Hands-on Lab: Deploying Fiori Apps with SAP Joule for Developers

Aleksandrs Antonuks, Vass, aleksandrs.antonuks@vasscompany.com

Michael Pytel, Vass, michael.pytel@vasscompany.com

Sheldon Lipshitz, Vass, sheldon.lipshitz@vasscompany.com

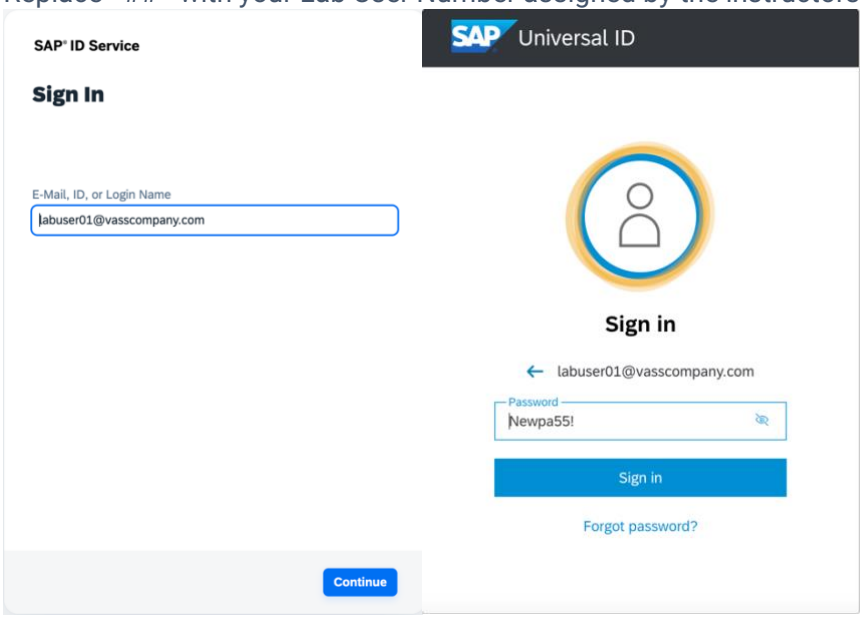
https://vasscompany.com

Index

Table of Contents

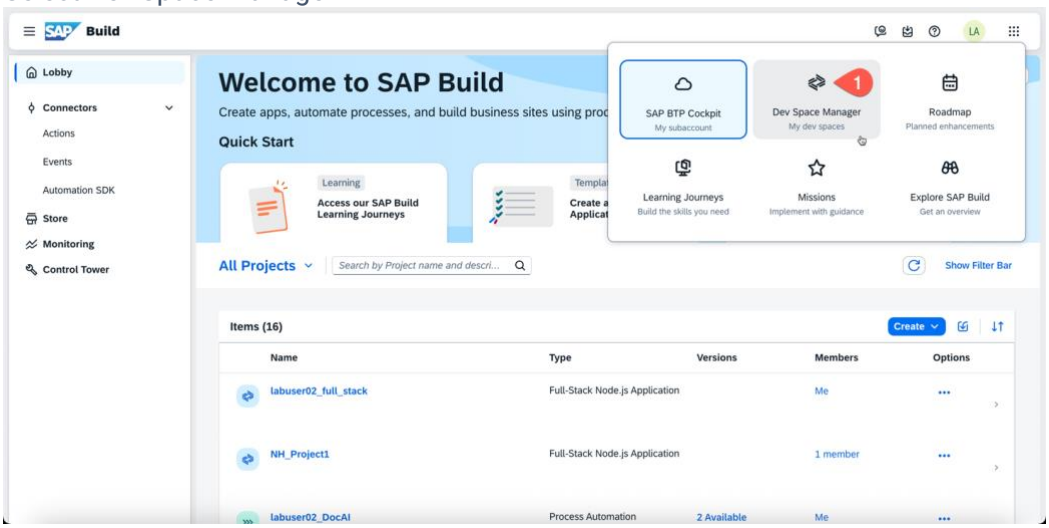
Environment Information

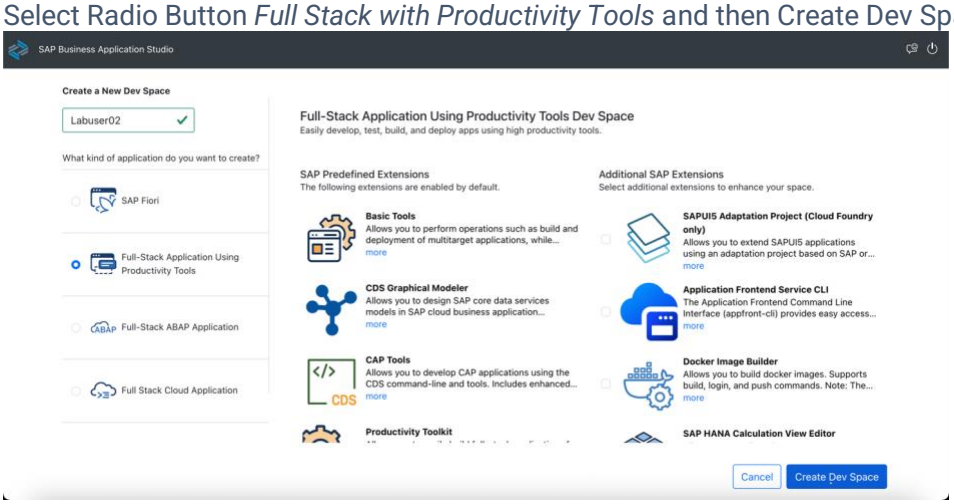
1. Please use GOOGLE CHROME as your browser
2. SAP Build App URLs
   - SAP Build App s
3. Lab User Information
   - User: **labuser<##>@vasscompany.com**
   - Password: Newpa55!
4. Replace <##> with your Lab User Number assigned by the instructors

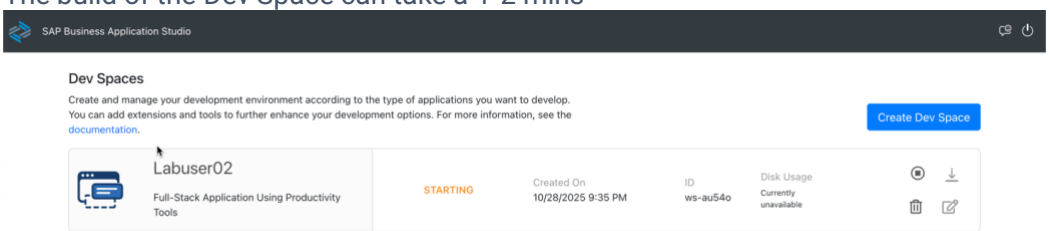


Create your Project in SAP Build

Our first step is to create a workspace for the new development. All prerequisites for SAP Joule for Developers have been deployed and you are ready to start work.

1. Open an Chrome Browser and navigate to SAP Build Apps
2. Enter your Lab UserID and Password
3. On the upper right, click the icon.
4. Select Dev Space Manager
5. 

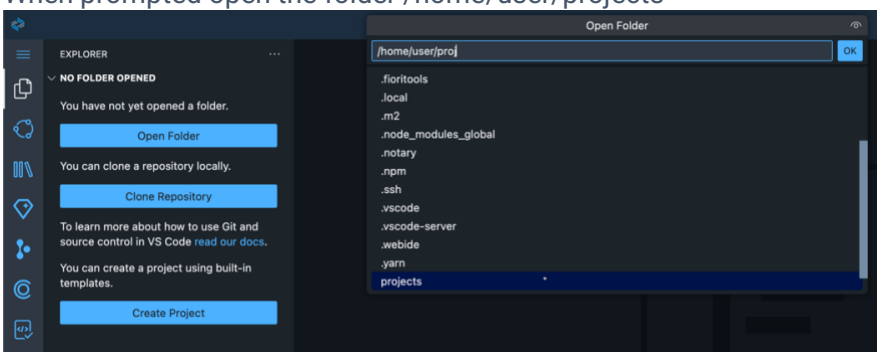
6. Select Create Dev Space
7. End a Name: Labuer##
8. Select Radio Button *Full Stack with Productivity Tools* and then Create Dev Space button.
9. 

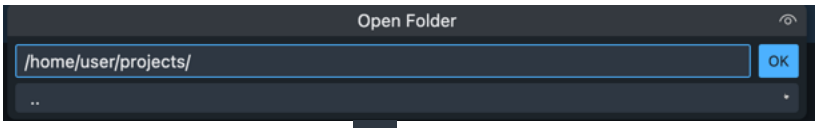
10. The build of the Dev Space can take a 1-2 mins
11. 

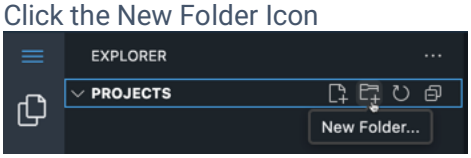
12. When it says *RUNNING* click on the Dev Space
13. Do Not Click on Create Project yet.

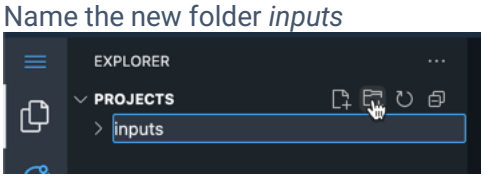
Prepare your prompt and image

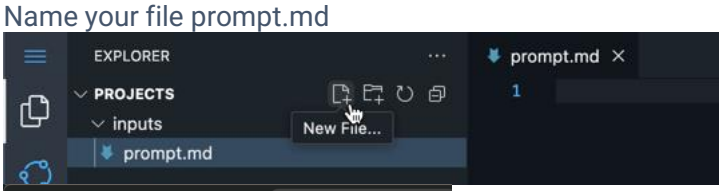
In this next section we're going to upload a previously prepared prompt for Joule and a rough wireframe for an application. This step is very important and requires some navigation at the terminal level to create directories and markdown files.

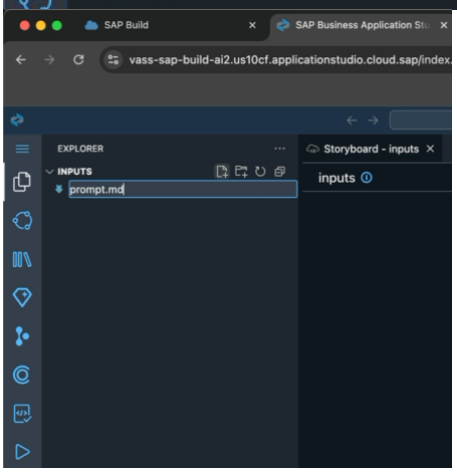
1. On your blank project, click the File Explorer icon  in the upper left.
2. Click on Open Folder
3. When prompted open the folder /home/user/projects
4. 

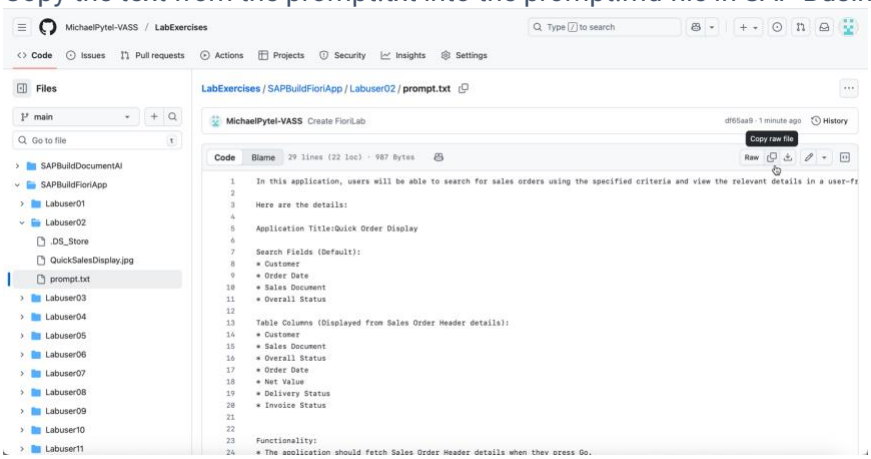
5. Press OK
6. 

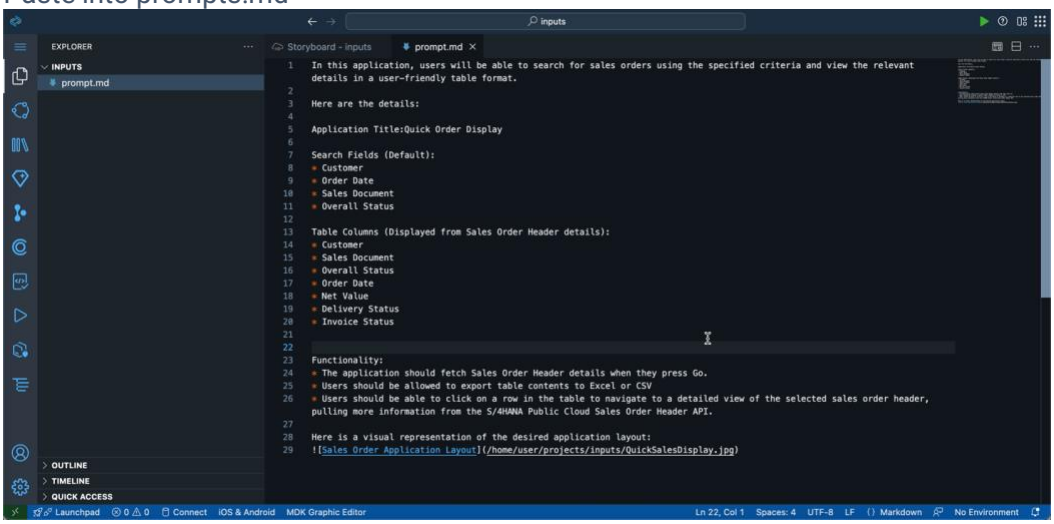
7. Select the File Explorer icon  again
8. Click the New Folder Icon
9. 

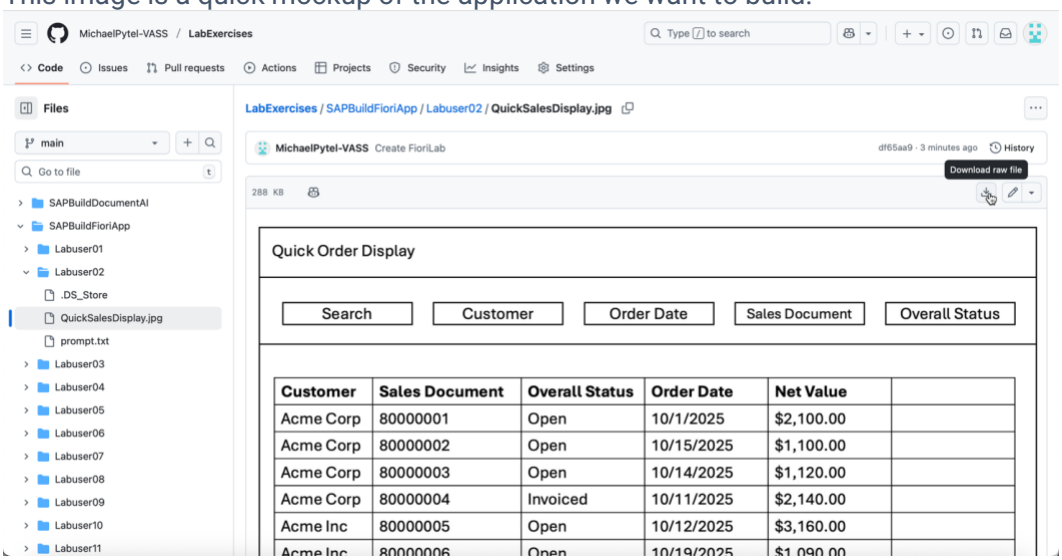
10. Name the new folder *inputs*
11. 

12. Open the folder, its empty, but the icon will change to down position
13. Click Add File button
14. Name your file prompt.md
15. 

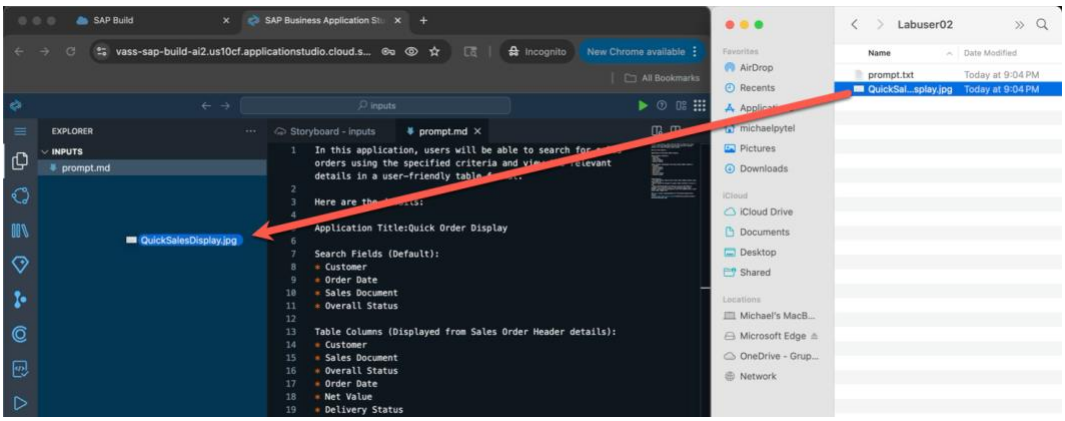
16. 

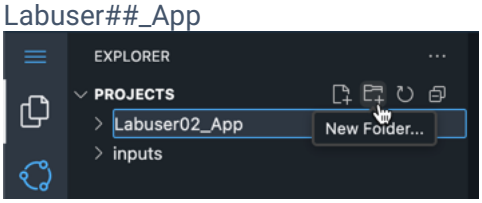
17. Click on your newly created markdown file
18. Navigate to Github (https://github.com/MichaelPytel-VASS/LabExercises/tree/main/SAPBuildFioriApp)
19. Select your Labuser## Folder
20. Copy the text from the prompt.txt into the prompt.md file in SAP Business Application Studio
21. 

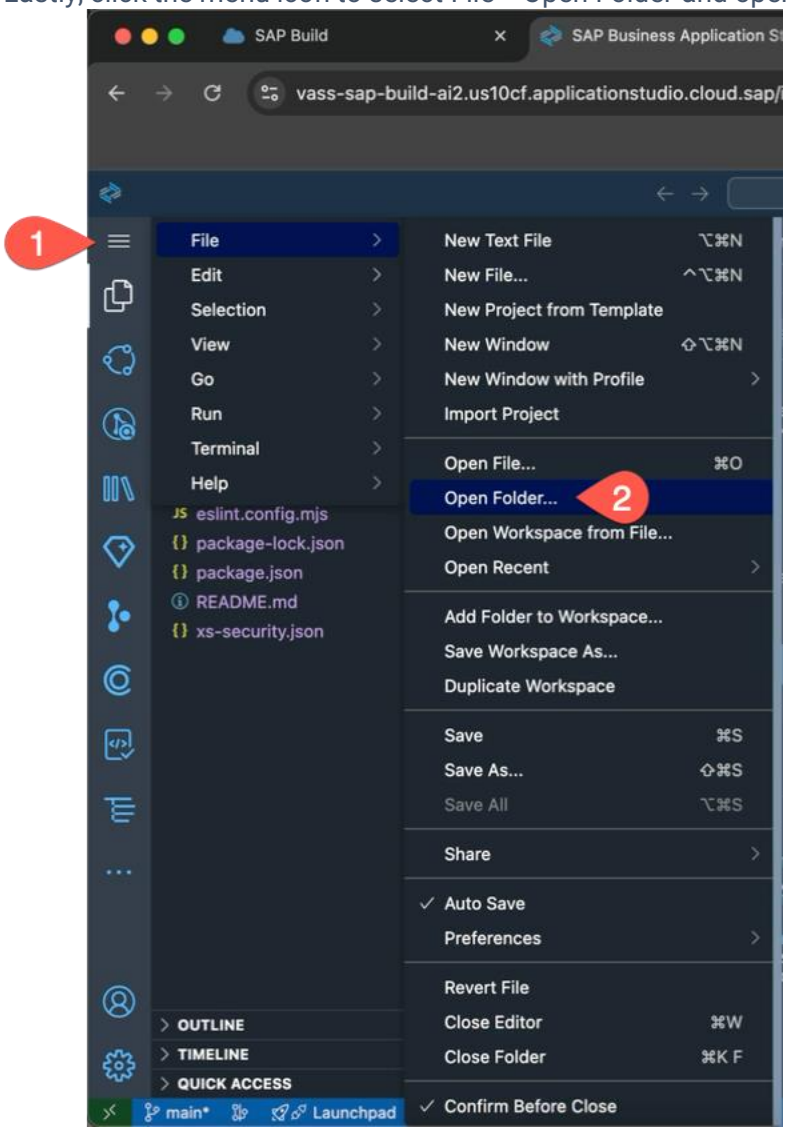
22. Paste into prompts.md
23. 

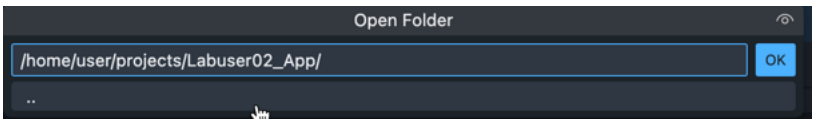
24. The file is saved automatically
25. Next, from Github, download the image in your Labuser## folder
26. This image is a quick mockup of the application we want to build.
27. 

28. Next, using your mouse, drag the image from a local folder to the Chrome Browser with SAP Business Application Studio. Drop the image on the folder 'inputs' we created above.

29. 

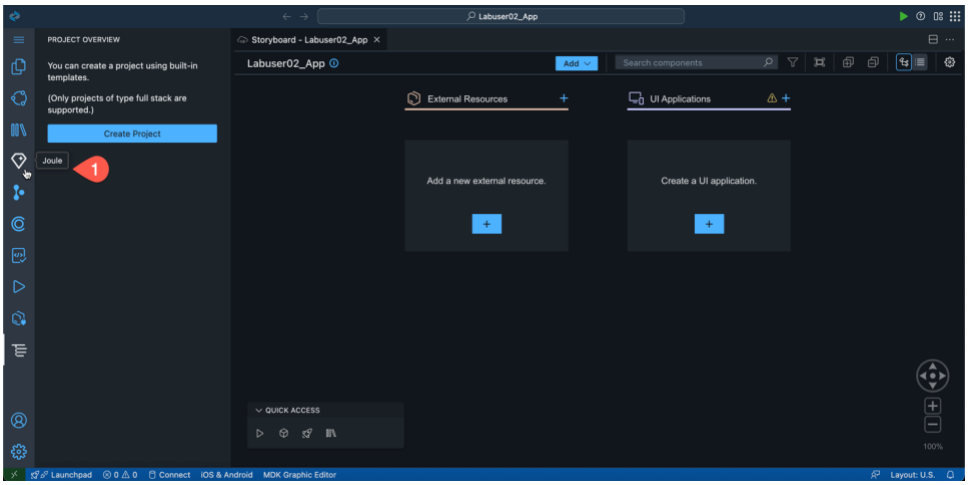
30. The last line in your prompt.md file should read like this
31. ![Sales Order Application Layout](/home/user/projects/inputs/QuickSalesDisplay.jpg)
32. From the File Explorer, you should have an *inputs* folder with two files.
33. Next, we're going to create a new Project Folder
34. From the Projects Folder, click the New Folder button and create a new folder named Labuser##_App
35. 

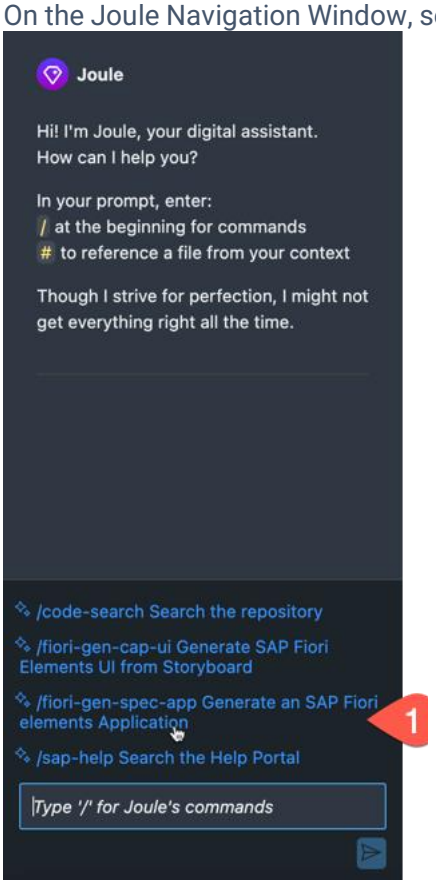
36. Lastly, click the menu icon to select File > Open Folder and open the newly created app folder.
37. 

38. 

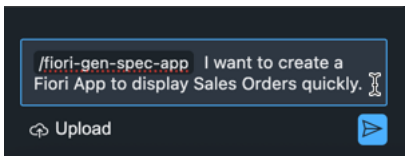
39. Click OK

Prepare your prompt and image

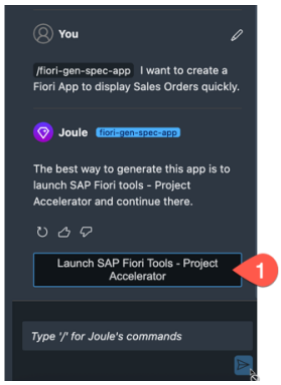
1. Within your empty folder create above.
2. Select the Joule Icon on the left.
3. 

4. On the Joule Navigation Window, select /Fiori-gen-spec-app
5. 

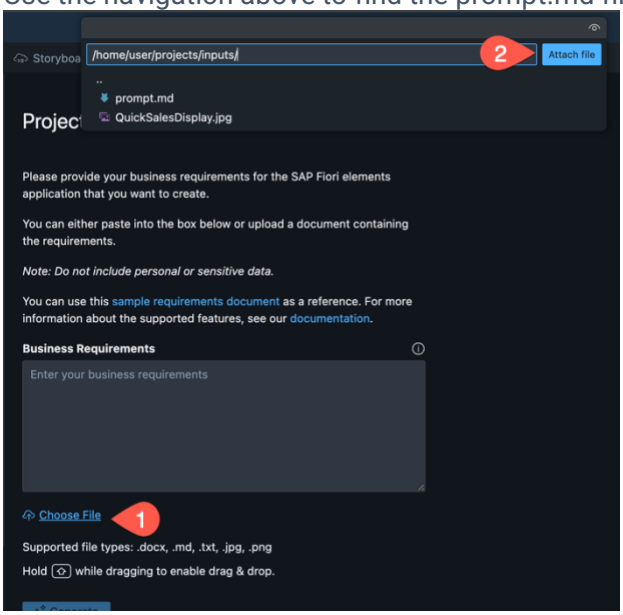
6. Enter the prompt below
    a. Text: */fiori-gen-spec-app I want to create a Fiori App to display Sales Orders quickly.*

b. 

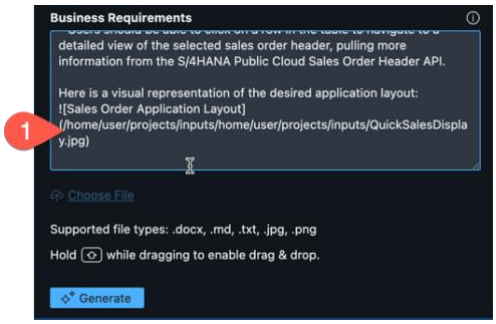
7. Send the command
8. The select the button for 'Launch SAP Fiori Tools – Project Accelerator'

9. 

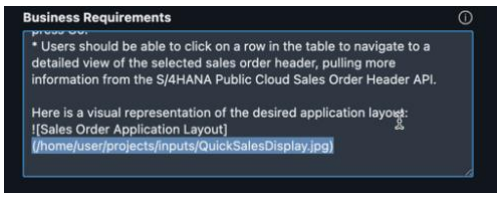
10. On the next screen, remove the text from the Business Requirements box.
11. Instead, we will use our prompt.md file we created before.
12. Click *Choose File*
13. Use the navigation above to find the prompt.md file we created.

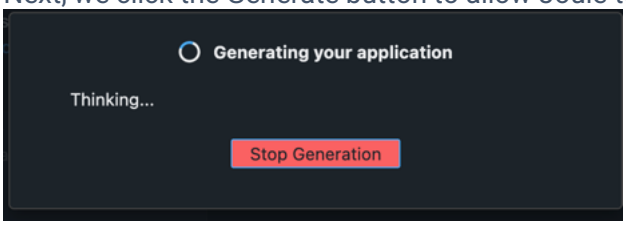
14. 

15. The file and text will be added.
16. We've seen a bug where the file path is not correct when importing the markdown file. Scroll to the bottom of the requirements and correct the path to the image.
17. Before

a. 

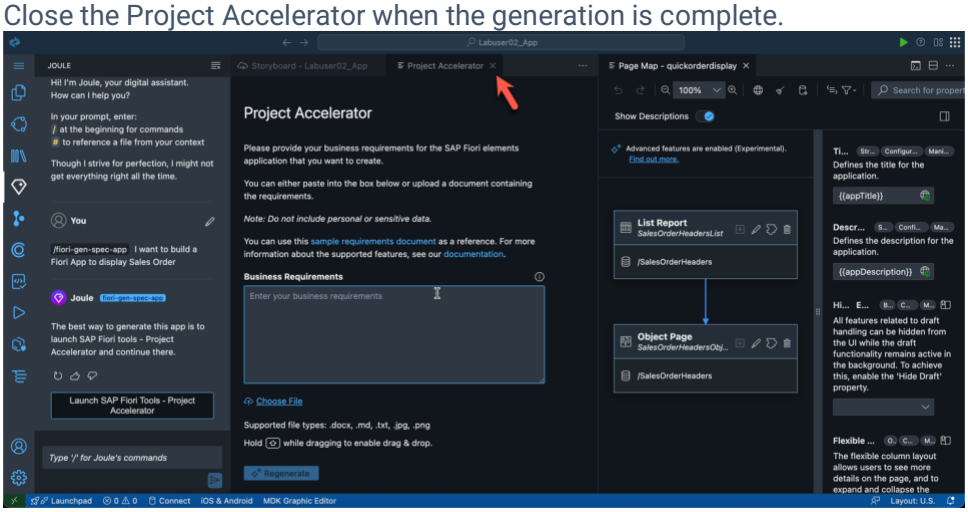
18. After
    a. (/home/user/projects/inputs/QuickSalesDisplay.jpg)

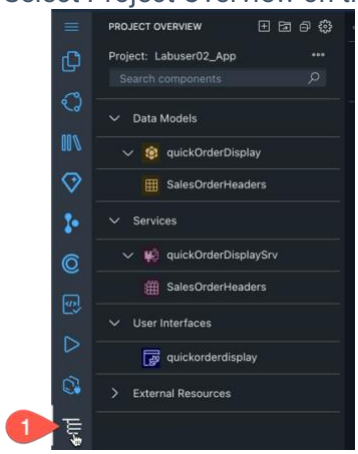
b. 

19. Next, we click the Generate button to allow Joule to create the app for us.
20. 

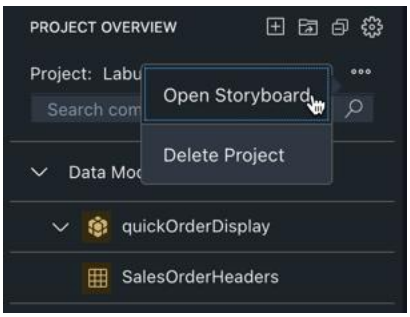
21. This process can take 3-5 minutes.

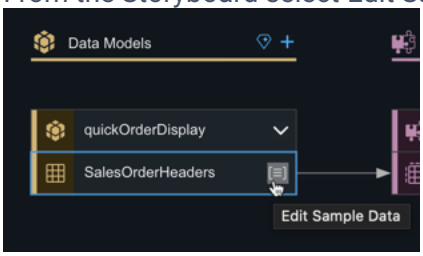
Generate Test Data & Demo your App

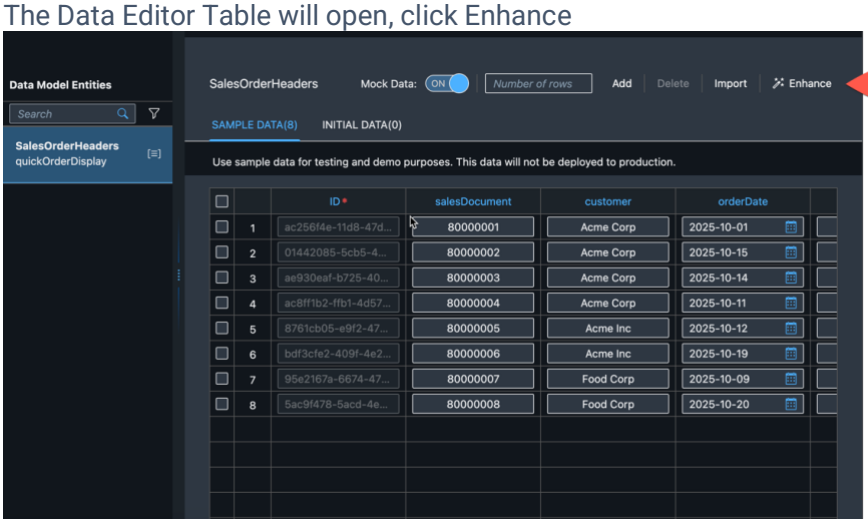
1. Close the Project Accelerator when the generation is complete.
2. 

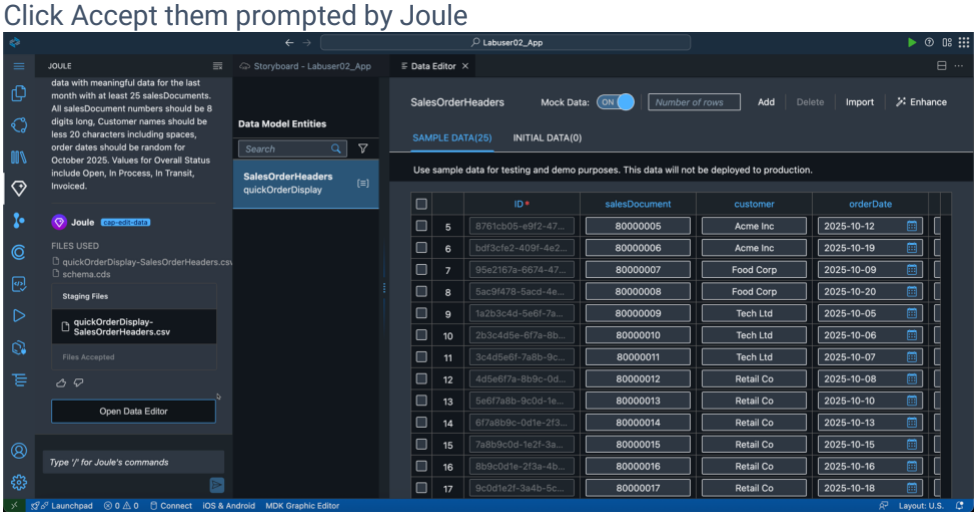
3. Click the Storyboard Tab
4. Select Project Overview on the lower left
5. 

6. From the Project Overview click the three dots and select Open Story Board

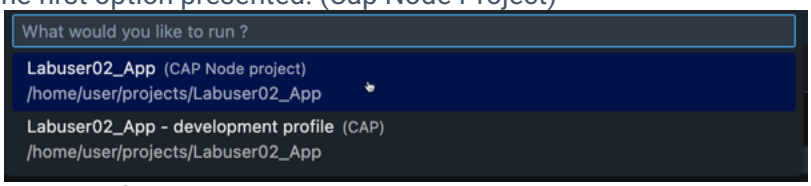
7. 

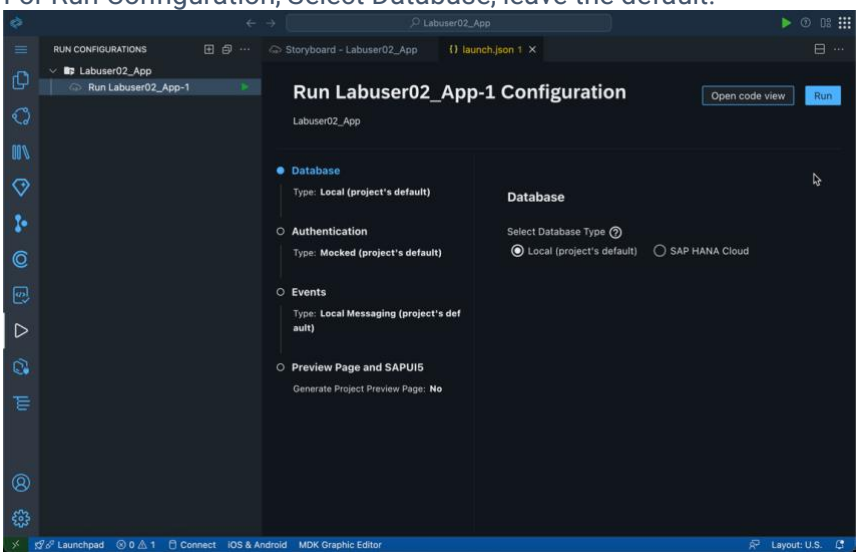
8. From the Storyboard select Edit Sample Data Button from the Data Model SalesOrderHeaders
9. 

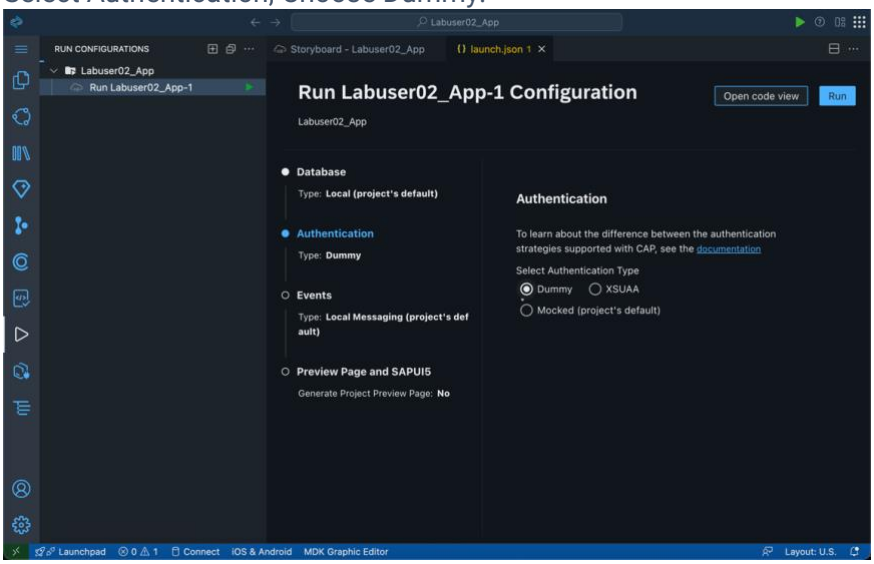
10. The Data Editor Table will open, click Enhance
11. 

12. Edit the Joule prompt with the following.
    1. Text: /cap-edit-data #test/data/quickOrderDisplay-SalesOrderHeaders.csv Create sample data with meaningful data for the last month with at least 25 salesDocuments. All salesDocument numbers should be 8 digits long, Customer names should be less 20 characters including spaces, order dates should be random for October 2025. Values for Overall Status include Open, In Process, In Transit, Invoiced.
13. Click Accept them prompted by Joule
14. 

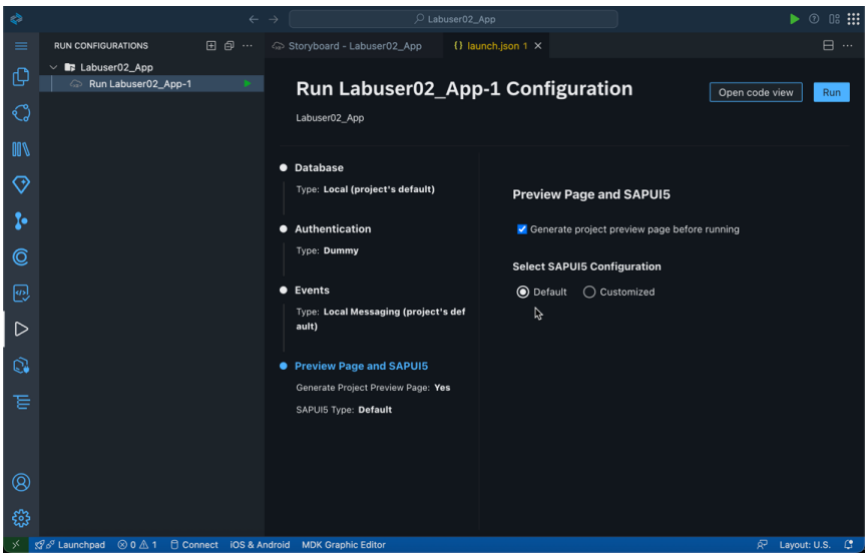
15. Confirm more data was added.
16. Close the Data Editor
17. Click the *Run Confiugration* button on the left side

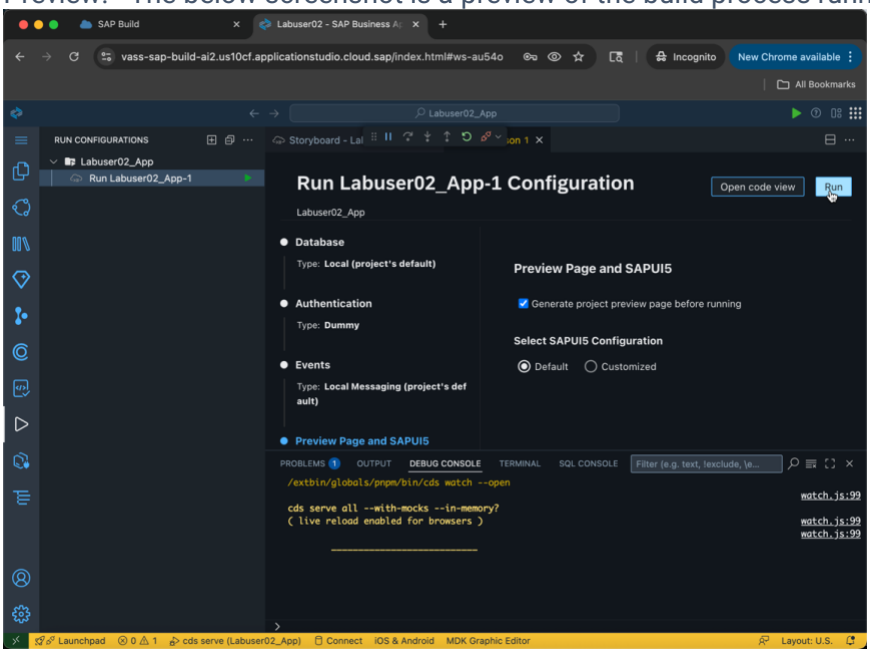
18. Click Create Configuration
19. Select the first option presented. (Cap Node Project)
    1. 

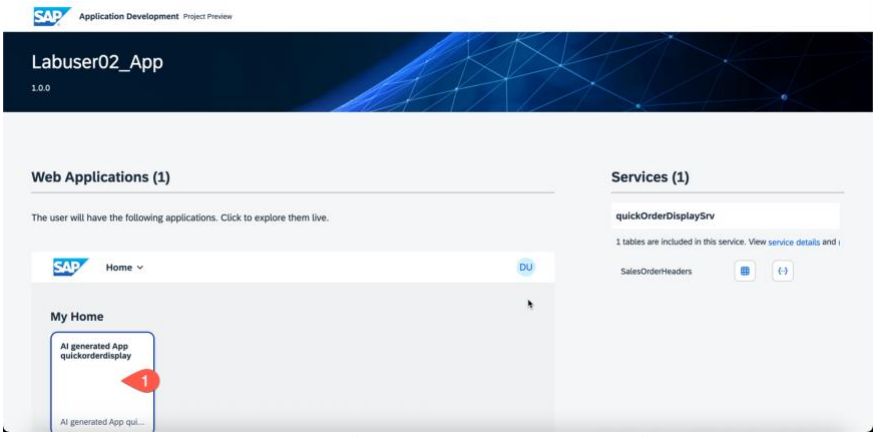
20. It will create a default name, click Enter to accept.
21. For Run Configuration, Select Database, leave the default.
22. 

23. Select Authentication, Choose Dummy.
24. 

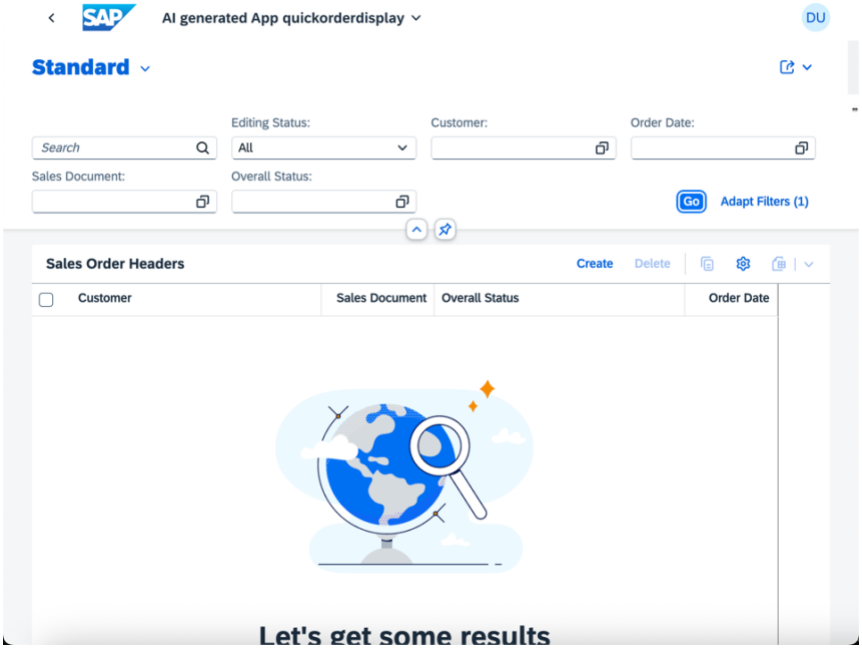
25. Leave Events as defaults
26. Select PreviewPage and SAPUI5, choose ‘Generate project preview page before running’

27. 

28. Click RUN
29. The first run can take 3-5 mins to process. A new browser tab will open with a Fiori App Preview. The below screenshot is a preview of the build process running.
30. 

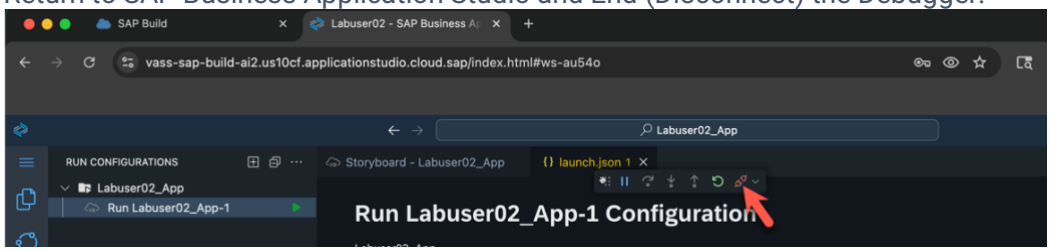
31. The Fiori Page Preview will open. Select your App Tile
32. 

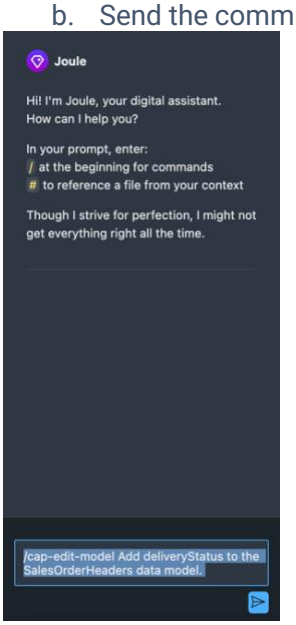
33. Click Go to display Records (test data stored locally)

34. 

35. Verify the App displays the mock data

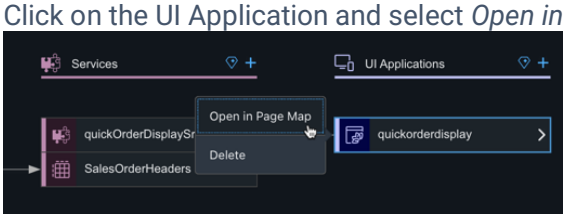
Edit your App with Joule

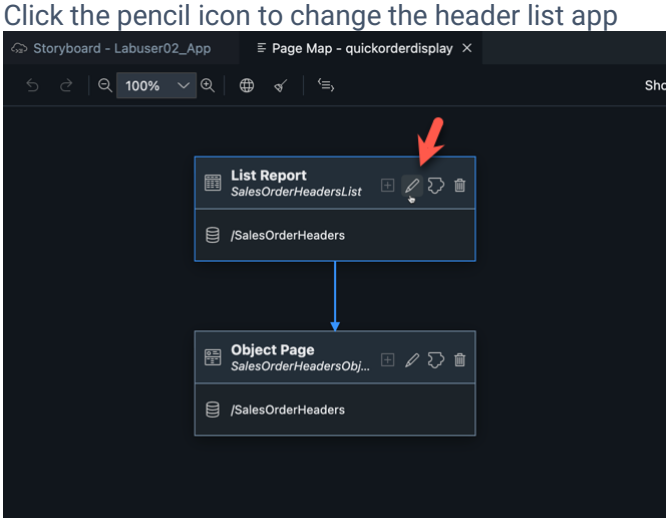
1. Once you're done with the app preview, close the browser tab.
2. Return to SAP Business Application Studio and End (Disconnect) the Debugger.
3. 

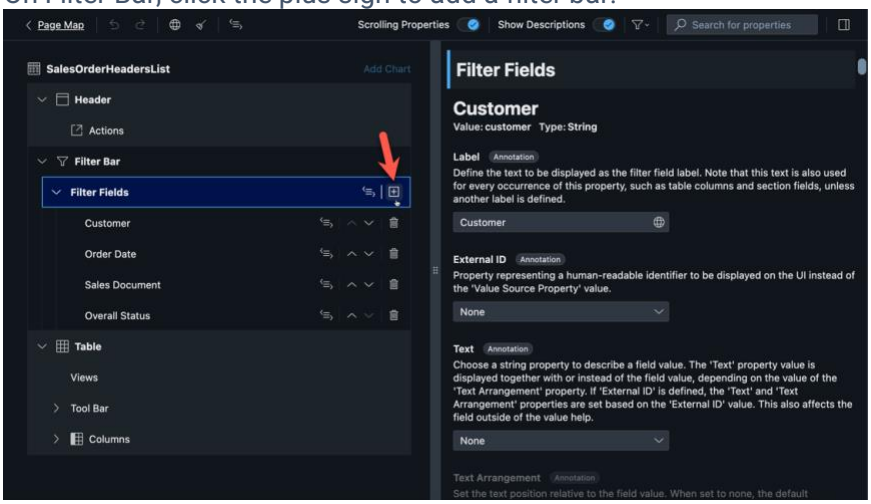
4. Return to the Joule menu on the left
5. Enter in the following text
    a. /cap-edit-model Add deliveryStatus to the SalesOrderHeaders data model.
    b. Send the command to Joule
6. 

7. Joule will recommend updates, click Accept

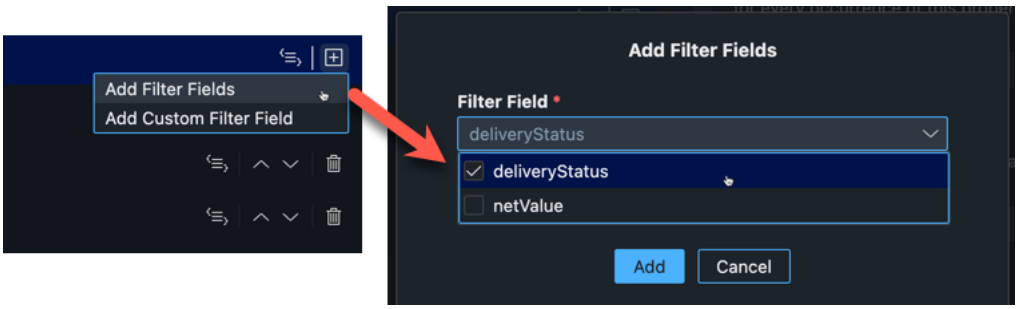
8. 

9. Joule has updated the Data Model and Service with the new deliveryStatus field.
10. Lets add this field to the Fiori Application
11. Click on the UI Application and select *Open in Page Map*
12. 

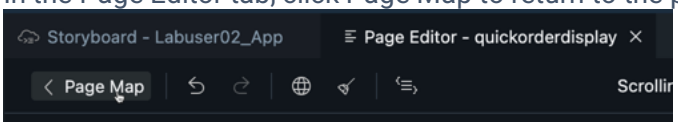
13. Click the pencil icon to change the header list app
14. 

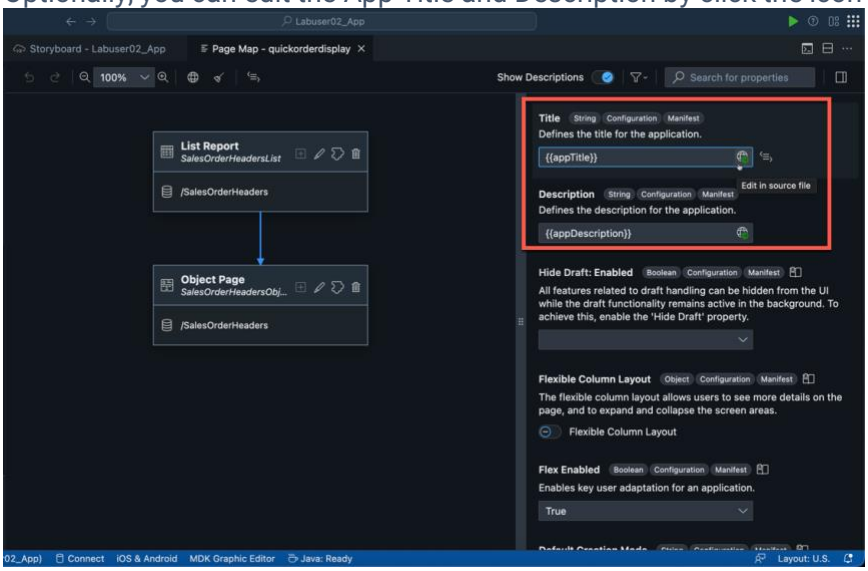
15. On Filter Bar, click the plus sign to add a filter bar.
16. 

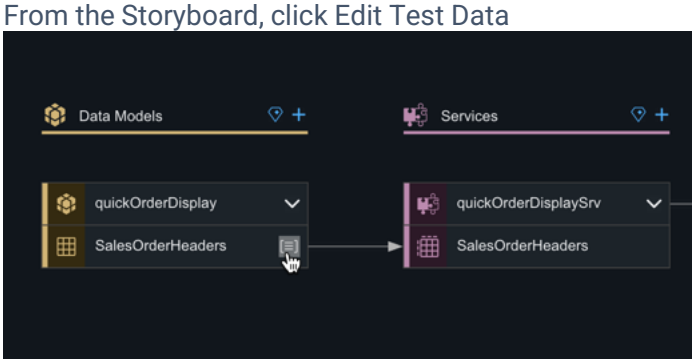
17. Select *Add Filter Fields* then select *deliveryStatus*



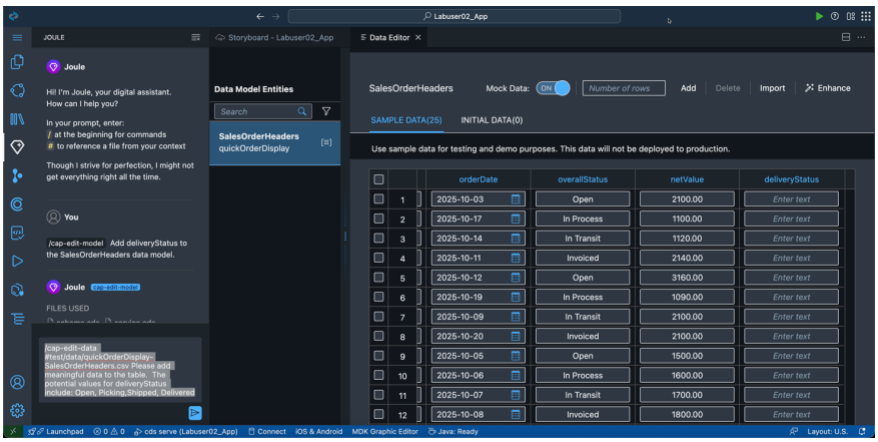
18.
19. Click Add
20. Repeat the Step under  to add the column to the table.

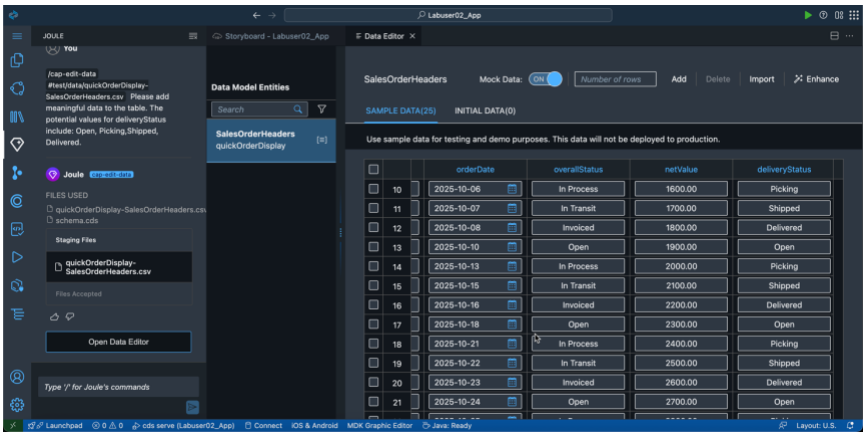
21. In the Page Editor tab, click Page Map to return to the prior screen.
22. 

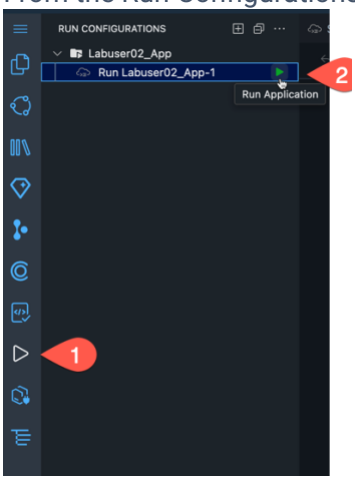
23. Optionally, you can edit the App Title and Description by click the icon to edit.
24. 

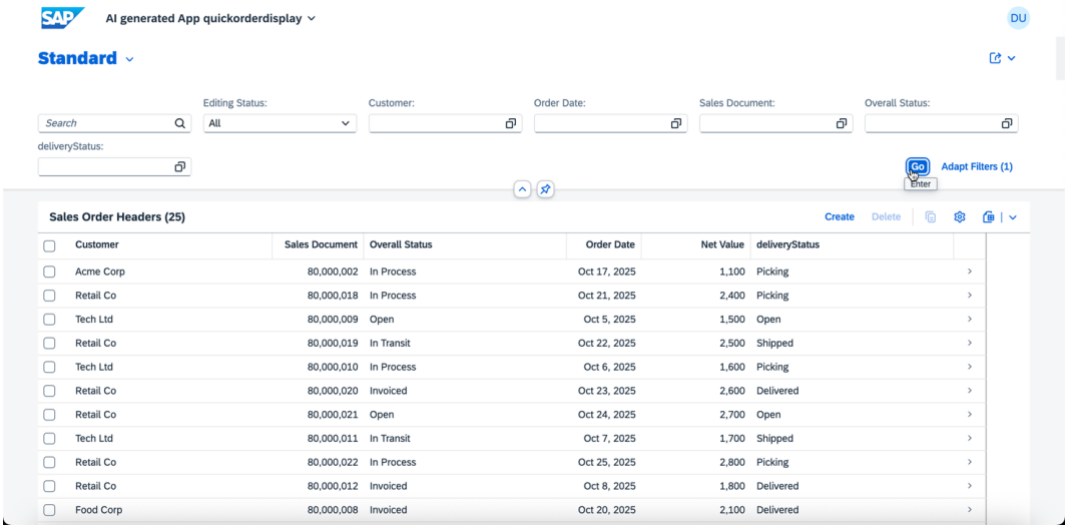
25. Next, we will regenerate test data for the new field.
26. From the Storyboard, click Edit Test Data
27. 

28. Click the Enhance Button

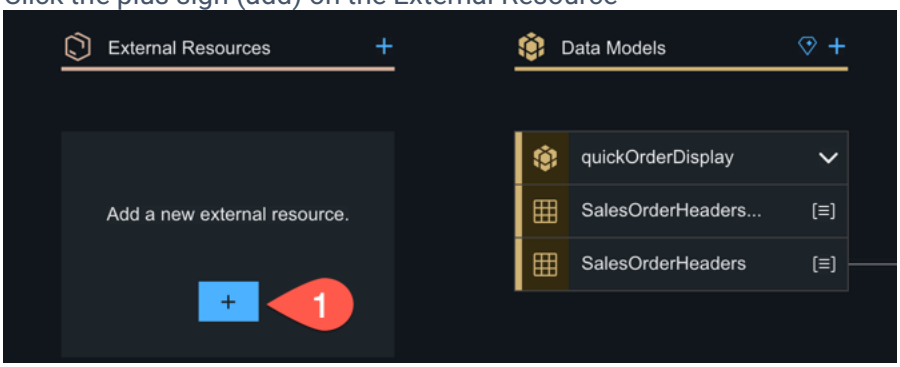
29. 

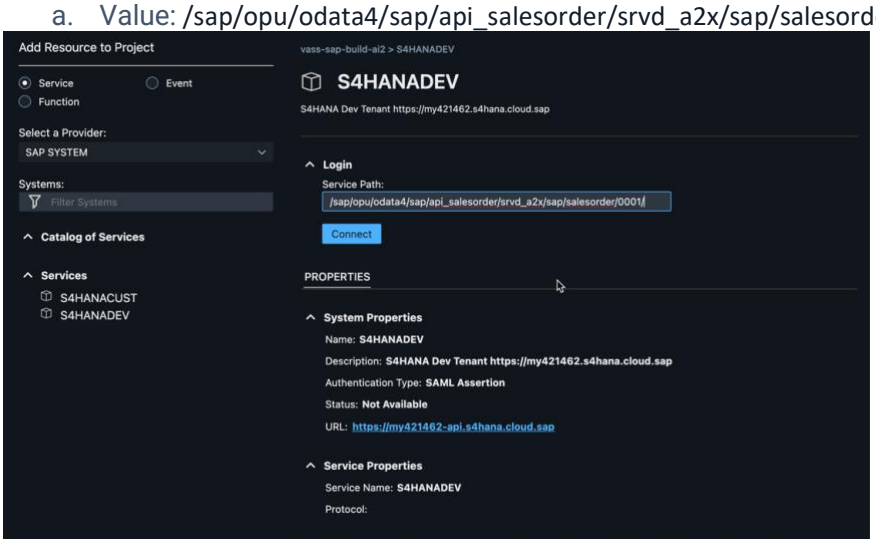
30. Enter the following tyext prompt
    a. Text: /cap-edit-data #test/data/quickOrderDisplay-SalesOrderHeaders.csv Please add meaningful data to the table. The potential values for deliveryStatus include: Open, Picking,Shipped, Delivered.
    b. Send the command
    c. Accept updates from Joule
31. Confirm Test Data
32. 

33. Next, we will test our app again to confirm the field was added.
34. From the Run Configurations Menu, select the Run button on the config we modified above.
35. 

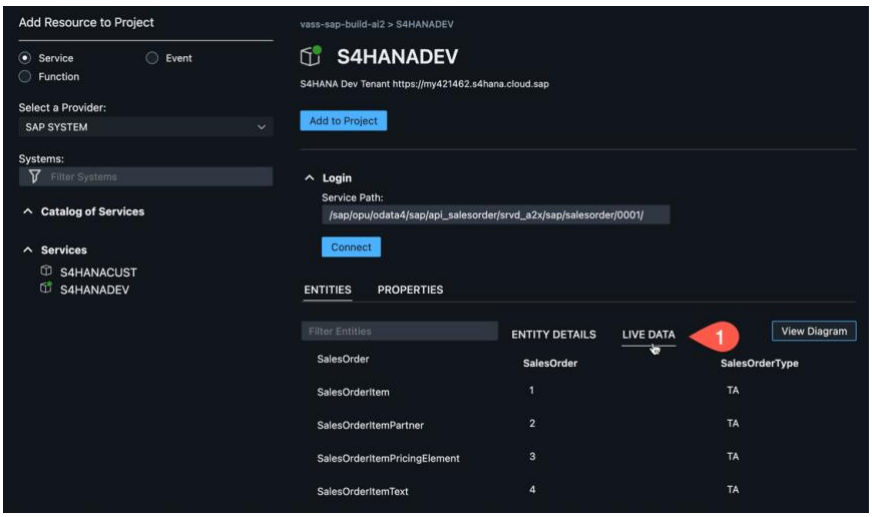
36. 

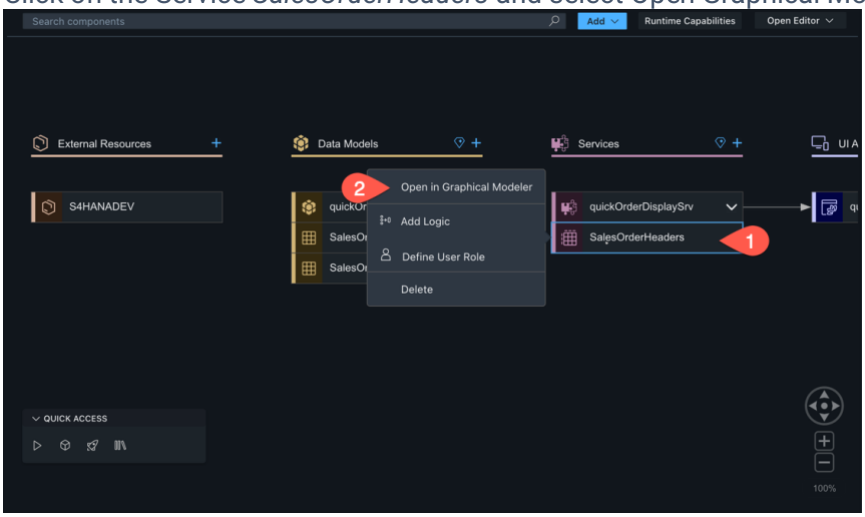
Add S/4HANA Live Connection to Your App

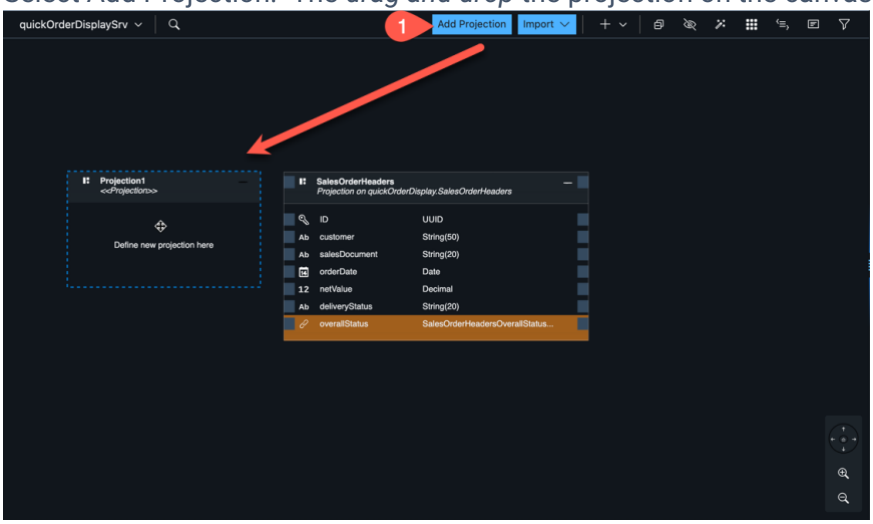
1. Return to the Story Board for your application.
2. Click the plus sign (add) on the External Resource
3. 

4. Select the System S4HANADEV
5. Under Service Path enter the following value. Then click Connect
    a. Value: /sap/opu/odata4/sap/api_salesorder/srvd_a2x/sap/salesorder/0001/
6. 

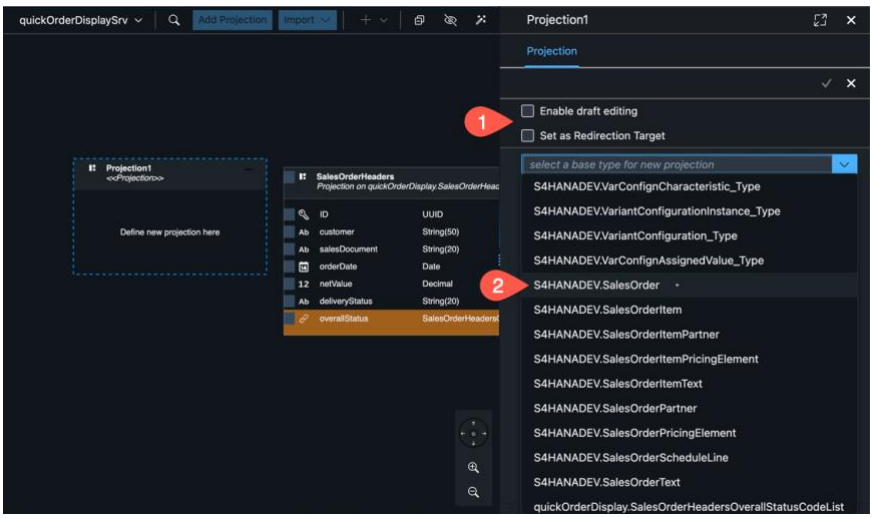
7. Select the *Live Data* Tab *to* browse data from S/4HANA

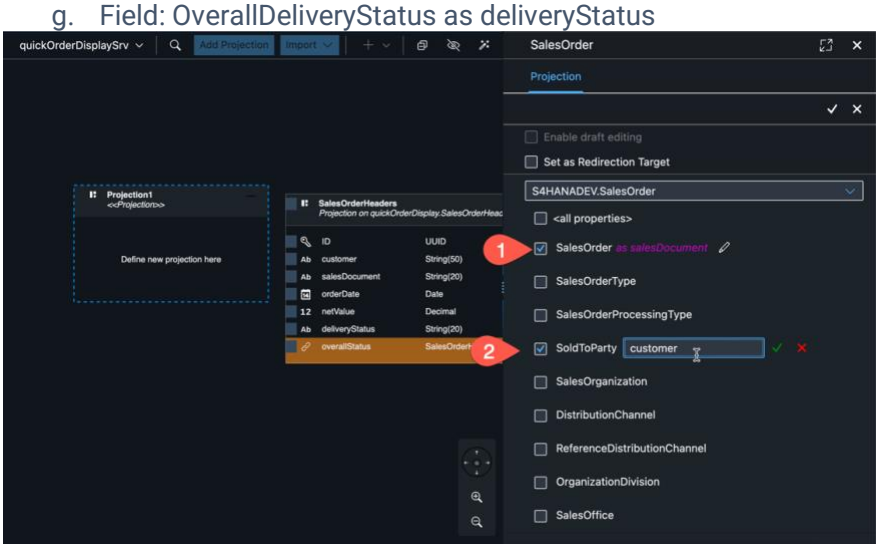
8. 

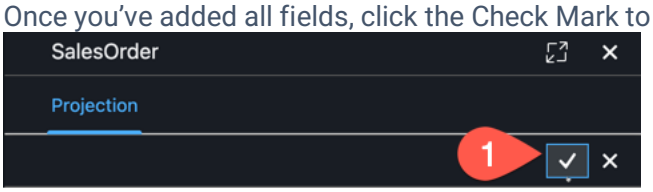
9. Click Add to Project.
10. Return to the Story Board once the External Resource has been added
11. Click on the Service *SalesOrderHeaders* and select Open Graphical Modeler
12. 

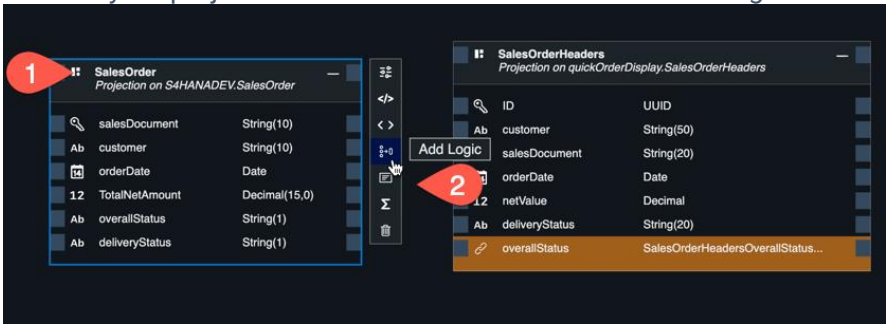
13. Select Add Projection. The *drag and drop* the projection on the canvas.
14. 

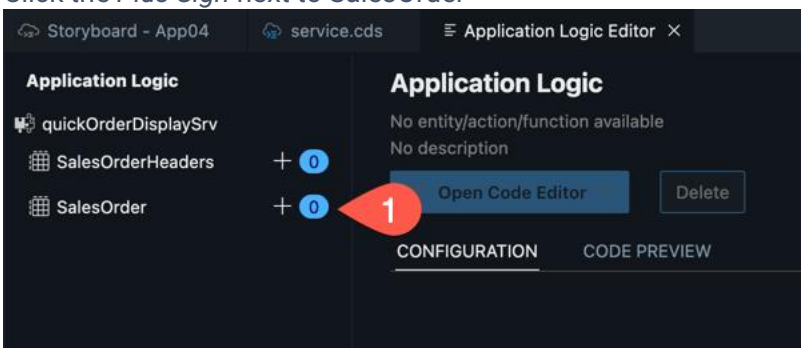
15. Uncheck boxes
    a. Unchecked: Enable draft editing
    b. Unchecked: Set as Redirection Target
16. Select S4HANADEV.SalesOrder

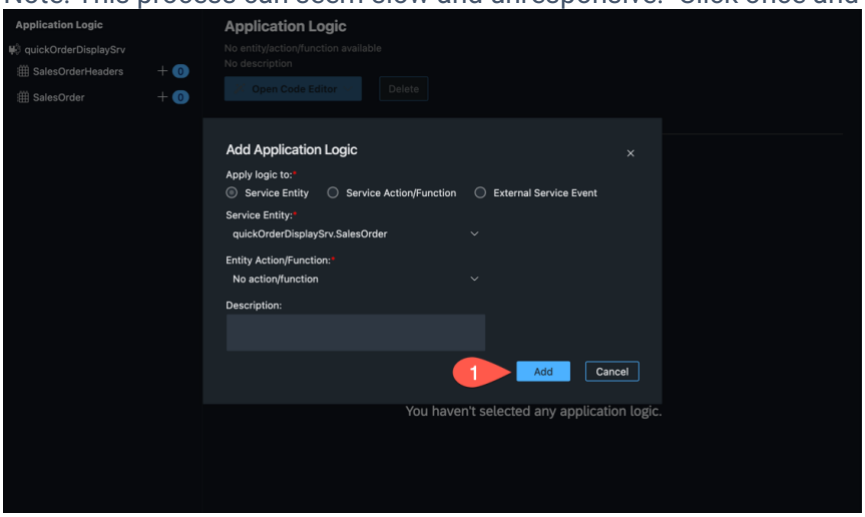
17. 

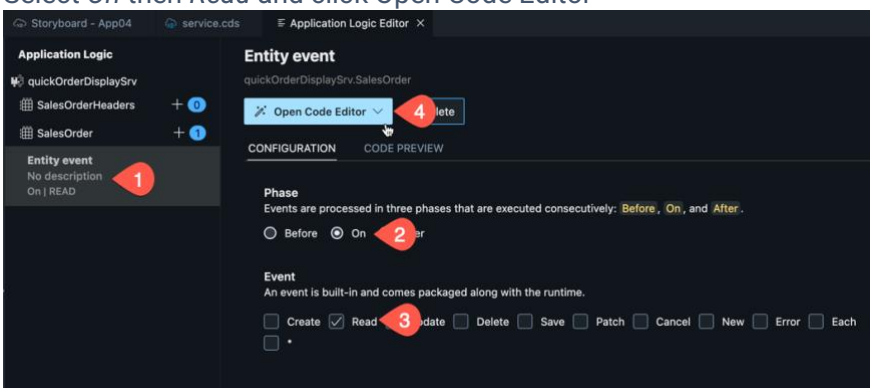
18. Next, we're going to select the Fields from the API we want to consume.
19. We're looking at SalesOrderHeader only in this lab.
20. Select the following fields, click the field rename pencil, and adapt as shown below.
    a. Uncheck: <all properties>
    b. Field: SalesOrder as salesDocument
    c. Field: SoldToParty as customer
    d. Field: CreationDate as orderDate
    e. Field: TotalNetAmount as netValue
    f. Field: OverallSDProcessStatus as overallStatus
    g. Field: OverallDeliveryStatus as deliveryStatus
21. 

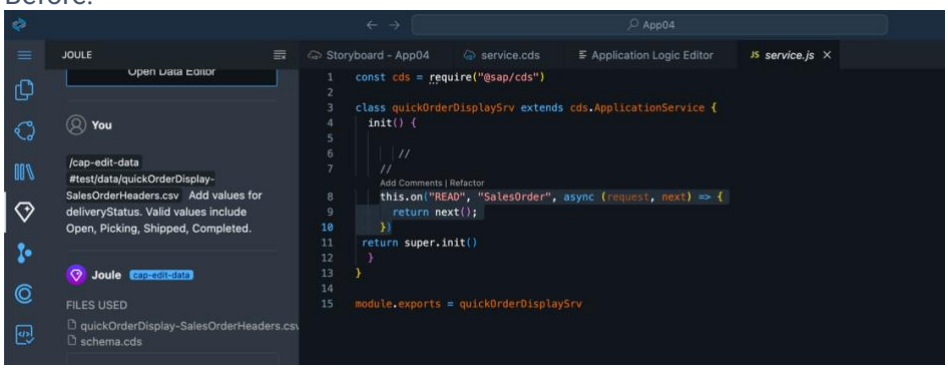
22. Once you've added all fields, click the Check Mark to confirm
23. 

24. Click the Properties Sheet on the right.
25. Click on your projection and select the small icon for Add Logic
26. 

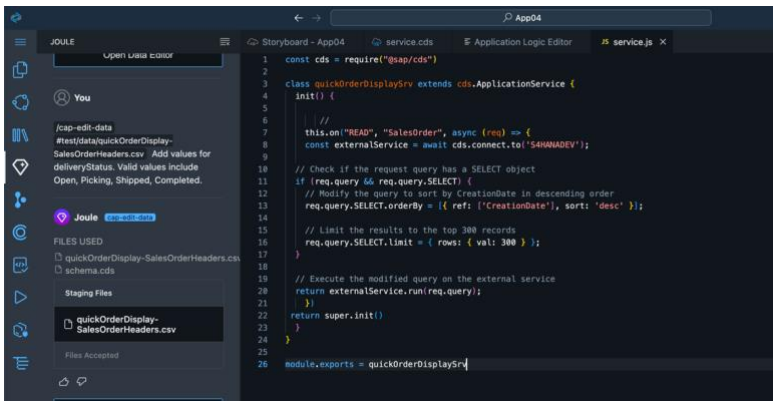
27. Click the *Plus Sign* next to SalesOrder
28. 

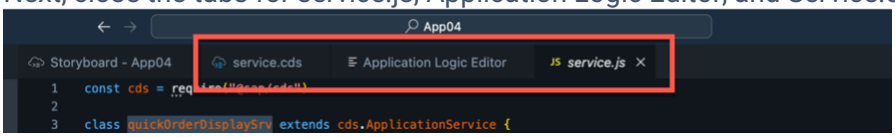
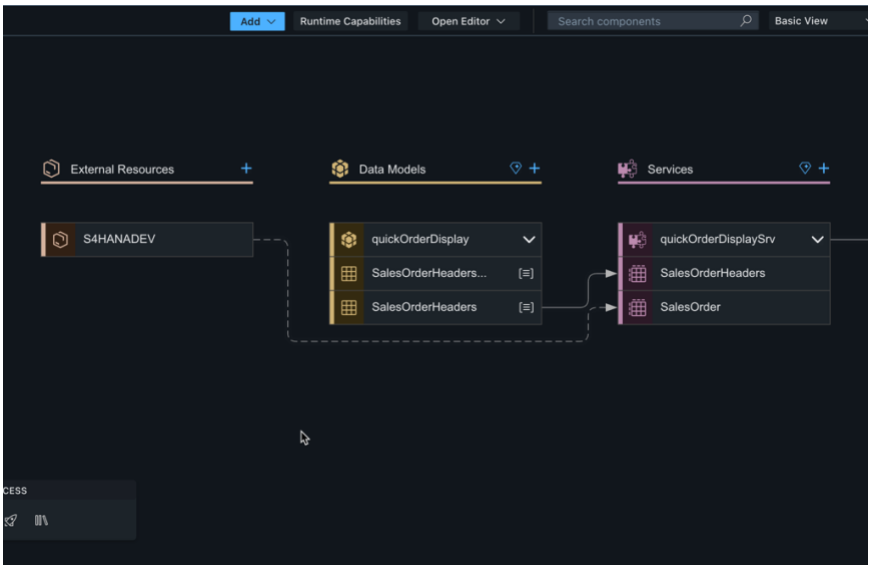
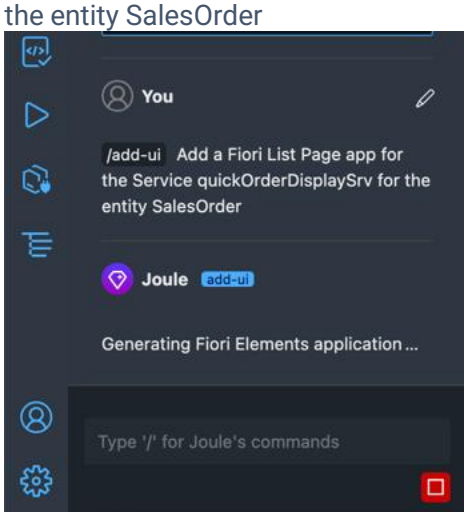
29. Accept the defaults shown and click Add
30. Note: This process can seem slow and unresponsive. Click once and wait 60secs.
31. 

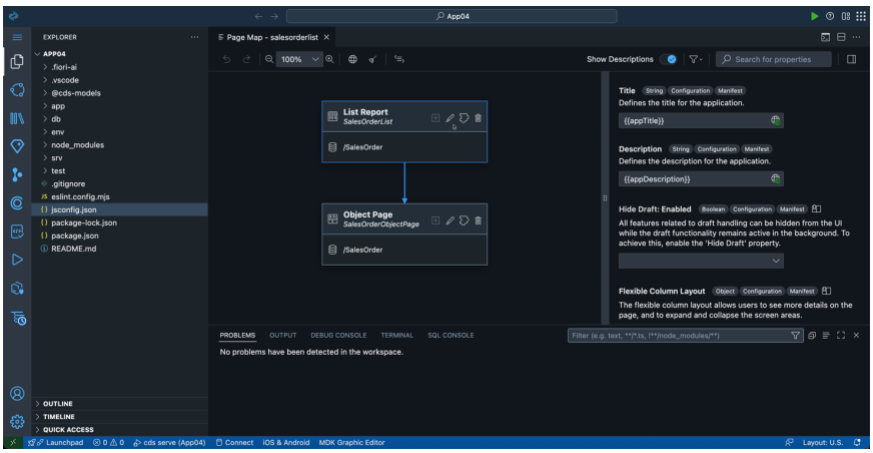
32. Once added, select the Entity Event on the left.
33. Select *On* then *Read* and click Open Code Editor
34. 

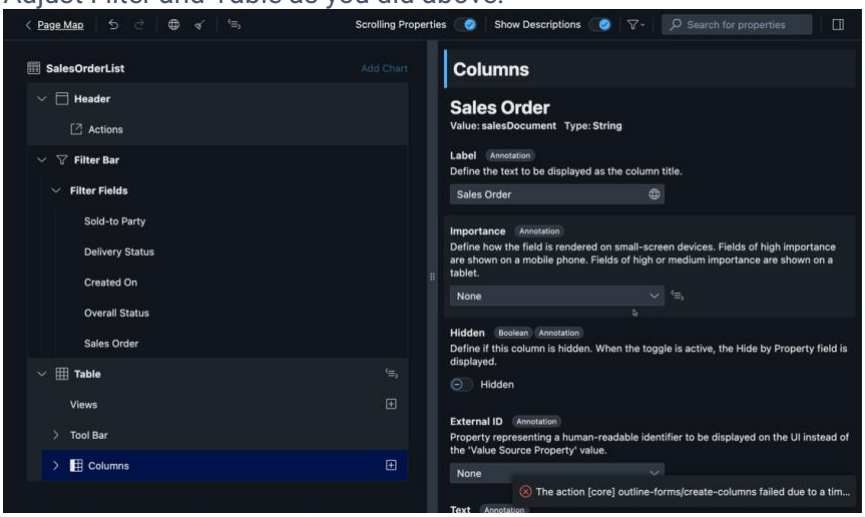
35. Paste in the code block from the Service.js file in your LABUSER## folder on Github.
    a. Before:
    b. 

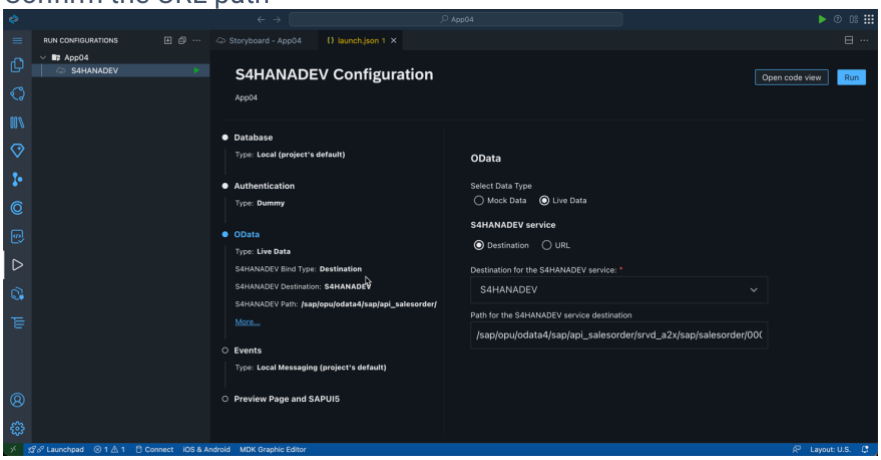
    c. After:

d. 

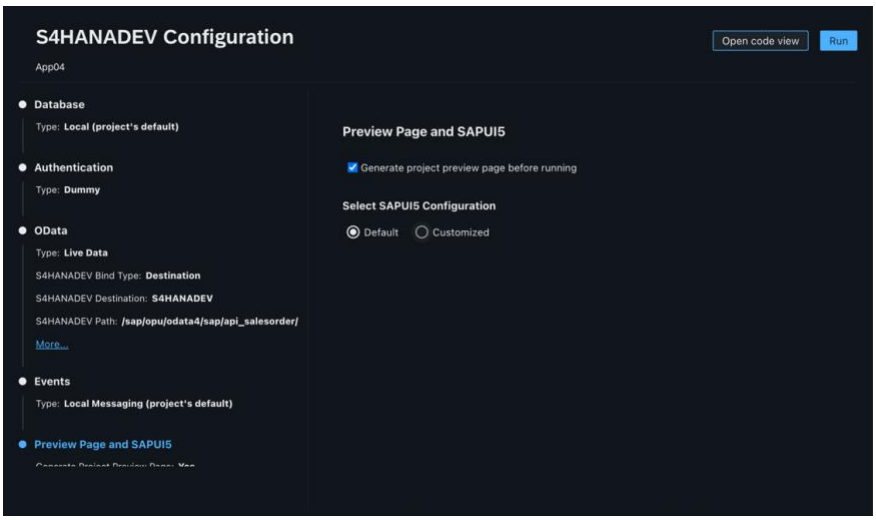
36. Next, close the tabs for service.js, Application Logic Editor, and Service.cds
37. 
38. Your storyboard should now show a dotted line from the External Service to a new SalesOrder service under Services.
39. 
40. Next, we're going to add a new UI just for the SalesOrder entity on the Service.
41. Click the Joule button on the left 
42. In the prompt, type the following command
    a. Command: /add-ui Add a Fiori List Page app for the Service quickOrderDisplaySrv for the entity SalesOrder
    b. 
43. After 2-3 minutes, the command will complete and you can view the page map.

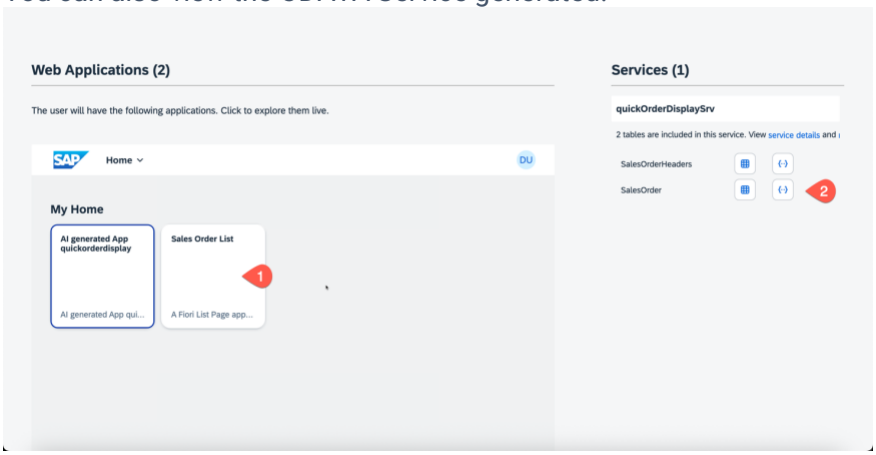
44. 

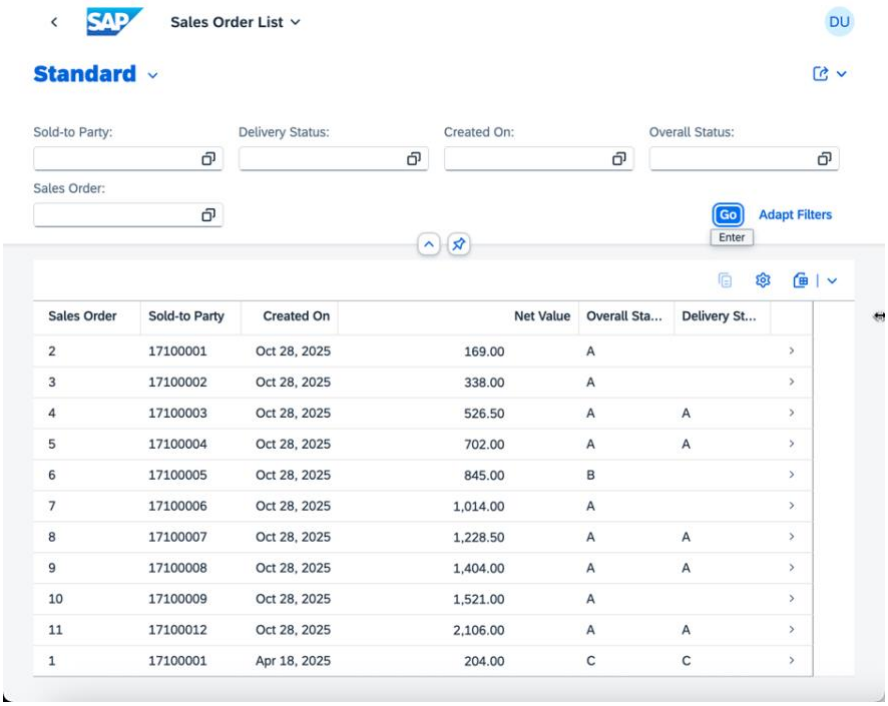
45. Edit the List Report page by clicking Pencil.
46. Adjust Filter and Table as you did above.
47. 

48. Next, we're going to test your new app with the live connection.
49. On the left, click on Run Configurations 
50. On your Run Configuration, select ODATA and ensure the radio button Live Data is selected.
51. Confirm the URL path
52. 

53. Select *Preview Page and SAPUI5* button
54. Confirm you have checked *Generate project preview page before running*

55. 

56. Click Run
57. This process can take 2-3
58. When the Page Preview Opens, you have another app to open.
59. You can also view the ODATA Service generated.
60. 

61. Click Go on your app to view live data!
62. 

Thank you for completing our Hands-On Lab! Please email if you have any questions!