

PUBLIC

Build a Workflow from Scratch with SAP Cloud Platform Workflow Management

DEV163

Exercise 1 | Workflow
Archana Shukla / SAP

TABLE OF CONTENTS

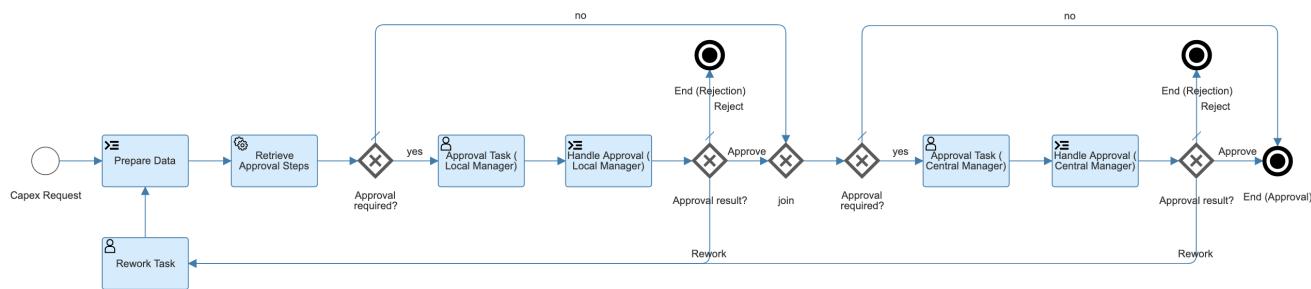
OVERVIEW	3
MODEL WORKFLOW USING BUSINESS APPLICATION STUDIO.....	4
BUILD AND DEPLOY WORKFLOW	45
EXECUTE AND MONITOR WORKFLOW.....	47
SUMMARY	53

OVERVIEW

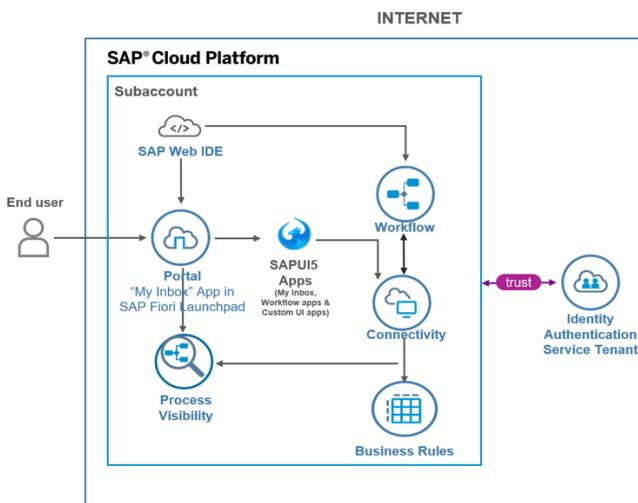
Estimated time: 90 minutes

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.

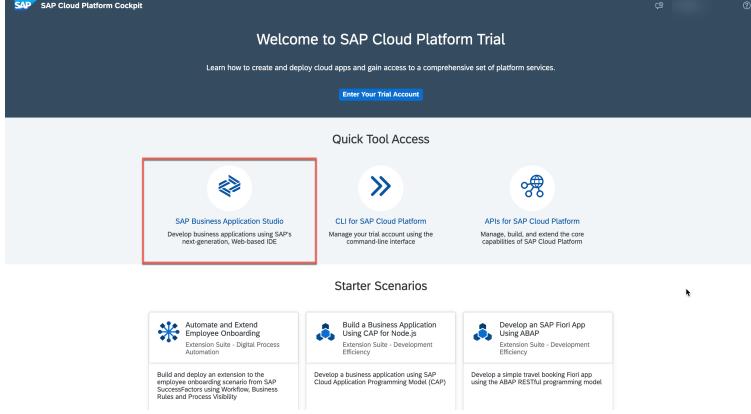
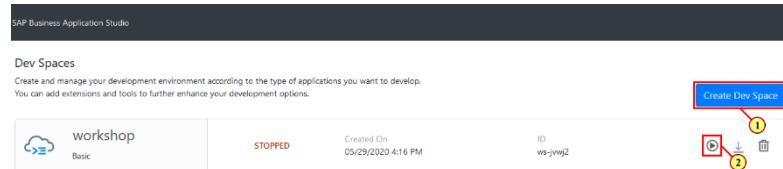
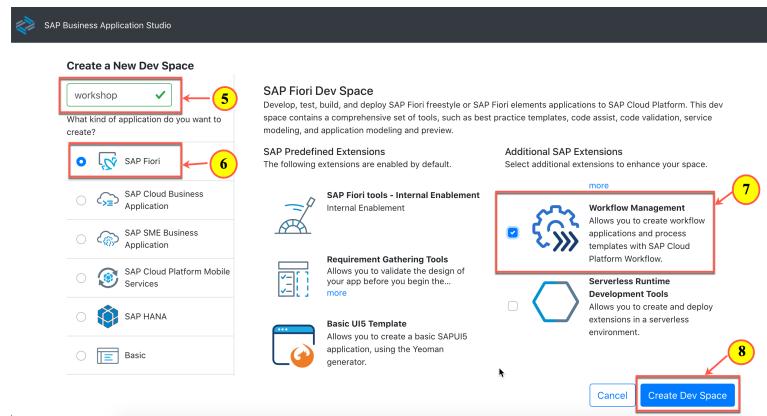
There are many challenges with capital expenditure process and companies are always looking for managing investment decisions in a transparent way. Automating capital expenditure approval process enables customer to accelerate investment decisions and to bring the agility into every step and decision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

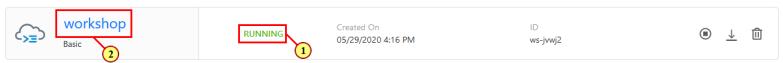
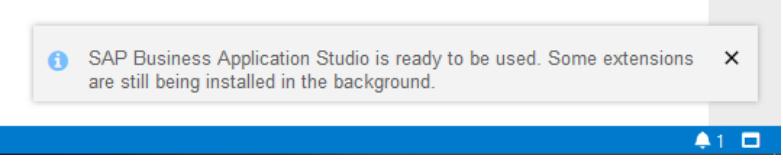
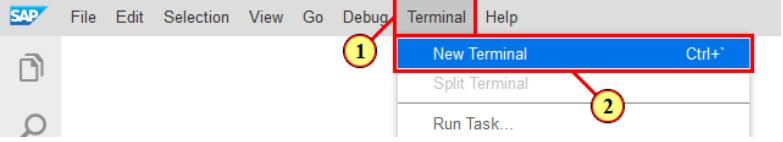
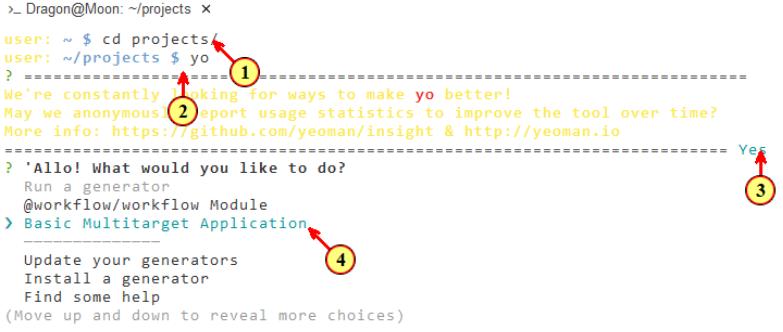
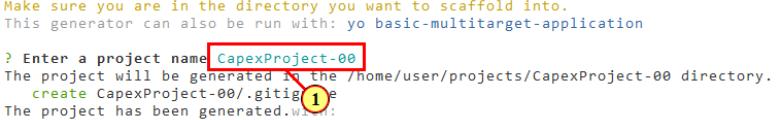


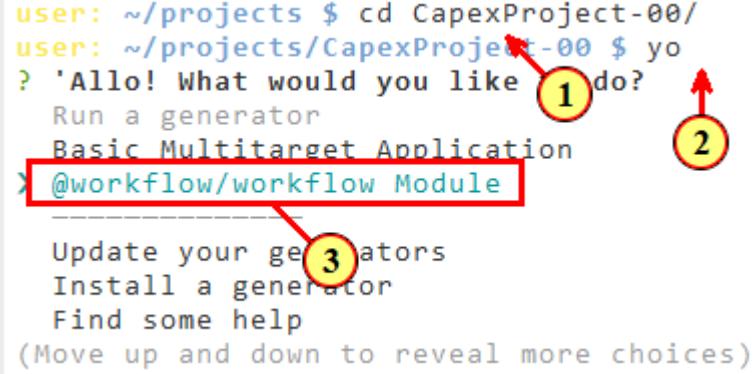
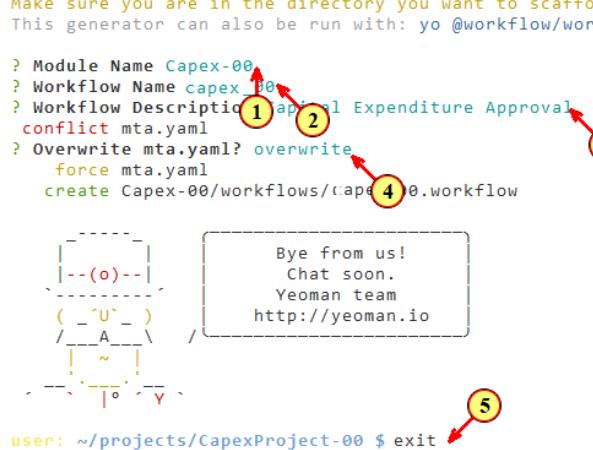
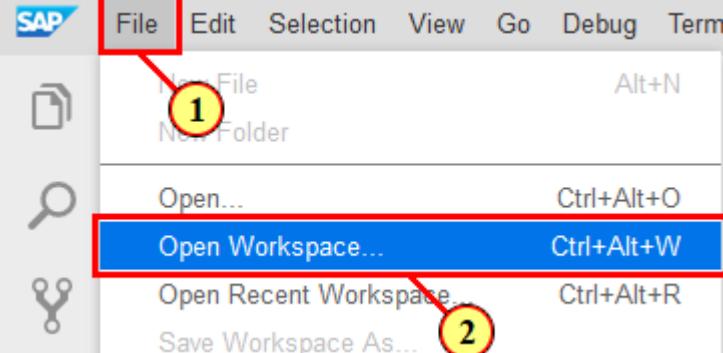
The below solution diagram explains how customer can automate CAPEX scenario using SAP Cloud Platform.

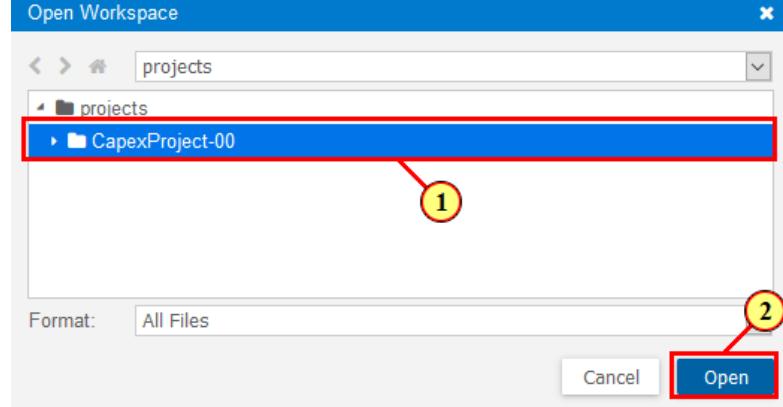
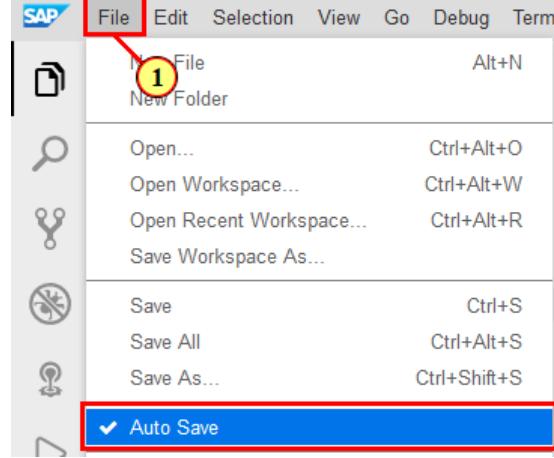
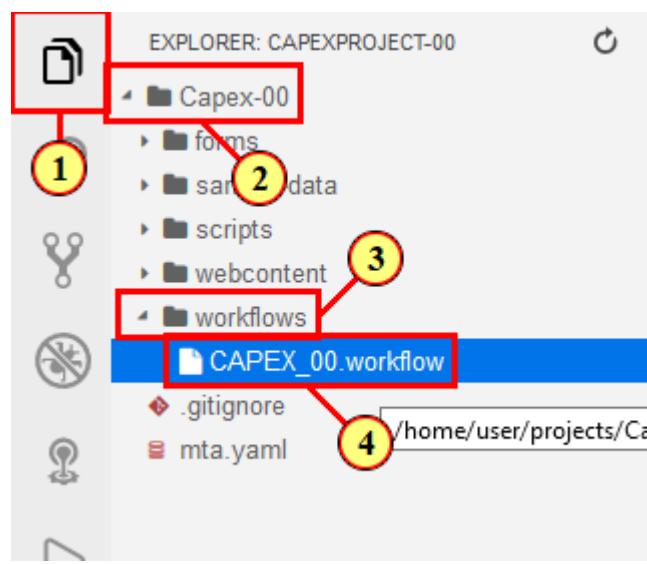


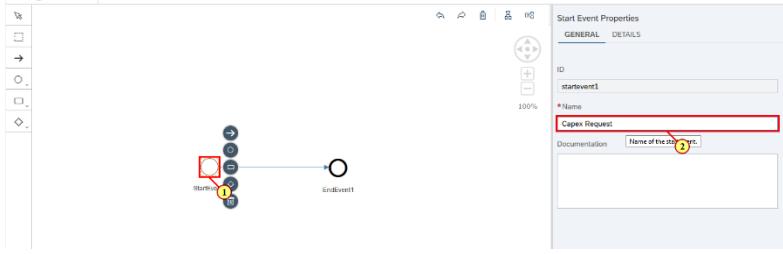
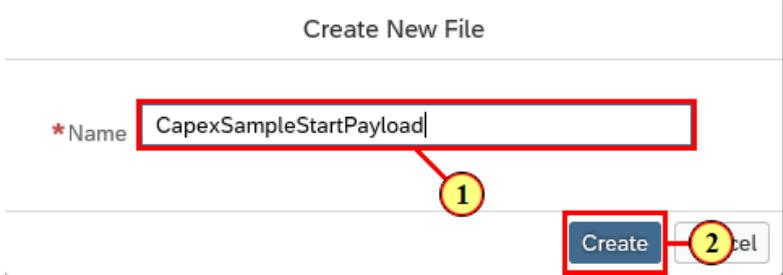
MODEL WORKFLOW USING BUSINESS APPLICATION STUDIO

Explanation	Screenshot
<ol style="list-style-type: none"> Logon to SAP Cloud Platform Trial Home Page with your trial username and password. https://cockpit.hanatrial.ondemand.com/cockpit/#/home/trial Choose SAP Business Application Studio. 	
<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"> If you have no dev space, click the Create Dev Space and follow instructions in step 5. Else select PLAY to start the space if you see the status as STOPPED. <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"> Enter workshop as the new Dev Space name. Ensure Application Type is selected as SAP Fiori. Select the extension Workflow Management. Note: This extension will enable the workflow modelling features. Select Create Dev Space. 	

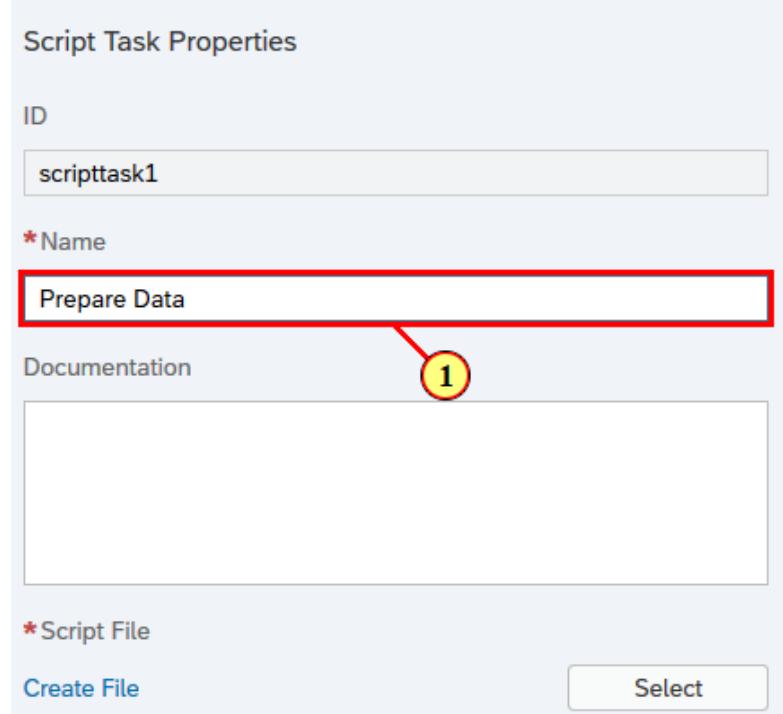
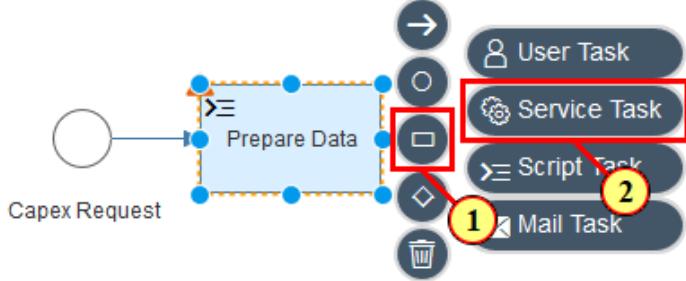
Explanation	Screenshot
<p>9. Wait until the virtual machine providing this dev space is in status RUNNING.</p> <p>10. Open the SAP Business Application Studio by clicking on the dev space name.</p>	
<p>Note: SAP Business Application Studio is a web application that loads the features of the extensions. There might be a short delay for all features to be operational when first accessed.</p>	
<p>11. From the menu, select Terminal.</p> <p>12. Select New Terminal to open a Linux Terminal window.</p>	
<p>13. Enter cd projects/ in the Terminal 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 yo 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 Basic Multitarget Application by using the up/down arrow keys and press Enter.</p>	
<p>17. Enter CapexProject-00 or any name of your choice as Project name.</p>	

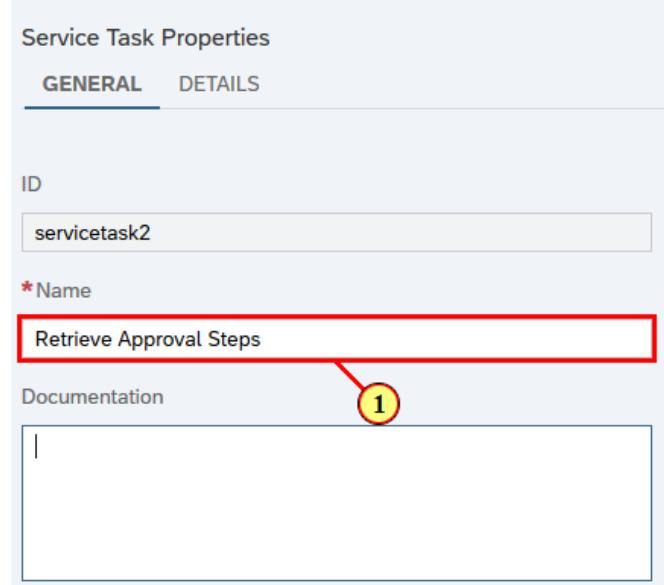
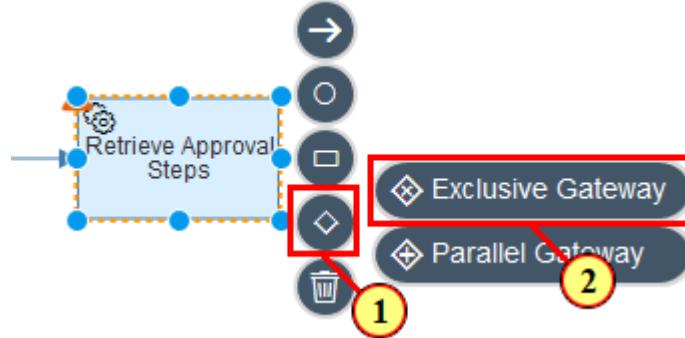
Explanation	Screenshot
<p>18. Change into the newly created directory by entering cd CapexProject-00. You can also use TAB for auto-completion, i.e. entering cd Cap<TAB> will be sufficient.</p> <p>19. Enter yo in the window to start Yeoman again.</p> <p>20. Select Workflow Module by using the up/down arrow keys and press Enter.</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 Capex-00 as Module Name.</p> <p>22. Enter Capex_00 as Workflow Name.</p> <p>23. Enter Capital Expenditure Approval as Workflow Description.</p> <p>24. Yeoman will warn you that there already is an mta.yaml file. Type Y to overwrite it.</p> <p>25. You can close the Terminal by typing exit 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 File.</p> <p>27. Select Open Workspace....</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>

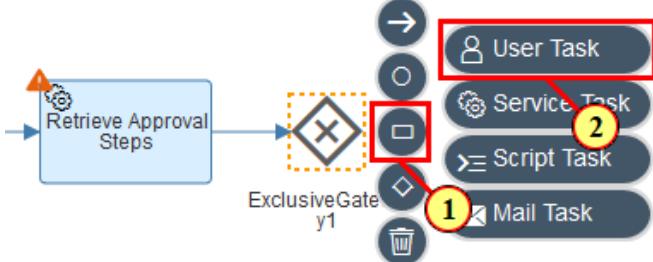
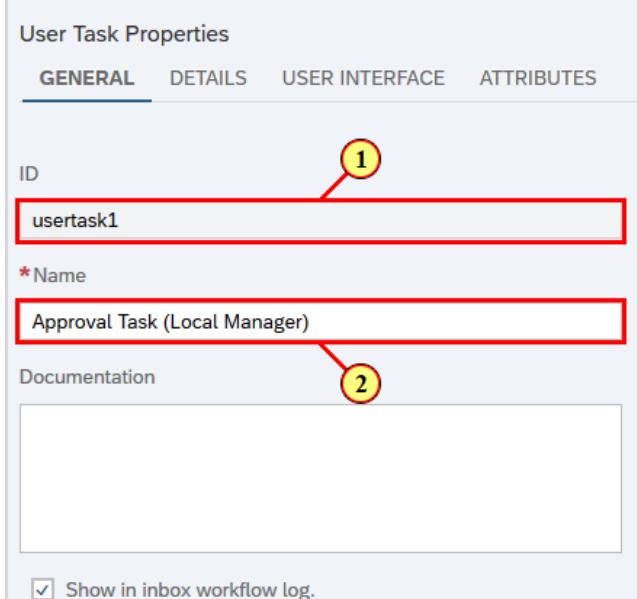
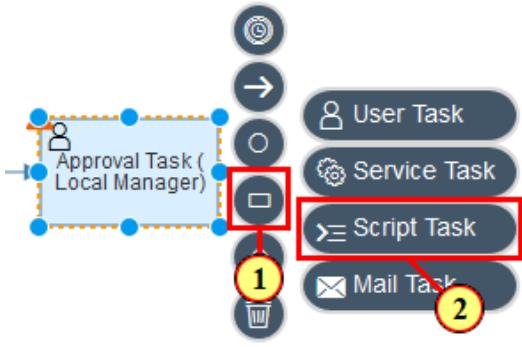
Explanation	Screenshot
<p>28. Select the directory containing your module CapexProject-00.</p> <p>29. Select Open.</p>	
<p>30. From the menu, select File. Note that there is an Auto Save functionality (enabled by default). If you choose to disable it, remember to manually save your files.</p>	
<p>31. If it is not already open, select the Explorer  from the sidebar.</p> <p>32. Open the Capex-00 module by selecting it.</p> <p>33. Open the workflows directory by selecting it.</p> <p>34. Open the CAPEX_00.workflow 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>	

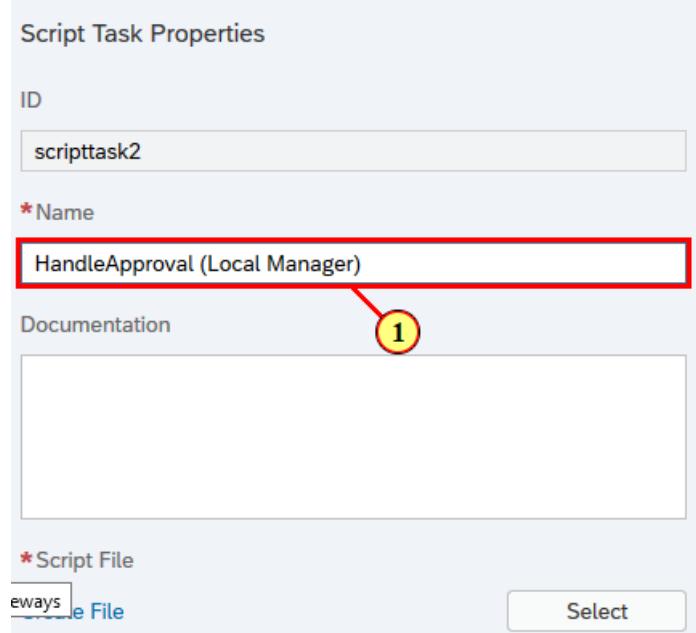
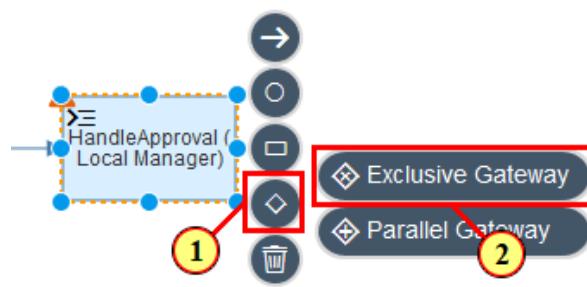
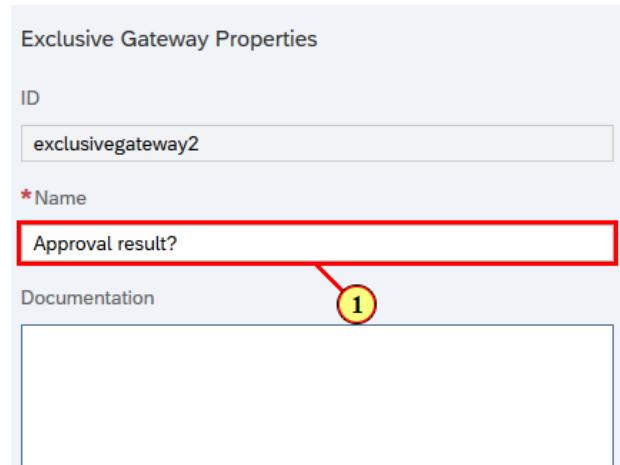
Explanation	Screenshot
<p>A new workflow with a default Start and End event is created.</p> <ul style="list-style-type: none"> ○ On the left side there is a toolbar for adding new flow elements. ○ On the right side is the properties pane for the currently selected element. <p>35. Click on the Start Event  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 Capex Request 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 Details of the start event.</p> <p>38. Select Configure Sample Context. 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 Create File.</p>	
<p>1. In the popup, enter CapexSampleStartPayload as File Name.</p> <p>2. Click Create.</p>	

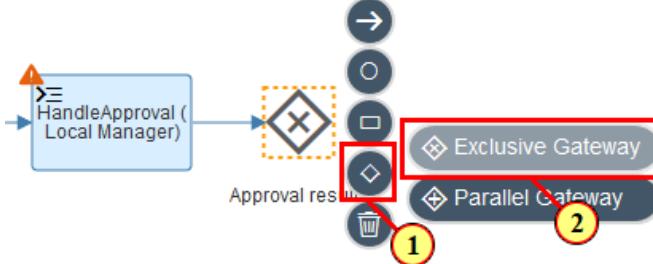
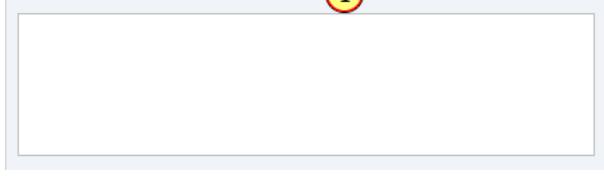
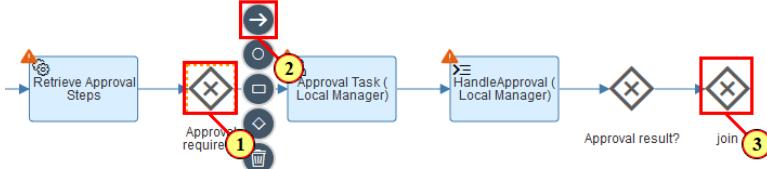
Explanation	Screenshot
<p>A new file with the name CapexSampleStartPayload.json is created in the project folder below sample-data. It is created with some default data as example.</p>	 <pre>* CAPEX_00.workflow x CapexSampleStartPayload.json x 1 { 2 "product": "Hamlet (Paperback)", 3 "inStock": true, 4 "inventory": 20000, 5 "price": 7.49, 6 "publishingDate": "1600-04-23T18:25:43.511Z", 7 "author": { "name": "William Shakespeare" }, 8 "publishers": ["Simon & Brown", "SparkNotes", "Dover Publications"] 9 }</pre>
<ol style="list-style-type: none"> 3. Remove the example content and Copy-Paste the example context given in the file CapexSampleStartPayload.json into the editor. 4. Set the RequestId to CAPEX_REQ_XX_001. Replace XX with the number assigned to you by the trainer. 5. Insert your name, email address and user id into the respective fields. Your UserId is your trial user email ID. 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. 	 <pre>* CAPEX_00.workflow x CapexSampleStartPayload.json x 1 { 2 "RequestId": "CAPEX_REQ_00_001", 1 3 "title": "Build mobile apps", 4 "Requester": { 5 "Name": "<your name>", 6 "Email": "<your email address>", 7 "UserId": "<your userId>", 2 8 "Comment": "Please Approve" 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"> 7. Click the Start event  to select it. 8. Click on Add Task . 9. Select the Script Task . 	

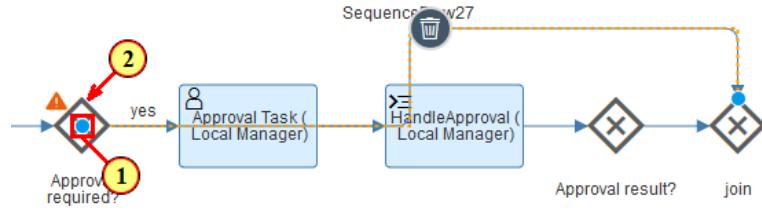
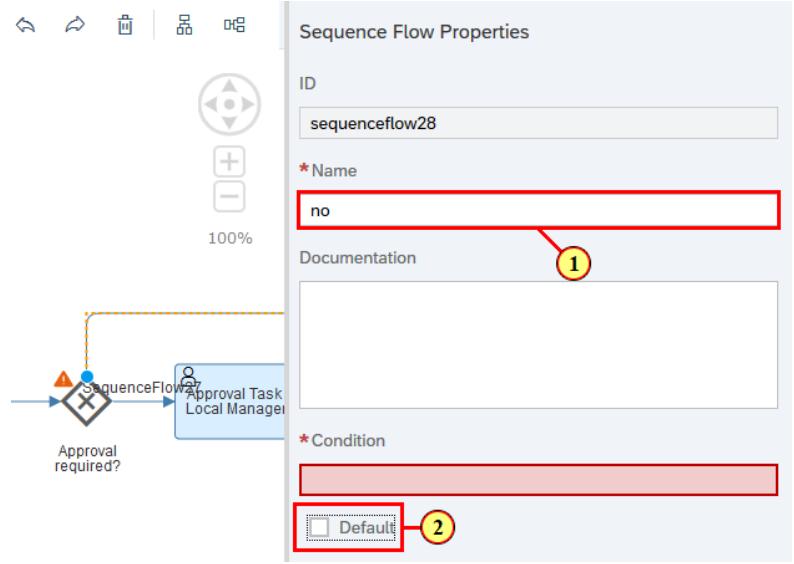
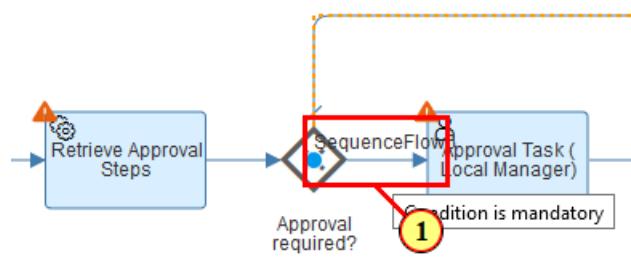
Explanation	Screenshot
<p>10. In the Script Task Properties, enter Prepare Data as Task name.</p> <p>Note: We will configure the task later.</p>	 <p>The screenshot shows the 'Script Task Properties' dialog box. The 'Name' field contains 'scripttask1'. The 'Task name' field is highlighted with a red box and contains 'Prepare Data'. A yellow circle with the number '1' points to the 'Task name' field.</p>
<p>11. On the selected Script Task 'Prepare Data', click Add Task  in the speed buttons.</p> <p>12. Select Service Task .</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 '2'.</p>

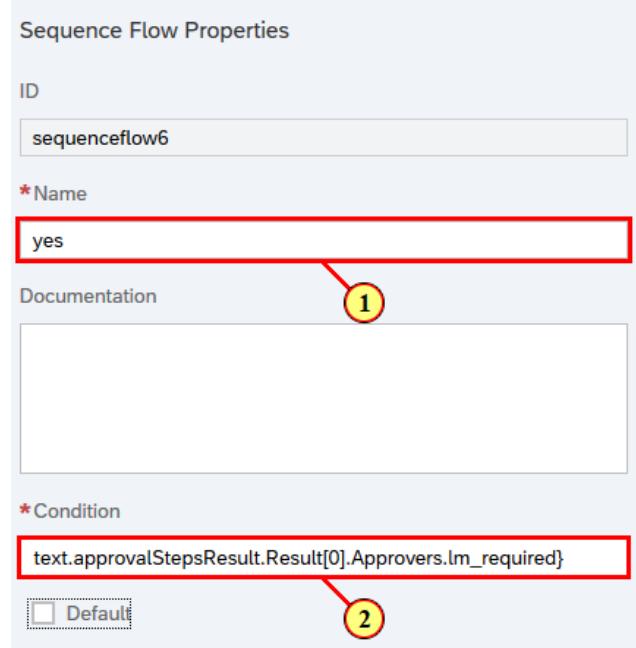
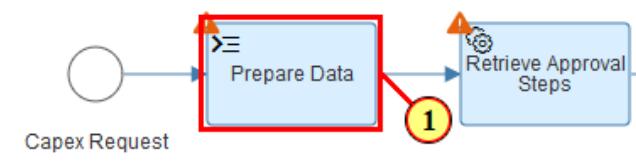
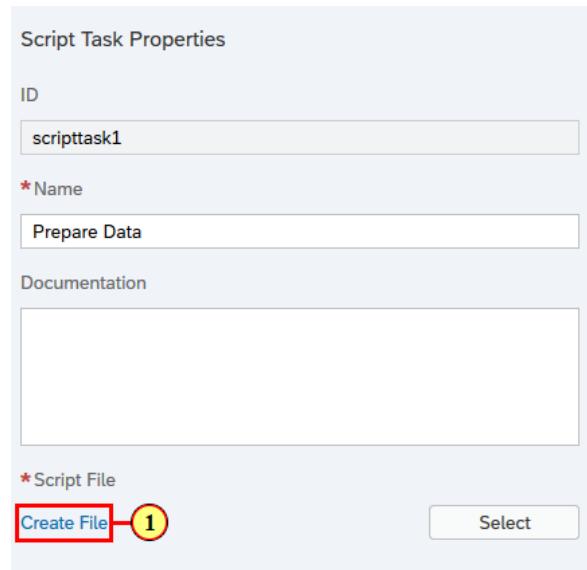
Explanation	Screenshot
<p>13. Enter Retrieve Approval Steps as Name.</p> <p>Note: We will configure the task later.</p>	 <p>The screenshot shows the 'Service Task Properties' dialog with the 'GENERAL' tab selected. The 'ID' field contains 'servicetask2'. The 'Name' field is highlighted with a red box and contains the text 'Retrieve Approval Steps'. A yellow circle with the number '1' is positioned next to the 'Documentation' input area.</p>
<p>14. On the selected 'Retrieve Approval Steps' service task, click Add Gateway .</p> <p>15. Select Exclusive Gateway .</p>	 <p>The screenshot shows a workflow diagram. A blue rounded rectangle represents a service task named 'Retrieve Approval Steps'. It is connected to a dark grey rounded rectangle representing an exclusive gateway. The exclusive gateway has two outgoing branches: one leading to a black circle with a right-pointing arrow, and another leading to a black circle with a circle inside. A third branch leads to a black circle with a square inside. A fourth branch leads to a black circle with a diamond inside. A fifth branch leads to a black circle with a trash can inside. A yellow circle with the number '1' is positioned near the connection between the service task and the exclusive gateway. A yellow circle with the number '2' is positioned near the parallel gateway icon.</p>
<p>16. Enter Approval required? as gateway Name.</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 is highlighted with a red box and contains the text 'Approval required?'. A yellow circle with the number '1' is positioned next to the 'Documentation' input area.</p>

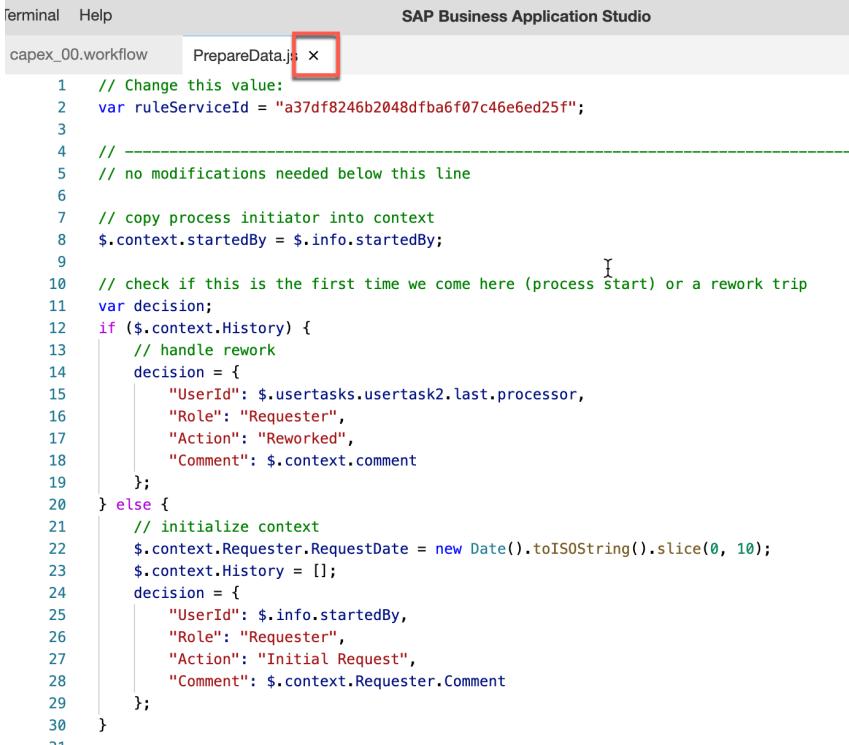
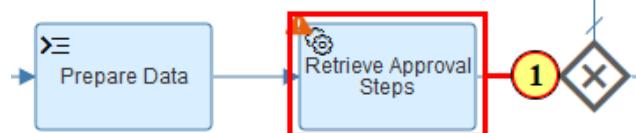
Explanation	Screenshot												
<p>17. Select Add Task .</p> <p>18. Select User Task .</p>	 <pre> graph LR A[Retrieve Approval Steps] --> B{ExclusiveGate y1} B --> C((User Task)) B --> D((Service Task)) B --> E((Script Task)) B --> F((Mail Task)) </pre>												
<p>Note that the Task ID is usertask1. This will be needed later.</p> <p>19. Enter Approval Task (Local Manager) as Task name.</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 Add Task .</p> <p>21. Select Script Task .</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>												

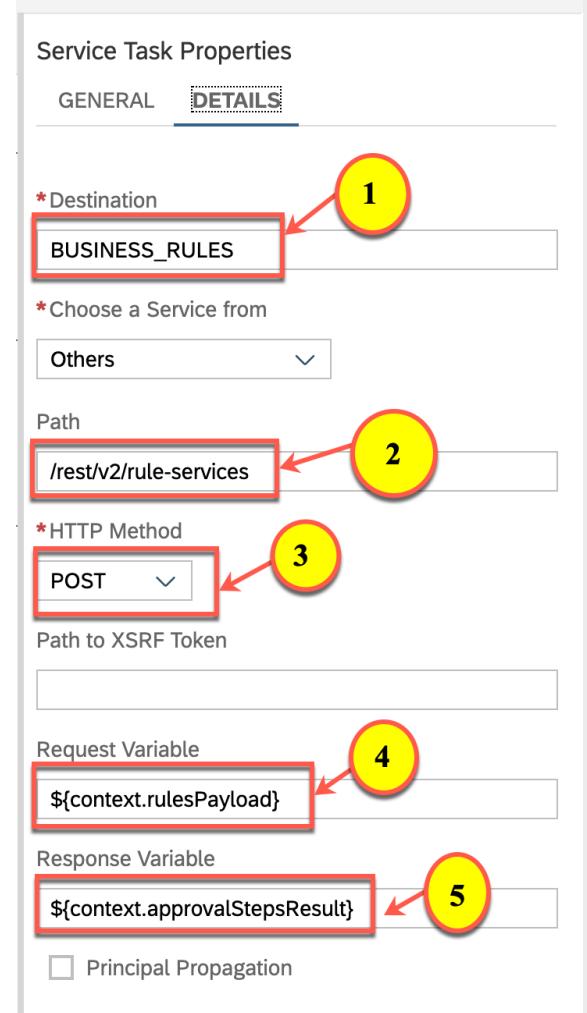
Explanation	Screenshot
<p>22. Enter Handle Approval (Local Manager) as Name.</p> <p>Note: We will configure the task later.</p>	 <p>The screenshot shows the 'Script Task Properties' dialog. The 'Name' field contains 'HandleApproval (Local Manager)' and is highlighted with a red box. A yellow circle with the number '1' points to the 'Documentation' field below it.</p>
<p>23. Click Add Gateway .</p> <p>24. Select Exclusive Gateway .</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 Approval result? as Name.</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>

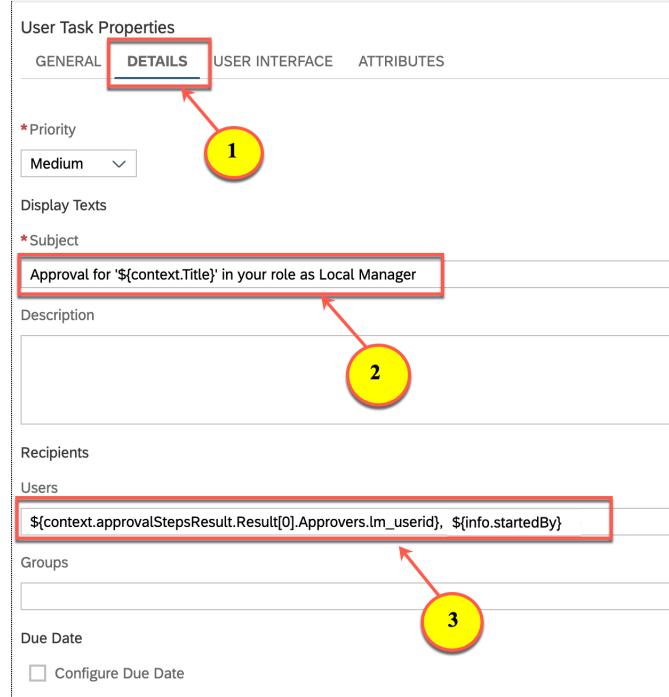
Explanation	Screenshot
<p>26. Select Add Gateway .</p> <p>27. Select Exclusive Gateway .</p>	
<p>28. Enter join as gateway Name.</p> <p>Note: This gateway will be joining the branches from a previous gateway together. In most cases, such joining gateway are optional and implicit, when sequence flows 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: join </p> <p>Documentation: </p>
<p>29. Select the Exclusive Gateway  with the name 'Approval required?'</p> <p>30. Click on the Sequence Flow Connector  and start dragging it</p> <p>31. Drop the Sequence Flow Connector onto the top corner of the Join gateway.</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>	

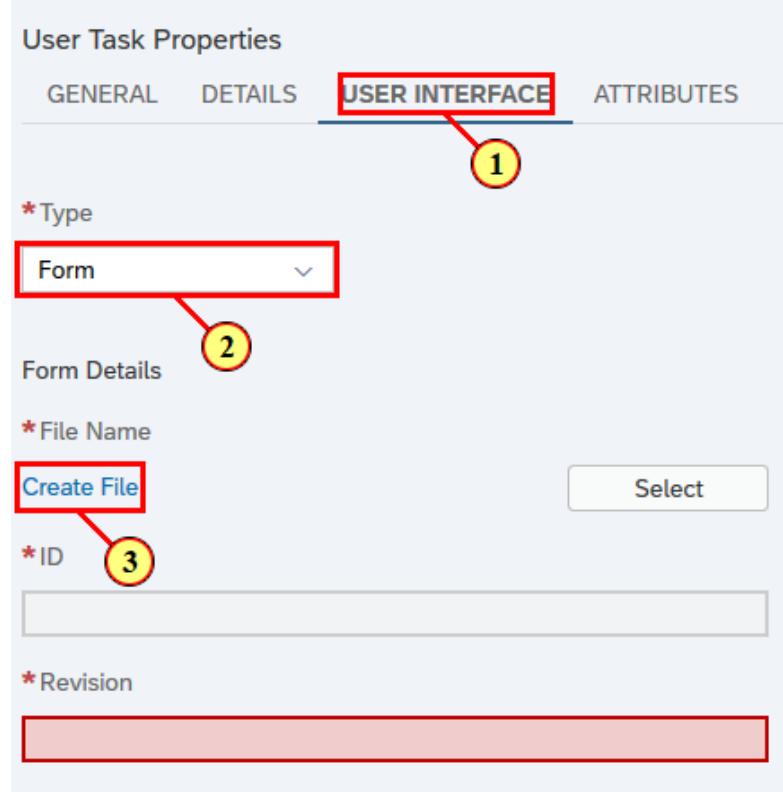
Explanation	Screenshot
<p>32. Drag the Connection Dot  from the middle of the Exclusive Gateway with name Approval required? to its upper corner.</p> <p>33. Drop on the upper corner of the icon.</p>	
<p>Since there is now more than one Sequence Flow, each flow has a name and branch condition.</p> <p>34. Enter no as Name for the newly created (upper) Sequence Flow.</p> <p>35. Select Default 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 Sequence Flow  between the gateway 'Approval required?' and the user task 'Approval Task (Local Manager)'</p>	

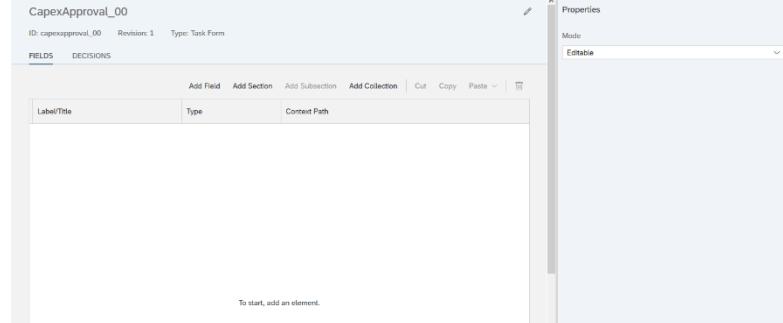
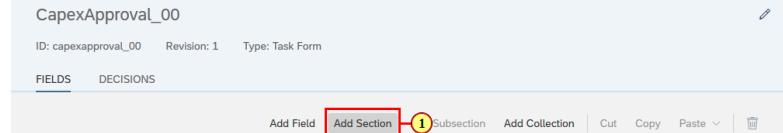
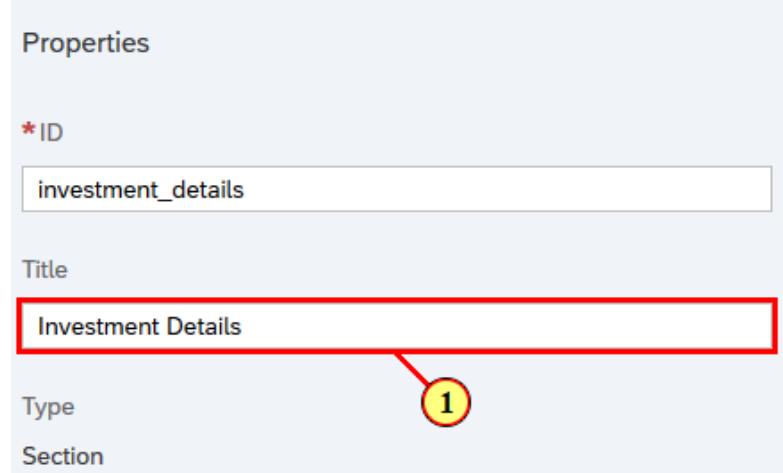
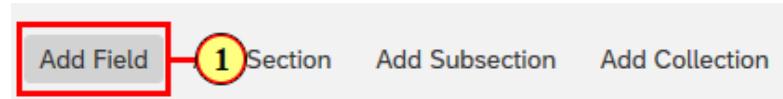
Explanation	Screenshot
<p>37. Enter yes as Sequence Flow name.</p> <p>38. Enter <code>#{context.approvalStepsResult.Result[0].Approvers.lm_required}</code> as Condition.</p> <p>⚠️ Be careful during copying. There are no spaces in the condition. Sometimes during copy, spaces are added automatically. In 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 Prepare Data.</p>	 <pre> graph LR Start(()) --> Capex Request PD[Prepare Data] PD --> RAS[Retrieve Approval Steps] </pre>
<p>40. In the properties, click Create File.</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>

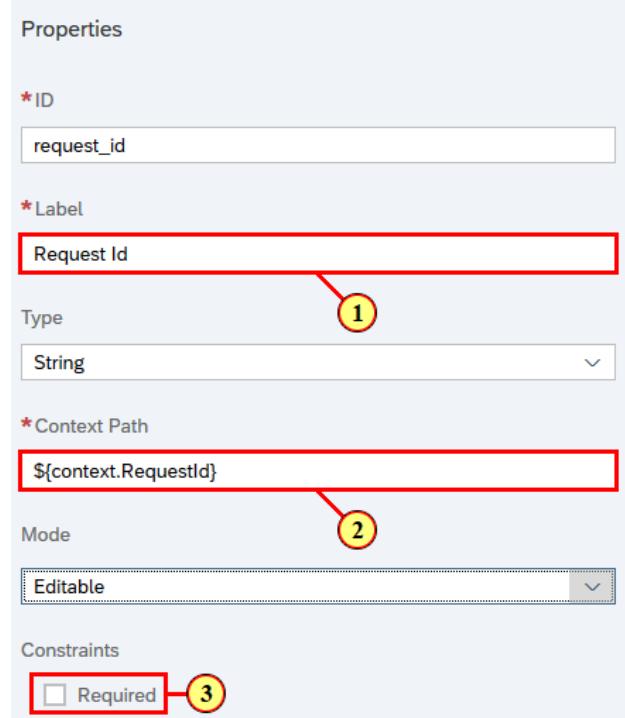
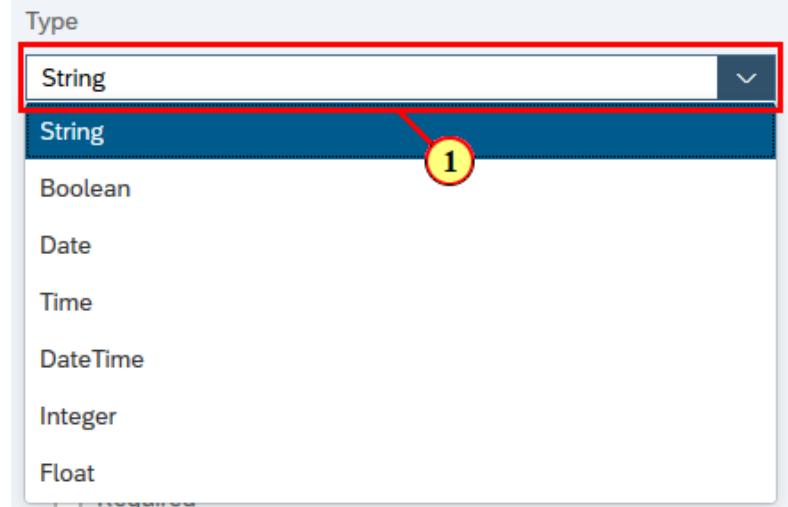
Explanation	Screenshot
<p>41. Enter PrepareData as Name.</p> <p>42. Click on Create.</p>	
<p>43. Delete all the default content and replace it with the script given in the file PrepareData.js.</p> <p>Note: This workflow calls a pre-build rule which you don't need to change now. In a business rules exercise, 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 = "a37df8246b2048dfba6f07c46e6ed25f"; 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 </pre>
<p>45. Select the Service Task Retrieve Approval Steps.</p>	

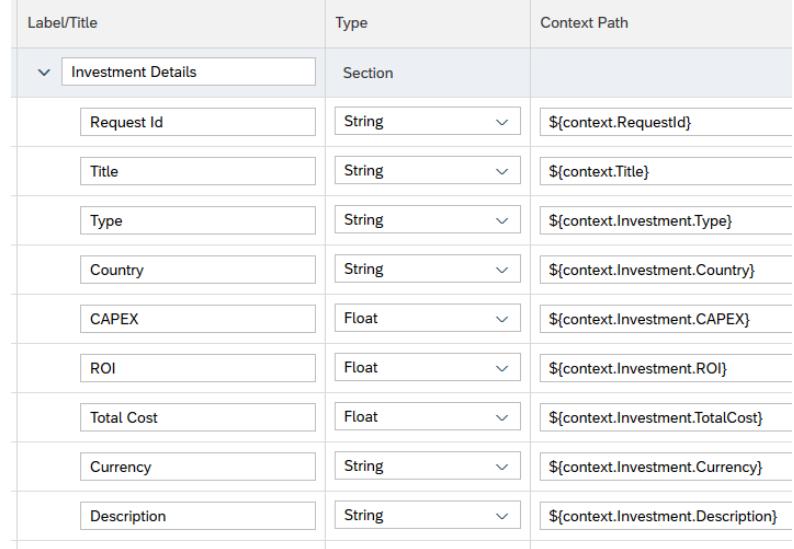
Explanation	Screenshot
<p>46. Select the DETAILS tab.</p> <p>47. Enter BUSINESS_RULES as Destination.</p> <p>48. Enter /rest/v2/rule-services as Path.</p> <p>49. Enter HTTP Method as POST.</p> <p>50. Enter \${context.rulesPayload} as Request Variable.</p> <p>51. Enter \${context.approvalStepsResult}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

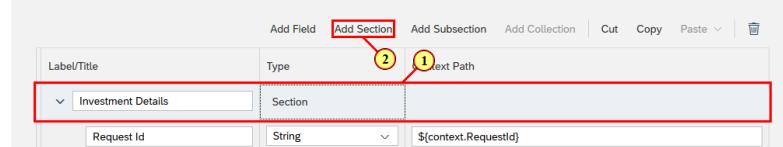
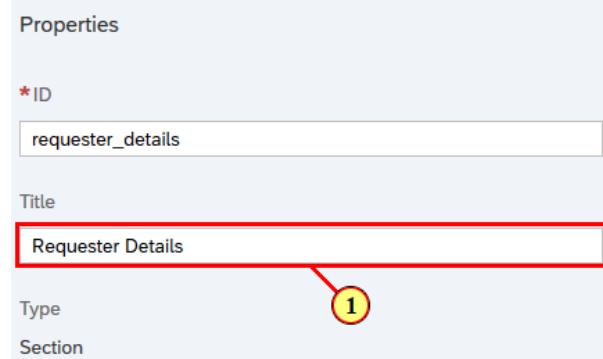
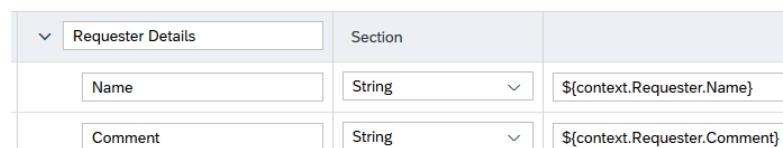
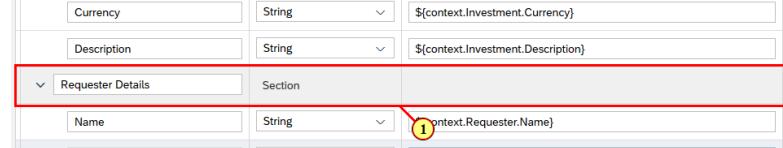
Explanation	Screenshot
<p>53. Select the DETAILS tab.</p> <p>54. Enter Approval for '\${context.Title}' in your role as Local Manager as Subject.</p> <p>55. Enter `\${context.approvalStepsResu lt.Result[0].Approvers.lm_u serid}, \${info.startedBy} as Recipients - Users.</p> <p>It means that this user 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>

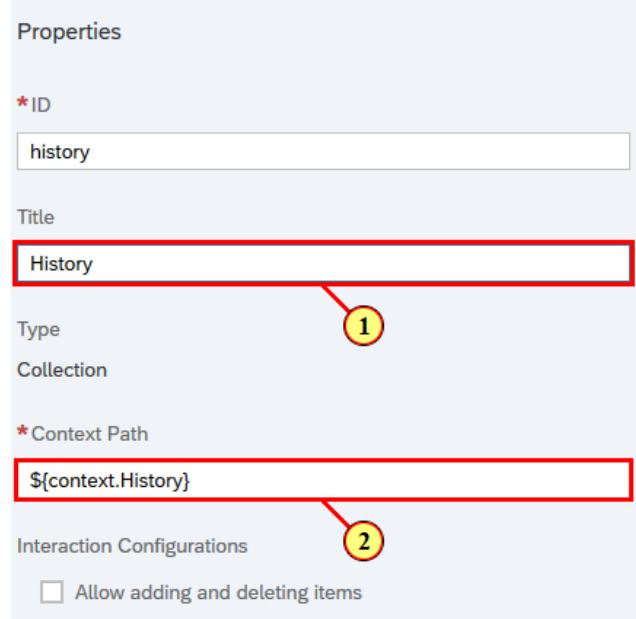
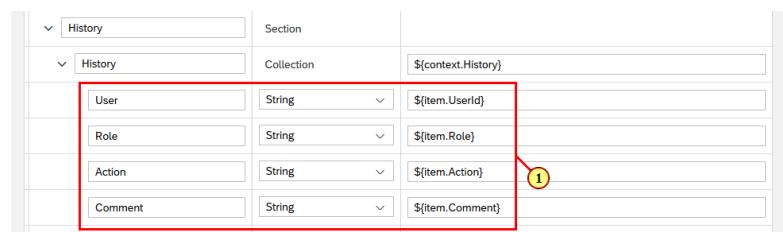
Explanation	Screenshot
<p>56. Select the USER INTERFACE tab.</p> <p>57. Select Form in the dropdown menu for Type.</p> <p>58. Click Create File.</p> <p>Note: Forms are a quick way to create simple task UIs.</p>	 <p>The screenshot shows the 'User Task Properties' dialog. The tabs at the top are GENERAL, DETAILS, USER INTERFACE (which is highlighted with a red box and has a yellow circle with '1' over it), and ATTRIBUTES. Below the tabs, there are fields for 'Type' (set to 'Form'), 'File Name' (labeled 'Create File' with a red box and yellow circle '2'), 'ID' (with a red box and yellow circle '3'), and 'Revision' (highlighted with a red box). A 'Select' button is also visible.</p>
<p>59. In the popup, enter CapexApproval_00 as Form name.</p> <p>60. Press TAB to accept the generated Id.</p> <p>61. Enter 1 as Revision.</p> <p>62. Click Create.</p>	 <p>The screenshot shows a 'New Form' dialog. It contains fields for 'Name' (set to 'CapexApproval_00' with a red box and yellow circle '1'), 'ID' (set to 'capexapproval_00' with a red box and yellow circle '2'), 'Revision' (set to '1' with a red box and yellow circle '3'), and 'Type' (set to 'Task Form'). At the bottom right, there is a 'Create' button (highlighted with a red box and yellow circle '4') and a 'Cancel' button.</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>The screenshot shows the SAP Cloud Platform Workflow Management Forms Editor interface. A task form titled "CapexApproval_00" is displayed. The "FIELDS" tab is selected. A toolbar at the top includes "Add Field", "Add Section", "Add Subsection", "Add Collection", "Cut", "Copy", "Paste", and a trash icon. On the right, a "Properties" panel shows "Mode: Editable". Below the toolbar, there's a table with columns "Label/Title", "Type", and "Context Path". A message at the bottom says "To start, add an element."</p>
<p>63. Click Add Section.</p>	 <p>The screenshot shows the same task form "CapexApproval_00". The "Add Section" button in the toolbar is highlighted with a yellow circle containing the number "1".</p>
<p>64. Enter Investment Details as Title. Note: The ID field is automatically filled out based on the given title.</p>	 <p>The screenshot shows the "Properties" section of the task form. The "ID" field is set to "investment_details". The "Title" field is set to "Investment Details", which is also highlighted with a red rectangle and a yellow circle containing the number "1". Other fields shown include "Type" and "Section".</p>
<p>65. Select Add Field.</p>	 <p>The screenshot shows the toolbar with the "Add Field" button highlighted with a yellow circle containing the number "1". Other buttons in the toolbar include "Section", "Add Subsection", and "Add Collection".</p>

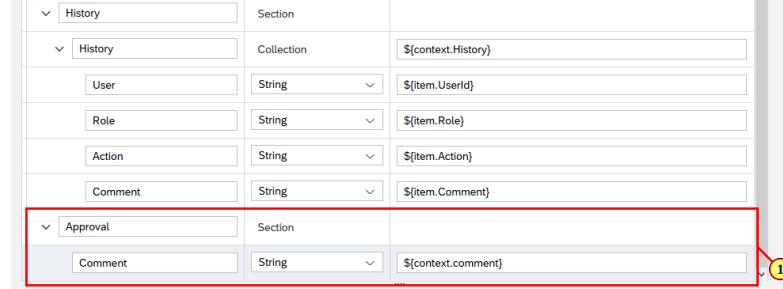
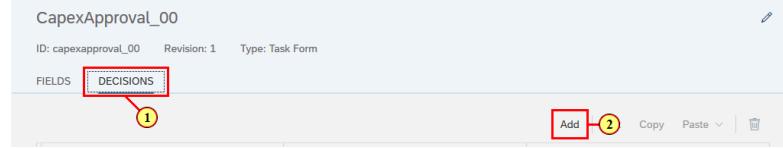
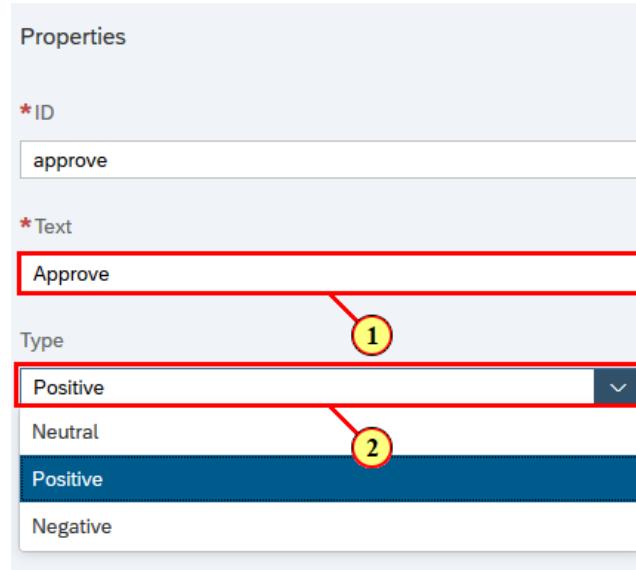
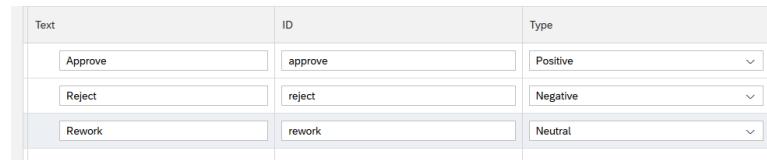
Explanation	Screenshot
<p>66. Enter Request Id as Label. The ID is then automatically derived from this.</p> <p>67. Enter <code> \${context.RequestId}</code> as Context Path.</p> <p>68. An input field can also be marked as Required. Since the Request Id is already given and should not be editable, leave it unchecked.</p>	 <p>Properties</p> <p>* ID request_id</p> <p>* Label Request Id 1</p> <p>Type String</p> <p>* Context Path \${context.RequestId} 2</p> <p>Mode Editable</p> <p>Constraints <input type="checkbox"/> Required 3</p>
<p>69. Clicking on the Type field opens a dropdown list. Select the type as String.</p>	 <p>Type</p> <p>String 1</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 Mode field opens a dropdown list. Select Display-Only.</p>	 <p>Mode</p> <p>Display-Only</p> <p>Editable</p> <p>Display-Only</p>

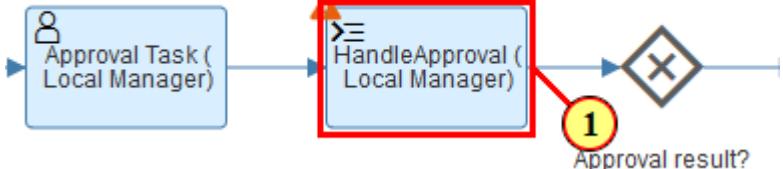
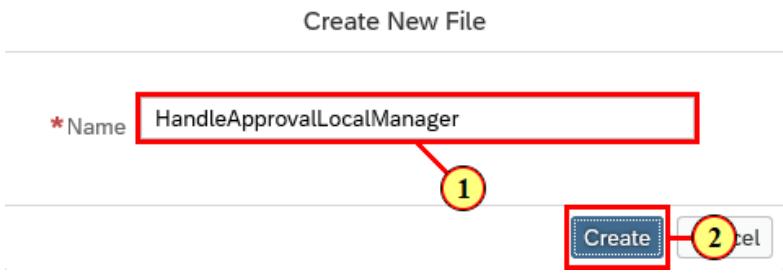
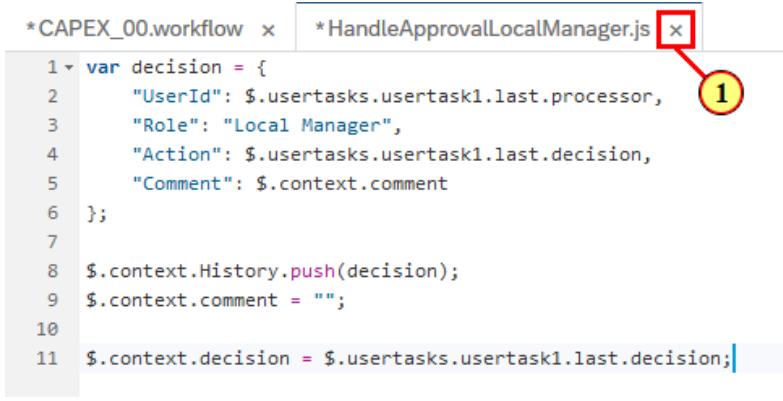
Explanation	Screenshot																														
<p>71. In the same way, add the following fields with given Label, Type and Context Path:</p> <ul style="list-style-type: none"> ○ Set the property of all of the fields as Editable and Required. <p>Label: Title Type: String Context Path: \${context.Title}</p> <p>Label: Type Type: String Context Path: \${context.Investment.Type}</p> <p>Label: Country Type: String Context Path: \${context.Investment.Country}</p> <p>Label: CAPEX Type: Float Context Path: \${context.Investment.CAPEX}</p> <p>Label: ROI Type: Float Context Path: \${context.Investment.ROI}</p> <p>Label: Total Cost Type: Float Context Path: \${context.Investment.TotalCost}</p> <p>Label: Currency Type: String Context path: \${context.Investment.Currency}</p> <p>Label: Description Type: String Context Path: \${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 1204 861">Type</th> <th data-bbox="1204 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 1204 903">String</td> <td data-bbox="1204 861 1481 903">\${context.RequestId}</td> </tr> <tr> <td data-bbox="687 903 1008 946">Title</td> <td data-bbox="1008 903 1204 946">String</td> <td data-bbox="1204 903 1481 946">\${context.Title}</td> </tr> <tr> <td data-bbox="687 946 1008 988">Type</td> <td data-bbox="1008 946 1204 988">String</td> <td data-bbox="1204 946 1481 988">\${context.Investment.Type}</td> </tr> <tr> <td data-bbox="687 988 1008 1030">Country</td> <td data-bbox="1008 988 1204 1030">String</td> <td data-bbox="1204 988 1481 1030">\${context.Investment.Country}</td> </tr> <tr> <td data-bbox="687 1030 1008 1072">CAPEX</td> <td data-bbox="1008 1030 1204 1072">Float</td> <td data-bbox="1204 1030 1481 1072">\${context.Investment.CAPEX}</td> </tr> <tr> <td data-bbox="687 1072 1008 1115">ROI</td> <td data-bbox="1008 1072 1204 1115">Float</td> <td data-bbox="1204 1072 1481 1115">\${context.Investment.ROI}</td> </tr> <tr> <td data-bbox="687 1115 1008 1157">Total Cost</td> <td data-bbox="1008 1115 1204 1157">Float</td> <td data-bbox="1204 1115 1481 1157">\${context.Investment.TotalCost}</td> </tr> <tr> <td data-bbox="687 1157 1008 1199">Currency</td> <td data-bbox="1008 1157 1204 1199">String</td> <td data-bbox="1204 1157 1481 1199">\${context.Investment.Currency}</td> </tr> <tr> <td data-bbox="687 1199 1008 1241">Description</td> <td data-bbox="1008 1199 1204 1241">String</td> <td data-bbox="1204 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}																													

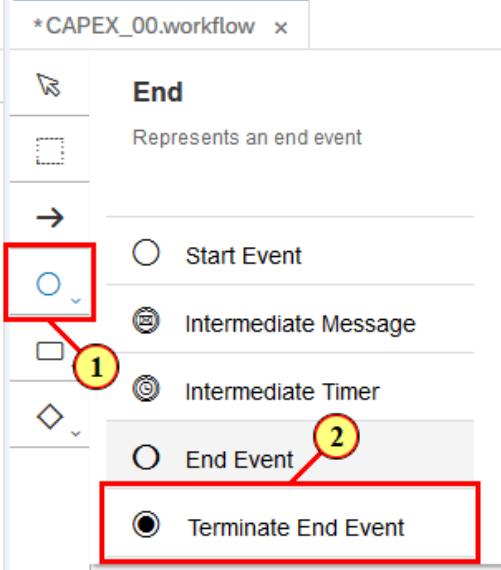
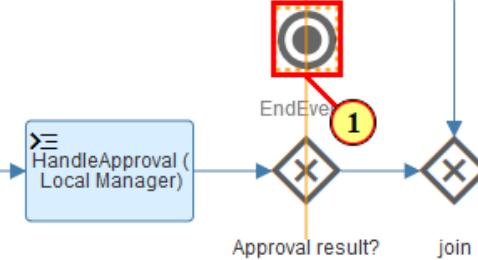
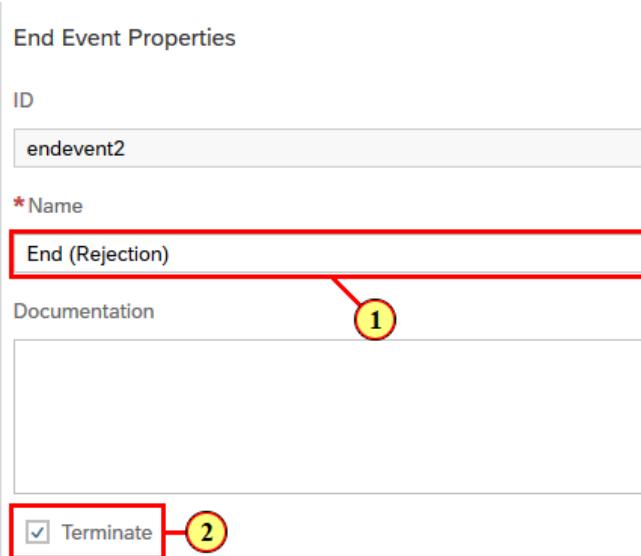
Explanation	Screenshot
<p>72. Select Investment Details section.</p> <p>73. Select Add Section.</p>	
<p>74. Enter Requester Details as Title.</p>	
<p>75. Add the following fields.</p> <p>Label: Name Type: String Mode: Display-Only Control: Input Context Path: \${context.Requester.Name}</p> <p>Label: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

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

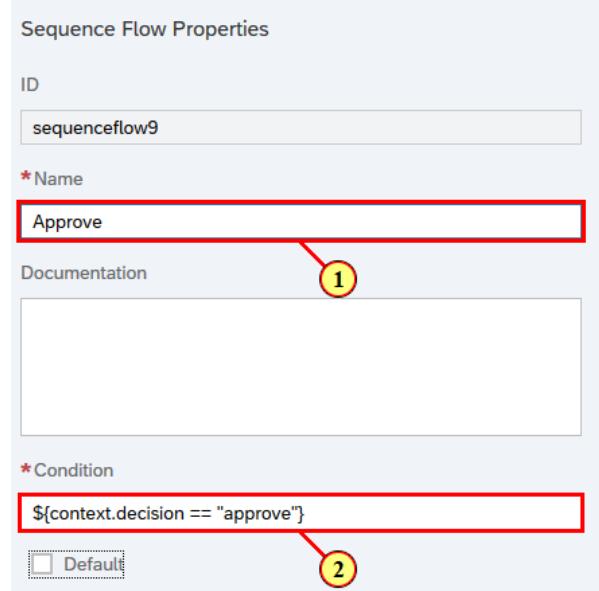
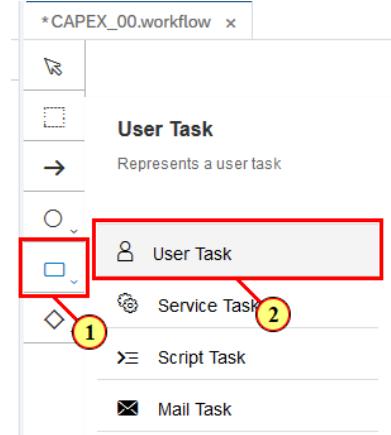
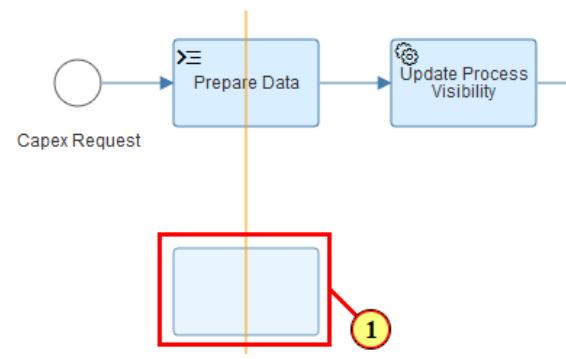
Explanation	Screenshot
<p>82. Add a new Section with title Approval.</p> <p>83. Add a new field and name it Comment.</p> <p>84. Enter <code>#{context.comment}</code> as Context Path</p> <p>85. Change the UI Control type from <i>Input field</i> to Text Area.</p>	<p>The screenshot shows the 'Properties' 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>

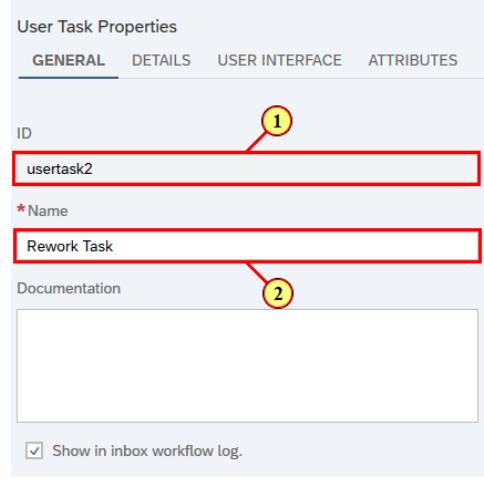
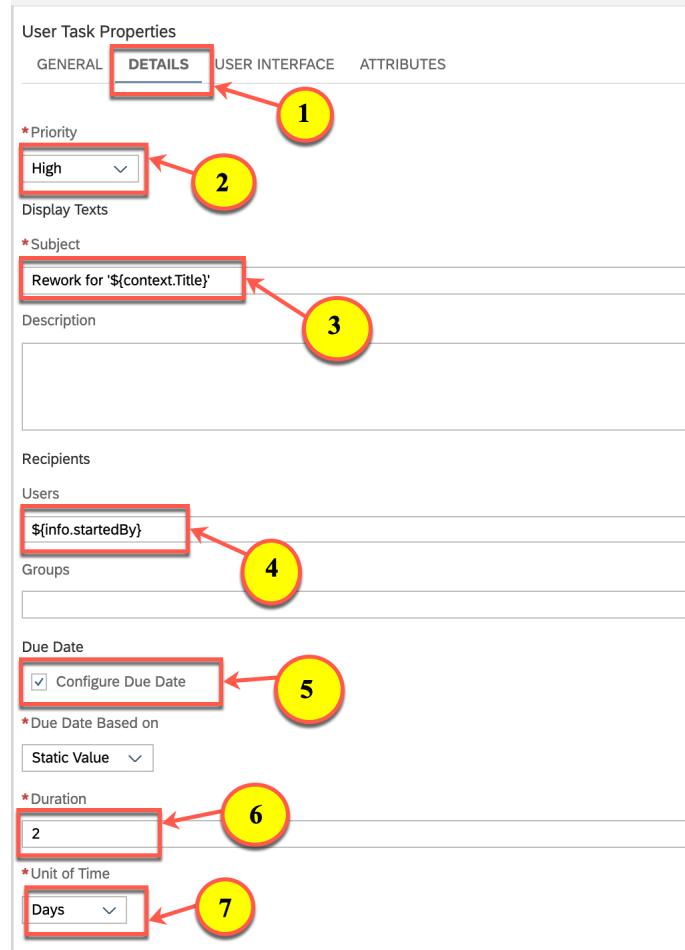
Explanation	Screenshot
86. Ensure your Form looks like this.	
87. Select the Decisions tab. 88. Select Add to add a new decision option.	
89. Enter Approve as Text of the decision button. 90. Select Positive as Type from the dropdown-list.	
91. In the same way, add a decision option Reject with type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

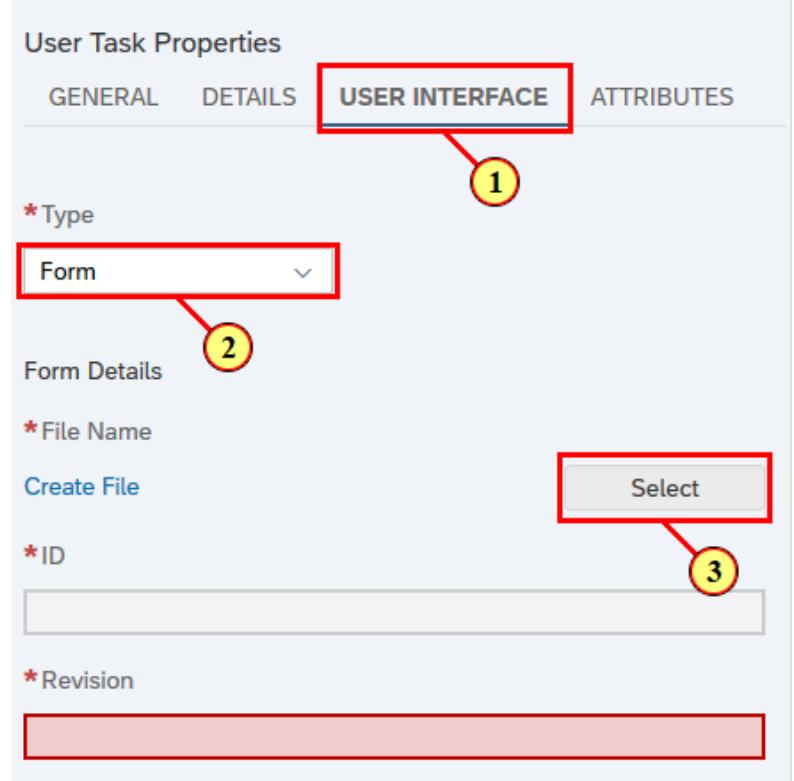
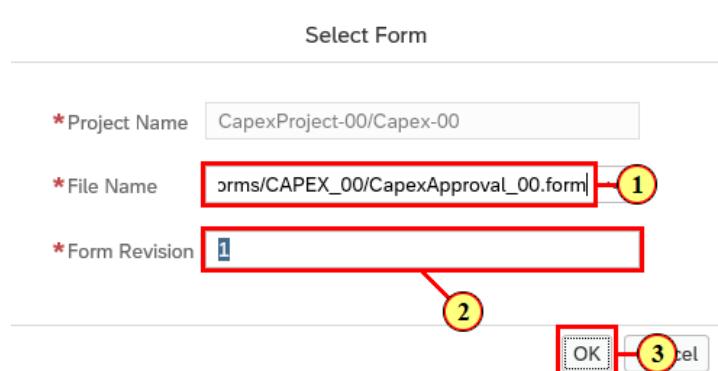
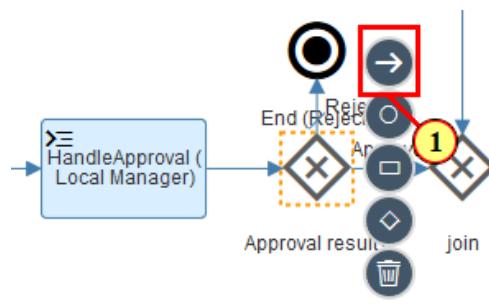
Explanation	Screenshot
<p>93. Select the script task Handle Approval (Local Manager).</p>	
<p>94. In the Script Task Properties, select Create File.</p>	
<p>95. Enter HandleApprovalLocalManager as Filename 96. Select Create.</p>	
<p>97. Replace the default content with the script from HandleApprovalLocalManager.js in the exercises folder. 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 Add Event.</p> <p>100. Select the Terminate End Event.</p>	
<p>101. Place the End Event above the exclusive gateway named Approval result?</p> <p>Note: The yellow guiding lines help you position it nicely.</p>	
<p>102. Enter End (Rejection) as Name.</p> <p>103. Ensure that Terminate is checked.</p>	

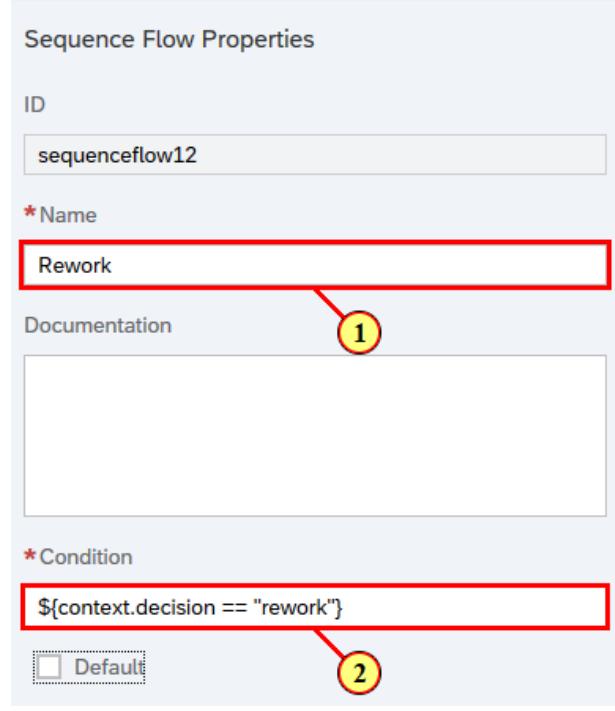
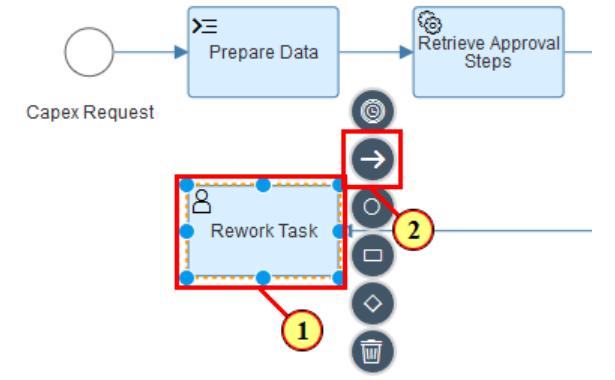
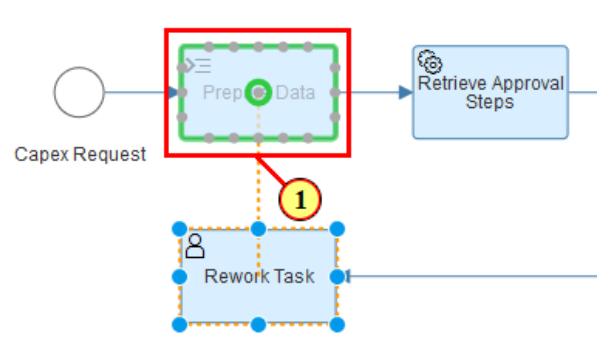
Explanation	Screenshot
<p>104. Select the exclusive gateway with name Approval result?</p> <p>105. Drag the sequence flow connector → onto the end event with name End (Rejection)</p>	
<p>106. Enter Reject as Name.</p> <p>107. Select the checkbox for Default condition.</p>	<p>Sequence Flow Properties</p> <p>ID: sequenceflow11</p> <p>* Name: Reject</p> <p>Documentation: 1</p> <p><input checked="" type="checkbox"/> Default 2</p>
<p>108. Select the sequence flow connector between the gateways named Approval result? and join.</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 Approve as Name.</p> <p>110. Enter <code>#{context.decision == "approve"}</code> as Condition.</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 Add Task .</p> <p>112. Select User Task .</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 User Task below the script task Prepare Data.</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'. Below 'Prepare Data', there is a light blue rounded rectangle labeled 'User Task'. A vertical yellow guiding line is positioned to the left of the 'User Task' box, and a horizontal yellow guiding line connects the bottom edge of 'Prepare Data' to the top edge of 'User Task'. A yellow circle labeled '1' points to the 'User Task' box.</p>

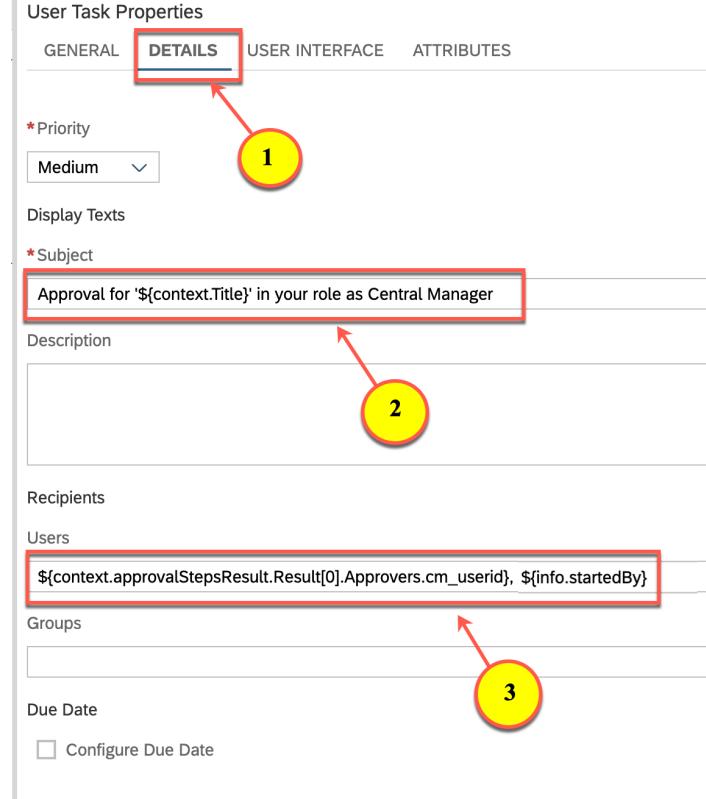
Explanation	Screenshot
<p>114. Verify that the ID is <code>usertask2</code>. If it differs, the references in the script file Prepare Data.js needs to be adapted accordingly.</p> <p>115. Enter Rework Task as Name.</p>	 <p>User Task Properties</p> <p>GENERAL DETAILS USER INTERFACE ATTRIBUTES</p> <p>ID: 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 DETAILS tab.</p> <p>117. Select the Priority as High.</p> <p>118. Enter Rework for '\${context.Title}' as Subject.</p> <p>119. Enter \${info.startedBy} as Recipients - User.</p> <p>120. Select Configure Due Date.</p> <p>121. Enter 2 as Duration and Days from the dropdown list Unit of Time, 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>

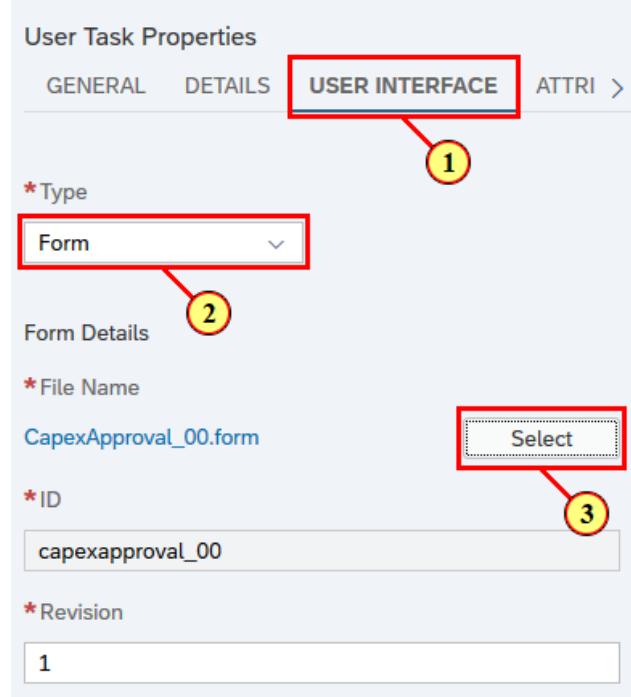
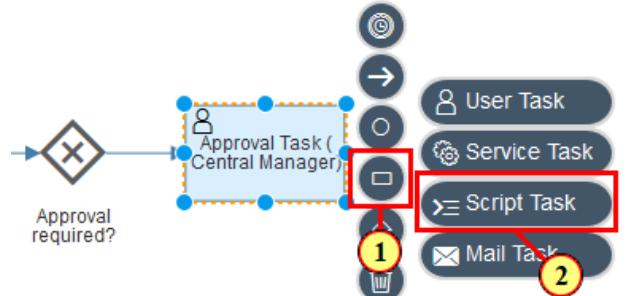
Explanation	Screenshot
<p>122. Select the USER INTERFACE tab.</p> <p>123. Select Form as Type.</p> <p>124. Click Select to choose an existing form.</p>	 <p>User Task Properties</p> <p>GENERAL DETAILS USER INTERFACE ATTRIBUTES</p> <p>* Type</p> <p>Form</p> <p>Form Details</p> <p>* File Name</p> <p>Create File Select</p> <p>* ID</p> <p>* Revision</p>
<p>125. Select the previously created form CapexApproval_00</p> <p>126. Ensure that the Revision is 1.</p> <p>127. Select OK.</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 Approval result? and drag an additional Sequence Flow Connector...</p>	 <pre> graph LR A[HandleApproval Local Manager] --> B{Approval result?} B --> C(()) B --> D(()) B --> E(()) B --> F(()) C --> G[End Reject] D --> H(()) E --> I(()) F --> J(()) </pre>

Explanation	Screenshot
<p>...onto the User Task Rework Task. Position it onto the highlighted middle dot.</p>	
<p>129. Click on the Connection Dot 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>	

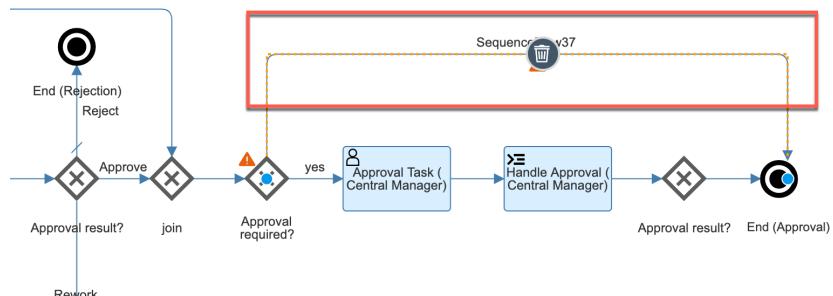
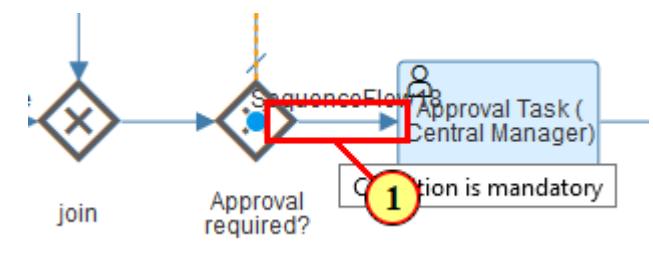
Explanation	Screenshot
<p>131. Enter Rework as Name.</p> <p>132. Enter <code>#{context.decision == "rework"}</code> as Condition.</p>	 <p>Sequence Flow Properties</p> <p>ID: sequenceflow12</p> <p>* Name: Rework</p> <p>Documentation: 1</p> <p>* Condition: \${context.decision == "rework"}</p> <p>Default: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

Explanation	Screenshot
<p>Ensure your workflow looks like this.</p>	<pre> graph LR Start(()) --> Prep[Prepare Data] Prep --> Retrieve[Retrieve Approval Steps] Retrieve --> Decision{Approval required?} Decision -- yes --> ApprovalTask[Approval Task Local Manager] ApprovalTask --> Handle[Handle Approval Local Manager] Handle --> Decision Decision -- no --> EndReject((End (Reject))) ApprovalTask --> EndApproved((End (Approved))) EndApproved --> Join(()) Decision --> Rework[Rework Task] Rework --> Prep </pre>
<p>135. Select the exclusive gateway with name join. 136. Select Add Gateway .</p> <p>137. Select the Exclusive Gateway option .</p> <p>138. Name it as Approval required?</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 'Parallel Gateway' option. Step 3 highlights the newly added 'Exclusive Gateway' node, which is redboxed and connected to a decision gateway labeled 'Approval required?'.</p>
<p>139. Select Add Task .</p> <p>140. Select User Task .</p> <p>141. Enter Approval Task (Central Manager) as Name.</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 'Service Task' icon. Step 3 highlights the 'Script Task' icon. Step 4 highlights the 'Mail Task' icon. The 'User Task' icon is redboxed.</p>

Explanation	Screenshot
<p>142. Select the Details tab.</p> <p>143. Enter Approval for '\${context.Title}' in your role as Central Manager as Subject.</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>

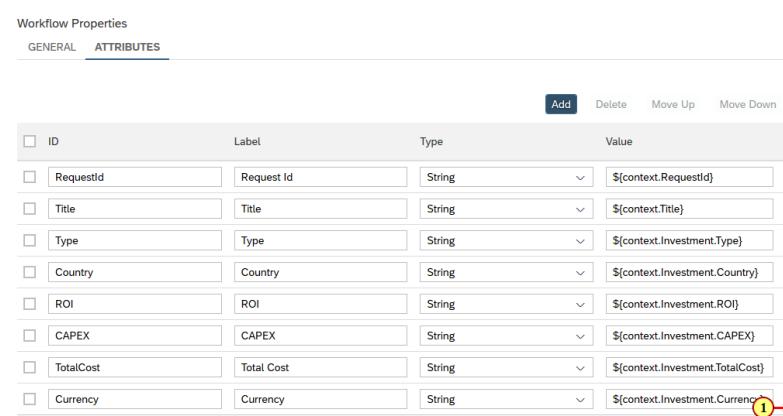
Explanation	Screenshot
<p>145. Select the User Interface tab.</p> <p>146. Select Form as Type.</p> <p>147. Use the Select button to choose the previously created form capexapproval_00.</p>	 <p>User Task Properties</p> <p>GENERAL DETAILS USER INTERFACE ATTRI ></p> <p>*Type Form</p> <p>Form Details</p> <p>*File Name CapexApproval_00.form</p> <p>Select</p> <p>*ID capexapproval_00</p> <p>*Revision 1</p>
<p>148. Select Add Task on the central manager approval.</p> <p>149. Select the Script Task.</p>	 <pre> graph LR Diamond{Approval required?} --> ApprovalTask[Approval Task (Central Manager)] ApprovalTask --> End(()) </pre> <p>1</p> <p>2</p> <p>User Task</p> <p>Service Task</p> <p>Script Task</p> <p>Mail Task</p>

Explanation	Screenshot
<p>150. Enter Handle Approval (Central Manager) as Name.</p> <p>151. Select Create File to create a new file. Name it HandleApprovalCentralManager 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 Task ID (<code>usertask3</code>) multiple times. This is the id of the User Task Approval Task (Central Manager) 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 Handle Approval scripts are which task they refer to, and the 'Role' stored in the history element of workflow context.</p>	<p>Script Task Properties</p> <p>ID: scripttask3</p> <p>* Name: Handle Approval (Central Manager) (1)</p> <p>Documentation:</p> <p>* Script File: HandleApprovalCentralManager.js (2)</p>
<p>152. Select Add Gateway (diamond icon).</p> <p>153. Select the Exclusive Gateway (diamond with a diagonal line icon).</p> <p>154. Name it as Approval result?</p>	<p>Handle Approval (Central Manager)</p> <p>Exclusive Gateway (1)</p> <p>Parallel Gateway (2)</p> <p>Approval result? (highlighted)</p>
<p>155. Click Approval result? and Select Add Event.</p> <p>156. Select the Terminate End Event. Name the event as End (Approval).</p>	<p>Intermediate Message</p> <p>Intermediate Timer</p> <p>End Event</p> <p>Terminate End Event (highlighted) (1)</p> <p>Approval result? (2)</p> <p>End (Approval)</p>

Explanation	Screenshot
<p>157. On the previous exclusive gateway with name Approval required? add a new Sequence Flow Connector and drag it onto the newly added Terminate End Event.</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(()) --> Join(()) Join -- "join" --> Gate{Approval required?} Gate -- "no" --> Reject(()) Reject -- "End (Rejection) Reject" --> End(()) Gate -- "yes" --> Task1[Approval Task (Central Manager)] Task1 --> Task2[Handle Approval (Central Manager)] Task2 --> Gate Gate -- "yes" --> End </pre>
<p>158. Enter no as Name.</p> <p>159. Select it as Default.</p>	<p>Sequence Flow Properties</p> <p>ID: sequenceflow23</p> <p>* Name: no (1)</p> <p>Documentation:</p> <p><input checked="" type="checkbox"/> Default (2)</p>
<p>160. Select the Sequence Flow Connector between Approval required? and Approval Task (Central Manager).</p>	

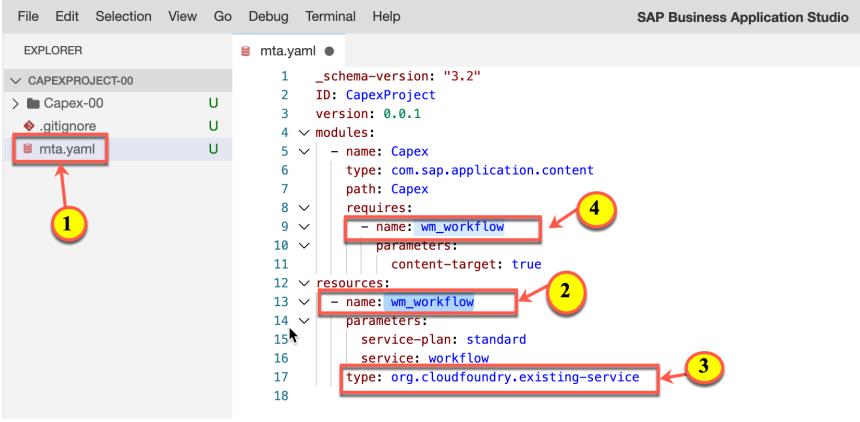
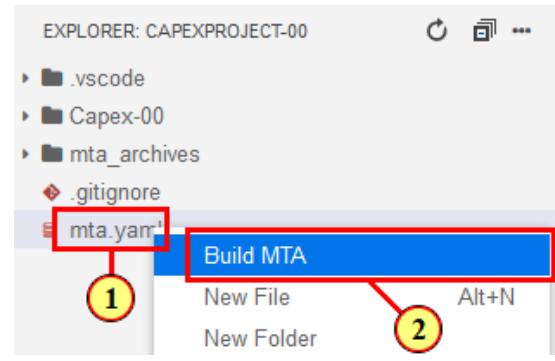
Explanation	Screenshot
<p>161. Enter Yes as Name.</p> <p>162. Enter <code>#{context.approvalStepsResult.Result[0].Approvers.cm_required}</code> as Condition.</p> <p>Note: Be careful during copying. There are no spaces in the condition. Sometimes during copy, spaces are added automatically. In this case, 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>Sequence Flow Properties</p> <p>ID: sequenceflow18</p> <p>*Name: yes</p> <p>Documentation: 1</p> <p>*Condition: approvalStepsResult.Result[0].Approvers.cm_required</p> <p><input type="checkbox"/> Default: 2</p>
<p>163. Create a new Terminating End Event and place it above the Approval result? gateway. Name it End (Rejection).</p> <p>164. Drag a new Sequence Flow Connection from Approval result? onto the End (Rejection) event. Name it Reject and select Default as condition.</p>	<pre> graph LR A{Approval required?} -- yes --> B[Approval Task (Central Manager)] B --> C{Handle Approval (Central Manager)} C --> D{Approval result?} C --> E(()) D -- no --> F(()) D -- yes --> G(()) E --> H(()) G --> I(()) </pre>
<p>165. Select the Sequence Flow Connection from Approval result? onto the End (Approval) event.</p> <p>166. Name it Approve and enter <code>#{context.decision == "approve"}</code> as condition.</p>	<p>Sequence Flow Properties</p> <p>ID: sequenceflow36</p> <p>*Name: Approve</p> <p>Documentation:</p> <p>*Condition: #{context.decision == "approve"} 3</p> <p><input type="checkbox"/> Default:</p>

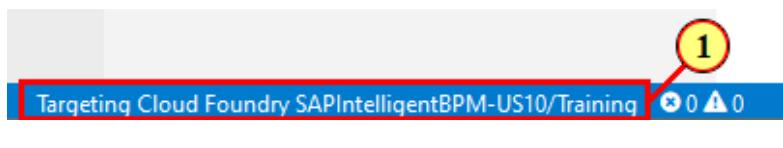
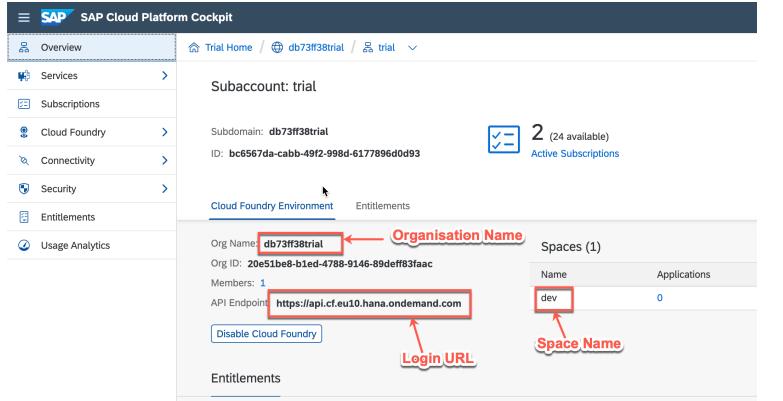
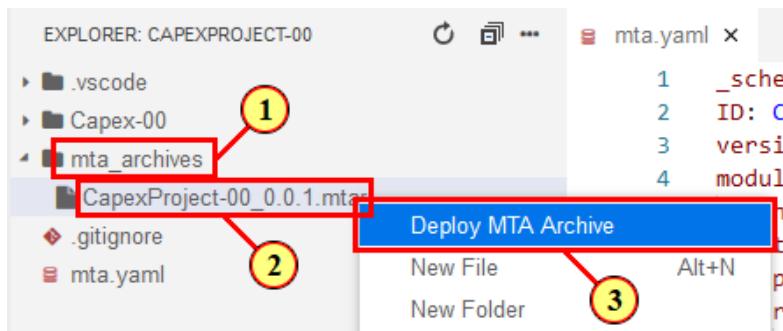
Explanation	Screenshot
<p>167. Drag a new Sequence Flow Connection from Approval result? onto the Rework Task.</p> <p>168. Adjust it that it starts directly downward by selecting the starting Connection Dot and move it to the lower edge of the Exclusive Gateway.</p>	<p>The screenshot shows a complex workflow involving multiple approval steps and a Rework Task. A new sequence flow has been added from the 'Approval result?' gateway to the Rework Task, which is highlighted with a red box. The flow starts at the top of the gateway and connects directly to the Rework Task, bypassing the upper connection point.</p>
<p>169. Name it as Rework and enter <code>\$(context.decision == "rework")</code> as condition.</p>	<p>The screenshot shows the 'Sequence Flow Properties' dialog for the newly created sequence flow. The name is set to 'Rework' (circled in yellow) and the condition is set to <code>\$(context.decision == "rework")</code> (circled in red). The sequence flow itself is also circled in yellow.</p>
<p>170. Click anywhere on an empty space in the workflow editor to open Workflow Properties.</p> <p>171. In Workflow Properties, enter Subject as CAPEX Approval for <code>'\$(context.Title)'</code></p> <p>172. Enter Business Key as <code>\$(context.RequestId)</code></p> <p>173. Select the ATTRIBUTES tab.</p>	<p>The screenshot shows the 'Workflow Properties' dialog for the entire workflow. The subject is set to 'CAPEX Approval for '\$(context.Title)' (circled in red), the business key is set to '\$(context.RequestId)' (circled in red), and the general tab is selected (circled in yellow).</p>
<p>Note: You can drag the vertical divider line of the Workflow Properties Pane to enlarge it.</p> <p>Now we will add attributes from the workflow context that will be exposed to be seen in Process Visibility scenario (in exercise 3 later)</p>	<p>The screenshot shows the 'Workflow Properties' dialog with the 'ATTRIBUTES' tab selected (circled in yellow). The 'GENERAL' tab is also visible.</p>

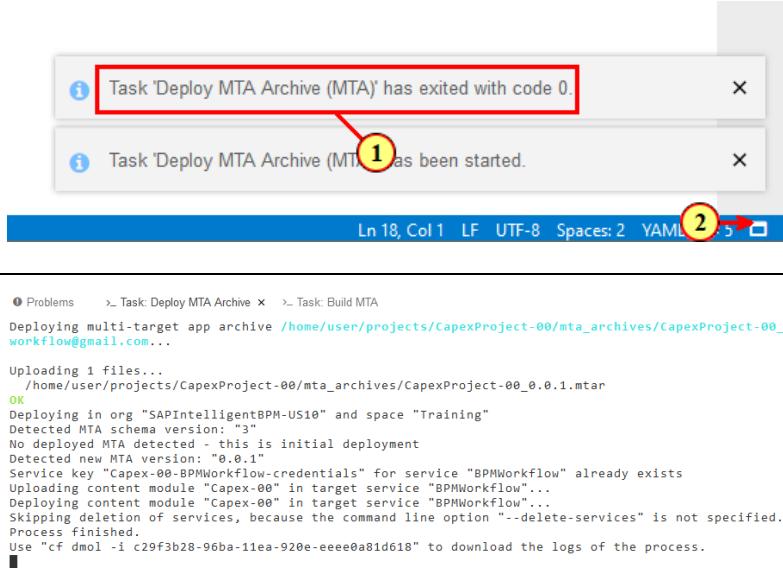
Explanation	Screenshot
<p>174. Select Add.</p> <p>175. Enter Currency as ID.</p> <p>176. Enter Currency as Label.</p> <p>177. Enter Value: \${context.Investment.Currency} as Value.</p>	
<p>178. In the same way, add the following fields:</p> <p>ID: TotalCost Label: Total Cost Value: \${context.Investment.TotalCost}</p> <p>ID: CAPEX Label: CAPEX Value: \${context.Investment.CAPEX}</p> <p>ID: ROI Label: ROI Value: \${context.Investment.ROI}</p> <p>ID: Country Label: Country Value: \${context.Investment.Country}</p> <p>ID: Type Label: Type Value: \${context.Investment.Type}</p> <p>ID: Title Label: Title Value: \${context.Title}</p> <p>ID: RequestId Label: Request Id Value: \${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)) --> PrepareData[Prepare Data] PrepareData --> RetrieveSteps[Retrieve Approval Steps] RetrieveSteps --> ApprovalReq{Approval required?} ApprovalReq -- yes --> LocalManagerTask[Approval Task (Local Manager)] LocalManagerTask --> LocalManagerResult{Approval result?} LocalManagerResult -- no --> EndRejection1((End (Rejection))) LocalManagerResult -- yes --> ReworkTask1[Rework Task] LocalManagerTask -. Rework .-> ReworkTask1 ApprovalReq -- no --> CentralManagerTask[Approval Task (Central Manager)] CentralManagerTask --> CentralManagerResult{Approval result?} CentralManagerResult -- no --> EndRejection2((End (Rejection))) CentralManagerResult -- yes --> ReworkTask2[Rework Task] CentralManagerTask -. Rework .-> ReworkTask2 LocalManagerResult -. join .-> CentralManagerResult CentralManagerResult -. join .-> EndApproval((End (Approval))) </pre>
<p>Note: If you did not use Auto Save, ensure that all files are saved.</p>	

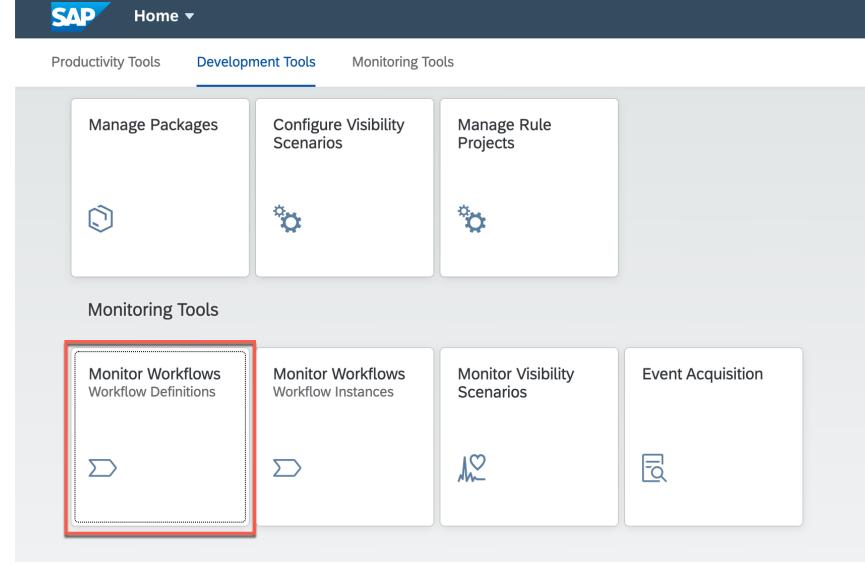
BUILD AND DEPLOY WORKFLOW

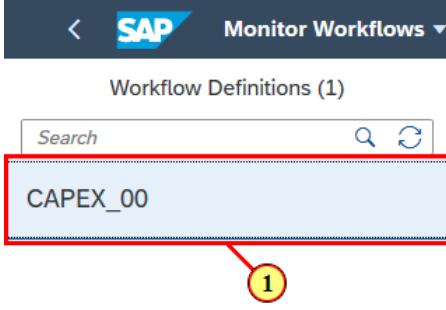
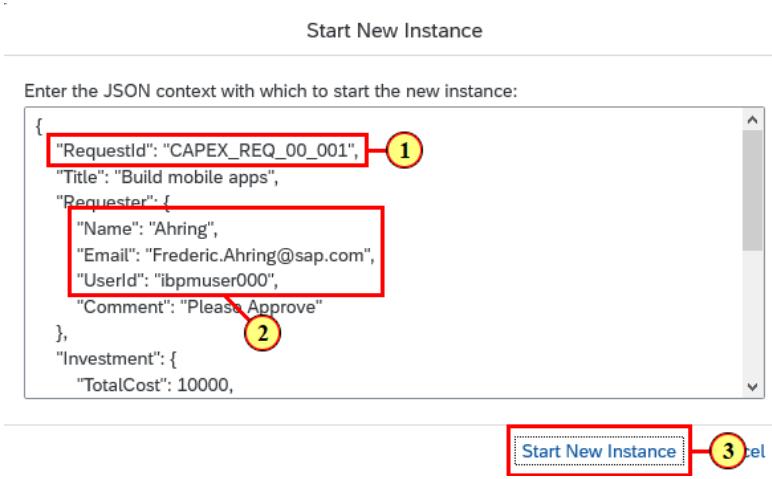
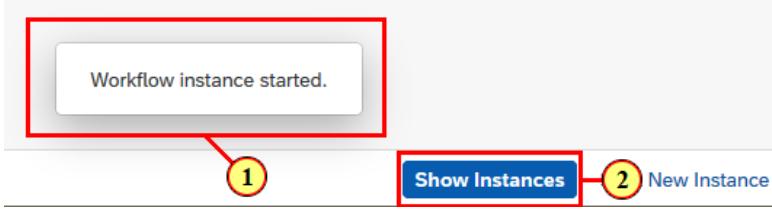
Explanation	Screenshot
179. Double-click the file mta.yaml to open it.	
<p>Note: The default MTA will create a reference to a new workflow service instance. When you ran the Booster in the setup the workflow service instance was already created so change the resource to the existing instance name.</p> <p>180. Change the resource name to wm_workflow.</p> <p>181. Change the resource type to org.cloudfoundry(existing-service).</p> <p>182. Change the resource reference to wm_workflow.</p>	 <pre data-bbox="572 629 1432 1051"> File Edit Selection View Go Debug Terminal Help SAP Business Application Studio EXPLORER mta.yaml CAPEXPROJECT-00 Capex-00 .gitignore 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>
183. Close the tab and ensure the file is saved.	
<p>184. To build the project, right-click on mta.yaml to open the context menu.</p> <p>185. Select Build MTA.</p>	

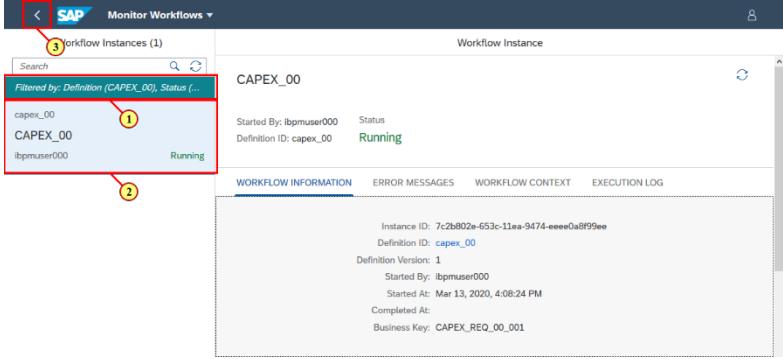
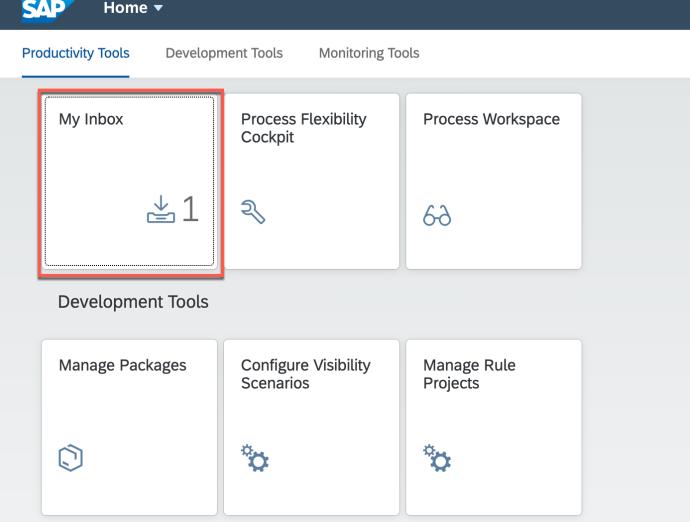
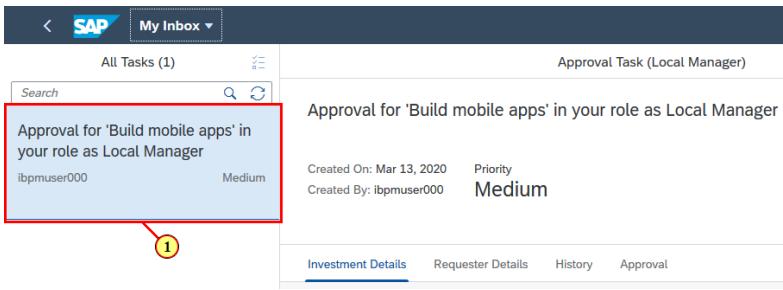
Explanation	Screenshot
<p>186. Ensure that the Build MTA task exits with code 0, indicating the build is completed.</p>	
<p>187. Ensure that a cloud tenant has already been configured. It will be shown in the bottom left corner on a blue bar. If it is not configured yet, then click on the blue bar and 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 mta_archives to open it. 189. Right-click on the created file CapexProject-00_0.0.1.mtar to open the context menu. 190. Select Deploy MTA Archive.</p>	

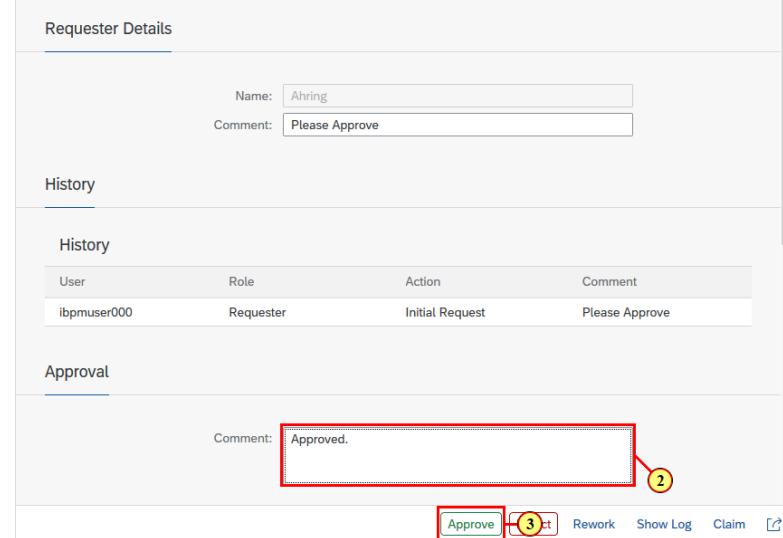
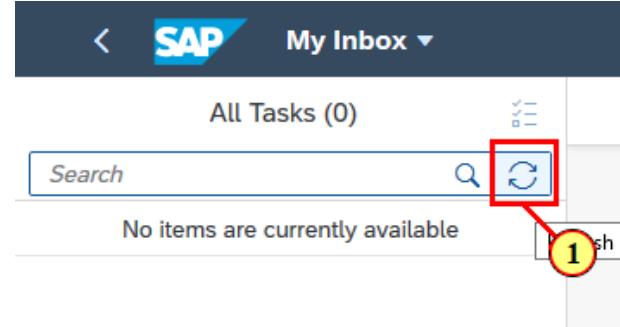
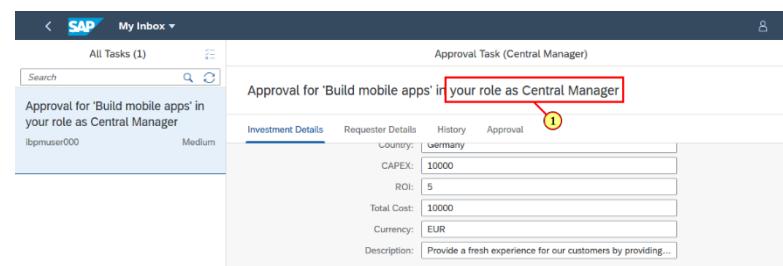
Explanation	Screenshot
<p>191. Ensure that the Deploy MTA Archive task exits with code 0 indicating that the deployment has finished.</p> <p>Note: You can also click on the  icon in the lower right corner to toggle display of the console panels.</p> <p>Note: Your deployment console should look similar to this with the "Process finished" 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"> Task 'Deploy MTA Archive (MTA)' has exited with code 0. Task 'Deploy MTA Archive (MTA)' has been started. <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>

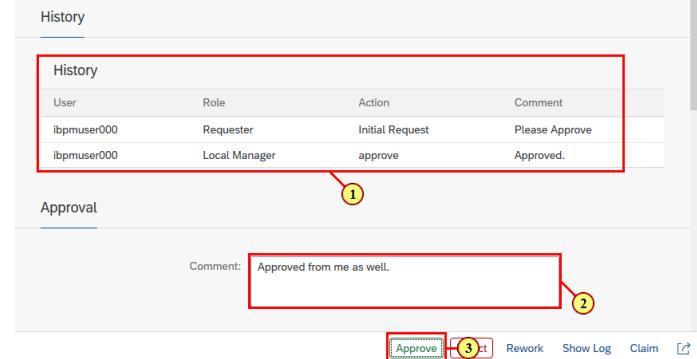
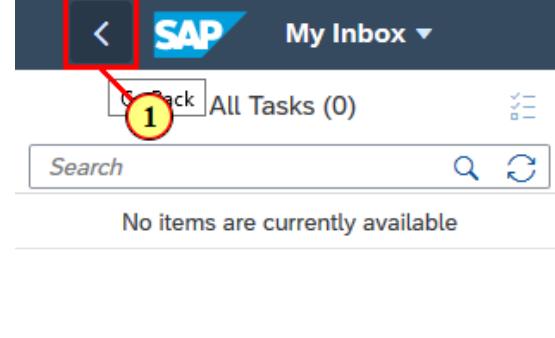
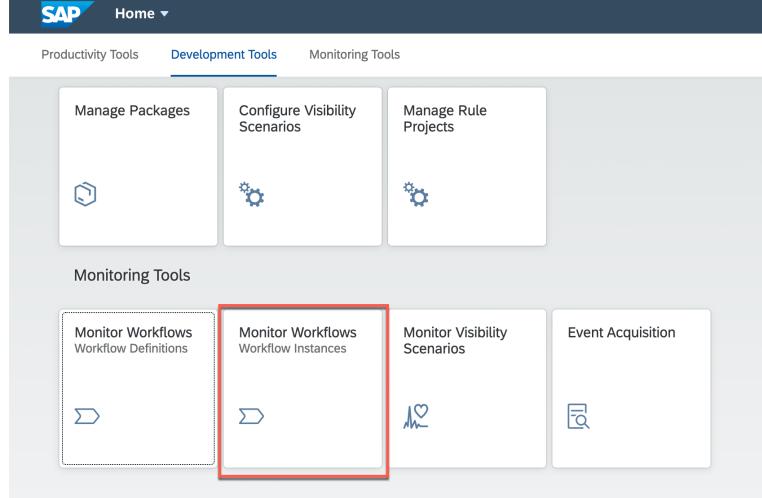
EXECUTE AND MONITOR WORKFLOW

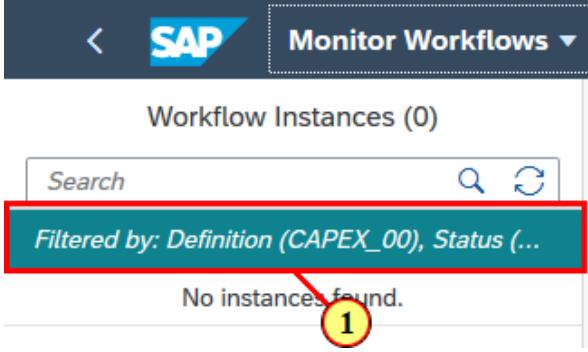
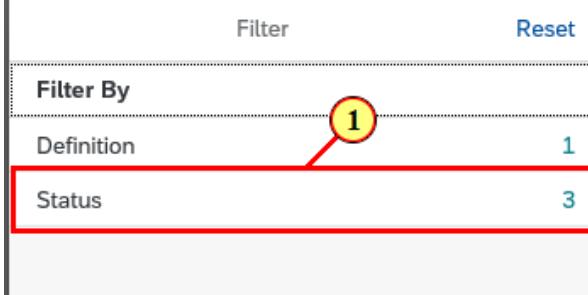
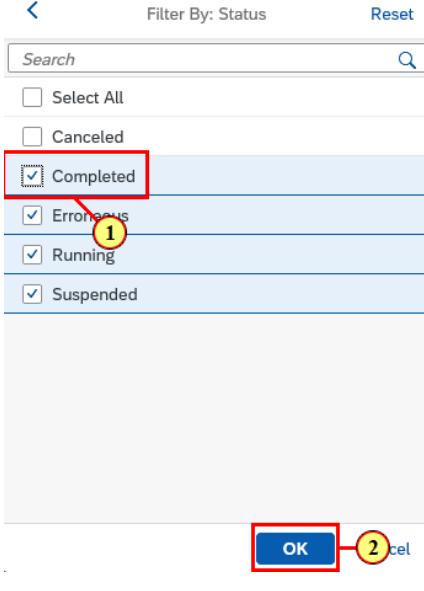
Explanation	Screenshot
<ol style="list-style-type: none"> Open the Workflow Management Launchpad Log On with your trial username and password. Select the tile Monitor Workflows - Workflow Definitions. 	 <p>The screenshot shows the SAP Cloud Platform Workflow Management Launchpad. The navigation bar at 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"> Manage Packages Configure Visibility Scenarios Manage Rule Projects Monitor Workflows Workflow Definitions (this tile is highlighted with a red border) Monitor Workflows Workflow Instances Monitor Visibility Scenarios Event Acquisition

Explanation	Screenshot
<p>Note: Monitor Workflows application has Master-Detail layout where you see all the workflows deployed on left. Ensure that the workflow you just deployed - CAPEX_00 - is existing.</p> <p>4. Select the workflow.</p>	
<p>5. Select Start New Instance.</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 Workflow API.</p>	
<p>6. Modify the RequestId 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 Requester:</p> <ul style="list-style-type: none"> ○ Name can be anything ○ Email should be your trial user email ID ○ UserId is your trial logon user id <p>8. Select Start New Instance.</p>	
<p>A popup confirms that the workflow instance has been started.</p> <p>9. Select Show Instances to switch to the Monitor Workflows - Workflow</p>	

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

Explanation	Screenshot
<p>14. Scroll down to the Approval section.</p> <p>15. Enter a Comment of your choice.</p> <p>16. Click Approve.</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 Refresh .</p>	
<p>Note: You can now see the task for the Central Manager.</p>	

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

Explanation	Screenshot
<p>22. Be default, Monitor Instances application only shows Running, Suspended and Erroneous workflow instance. To see the completed workflow, click on the line showing the currently active Filters or select the Filter icon at the bottom of the screen.</p>	
<p>23. In the popup, select Status.</p>	
<p>24. Select Completed to also show already completed workflow instances.</p> <p>25. Select OK 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 is 7c2b802e-653c-11ea-9474-e00a0bf99ee, the definition ID is capex_00, and the version is 1. It was started by ibpmuser000 on March 13, 2020, at 4:08:24 PM, and completed on the same day at 4:12:35 PM. The business key is CAPEX_REQ_00_001. The 'Workflow Information' tab is selected, showing the JSON representation of the workflow definition:</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: "User": "ibpmuser000", 15: "WorkflowResult": [16: { 17: "Result": [18: { 19: "Approver": { 20: "cm_required": true 21: } 22: } 23:] 24: } 25:] 26: } </pre>
<p>Congratulations! You completed the Workflow exercise.</p> <p>In this exercise, you learnt how to:</p> <ul style="list-style-type: none"> ○ Model workflow from SAP Business Application Studio. ○ Build & deploy workflow ○ Start workflow instance from Monitor Definitions ○ Track the workflow instances from Monitor Instances ○ Process your work items from My Inbox 	

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

© 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 for additional trademark information and notices.