dot</b>  from the middle of the Exclusive Gateway with name <b>Approval required?</b> to its upper corner.</p> <p>33. Drop on <b>the upper corner</b> of the icon.</p>	
<p>Since there is now more than one Sequence Flow, each flow has a name and branch condition.</p> <p>34. Enter <b>no</b> as <b>Name</b> for the newly created (upper) Sequence Flow.</p> <p>35. Select <b>Default</b> to make this as the default branch. This will mean that if none of the condition will meet then this default route will be executed.</p>	
<p>36. Select the <b>Sequence Flow</b>  between the gateway 'Approval required?' and the user task 'Approval Task (Local Manager)'</p>	

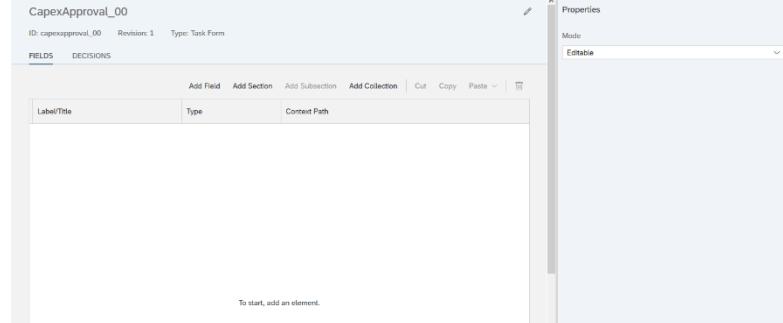
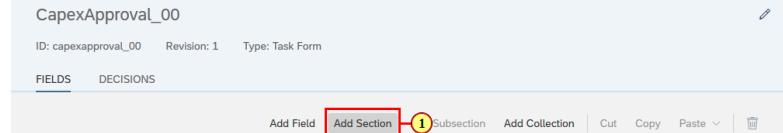
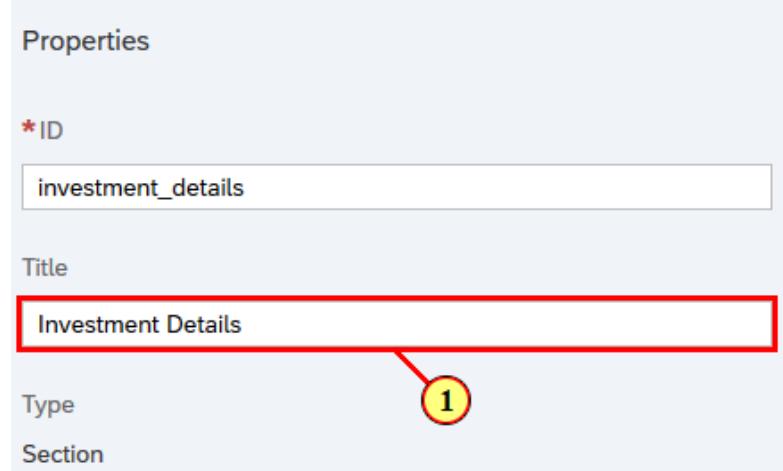
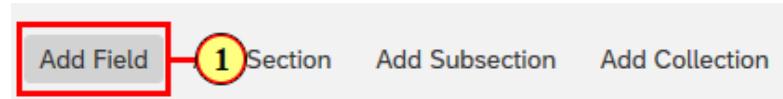
Explanation	Screenshot
<p>37. Enter <b>yes</b> as <b>Sequence Flow name</b>.</p> <p>38. Enter  <code>#{context.approvalStepsResult.Result[0].Approvers.lm_required}</code> as <b>Condition</b>.</p> <p><b>⚠️</b> Be careful during copying. There are no spaces in the condition. Sometimes during copy, spaces are added automatically. In this case, they need to be removed.</p> <p>Note: This expression uses data in the workflow context to evaluate whether this branch should be taken. This data was retrieved in previous step by a service call to Business Rules.</p>	 <p>Sequence Flow Properties</p> <p>ID: sequenceflow6</p> <p>*Name: yes (highlighted with red box)</p> <p>Documentation: (empty)</p> <p>*Condition: text.approvalStepsResult.Result[0].Approvers.lm_required (highlighted with red box)</p> <p>Default: (unchecked)</p>
<p>39. Select the script task <b>Prepare Data</b>.</p>	 <pre> graph LR     Start(( )) --&gt; Capex Request  PD[Prepare Data]     PD --&gt; RAS[Retrieve Approval Steps]   </pre>
<p>40. In the properties, click <b>Create File</b>.</p>	 <p>Script Task Properties</p> <p>ID: scripttask1</p> <p>*Name: Prepare Data</p> <p>Documentation: (empty)</p> <p>*Script File: Create File (highlighted with red box)</p> <p>Select: (button)</p>

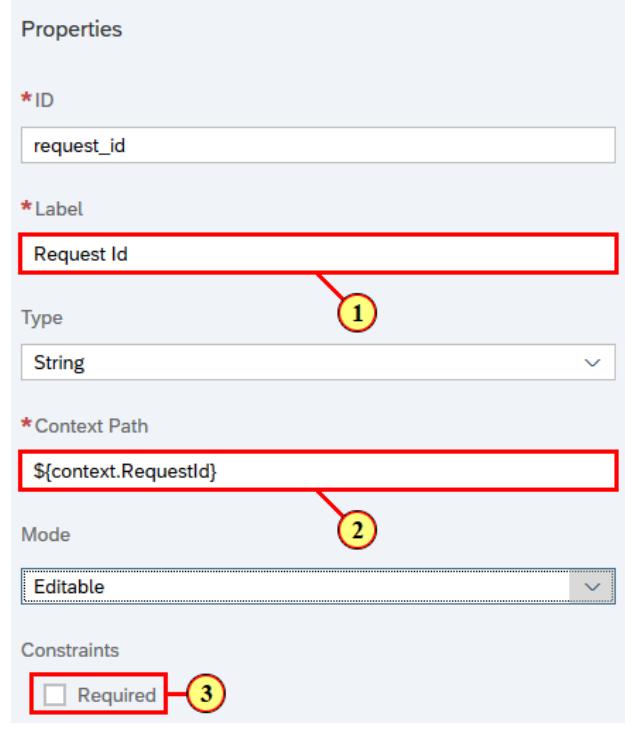
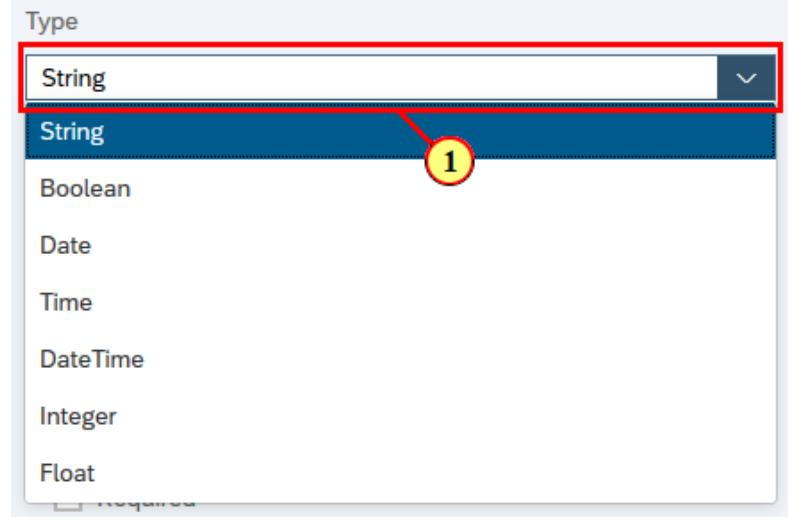
Explanation	Screenshot
<p>41. Enter <b>PrepareData</b> as Name.</p> <p>42. Click on <b>Create</b>.</p>	
<p>43. Delete all the default content and replace it with the script given in the file <b>PrepareData.js</b>.</p> <p>Note: This workflow calls a pre-build rule which you don't need to change now. In a business rules exercise, a custom rule will be created and then you can replace it with your rule service id.</p> <p>44. Close the file by clicking on .</p>	 <pre> 1 // Change this value: 2 var ruleServiceId = "48e84629d7a442bcb43dfc05c7746b36"; 3 4 // 5 // no modifications needed below this line 6 7 // copy process initiator into context 8 \$.context.startedBy = \$.info.startedBy; 9 10 // check if this is the first time we come here (process start) or a rework trip 11 var decision; 12 if (\$.context.History) { 13     // handle rework 14     decision = { 15         "UserId": \$.usertasks.usertask2.last.processor, 16         "Role": "Requester", 17         "Action": "Reworked", 18         "Comment": \$.context.comment 19     }; 20 } else { 21     // initialize context 22     \$.context.Requester.RequestDate = new Date().toISOString().slice(0, 10); 23     \$.context.History = []; 24     decision = { 25         "UserId": \$.info.startedBy, 26         "Role": "Requester", 27         "Action": "Initial Request", 28         "Comment": \$.context.Requester.Comment 29     }; 30 } 31 32 \$.context.History.push(decision); 33 \$.context.comment = ""; 34 </pre>
<p>45. Select the Service Task <b>Retrieve Approval Steps</b>.</p>	

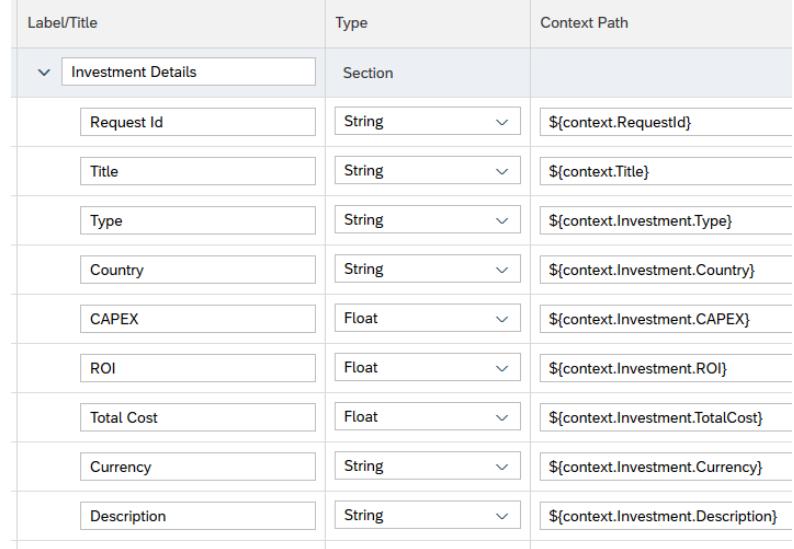
Explanation	Screenshot
<p>46. Select the <b>DETAILS</b> tab.</p> <p>47. Enter <b>BUSINESS_RULES</b> as <b>Destination</b>.</p> <p>48. Enter <b>/rest/v2/rule-services</b> as <b>Path</b>.</p> <p>49. Enter HTTP Method as <b>POST</b>.</p> <p>50. Enter <b> \${context.rulesPayload}</b> as <b>Request Variable</b>.</p> <p>51. Enter <b> \${context.approvalStepsResult}</b> as <b>Response Variable</b>.</p>	 <p>Service Task Properties</p> <p>GENERAL DETAILS</p> <p>*Destination <b>BUSINESS_RULES</b> <span style="color: red;">1</span></p> <p>*Choose a Service from Others</p> <p>Path <b>/rest/v2/rule-services</b> <span style="color: red;">2</span></p> <p>*HTTP Method <b>POST</b> <span style="color: red;">3</span></p> <p>Path to XSRF Token</p> <p>Request Variable <b> \${context.rulesPayload}</b> <span style="color: red;">4</span></p> <p>Response Variable <b> \${context.approvalStepsResult}</b> <span style="color: red;">5</span></p> <p><input type="checkbox"/> Principal Propagation</p>
<p>52. Select the User Task <b>Approval Task (Local Manager)</b>.</p>	 <pre> graph LR     A{Approval required?} -- yes --&gt; B[Approval Task (Local Manager)]     B --&gt; C[HandleApproval (Local Manager)]     </pre> <p>Approval required?</p> <p>yes</p> <p>Approval Task (Local Manager) <span style="color: red;">1</span></p> <p>HandleApproval (Local Manager)</p>

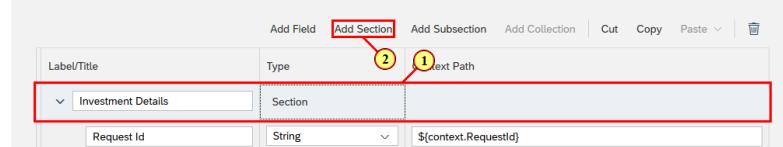
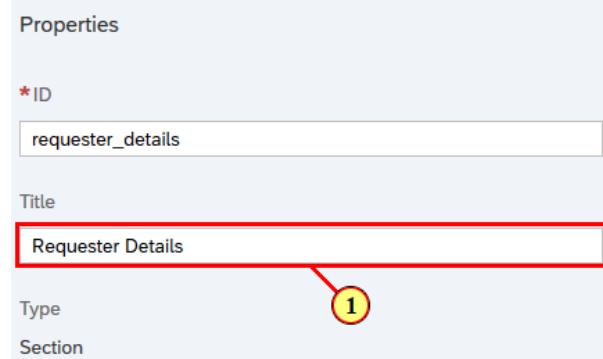
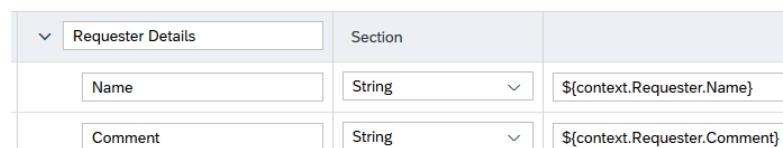
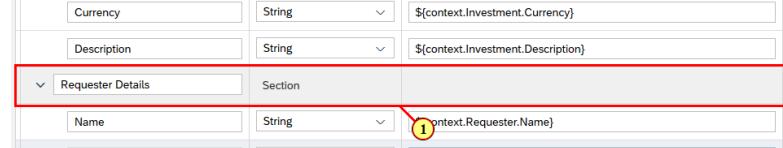
Explanation	Screenshot
<p>53. Select the <b>DETAILS</b> tab.</p> <p>54. Enter <b>Approval for '\${context.Title}' in your role as Local Manager</b> as <b>Subject</b>.</p> <p>55. Enter <b>`\${context.approvalStepsResu lt.Result[0].Approvers.lm_u serid}, \${info.startedBy}</b> as <b>Recipients - Users</b>.</p> <p>It means that this user task will be sent to the users retrieved from Business Rules and the workflow initiator.</p>	 <p>User Task Properties</p> <p>GENERAL DETAILS USER INTERFACE ATTRIBUTES</p> <p>*Priority Medium</p> <p>Display Texts</p> <p>*Subject Approval for '\${context.Title}' in your role as Local Manager</p> <p>Description</p> <p>Recipients</p> <p>Users \${context.approvalStepsResult.Result[0].Approvers.lm_userid}, \${info.startedBy}</p> <p>Groups</p> <p>Due Date</p> <p><input type="checkbox"/> Configure Due Date</p>

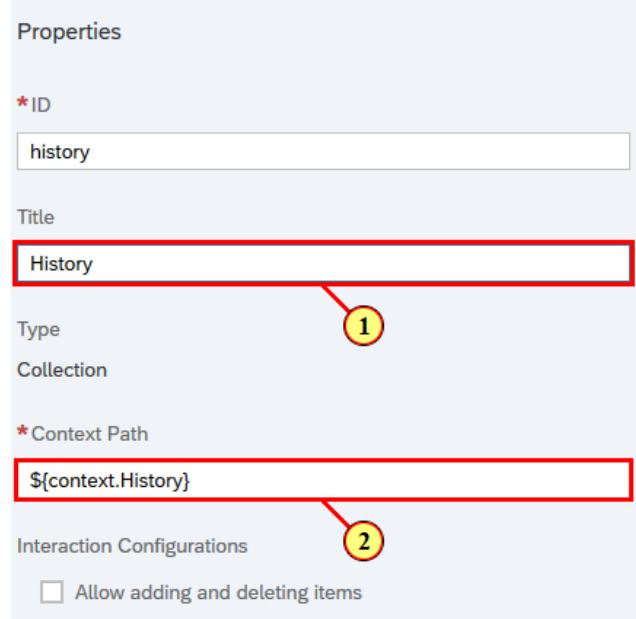
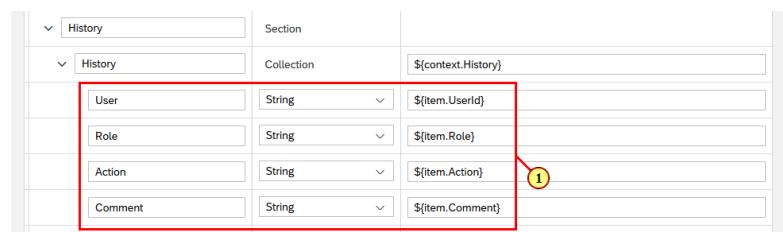
Explanation	Screenshot
<p>56. Select the <b>USER INTERFACE</b> tab.</p> <p>57. Select <b>Form</b> in the dropdown menu for <b>Type</b>.</p> <p>58. Click <b>Create File</b>.</p> <p>Note: Forms are a quick way to create simple task UIs.</p>	 <p>User Task Properties</p> <p>GENERAL DETAILS <b>USER INTERFACE</b> ATTRIBUTES</p> <p>*Type</p> <p>Form</p> <p>Form Details</p> <p>*File Name</p> <p>Create File</p> <p>Select</p> <p>*ID</p> <p>*Revision</p>
<p>59. In the popup, enter <b>CapexApproval_00</b> as <b>Form name</b>.</p> <p>60. Press <b>TAB</b> to accept the generated Id.</p> <p>61. Enter <b>1</b> as <b>Revision</b>.</p> <p>62. Click <b>Create</b>.</p>	 <p>New Form</p> <p>*Name</p> <p>CapexApproval_00</p> <p>*ID</p> <p>capexapproval_00</p> <p>*Revision</p> <p>1</p> <p>Type</p> <p>Task Form</p> <p>Create</p>

Explanation	Screenshot
<p>Note: With Workflow Forms, building Fiori form UIs becomes as easy as building workflows. The Forms Editor enables a declarative low-code approach to rapidly develop simple form UIs. Due to its full integration into Workflow a form can directly work on the values of the workflow context.</p>	
<p><b>63. Click Add Section.</b></p>	
<p><b>64. Enter Investment Details as Title.</b> Note: The ID field is automatically filled out based on the given title.</p>	
<p><b>65. Select Add Field.</b></p>	

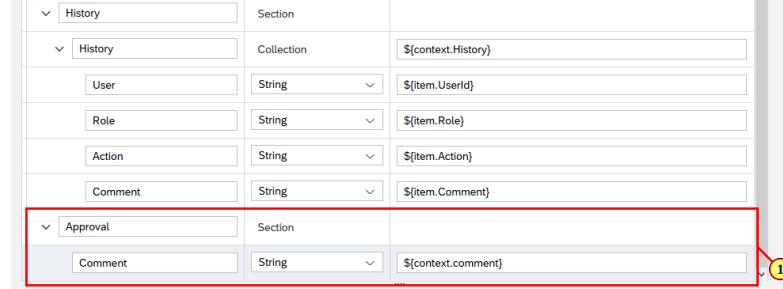
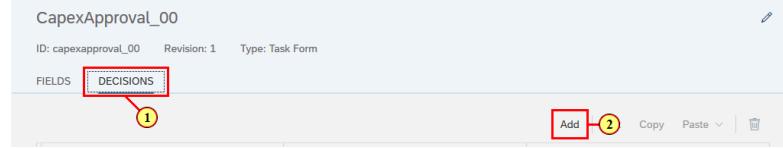
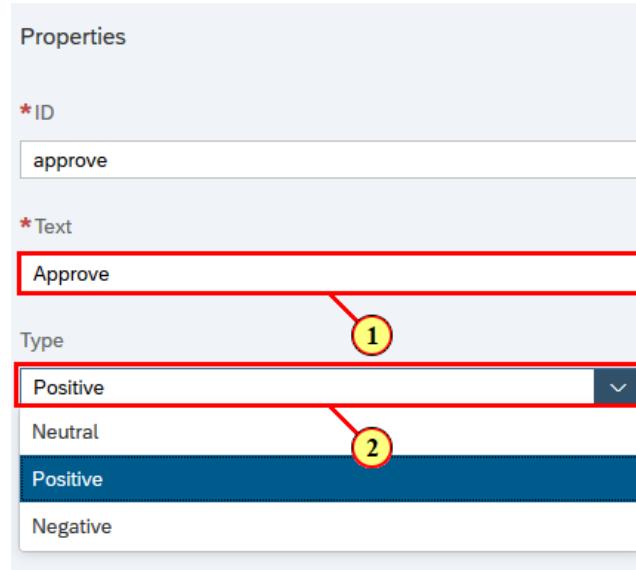
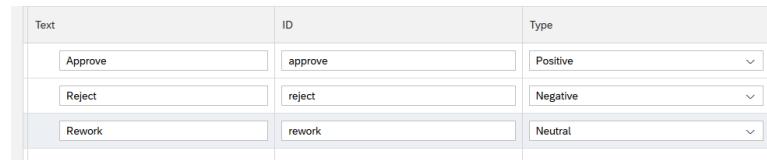
Explanation	Screenshot
<p>66. Enter <b>Request Id</b> as <b>Label</b>. The ID is then automatically derived from this.</p> <p>67. Enter <code> \${context.RequestId}</code> as <b>Context Path</b>.</p> <p>68. An input field can also be marked as <b>Required</b>. Since the <b>Request Id</b> is already given and should not be editable, leave it unchecked.</p>	 <p>Properties</p> <p>* ID request_id</p> <p>* Label Request Id <span style="border: 2px solid red; border-radius: 50%; padding: 2px;">1</span></p> <p>Type String</p> <p>* Context Path \${context.RequestId} <span style="border: 2px solid red; border-radius: 50%; padding: 2px;">2</span></p> <p>Mode Editable</p> <p>Constraints <input type="checkbox"/> Required <span style="border: 2px solid red; border-radius: 50%; padding: 2px;">3</span></p>
<p>69. Clicking on the <b>Type</b> field opens a dropdown list. Select the type as <b>String</b>.</p>	 <p>Type</p> <p>String <span style="border: 2px solid red; border-radius: 50%; padding: 2px;">1</span></p> <p>String</p> <p>Boolean</p> <p>Date</p> <p>Time</p> <p>DateTime</p> <p>Integer</p> <p>Float</p>
<p>70. Clicking on the <b>Mode</b> field opens a dropdown list. Select <b>Display-Only</b>.</p>	 <p>Mode</p> <p>Display-Only</p> <p>Editable</p> <p>Display-Only</p>

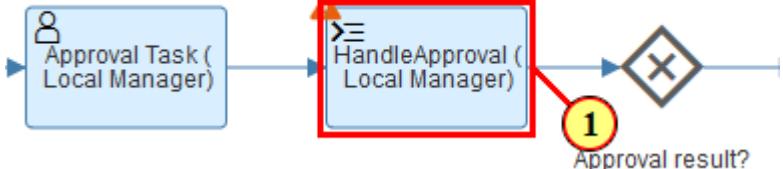
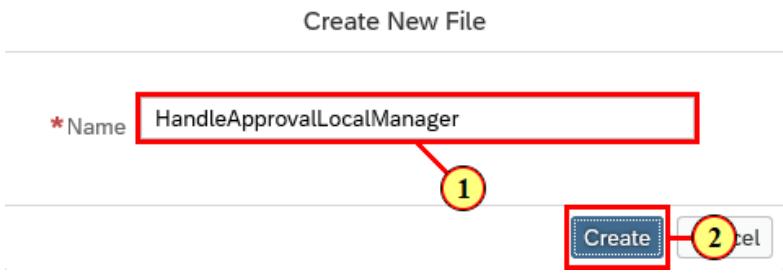
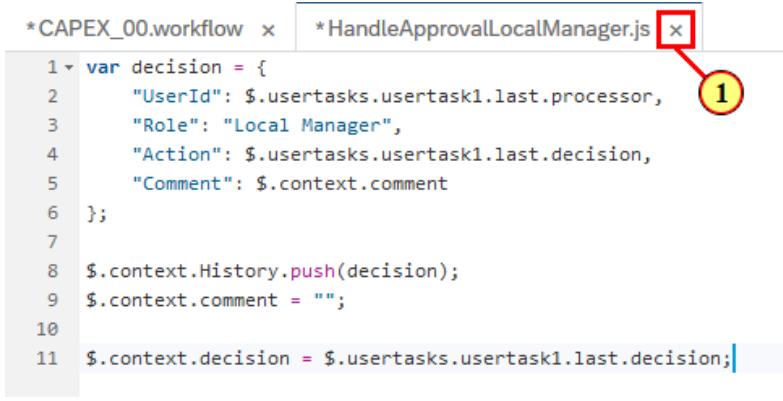
Explanation	Screenshot																														
<p>71. In the same way, add the following fields with given Label, Type and Context Path:</p> <ul style="list-style-type: none"> <li>○ Set the property of all of the fields as <b>Editable</b> and <b>Required</b>.</li> </ul> <p><b>Label: Title</b>  <b>Type:</b> String  <b>Context Path:</b> \${context.Title}</p> <p><b>Label: Type</b>  <b>Type:</b> String  <b>Context Path:</b>  \${context.Investment.Type}</p> <p><b>Label: Country</b>  <b>Type:</b> String  <b>Context Path:</b>  \${context.Investment.Country}</p> <p><b>Label: CAPEX</b>  <b>Type:</b> Float  <b>Context Path:</b>  \${context.Investment.CAPEX}</p> <p><b>Label: ROI</b>  <b>Type:</b> Float  <b>Context Path:</b>  \${context.Investment.ROI}</p> <p><b>Label: Total Cost</b>  <b>Type:</b> Float  <b>Context Path:</b>  \${context.Investment.TotalCost}</p> <p><b>Label: Currency</b>  <b>Type:</b> String  <b>Context path:</b>  \${context.Investment.Currency}</p> <p><b>Label: Description</b>  <b>Type:</b> String  <b>Context Path:</b>  \${context.Investment.Description}</p>	 <table border="1" data-bbox="687 798 1481 1347"> <thead> <tr> <th data-bbox="687 798 1008 861">Label/Title</th> <th data-bbox="1008 798 1176 861">Type</th> <th data-bbox="1176 798 1481 861">Context Path</th> </tr> </thead> <tbody> <tr> <td data-bbox="687 861 1008 903">Request Id</td> <td data-bbox="1008 861 1176 903">String</td> <td data-bbox="1176 861 1481 903">\${context.RequestId}</td> </tr> <tr> <td data-bbox="687 903 1008 946">Title</td> <td data-bbox="1008 903 1176 946">String</td> <td data-bbox="1176 903 1481 946">\${context.Title}</td> </tr> <tr> <td data-bbox="687 946 1008 988">Type</td> <td data-bbox="1008 946 1176 988">String</td> <td data-bbox="1176 946 1481 988">\${context.Investment.Type}</td> </tr> <tr> <td data-bbox="687 988 1008 1030">Country</td> <td data-bbox="1008 988 1176 1030">String</td> <td data-bbox="1176 988 1481 1030">\${context.Investment.Country}</td> </tr> <tr> <td data-bbox="687 1030 1008 1072">CAPEX</td> <td data-bbox="1008 1030 1176 1072">Float</td> <td data-bbox="1176 1030 1481 1072">\${context.Investment.CAPEX}</td> </tr> <tr> <td data-bbox="687 1072 1008 1115">ROI</td> <td data-bbox="1008 1072 1176 1115">Float</td> <td data-bbox="1176 1072 1481 1115">\${context.Investment.ROI}</td> </tr> <tr> <td data-bbox="687 1115 1008 1157">Total Cost</td> <td data-bbox="1008 1115 1176 1157">Float</td> <td data-bbox="1176 1115 1481 1157">\${context.Investment.TotalCost}</td> </tr> <tr> <td data-bbox="687 1157 1008 1199">Currency</td> <td data-bbox="1008 1157 1176 1199">String</td> <td data-bbox="1176 1157 1481 1199">\${context.Investment.Currency}</td> </tr> <tr> <td data-bbox="687 1199 1008 1241">Description</td> <td data-bbox="1008 1199 1176 1241">String</td> <td data-bbox="1176 1199 1481 1241">\${context.Investment.Description}</td> </tr> </tbody> </table>	Label/Title	Type	Context Path	Request Id	String	\${context.RequestId}	Title	String	\${context.Title}	Type	String	\${context.Investment.Type}	Country	String	\${context.Investment.Country}	CAPEX	Float	\${context.Investment.CAPEX}	ROI	Float	\${context.Investment.ROI}	Total Cost	Float	\${context.Investment.TotalCost}	Currency	String	\${context.Investment.Currency}	Description	String	\${context.Investment.Description}
Label/Title	Type	Context Path																													
Request Id	String	\${context.RequestId}																													
Title	String	\${context.Title}																													
Type	String	\${context.Investment.Type}																													
Country	String	\${context.Investment.Country}																													
CAPEX	Float	\${context.Investment.CAPEX}																													
ROI	Float	\${context.Investment.ROI}																													
Total Cost	Float	\${context.Investment.TotalCost}																													
Currency	String	\${context.Investment.Currency}																													
Description	String	\${context.Investment.Description}																													

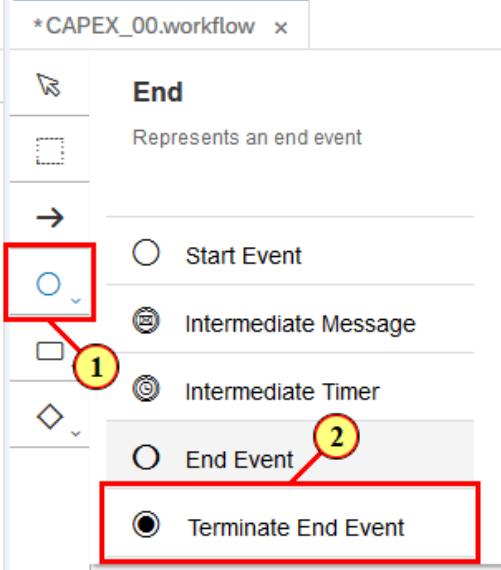
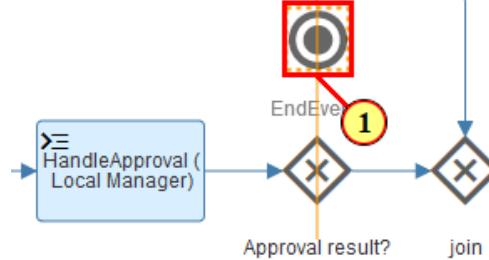
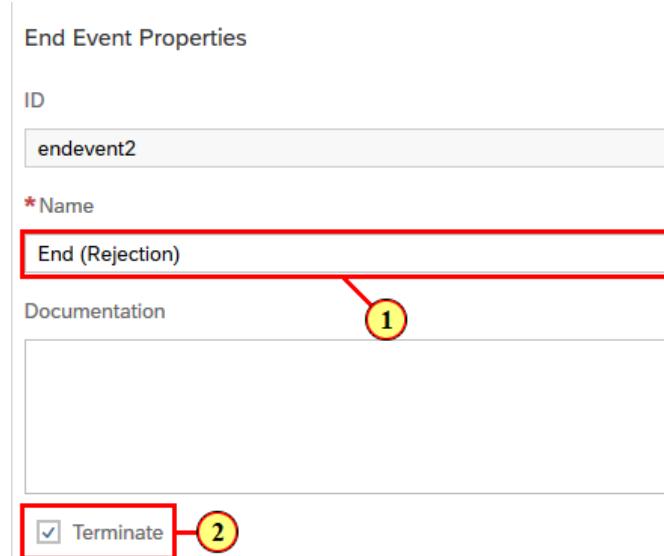
Explanation	Screenshot
<p>72. Select <b>Investment Details</b> section.</p> <p>73. Select <b>Add Section</b>.</p>	
<p>74. Enter <b>Requester Details</b> as Title.</p>	
<p>75. Add the following fields.</p> <p><b>Label:</b> Name  <b>Type:</b> String  <b>Mode:</b> Display-Only  <b>Control:</b> Input  <b>Context Path:</b> \${context.Requester.Name}</p> <p><b>Label:</b> Comment  <b>Type:</b> String  <b>Mode:</b> Editable  <b>Control:</b> Text Area  <b>Required:</b> no  <b>Context Path:</b> \${context.Requester.Comment}</p>	
<p>76. Select the <b>Requester Details</b> section.</p>	
<p>77. Select <b>Add Section</b>. Name the new Section <b>History</b>.</p>	

Explanation	Screenshot
<p>78. Select <b>Add Collection</b>.</p> <p>Note: Collections allow for an array of items to be displayed as table.</p>	
<p>79. Enter <b>History</b> as <b>Title</b>.</p> <p>80. Enter <code>#{context.History}</code> as <b>Context Path</b>.</p>	
<p>81. Add the following values for <b>Label</b> and <b>Context Path</b> and select Type <b>String</b> and mark them as <b>Display-Only</b>.</p> <p><b>Label: User</b> Context Path: \${item.UserId}</p> <p><b>Label: Role</b> Context Path: \${item.Role}</p> <p><b>Label: Action</b> Context Path: \${item.Action}</p> <p><b>Label: Comment</b> Context Path: \${item.Comment}</p> <p>Note that inside a collection, the variable <b>item</b> refers to a single row.</p>	

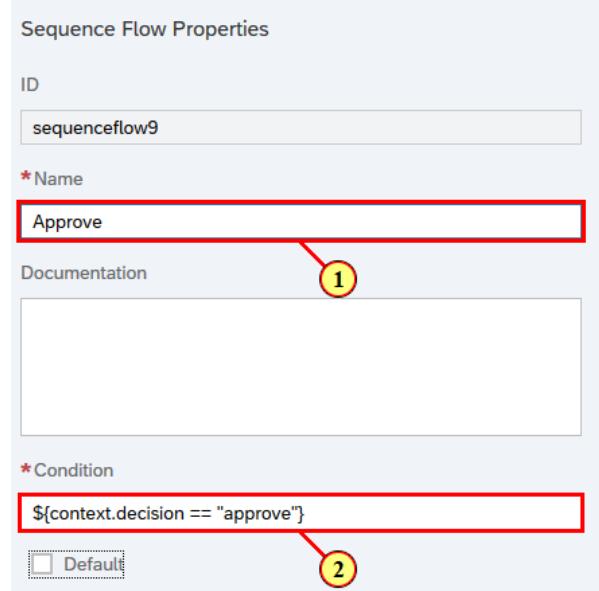
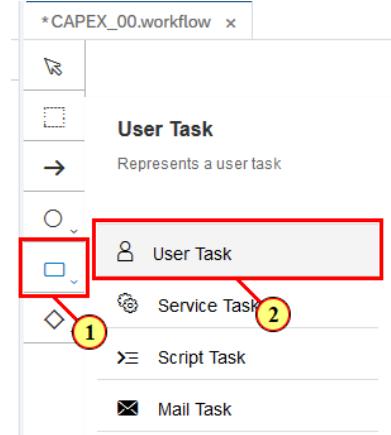
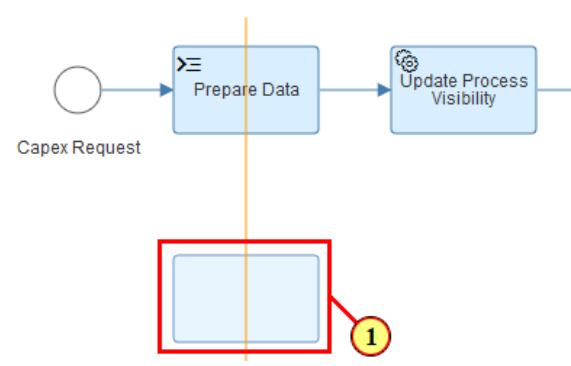
Explanation	Screenshot
<p>82. Add a new Section with title <b>Approval</b>.</p> <p>83. Add a new field and name it <b>Comment</b>.</p> <p>84. Enter <code>#{context.comment}</code> as <b>Context Path</b></p> <p>85. Change the <b>UI Control type</b> from <i>Input field</i> to <b>Text Area</b>.</p>	<p>The screenshot shows the 'Properties' dialog for a new field. The 'Label' field contains 'Comment' (circled in red, step 2). The 'Type' dropdown is set to 'String' (circled in red, step 2). The 'Context Path' field contains the expression '#{context.comment}' (circled in red, step 3). The 'Mode' dropdown is set to 'Editable' (circled in red, step 3). In the 'UI Configuration' section, the 'Control' dropdown is set to 'Text Area' (circled in red, step 4), which is highlighted in blue. Other options like 'Input' and 'Dropdown' are also listed.</p>

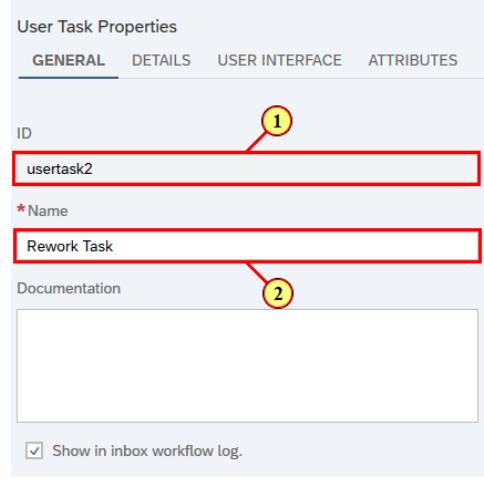
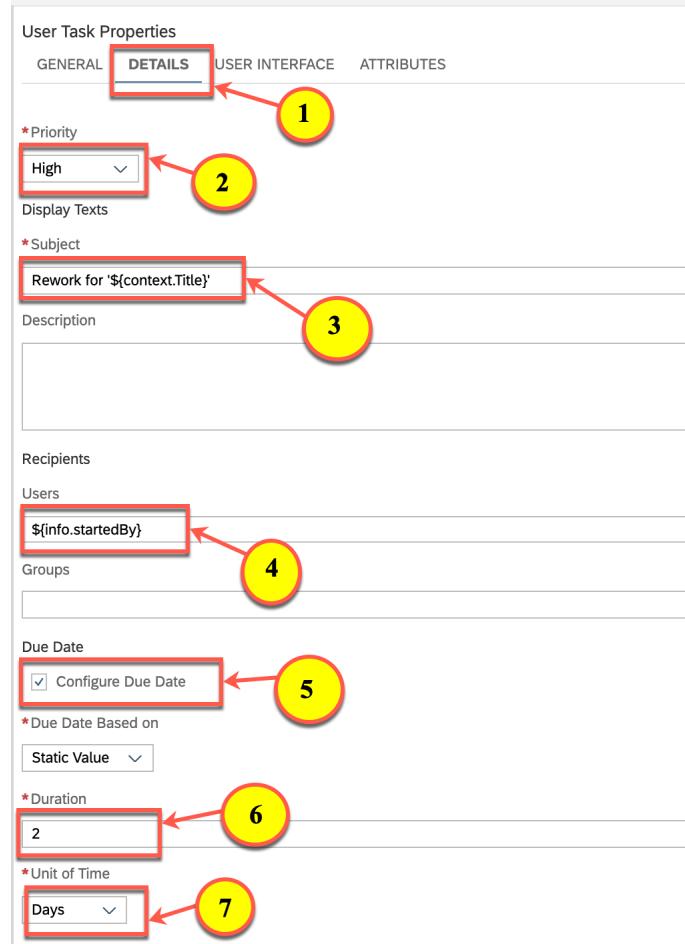
Explanation	Screenshot
86. Ensure your Form looks like this.	
87. Select the <b>Decisions</b> tab. 88. Select <b>Add</b> to add a new decision option.	
89. Enter <b>Approve</b> as <b>Text</b> of the decision button. 90. Select <b>Positive</b> as <b>Type</b> from the dropdown-list.	
91. In the same way, add a decision option <b>Reject</b> with type <b>Negative</b> and one with text <b>Rework</b> or <b>Neutral</b> type.	
92. <b>Close</b> the file by clicking <b>X</b> .	

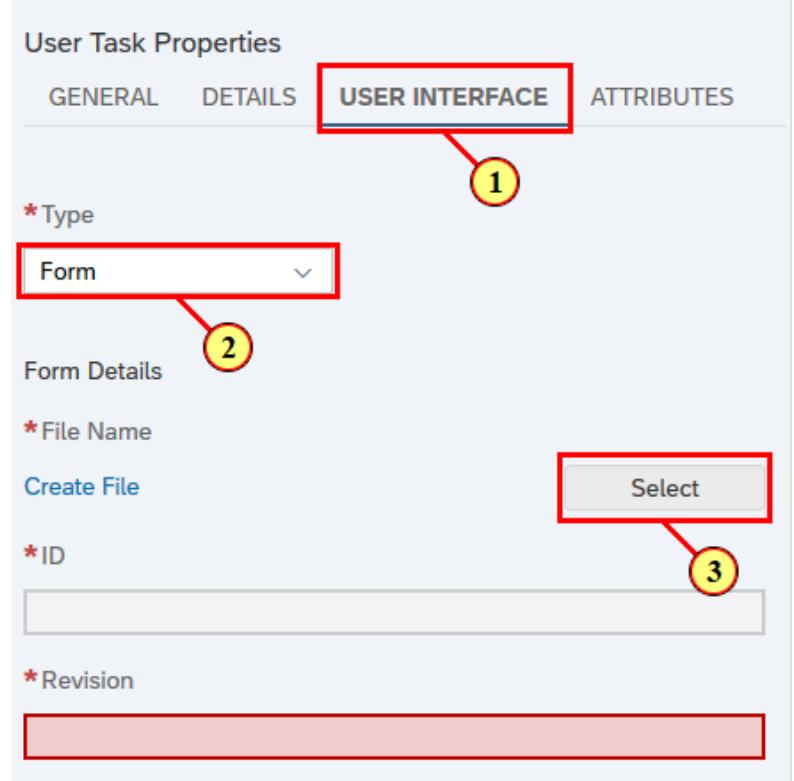
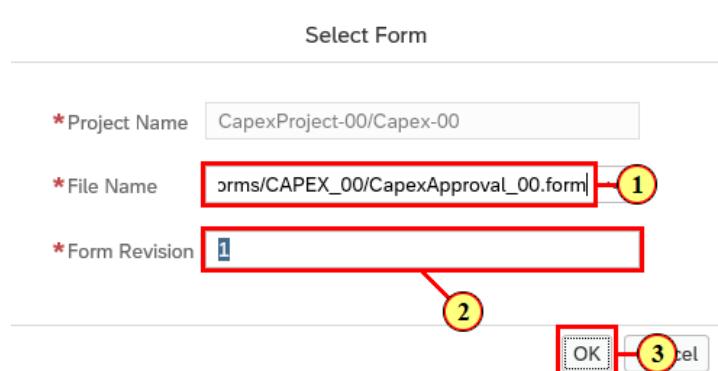
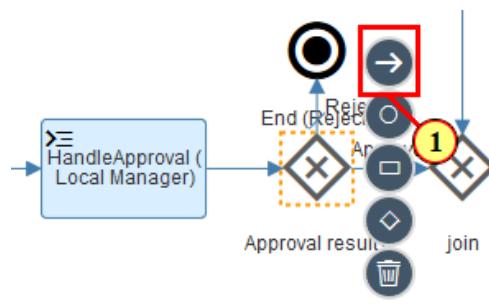
Explanation	Screenshot
<p>93. Select the script task <b>Handle Approval (Local Manager)</b>.</p>	
<p>94. In the <b>Script Task Properties</b>, select <b>Create File</b>.</p>	
<p>95. Enter <b>HandleApprovalLocalManager</b> as Filename  96. Select <b>Create</b>.</p>	
<p>97. Replace the default content with the script from <b>HandleApprovalLocalManager.js</b> in the exercises folder.</p> <p>98. Close the file by clicking .</p> <p>Note: This script refers to the Task ID (usertask1) multiple times. This is the id of the User Task "Approval Task (Local Manager)" which was mentioned earlier. It is automatically assigned to each task. If your ID is different, it needs to be replaced here. Please note only replace if your user task has a different id.</p>	 <pre>* CAPEX_00.workflow * HandleApprovalLocalManager.js 1 var decision = { 2   "UserId": \$.usertasks.usertask1.last.processor, 3   "Role": "Local Manager", 4   "Action": \$.usertasks.usertask1.last.decision, 5   "Comment": \$.context.comment 6 }; 7 8 \$.context.History.push(decision); 9 \$.context.comment = ""; 10 11 \$.context.decision = \$.usertasks.usertask1.last.decision;</pre>

Explanation	Screenshot
<p>99. From the left sidebar, select <b>Add Event</b>.</p> <p>100. Select the <b>Terminate End Event</b>.</p>	
<p>101. Place the <b>End Event</b> above the exclusive gateway named <b>Approval result?</b></p> <p>Note: The yellow guiding lines help you position it nicely.</p>	
<p>102. Enter <b>End (Rejection)</b> as <b>Name</b>.</p> <p>103. Ensure that <b>Terminate</b> is checked.</p>	

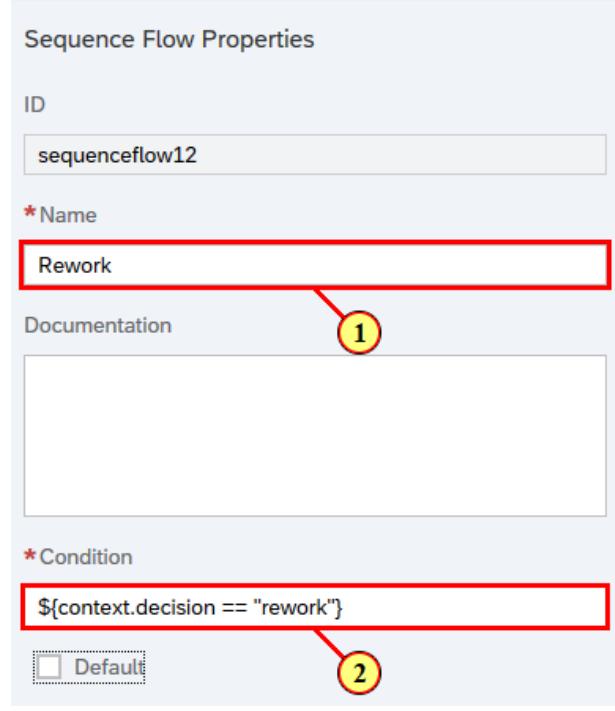
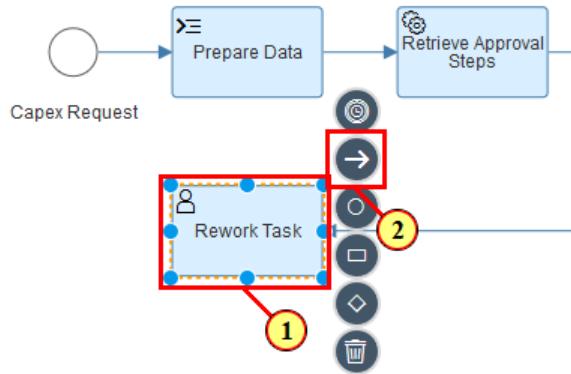
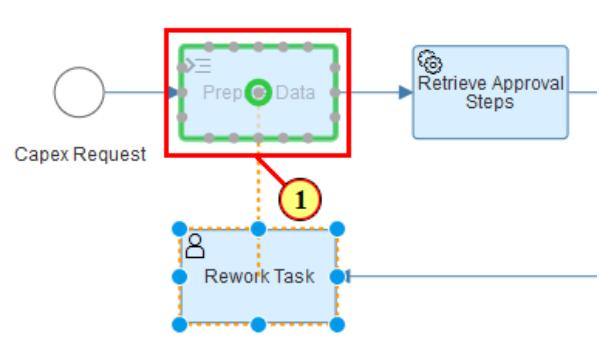
Explanation	Screenshot
<p>104. Select the exclusive gateway with name <b>Approval result?</b></p> <p>105. Drag the sequence flow connector → onto the end event with name <b>End (Rejection)</b></p>	
<p>106. Enter <b>Reject</b> as Name.</p> <p>107. Select the <b>checkbox for Default</b> condition.</p>	<p>Sequence Flow Properties</p> <p>ID: sequenceflow11</p> <p>* Name: <b>Reject</b></p> <p>Documentation: 1</p> <p><input checked="" type="checkbox"/> Default 2</p>
<p>108. Select the sequence flow connector between the gateways named <b>Approval result?</b> and <b>join</b>.</p> <p>Note: Once there is more than one outgoing sequence flow from an Exclusive Gateway, all of them can be named. Exactly one must have the 'default', and all others customized trigger conditions based on the workflow context.</p>	

Explanation	Screenshot
<p>109. Enter <b>Approve</b> as <b>Name</b>.</p> <p>110. Enter <code>#{context.decision == "approve"}</code> as <b>Condition</b>.</p>	 <p>The screenshot shows the 'Sequence Flow Properties' dialog. The 'Name' field contains 'Approve' and the 'Condition' field contains <code>#{context.decision == "approve"}</code>. Both fields are highlighted with red boxes. A yellow circle labeled '1' points to the 'Name' field, and a yellow circle labeled '2' points to the 'Condition' field.</p>
<p>111. From the left sidebar, select <b>Add Task</b> .</p> <p>112. Select <b>User Task</b> .</p>	 <p>The screenshot shows the workflow sidebar with 'User Task' selected. A red box highlights the 'User Task' icon, and a yellow circle labeled '1' points to it. Another yellow circle labeled '2' points to the 'Service Task' icon below it.</p>
<p>113. Place the <b>User Task</b> below the script task <b>Prepare Data</b>.</p> <p>Note: The yellow guiding lines help you position it nicely.</p>	 <p>The screenshot shows the workflow diagram. A blue rounded rectangle labeled 'Prepare Data' is connected by a sequence flow to a blue rounded rectangle labeled 'Update Process Visibility'. A yellow vertical guiding line is positioned to the left of the 'Update Process Visibility' task. A red box highlights a user task icon, and a yellow circle labeled '1' points to its connection point on the vertical line.</p>

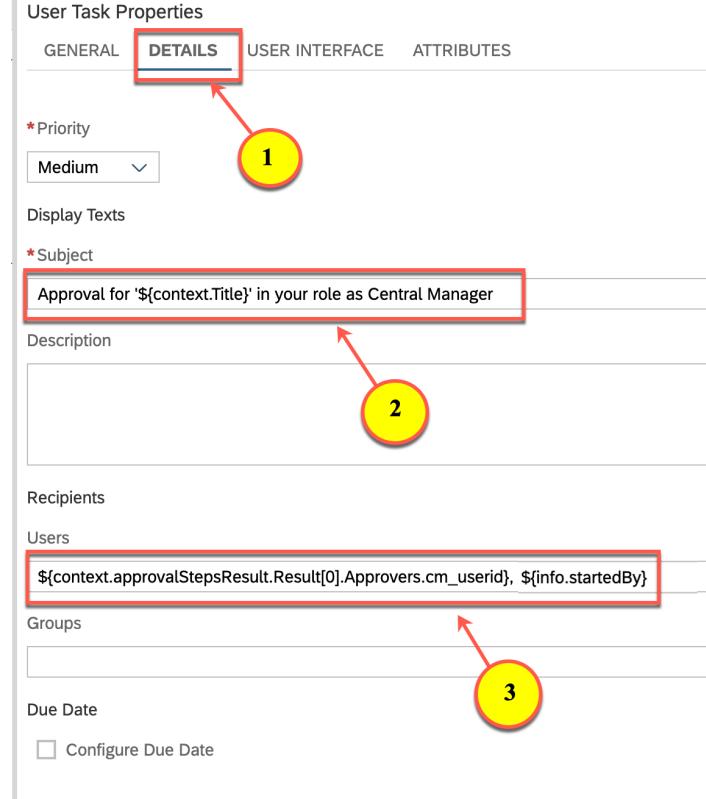
Explanation	Screenshot
<p>114. Verify that the <b>ID</b> is <code>usertask2</code>. If it differs, the references in the script file <b>Prepare Data.js</b> needs to be adapted accordingly.</p> <p>115. Enter <b>Rework Task</b> as <b>Name</b>.</p>	 <p>User Task Properties</p> <p>GENERAL DETAILS USER INTERFACE ATTRIBUTES</p> <p>ID: usertask2 (1)</p> <p>*Name: Rework Task (2)</p> <p>Documentation:</p> <p>Show in inbox workflow log: <input checked="" type="checkbox"/></p>
<p>116. Select the <b>DETAILS</b> tab.</p> <p>117. Select the <b>Priority</b> as <b>High</b>.</p> <p>118. Enter <b>Rework for '\${context.Title}'</b> as <b>Subject</b>.</p> <p>119. Enter <b>\${info.startedBy}</b> as <b>Recipients - User</b>.</p> <p>120. Select <b>Configure Due Date</b>.</p> <p>121. Enter <b>2</b> as <b>Duration</b> and <b>Days</b> from the dropdown list <b>Unit of Time</b>, to mark the task as being due 2 days after it started.</p>	 <p>User Task Properties</p> <p>GENERAL DETAILS USER INTERFACE ATTRIBUTES</p> <p>*Priority: High (1)</p> <p>Display Texts</p> <p>*Subject: Rework for '\${context.Title}' (2)</p> <p>Description:</p> <p>Recipients</p> <p>Users: \${info.startedBy} (4)</p> <p>Groups:</p> <p>Due Date</p> <p><input checked="" type="checkbox"/> Configure Due Date (5)</p> <p>*Due Date Based on: Static Value</p> <p>*Duration: 2 (6)</p> <p>*Unit of Time: Days (7)</p>

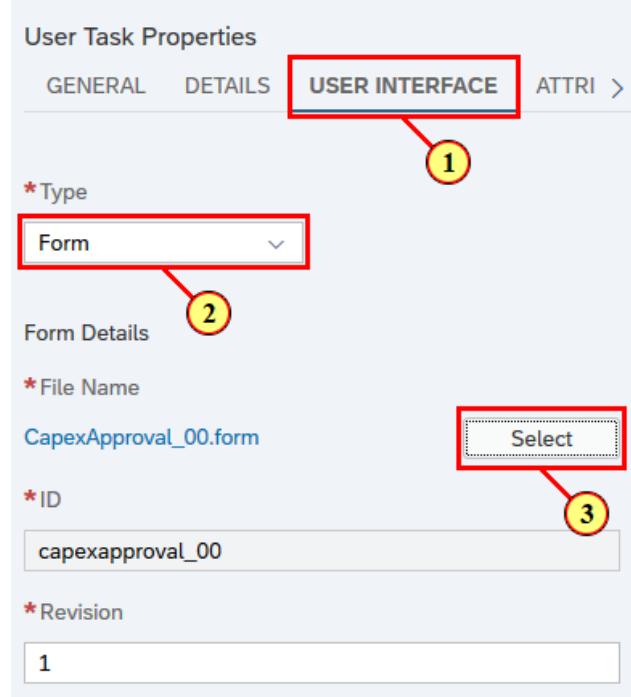
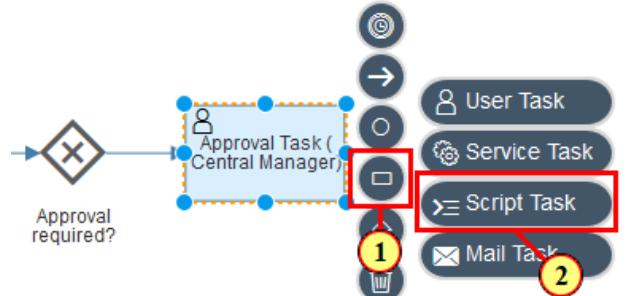
Explanation	Screenshot
<p>122. Select the <b>USER INTERFACE</b> tab.</p> <p>123. Select <b>Form</b> as <b>Type</b>.</p> <p>124. Click <b>Select</b> to choose an existing form.</p>	 <p>User Task Properties</p> <p>GENERAL DETAILS <b>USER INTERFACE</b> ATTRIBUTES</p> <p>* Type</p> <p>Form</p> <p>Form Details</p> <p>* File Name</p> <p>Create File Select</p> <p>* ID</p> <p>* Revision</p>
<p>125. Select the previously created form <b>CapexApproval_00</b></p> <p>126. Ensure that the <b>Revision</b> is 1.</p> <p>127. Select <b>OK</b>.</p>	 <p>Select Form</p> <p>* Project Name CapexProject-00/Capex-00</p> <p>* File Name forms/CAPEX_00/CapexApproval_00.form</p> <p>* Form Revision 1</p> <p>OK Cancel</p>
<p>128. Select the Exclusive Gateway <b>Approval result?</b> and drag an additional <b>Sequence Flow Connector</b>...</p>	 <pre> graph LR     A[HandleApproval Local Manager] --&gt; B{Approval result?}     B --&gt; C(( ))     B --&gt; D(( ))     B --&gt; E(( ))     B --&gt; F(( ))     C --&gt; G(( ))     D --&gt; H(( ))     E --&gt; I(( ))     F --&gt; J(( ))     G --&gt; K(( ))     H --&gt; L(( ))     I --&gt; M(( ))     J --&gt; N(( ))     K --&gt; O(( ))     L --&gt; P(( ))     M --&gt; Q(( ))     N --&gt; R(( ))     O --&gt; S(( ))     P --&gt; T(( ))     Q --&gt; U(( ))     R --&gt; V(( ))     S --&gt; W(( ))     T --&gt; X(( ))     U --&gt; Y(( ))     V --&gt; Z(( ))     </pre>

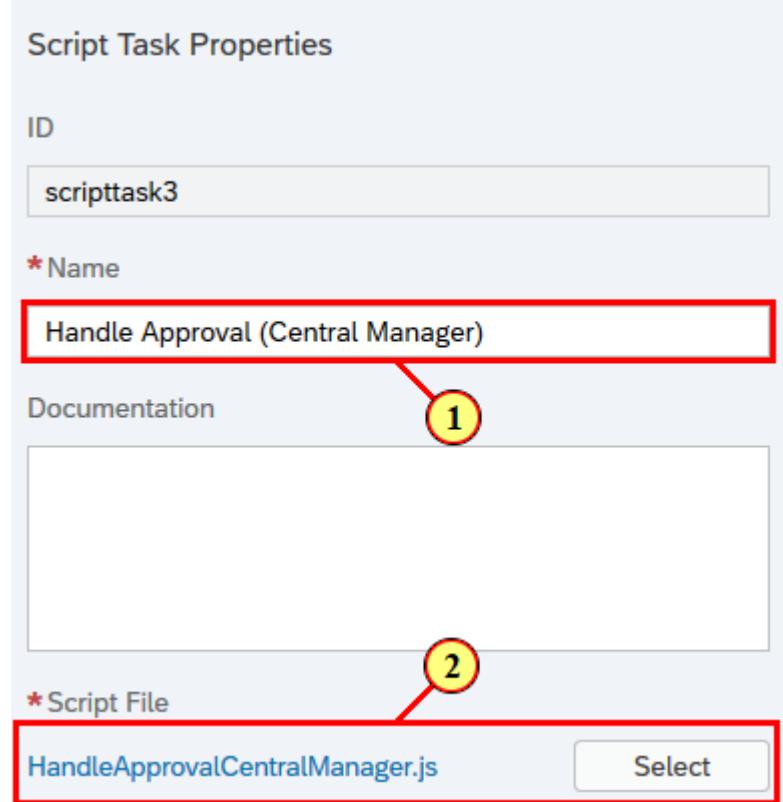
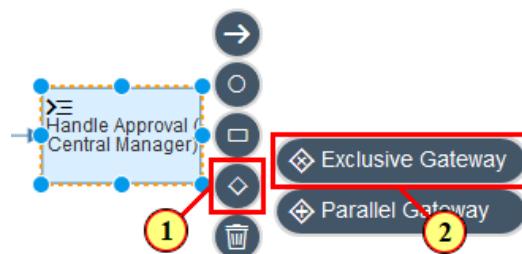
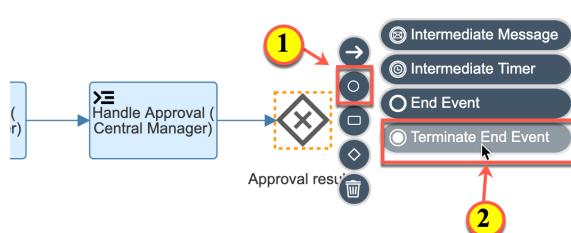
Explanation	Screenshot
<p>...onto the User Task <b>Rework Task</b>. Position it onto the highlighted middle dot.</p>	
<p>129. Click on the <b>Connection Dot</b>  of the new Sequence Flow and drag it from the middle of the Exclusive Gateway... onto the lower edge of the Exclusive Gateway.</p>	
<p>130. Ensure your outgoing Sequence Flow Connectors look like this.</p>	

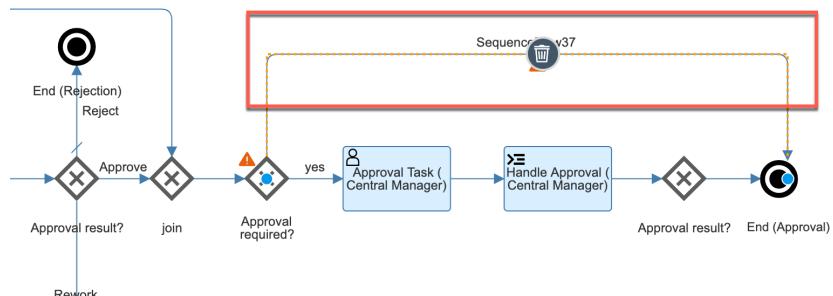
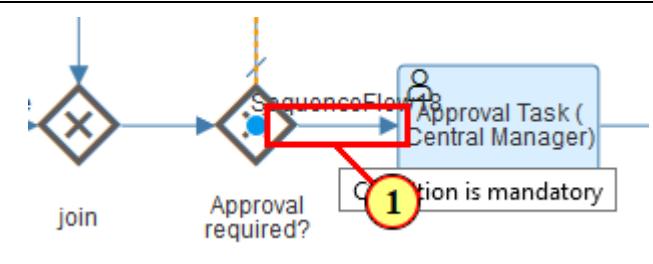
Explanation	Screenshot
<p>131. Enter <b>Rework</b> as <b>Name</b>.</p> <p>132. Enter <code>#{context.decision == "rework"}</code> as <b>Condition</b>.</p>	 <p>Sequence Flow Properties</p> <p>ID: sequenceflow12</p> <p>* Name: Rework</p> <p>Documentation: 1</p> <p>* Condition: \${context.decision == "rework"}</p> <p>Default: 2</p>
<p>133. Select User Task <b>Rework Task</b></p> <p>134. Select <b>Sequence Flow Connector</b> and drag it...</p>	 <p>Capex Request</p> <p>Prepare Data → Retrieve Approval Steps</p> <p>Rework Task</p>
<p>...onto the Script Task <b>Prepare Data</b>.</p>	 <p>Capex Request</p> <p>Prep Data → Retrieve Approval Steps</p> <p>Rework Task</p>

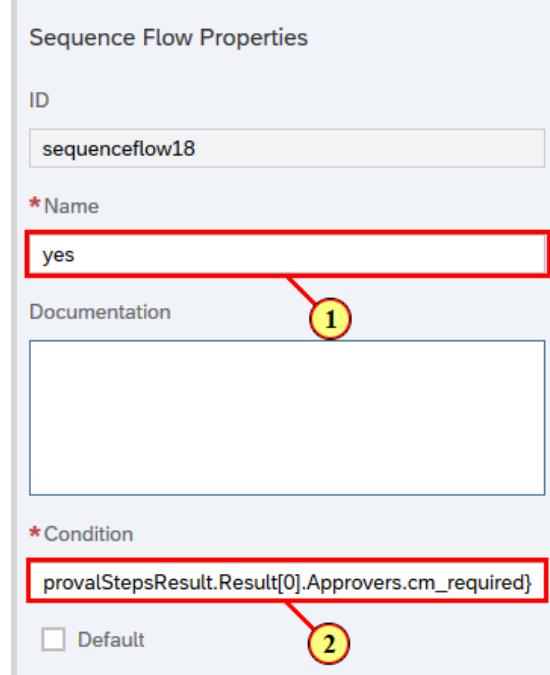
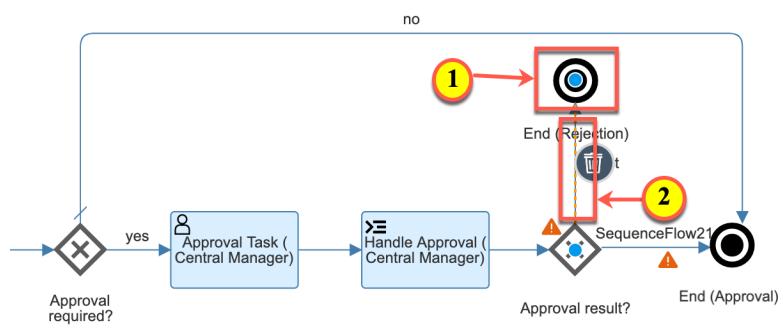
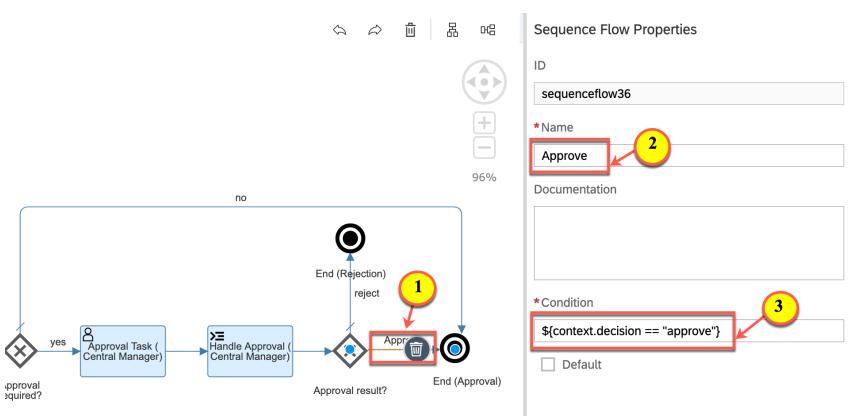
Explanation	Screenshot
<p>Ensure your workflow looks like this.</p>	<pre> graph LR     Start(( )) --&gt; Prep[Prepare Data]     Prep --&gt; Retrieve[Retrieve Approval Steps]     Retrieve --&gt; Decision{Approval required?}     Decision -- yes --&gt; ApprovalTask[Approval Task (Local Manager)]     ApprovalTask --&gt; Handle[Handle Approval Local Manager]     Handle --&gt; Decision     Decision -- no --&gt; EndReject((End (Reject)))     ApprovalTask --&gt; EndApproved((End (Approved)))     EndApproved --&gt; Join(( ))     Decision --&gt; Rework[Rework Task]     Rework --&gt; Prep     </pre>
<p>135. Select the exclusive gateway with name <b>join</b>.          136. Select <b>Add Gateway</b> .          137. Select the <b>Exclusive Gateway</b> option .          138. Name it as <b>Approval required?</b></p>	<p>The screenshot shows the SAP Cloud Platform Workflow Management interface. A user is adding a new gateway. Step 1 highlights the 'Exclusive Gateway' option in the list. Step 2 highlights the gateway icon being added. Step 3 highlights the newly added gateway in the palette.</p>
<p>139. Select <b>Add Task</b> .          140. Select <b>User Task</b> .          141. Enter <b>Approval Task (Central Manager)</b> as <b>Name</b>.</p>	<p>The screenshot shows the SAP Cloud Platform Workflow Management interface. A user is selecting a task type. Step 1 highlights the 'User Task' icon in the list. Step 2 highlights the task icon being selected. Step 1 also points to the 'Approval required?' decision gateway.</p>

Explanation	Screenshot
<p>142. Select the <b>Details</b> tab.</p> <p>143. Enter <b>Approval for '\${context.Title}' in your role as Central Manager as Subject.</b></p> <p>144. Enter  <code> \${context.approvalStepsResult.Result[0].Approvers.cm_userid}, \${info.startedBy} as Recipients - Users.</code></p> <p>Note: During development, the user tasks will be sent to the process initiator. Later, the user Id will be retrieved from SAP Cloud Platform Business Rules.</p>	 <p>User Task Properties</p> <p>GENERAL DETAILS USER INTERFACE ATTRIBUTES</p> <p>* Priority Medium</p> <p>* Subject Approval for '\${context.Title}' in your role as Central Manager</p> <p>Description</p> <p>Recipients</p> <p>Users \${context.approvalStepsResult.Result[0].Approvers.cm_userid}, \${info.startedBy}</p> <p>Groups</p> <p>Due Date <input type="checkbox"/> Configure Due Date</p>

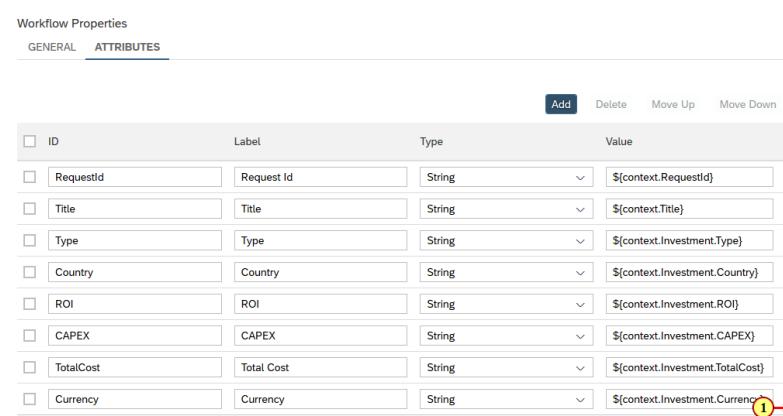
Explanation	Screenshot
<p>145. Select the <b>User Interface</b> tab.</p> <p>146. Select <b>Form</b> as <b>Type</b>.</p> <p>147. Use the <b>Select</b> button to choose the previously created form <b>capexapproval_00</b>.</p>	 <p>User Task Properties</p> <p>GENERAL DETAILS <b>USER INTERFACE</b> ATTRI &gt;</p> <p>*Type <b>Form</b></p> <p>Form Details</p> <p>*File Name CapexApproval_00.form</p> <p>Select</p> <p>*ID capexapproval_00</p> <p>*Revision 1</p>
<p>148. Select <b>Add Task</b> on the central manager approval.</p> <p>149. Select the <b>Script Task</b>.</p>	 <pre> graph LR     Start(( )) --&gt; Decision{Approval required?}     Decision -- No --&gt; End(( ))     Decision -- Yes --&gt; ApprovalTask[Approval Task (Central Manager)]     ApprovalTask --&gt; End     </pre> <p>Approval required?</p> <p>Approval Task (Central Manager)</p> <p>User Task</p> <p>Service Task</p> <p>Script Task</p> <p>Mail Task</p>

Explanation	Screenshot
<p>150. Enter <b>Handle Approval (Central Manager)</b> as <b>Name</b>.</p> <p>151. Select <b>Create File</b> to create a new file. Name it <b>HandleApprovalCentralManager</b> and paste the content of the file with the same name from the exercise folder. You can then close it again.</p> <p>Note: This script refers to the <b>Task ID</b> (<code>usertask3</code>) multiple times. This is the id of the User Task <b>Approval Task (Central Manager)</b> which was mentioned earlier. It is automatically assigned to each task. If your ID is different, then it needs to be replaced here.</p> <p>Note: To access the task decision the ID of the User Task is used. The difference in the two <b>Handle Approval</b> scripts are which task they refer to, and the 'Role' stored in the history element of workflow context.</p>	 <p>The screenshot shows the 'Script Task Properties' dialog. The 'Name' field is highlighted with a red box and contains the text 'Handle Approval (Central Manager)'. A yellow circle labeled '1' points to the 'Name' field. A yellow circle labeled '2' points to the 'Select' button next to the 'Script File' input field.</p>
<p>152. Select <b>Add Gateway</b> .</p> <p>153. Select the <b>Exclusive Gateway</b> .</p> <p>154. Name it as <b>Approval result?</b></p>	
<p>155. Click <b>Approval result?</b> and Select <b>Add Event</b>.</p> <p>156. Select the <b>Terminate End Event</b>. Name the event as <b>End (Approval)</b>.</p>	

Explanation	Screenshot
<p>157. On the previous exclusive gateway with name <b>Approval required?</b> add a new <b>Sequence Flow Connector</b> and drag it onto the newly added <b>Terminate End Event</b>.</p> <p>Note: To make it look nice, move the blue connection dots on both sides to the upper edge of the icons and optionally drag the line upwards further upwards.</p>	 <pre> graph LR     Start(( )) --&gt; Join(( ))     Join -- "Approval result?" --&gt; Reject(( ))     Join -- "join" --&gt; Gateway{Approval required?}     Gateway -- "yes" --&gt; Task1[Approval Task (Central Manager)]     Task1 --&gt; Task2[Handle Approval (Central Manager)]     Task2 --&gt; End(( ))     End -- "End (Approval)" --&gt; Terminate(( ))     </pre>
<p>158. Enter <b>no</b> as <b>Name</b>.</p> <p>159. Select it as <b>Default</b>.</p>	<p><b>Sequence Flow Properties</b></p> <p>ID: sequenceflow23</p> <p>* Name: no (1)</p> <p>Documentation:</p> <p>Default: checked (2)</p>
<p>160. Select the Sequence Flow Connector between <b>Approval required?</b> and <b>Approval Task (Central Manager)</b>.</p>	

Explanation	Screenshot
<p>161. Enter <b>Yes</b> as <b>Name</b>.</p> <p>162. Enter  <b><code>#{context.approvalStepsResult.Result[0].Approvers.cm_required}</code></b> as <b>Condition</b>.</p> <p>Note: Be careful during copying. There are no spaces in the condition.      Sometimes during copy, spaces are added automatically. In this case, then remove them.</p> <p>Note: how this accesses the field <i>cm_required</i>, for "approval required from Central Manager", whereas for the first approval, the field was called <i>lm_required</i>, for the <i>Local Manager</i>.</p>	 <p>The screenshot shows the 'Sequence Flow Properties' dialog. The 'Name' field is highlighted with a red box and contains the value 'yes'. The 'Condition' field is also highlighted with a red box and contains the value 'approvalStepsResult.Result[0].Approvers.cm_required'. A yellow circle labeled '1' points to the 'Name' field, and another yellow circle labeled '2' points to the 'Condition' field.</p>
<p>163. Create a new <b>Terminating End Event</b> and place it above the <b>Approval result?</b> gateway. Name it <b>End (Rejection)</b>.</p> <p>164. Drag a new Sequence Flow Connection from <b>Approval result?</b> onto the <b>End (Rejection)</b> event. Name it <b>Reject</b> and select <b>Default</b> as condition.</p>	 <p>The workflow diagram illustrates the process flow. It starts with an 'Approval required?' gateway. If 'yes', it leads to an 'Approval Task (Central Manager)' task. Following this is a 'Handle Approval (Central Manager)' task. The next step is an 'Approval result?' gateway. If 'no', the flow continues. If 'yes', it leads to the 'End (Rejection)' event, which is highlighted with a red box and circled by a yellow circle labeled '1'. From the 'End (Rejection)' event, a sequence flow labeled 'Reject' (highlighted with a red box) connects back to the 'Approval result?' gateway. If 'no', the flow continues to the 'End (Approval)' event, which is highlighted with a black box and circled by a yellow circle labeled '2'.</p>
<p>165. Select the Sequence Flow Connection from <b>Approval result?</b> onto the <b>End (Approval)</b></p> <p>166. Name it <b>Approve</b> and enter <b><code>#{context.decision == "approve"}</code></b> as condition.</p>	 <p>The screenshot shows the 'Sequence Flow Properties' dialog for the 'Approve' connection. The 'Name' field is highlighted with a red box and contains the value 'Approve'. The 'Condition' field is also highlighted with a red box and contains the value ' #{context.decision == "approve"}'. A yellow circle labeled '2' points to the 'Name' field, and another yellow circle labeled '3' points to the 'Condition' field.</p>

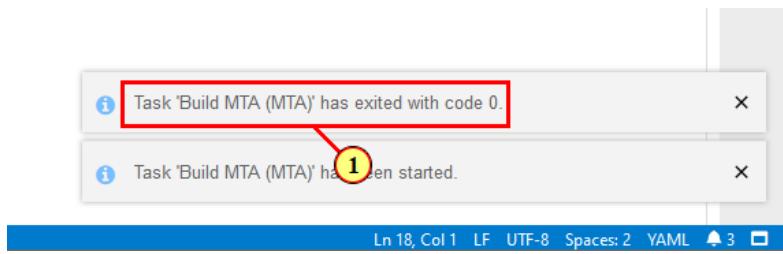
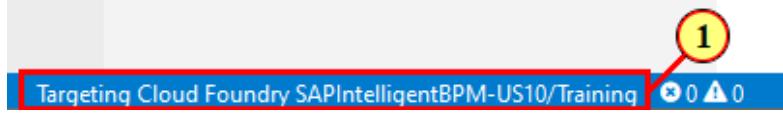
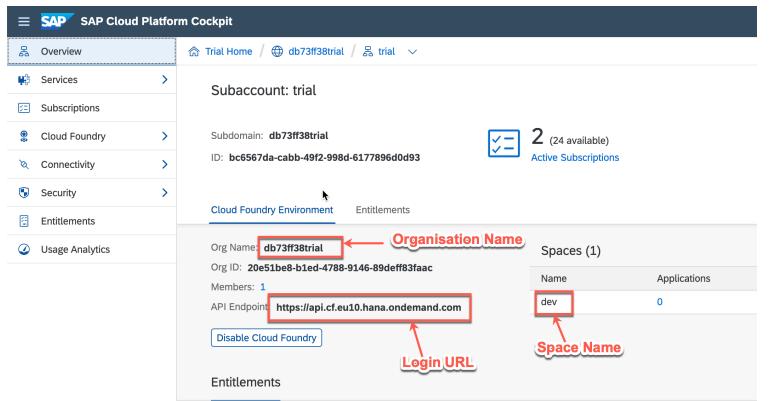
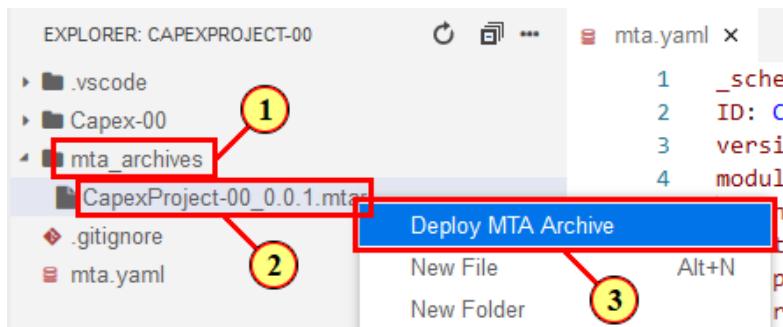
Explanation	Screenshot
<p>167. Drag a new Sequence Flow Connection from <b>Approval result?</b> onto the <b>Rework Task</b>.</p> <p>168. Adjust it that it starts directly downward by selecting the starting Connection Dot and move it to the lower edge of the Exclusive Gateway.</p>	<p>The screenshot shows a BPMN diagram with a sequence flow from an exclusive gateway labeled "Approval required?" to a task labeled "Handle Approval (Local Manager)". From this task, two parallel paths emerge: one leading to an "End (Rejection)" node and another leading to a decision diamond "Approve". From "Approve", the flow continues through another decision diamond "Approval result?" to a "join" node, which then leads to a "Rework" task. A new sequence flow is being drawn from the "Approval result?" decision diamond to the "Rework" task. The "Rework" task is highlighted with a red box and a yellow circle labeled 1.</p>
<p>169. Name it as <b>Rework</b> and enter <b>#{context.decision == "rework"}</b> as condition.</p>	<p>The screenshot shows the BPMN diagram with the newly named sequence flow "Rework" highlighted with a red box and a yellow circle labeled 2. The condition for this sequence flow is set to "#{context.decision == "rework"}", indicated by a red box and a yellow circle labeled 3.</p>
<p>170. Click anywhere on an empty space in the workflow editor to open <b>Workflow Properties</b>.</p> <p>171. In <b>Workflow Properties</b>, enter <b>Subject</b> as <b>CAPEX Approval for '#{context.Title}'</b></p>	<p>The screenshot shows the "Workflow Properties" pane with the "GENERAL" tab selected. The "Subject" field is highlighted with a red box and a yellow circle labeled 2, containing the value "CAPEX Approval for '#{context.Title}'". The "Business Key" field contains "\${context.RequestId}" and the "Documentation" field contains "Capital Expenditure Approval". A vertical divider line is being moved with a red box and a yellow circle labeled 1.</p>
<p>172. Enter <b>Business Key</b> as <b>#{context.RequestId}</b></p> <p>173. Select the <b>ATTRIBUTES</b> tab.</p> <p>Note: You can drag the vertical divider line of the <b>Workflow Properties</b> Pane to enlarge it.</p>	<p>The screenshot shows the "Workflow Properties" pane with the "ATTRIBUTES" tab selected. The "Business Key" field is highlighted with a red box and a yellow circle labeled 3, containing the value "\${context.RequestId}". The "GENERAL" tab is also highlighted with a red box and a yellow circle labeled 4.</p>

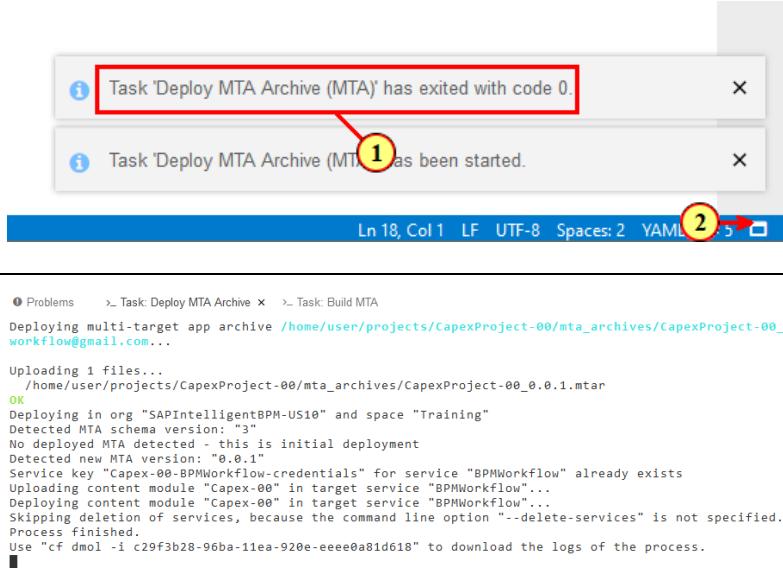
Explanation	Screenshot
<p>174. Select <b>Add</b>.</p> <p>175. Enter <b>Currency</b> as <b>ID</b>.</p> <p>176. Enter <b>Currency</b> as <b>Label</b>.</p> <p>177. Enter  <b>Value:</b> \${context.Investment.Currency} as  <b>Value.</b></p>	
<p>178. In the same way, add the following fields:</p> <p><b>ID:</b> TotalCost  <b>Label:</b> Total Cost  <b>Value:</b> \${context.Investment.TotalCost}</p> <p><b>ID:</b> CAPEX  <b>Label:</b> CAPEX  <b>Value:</b> \${context.Investment.CAPEX}</p> <p><b>ID:</b> ROI  <b>Label:</b> ROI  <b>Value:</b> \${context.Investment.ROI}</p> <p><b>ID:</b> Country  <b>Label:</b> Country  <b>Value:</b> \${context.Investment.Country}</p> <p><b>ID:</b> Type  <b>Label:</b> Type  <b>Value:</b> \${context.Investment.Type}</p> <p><b>ID:</b> Title  <b>Label:</b> Title  <b>Value:</b> \${context.Title}</p> <p><b>ID:</b> RequestId  <b>Label:</b> Request Id  <b>Value:</b> \${context.RequestId}</p> <p>Note: Currently only String type is supported.</p>	

Explanation	Screenshot
<p>Your workflow should finally look like this.</p>	<pre> graph LR     Start((Capex Request)) --&gt; Prep[Prepare Data]     Prep --&gt; Retrieve[Retrieve Approval Steps]     Retrieve --&gt; ApprovalReq{Approval required?}     ApprovalReq -- yes --&gt; LocalManager[Approval Task (Local Manager)]     LocalManager --&gt; LocalResult{Approval result?}     LocalResult -- no --&gt; EndRejection1((End (Rejection)))     LocalResult -- yes --&gt; Rework1[Rework Task]     ApprovalReq -- no --&gt; CentralManager[Approval Task (Central Manager)]     CentralManager --&gt; CentralResult{Approval result?}     CentralResult -- no --&gt; EndRejection2((End (Rejection)))     CentralResult -- yes --&gt; Rework2[Rework Task]     LocalResult -- join --&gt; EndApproval((End (Approval)))     CentralResult -- join --&gt; EndApproval     </pre>
<p>Note: If you did not use <b>Auto Save</b>, ensure that all files are saved.</p>	

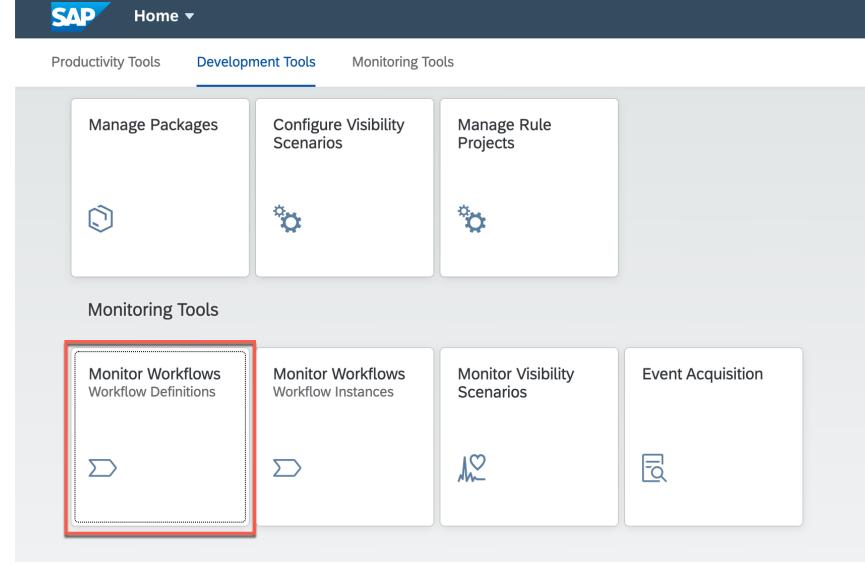
**BUILD AND DEPLOY WORKFLOW**

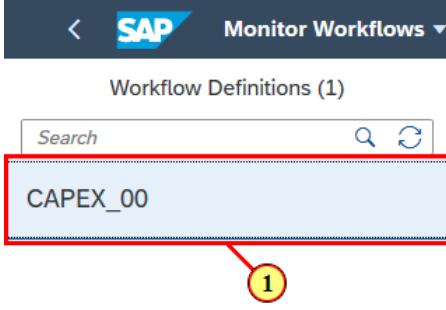
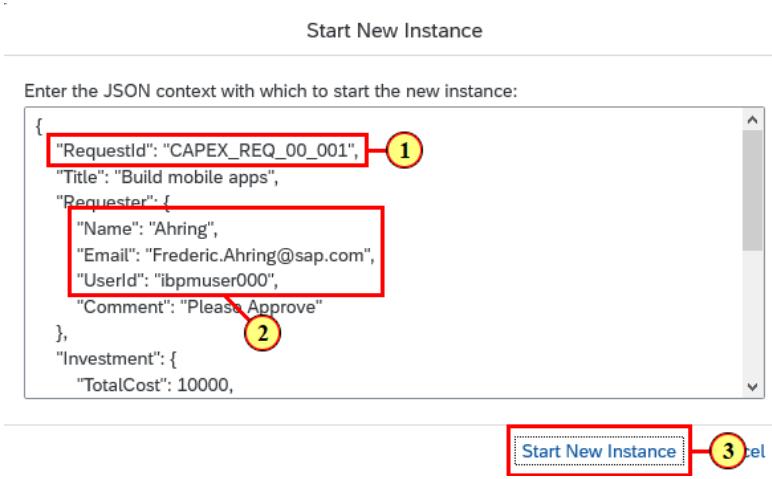
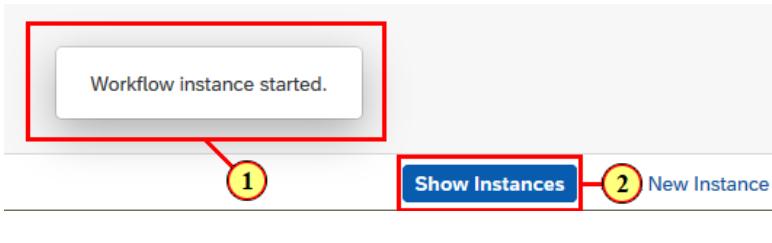
Explanation	Screenshot
179. Double-click the file <b>mta.yaml</b> to open it.	
<p>Note: The default MTA will create a reference to a new workflow service instance. When you ran the Booster in the setup the workflow service instance was already created so change the resource to the existing instance name.</p> <p>When you ran the Booster in the setup the workflow service instance was already created so change the resource to the existing instance name.</p>	<pre> File Edit Selection View Go Debug Terminal Help SAP Business Application Studio EXPLORER mta.yaml 1 _schema-version: "3.2" 2 ID: CapexProject 3 version: 0.0.1 4 modules: 5   - name: Capex 6     type: com.sap.application.content 7     path: Capex 8   requires: 9     - name: wm_workflow 10    parameters: 11      content-target: true 12 resources: 13   - name: wm_workflow 14     parameters: 15       service-plan: standard 16       service: workflow 17     type: org.cloudfoundry(existing-service) 18   </pre>
180. Change the resource name to <b>wm_workflow</b> .	
181. Change the resource type to <b>org.cloudfoundry(existing-service)</b> .	
182. Change the resource reference to <b>wm_workflow</b> .	
183. Close the tab and ensure the file is saved.	
184. To build the project, right-click on <b>mta.yaml</b> to open the context menu.	
185. Select <b>Build MTA</b> .	

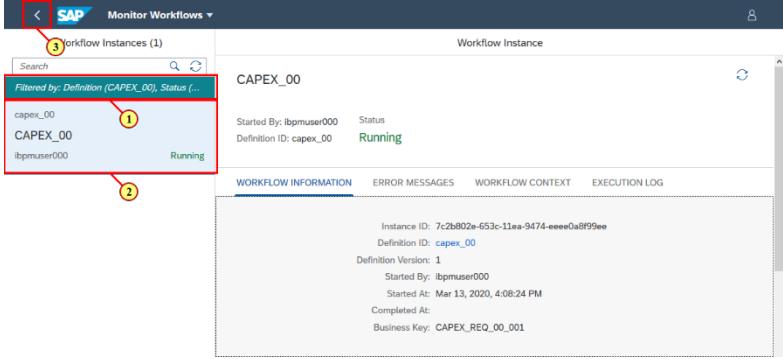
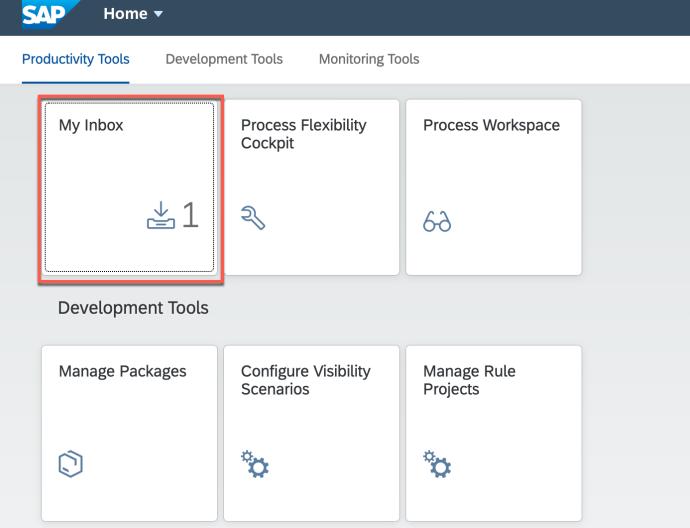
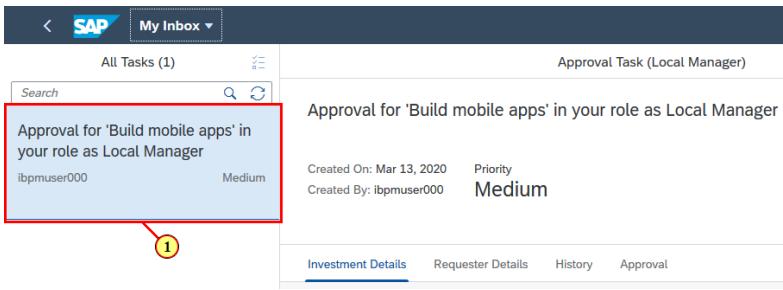
Explanation	Screenshot
<p>186. Ensure that the <b>Build MTA</b> task exits with code 0, indicating the build is completed.</p>	
<p>187. Ensure that a cloud tenant has already been configured. It will be shown in the bottom left corner on a blue bar. If it is not configured yet, then click on the blue bar and click on the blue-bar and follow the wizard to enter the details.</p>	
<p>You can get the need information about the Login URL, organization name and space name from your trial account.  You have to login with your trial user email and password.</p>	
<p>188. Select the newly created directory <b>mta_archives</b> to open it.  189. Right-click on the created file <b>CapexProject-00_0.0.1.mtar</b> to open the context menu.  190. Select <b>Deploy MTA Archive</b>.</p>	

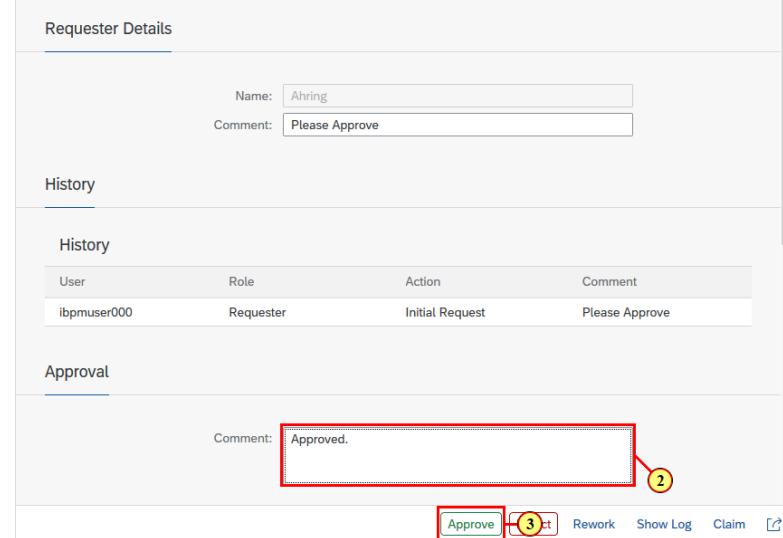
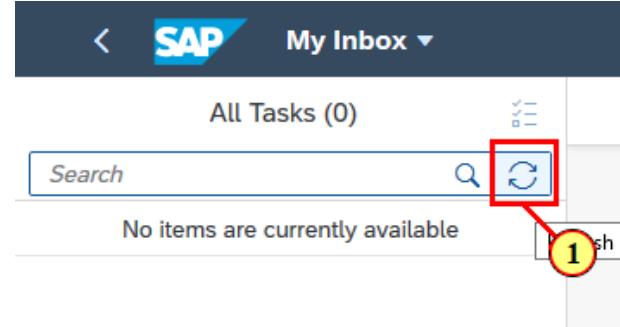
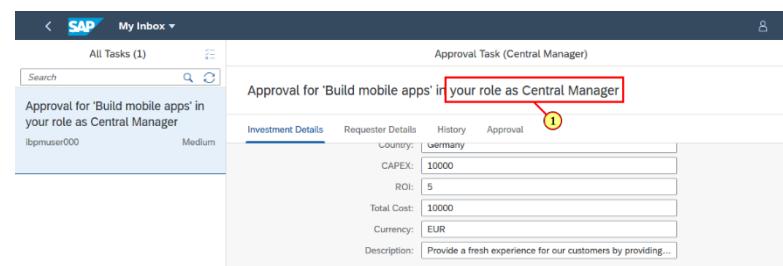
Explanation	Screenshot
<p>191. Ensure that the <b>Deploy MTA Archive</b> task exits with code 0 indicating that the deployment has finished.</p> <p>Note: You can also click on the  icon in the lower right corner to toggle display of the console panels.</p> <p>Note: Your deployment console should look similar to this with the "<b>Process finished</b>" in the second-to-last line, indicating success.</p> <p>If there are any errors then fix the error, build and deploy the workflow project again.</p>	 <p>The screenshot shows a deployment console window with two messages:</p> <ul style="list-style-type: none"> <li>Task 'Deploy MTA Archive (MTA)' has exited with code 0.</li> <li>Task 'Deploy MTA Archive (MTA)' has been started.</li> </ul> <p>Below the messages, the deployment log is displayed:</p> <pre>Problems   ... Task: Deploy MTA Archive  ... Task: Build MTA Deploying multi-target app archive /home/user/projects/CapexProject-00/mta_archives/CapexProject-00_workflow@gmail.com... Uploading 1 files... OK Deploying in org "SAPIntelligentBPM-US10" and space "Training" Detected MTA schema version: "3" No deployed MTA detected - this is initial deployment Detected new MTA version: "0.0.1" Service key "Capex-00-BPMWorkflow-credentials" for service "BPMWorkflow" already exists Uploading content module "Capex-00" in target service "BPMWorkflow"... Deploying content module "Capex-00" in target service "BPMWorkflow"... Skipping deletion of services, because the command line option "--delete-services" is not specified. Process finished. Use "cf logs -i c29f3b28-96ba-11ea-eeee0a81d618" to download the logs of the process.</pre>

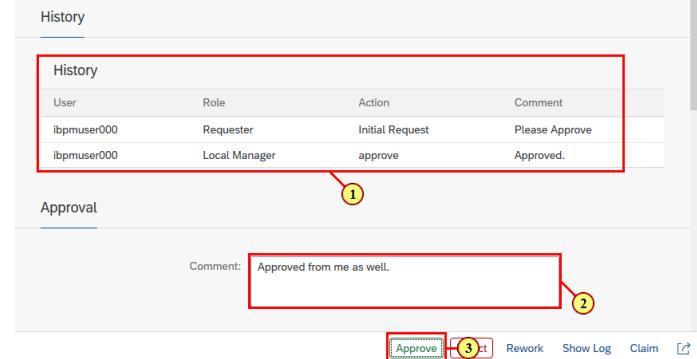
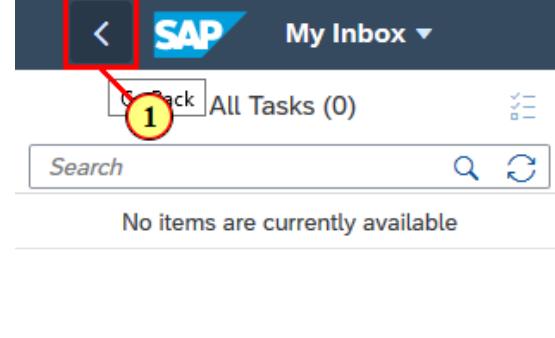
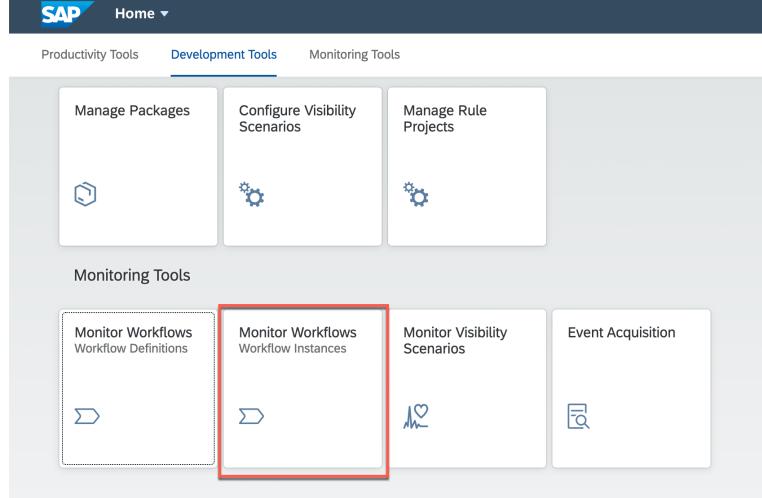
## EXECUTE AND MONITOR WORKFLOW

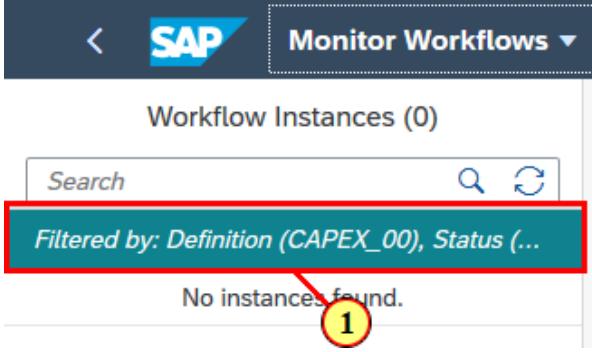
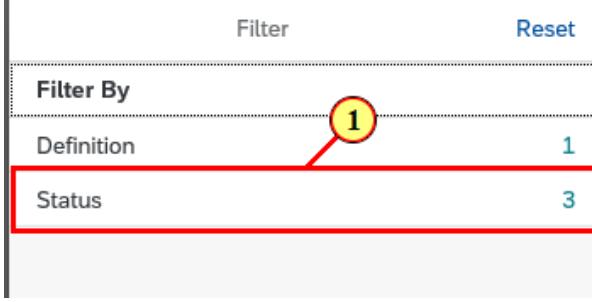
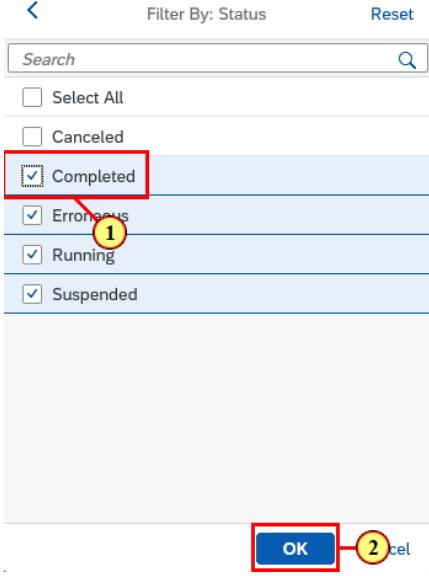
Explanation	Screenshot
<ol style="list-style-type: none"> <li>Open the Workflow Management Launchpad</li> <li><b>Log On</b> with your trial username and password.</li> <li>Select the tile <b>Monitor Workflows - Workflow Definitions</b>.</li> </ol>	 <p>The screenshot shows the SAP Cloud Platform Workflow Management Launchpad. The navigation bar at the top includes links for Productivity Tools, Development Tools (which is currently selected), and Monitoring Tools. Below the navigation bar, there are several tiles:</p> <ul style="list-style-type: none"> <li>Manage Packages</li> <li>Configure Visibility Scenarios</li> <li>Manage Rule Projects</li> <li><b>Monitor Workflows Workflow Definitions</b> (this tile is highlighted with a red border)</li> <li>Monitor Workflows Workflow Instances</li> <li>Monitor Visibility Scenarios</li> <li>Event Acquisition</li> </ul>

Explanation	Screenshot
<p>Note: <b>Monitor Workflows</b> application has Master-Detail layout where you see all the workflows deployed on left. Ensure that the workflow you just deployed - <b>CAPEX_00</b> - is existing.</p> <p>4. Select the workflow.</p>	
<p>5. Select <b>Start New Instance</b>.</p> <p>Note: This way of starting a workflow is mainly used for testing. Workflows can be started by end users using a Start Form or a custom UI5 application, and by other systems using the <a href="#">Workflow API</a>.</p>	
<p>6. Modify the <b>RequestId</b> in the JSON data to be unique: use CAPEX_REQ_00_YYY where YYY is a different number each time you start a workflow, e.g. counting up from 001.</p> <p>7. Also adapt the following fields in the JSON element <b>Requester</b>:</p> <ul style="list-style-type: none"> <li>○ <b>Name</b> can be anything</li> <li>○ <b>Email</b> should be your trial user email ID</li> <li>○ <b>UserId</b> is your trial logon user id</li> </ul> <p>8. Select <b>Start New Instance</b>.</p>	
<p>A popup confirms that the workflow instance has been started.</p> <p>9. Select <b>Show Instances</b> to switch to the <b>Monitor Workflows - Workflow</b></p>	

Explanation	Screenshot
<p><b>Instances</b> application where you can view all the running, completed, suspended workflow instances and their details.</p>	
<p>Note: When coming from <b>Show Instances</b>, a <b>Filter</b> is automatically added to the specific <b>Workflow Definition</b>.</p> <p>10. Select the newly started workflow instance and look at its details.</p> <p>11. Select <b>SAP icon</b> to go back to the Home Page of the Workflow Management Launchpad.</p>	
<p>12. Select the tile for <b>My Inbox</b>. It should show at least one new task.</p>	
<p>13. Ensure that in the list view the new task "<b>Approval for...</b>" is selected.</p>	

Explanation	Screenshot
<p>14. Scroll down to the <b>Approval</b> section.</p> <p>15. Enter a <b>Comment</b> of your choice.</p> <p>16. Click <b>Approve</b>.</p>	
<p>17. In this exercise, the user task in the workflow has been configured to send all tasks in your inbox. To see the second approval task, select <b>Refresh</b> .</p>	
<p>Note: You can now see the task for the Central Manager.</p>	

Explanation	Screenshot
<p>Note: The <b>History</b> table is filled with the comment entered in the previous task.</p> <p>18. Enter again a <b>Comment</b> of your choice.</p> <p>19. Click <b>Approve</b>.</p>	
<p>20. Select <b>SAP icon</b> to go back to the Workflow Management Home Page.</p>	
<p>21. Select the <b>Monitor Workflows - Workflow Instances</b> tile.</p>	

Explanation	Screenshot
<p>22. Be default, <b>Monitor Instances</b> application only shows Running, Suspended and Erroneous workflow instance. To see the completed workflow, click on the line showing the currently active <b>Filters</b> or select the <b>Filter</b> icon at the bottom of the screen.</p>	
<p>23. In the popup, select <b>Status</b>.</p>	
<p>24. Select <b>Completed</b> to also show already completed workflow instances.      25. Select <b>OK</b> to activate the changed filter settings.</p>	

Explanation	Screenshot
<p>26. Ensure the correct instance is selected.</p> <p>27. Look at the details of the completed workflow instance.</p>	<p>The screenshot shows the SAP Monitor Workflows interface. On the left, a list of workflow instances is displayed, filtered by definition (CAPEX_00) and status (Completed). One instance, 'CAPEX_00' started by 'ibpmuser000', is highlighted with a red box and a yellow circle with the number 1. On the right, a detailed view of this instance is shown under the 'Workflow Instance' tab. The status is listed as 'Completed' with a red box and a yellow circle with the number 2. Below the status, the instance ID, definition ID, and version information are provided. Under the 'WORKFLOW CONTEXT' tab, a JSON snippet of the workflow context is shown:</p> <pre> 1: { 2:   "Request": { 3:     "Comment": "Please Approve", 4:     "RequestDate": "2020-03-13", 5:     "Title": "Build mobile apps", 6:     "UserId": "ibpmuser000", 7:     "Name": "Ahrim" 8:   }, 9:   "decision": "approve" 10:  "RequestId": "CAPEX_REQ_00_001", 11:  "Status": "Completed", 12:  "Title": "Build mobile apps", 13:  "Comment": "", 14:  "WorkflowContextResult": { 15:    "Result": [ 16:      { 17:        "Approver": { 18:          "cm_required": true 19:        } 20:      } 21:    ] 22:  } 23: } </pre>
<p><b>Congratulations! You completed the Workflow exercise.</b></p> <p>In this exercise, you learnt how to:</p> <ul style="list-style-type: none"> <li>○ Model workflow from <b>SAP Business Application Studio</b>.</li> <li>○ Build &amp; deploy workflow</li> <li>○ Start workflow instance from <b>Monitor Definitions</b></li> <li>○ Track the workflow instances from <b>Monitor Instances</b></li> <li>○ Process your work items from <b>My Inbox</b></li> </ul>	

## SUMMARY

You have completed the exercise!

You are now able to:

1. Create a new workflow in **Business Application Studio**.
2. Add & configure **service task**, **script task**, **user task** and **gateways** in the workflow model.
3. **Build** and **deploy** workflow from Business Application Studio
4. Test the workflow using **Monitor Workflow | Workflow Definition** application.
5. Track and monitor the workflow using **Monitor Workflow | Workflow Instances** application.
6. View user tasks and take action from **My Inbox** application.

[www.sap.com/contactsap](http://www.sap.com/contactsap)

© 2020 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. See [www.sap.com/copyright](http://www.sap.com/copyright) for additional trademark information and notices.