Module IV

Foundations of SAP ABAP and Build Code with Joule

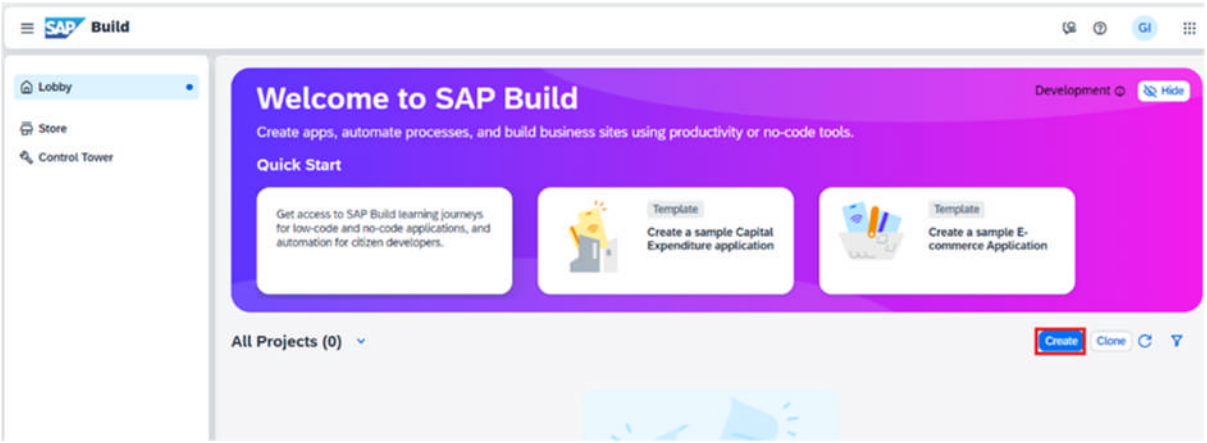
Title

Hand Book

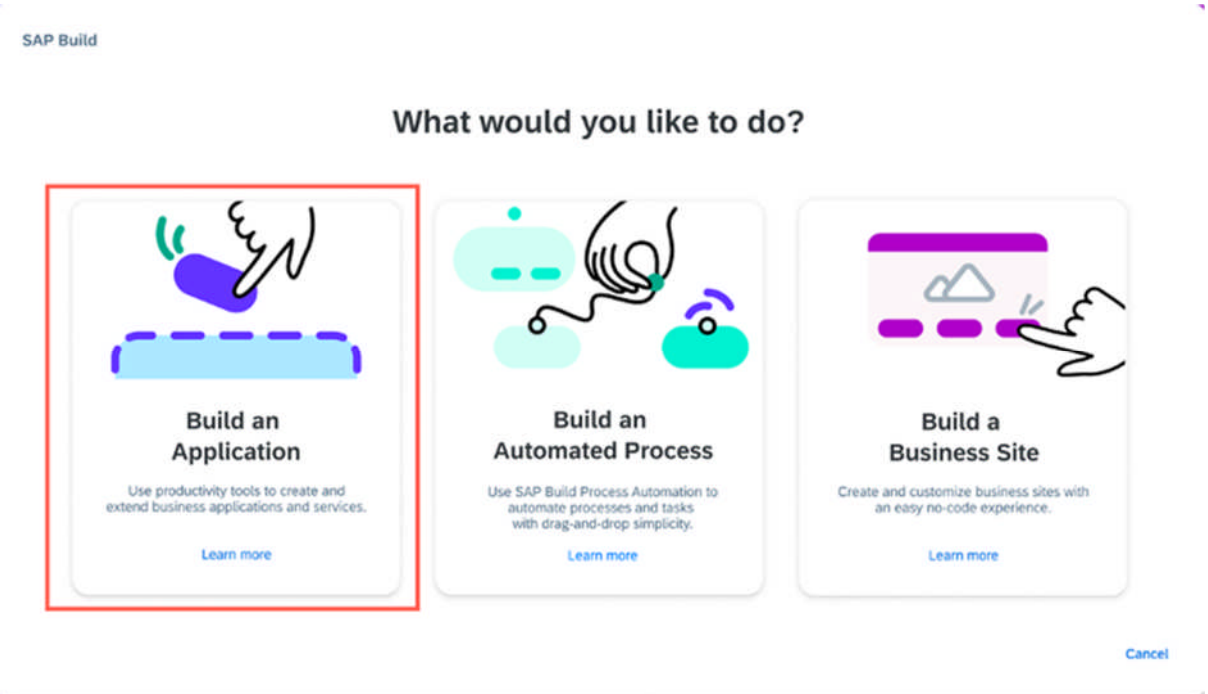
Lab:-2 Create a Full-Stack SAP Fiori Application with Joule in SAP Build Code

1 Create a New Project Using SAP Build Code

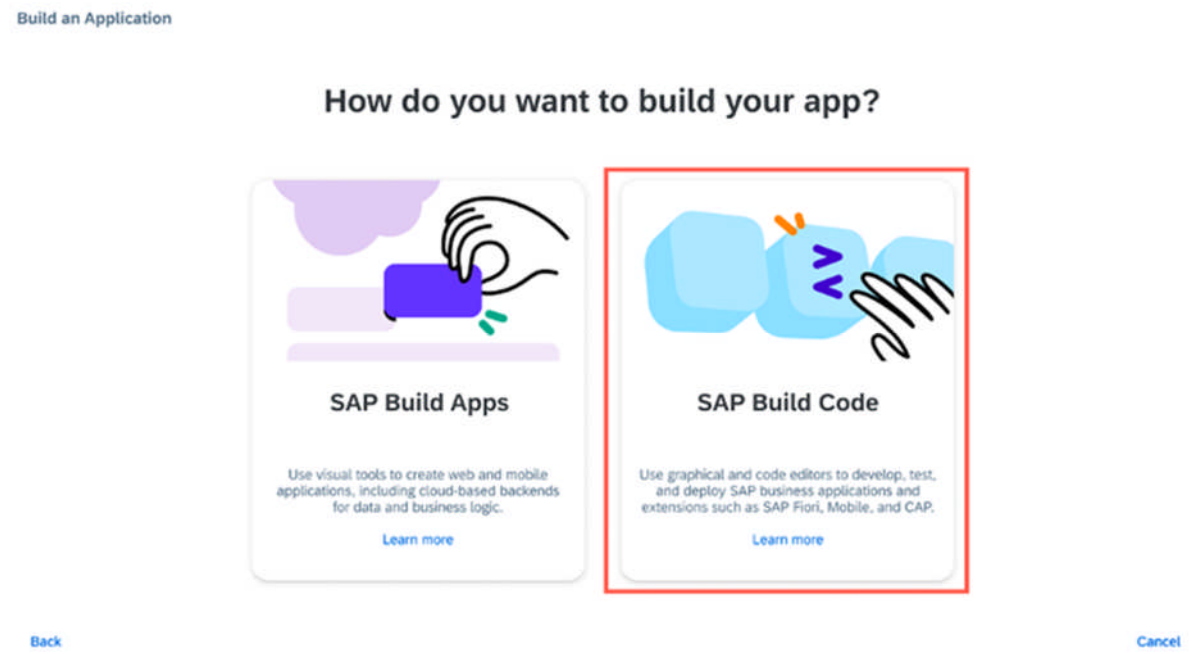
1. Navigate to the SAP Build lobby.
2. Click **Create** to start the creation process



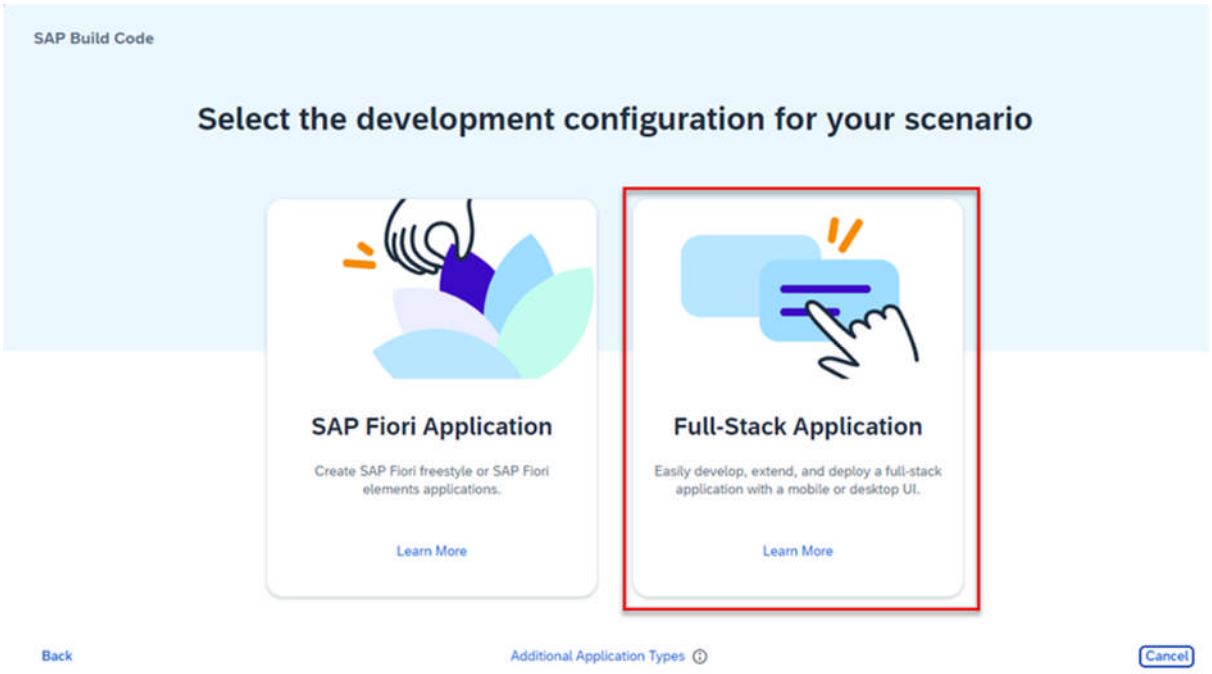
3. Click the **Build an Application** tile.



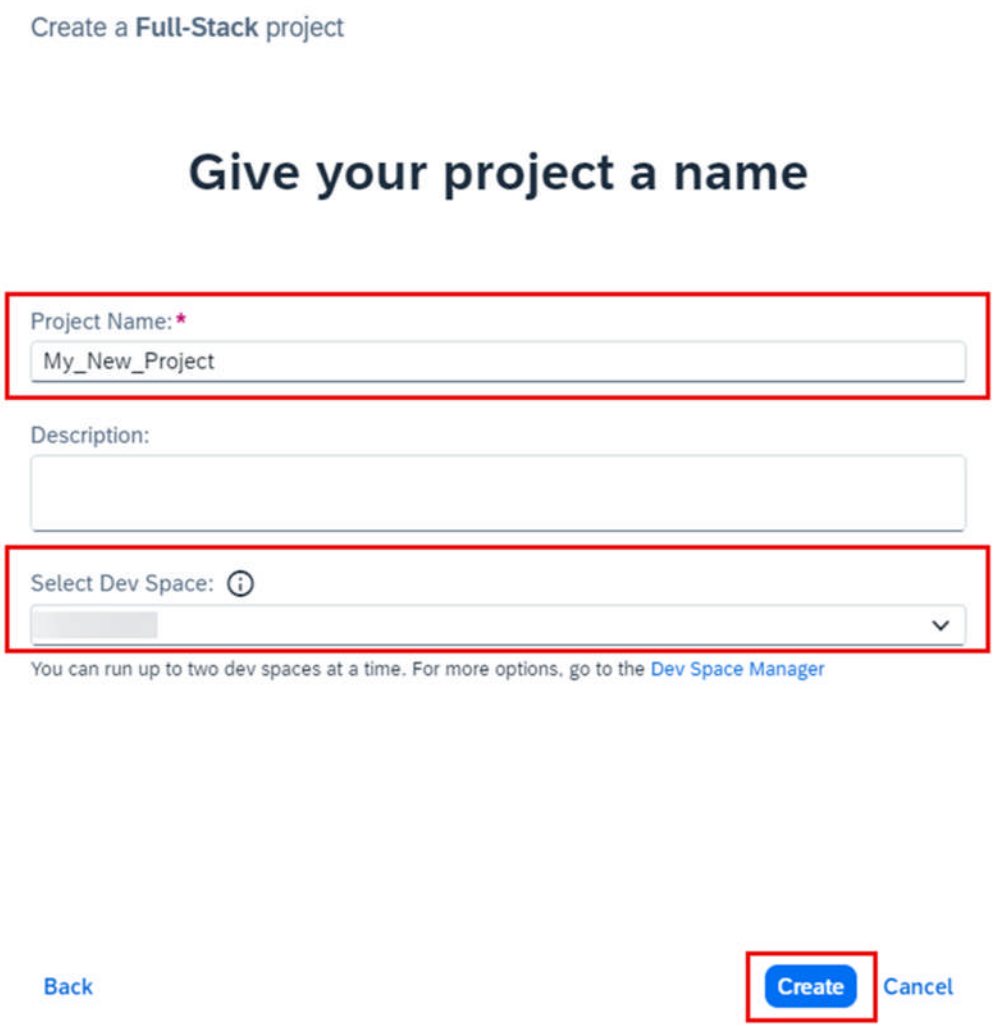
4.Click the **SAP Build Code** tile to develop your project in SAP Business Application Studio, the SAP Build Code development environment, leveraging the capabilities of the services included in SAP Build Code.



5.Click the **Full-Stack Application** tile.

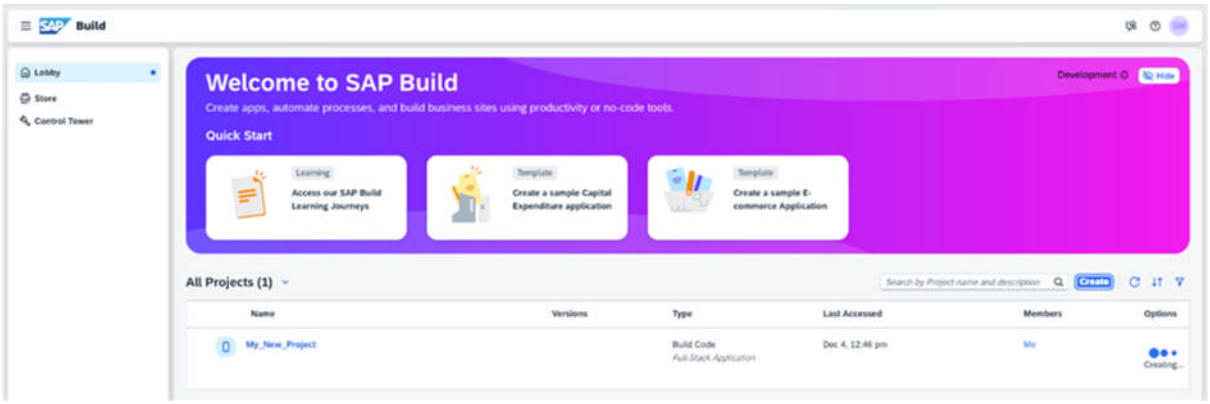


6. Enter a name for your project.
7. Select the dev space where you want the project to reside.
8. Click Create.

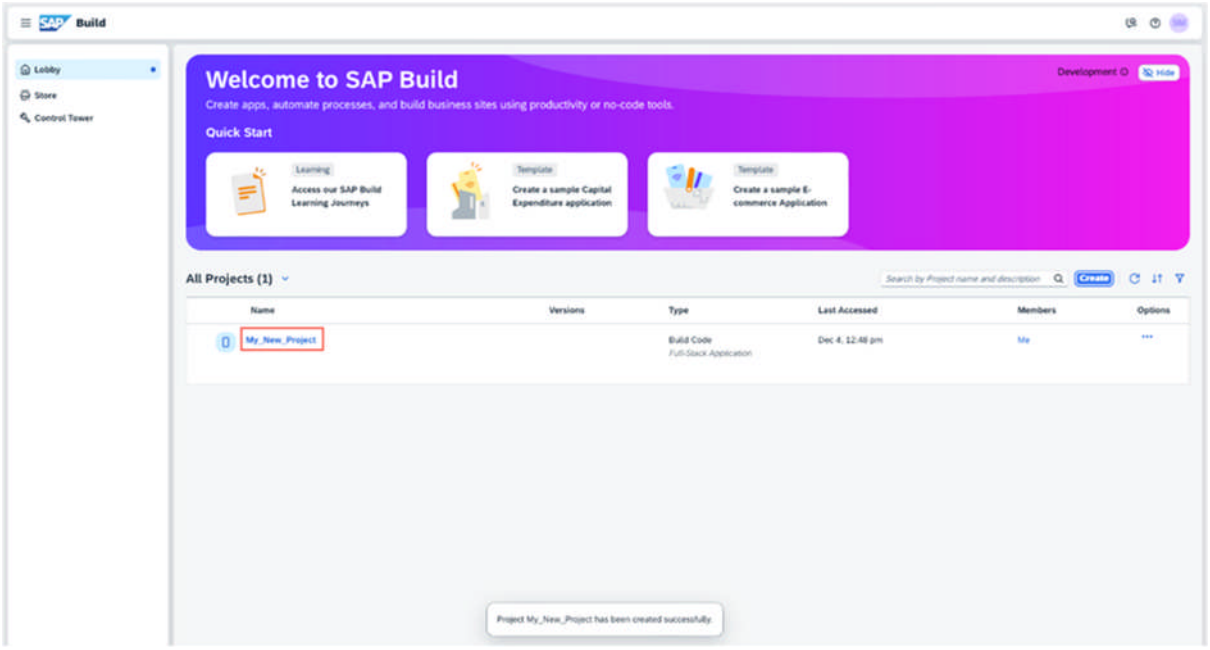


9.You can see the project being created in the Project table of the lobby.

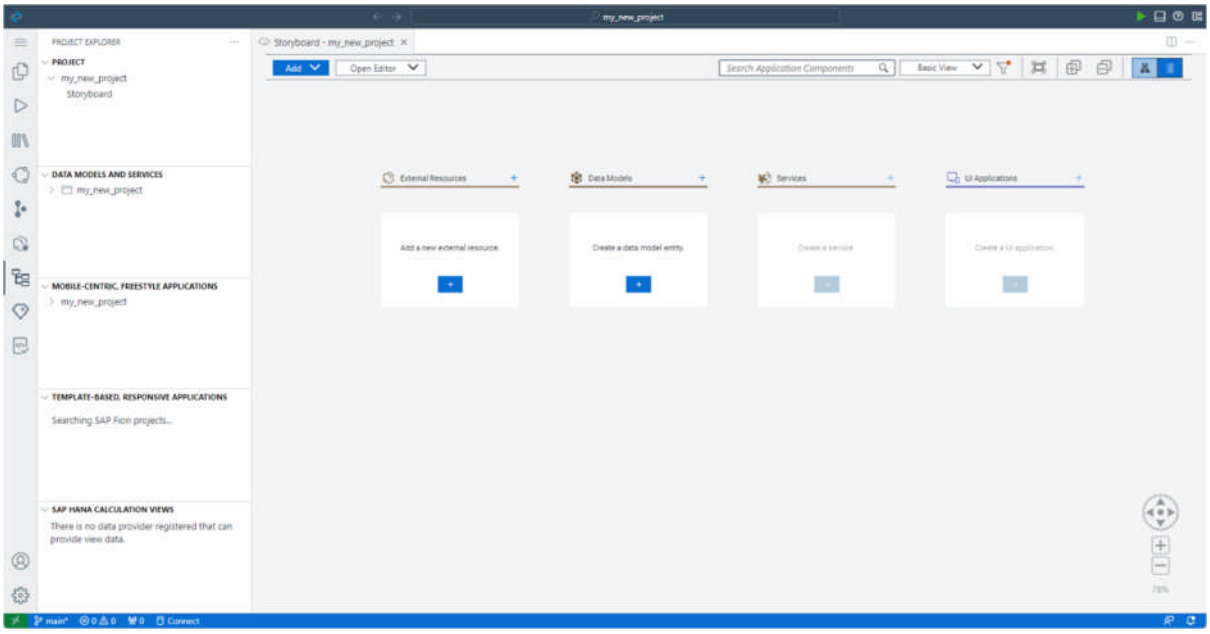
The creation of the project may take a few moments.



10. After you see a message stating that the project has been created successfully, click the project to open it.



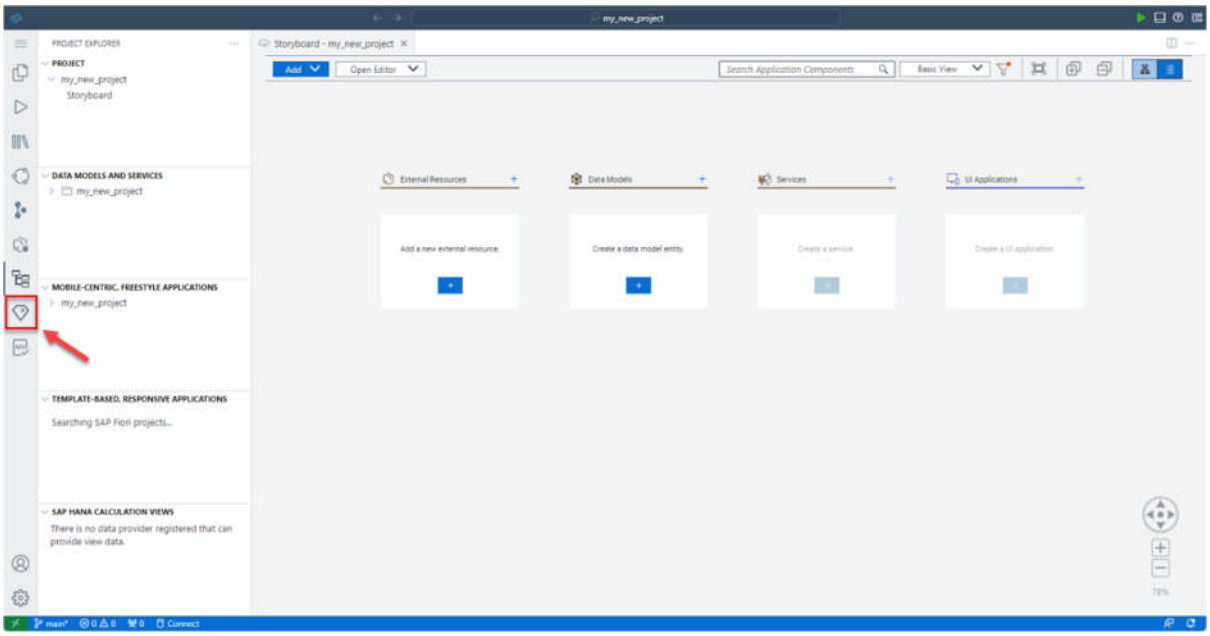
The project opens in SAP Business Application Studio, the SAP Build Code development environment.



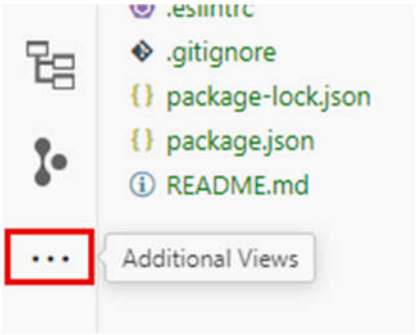
2 Create Data Entities with Joule

In SAP Business Application Studio, the SAP Build Code development environment, open the digital assistant, Joule, from the activity bar.

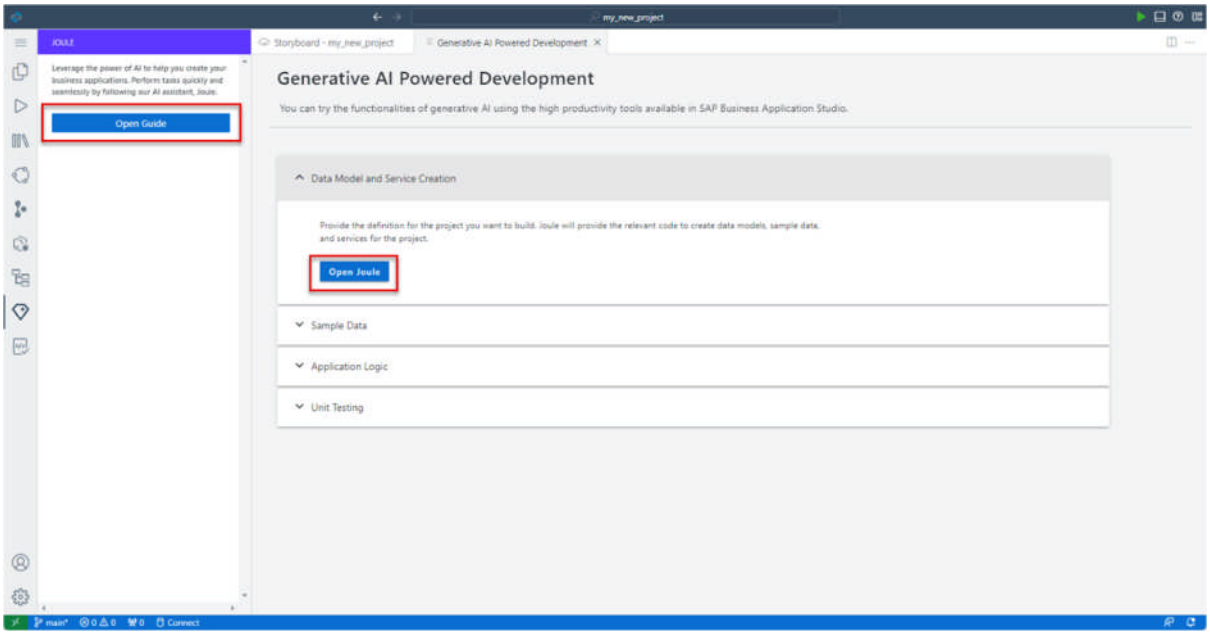
1. In SAP Business Application Studio, the SAP Build Code development environment, open the digital assistant, Joule, from the activity bar.



Note: If you do not see the icon, click Additional Views and select **Joule** from the list.



2. Click **Open Guide**.
3. Expand the **Data Model and Service Creation** section, and click **Open Joule**.

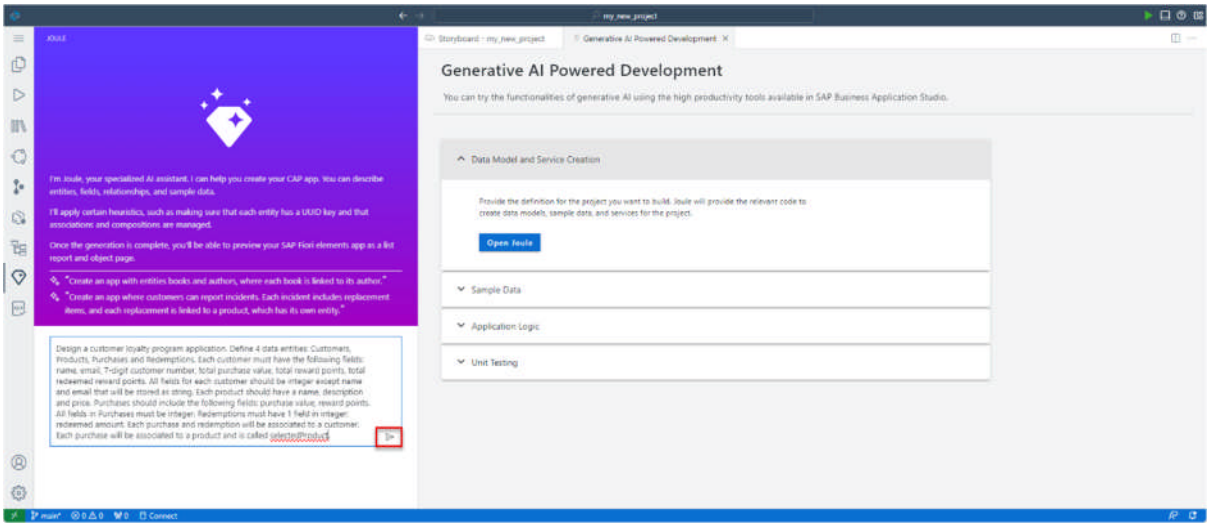


Copy the prompt below.

```
Design a customer loyalty program application.
Define 4 data entities: Customers, Products, Purchases and Redemptions.
Each customer must have the following fields: name, email, 7-digit
customer number, total purchase value, total reward points, total
redeemed reward points.
All fields for each customer should be integer except name and email
that will be stored as string.
Each product should have a name, description and price.
Purchases should include the following fields: purchase value, reward
points.
All fields in Purchases must be integer.
Redemptions must have 1 field in integer: redeemed amount.
```

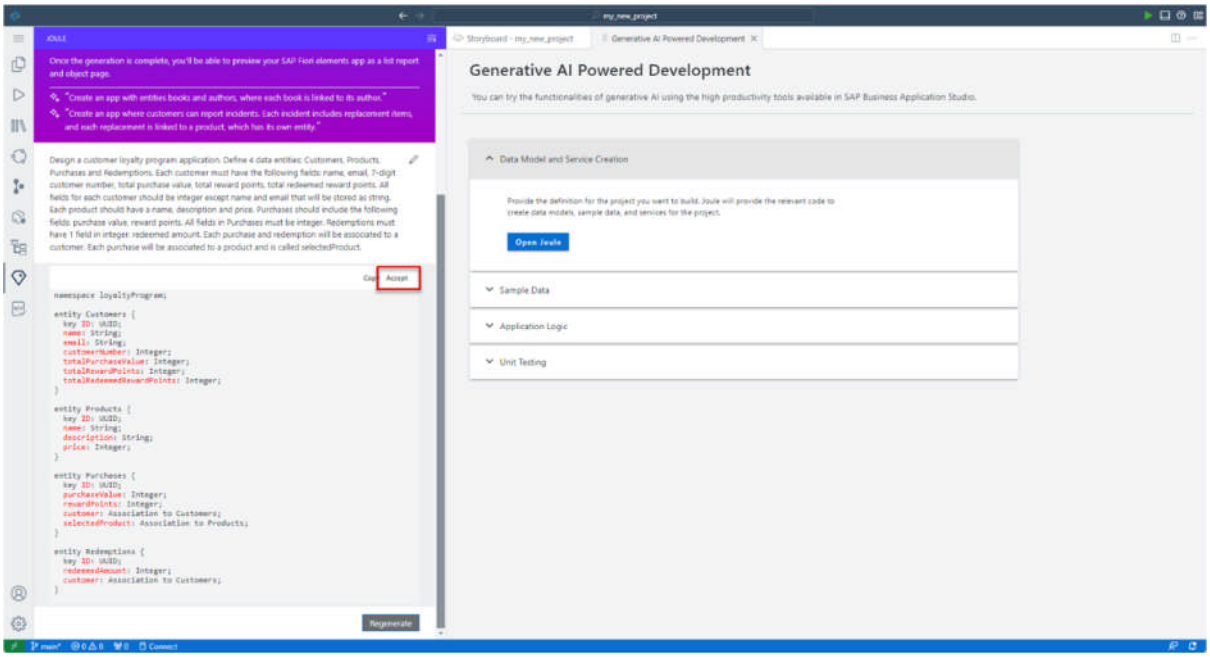
4. Paste the code in the text field, and click the arrow

to send the prompt to Joule.



The code is generated and is displayed below your prompt.

6. Accept the code.



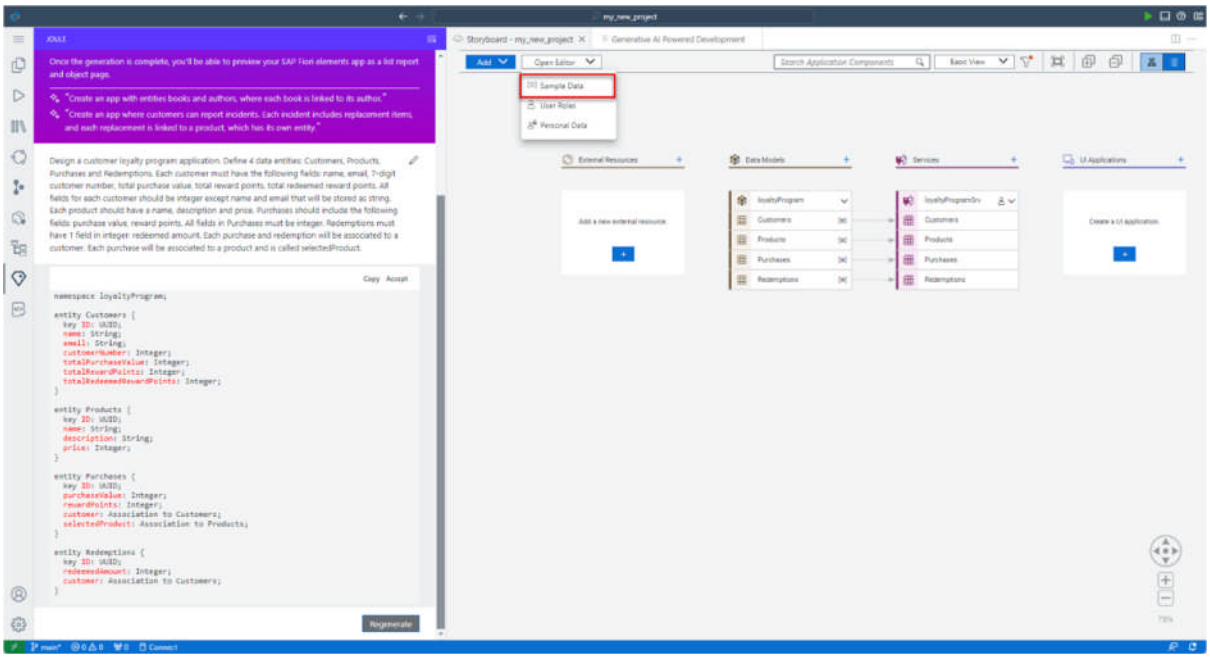
Depending on the server, it may take a few moments for Joule to create the data models and services for you.

Once you accept the code, you will see the update on the right side in the Storyboard tab.

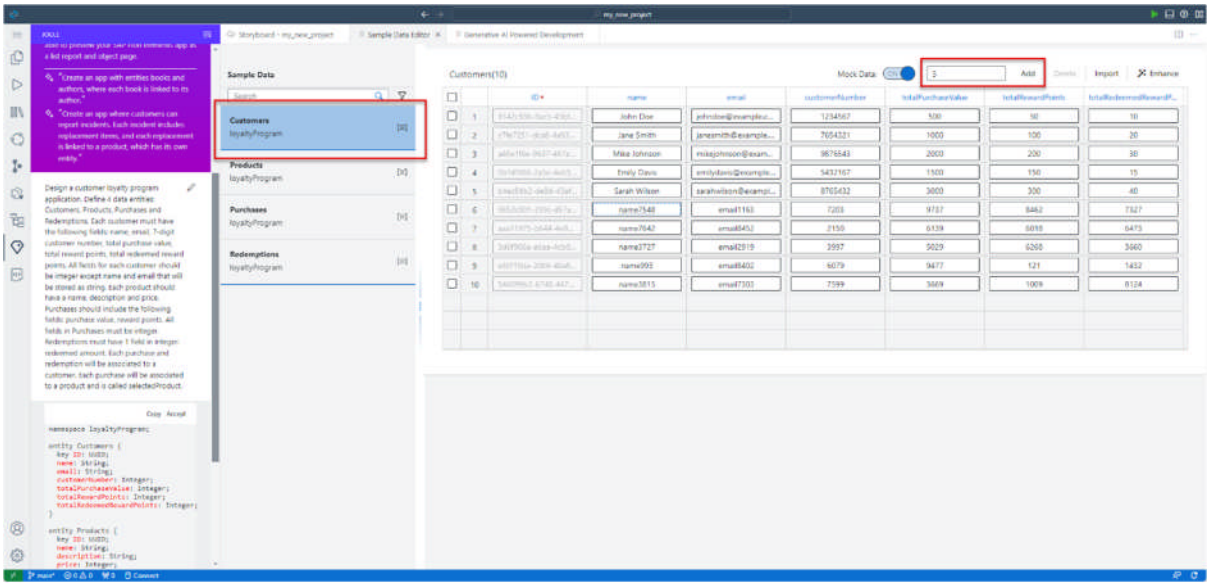
3 Enhance the Sample Data Using Joule

Joule created the CAP data model and the OData service. In addition, Joule created some sample data by default. We will now ask Joule to provide additional sample data.

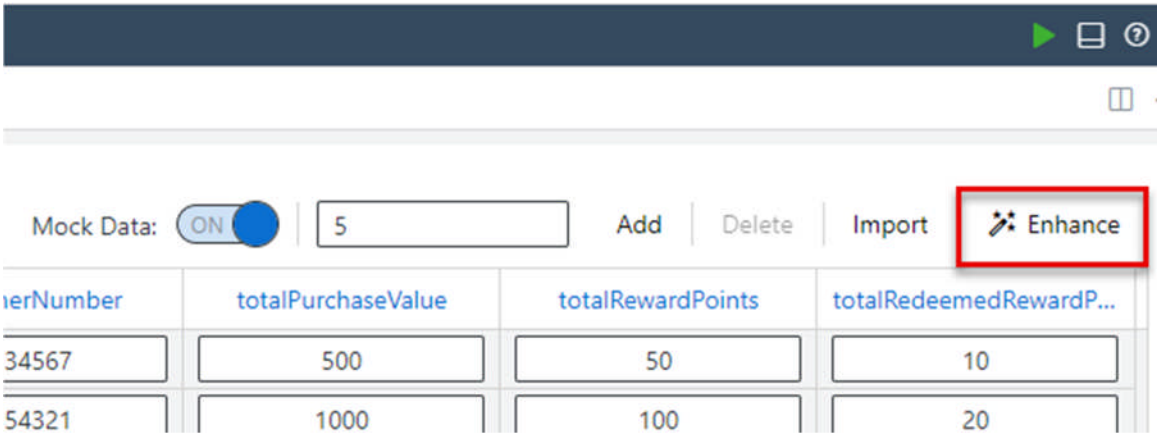
1. Open the Sample Data editor in the Storyboard by selecting **Open Editor** -> **Sample Data**.



2. In the Sample Data Editor, select the **Customers** data entity, and add 5 more rows. Click **Add**.



3. Click **Enhance**. This will reopen Joule to modify the sample data.

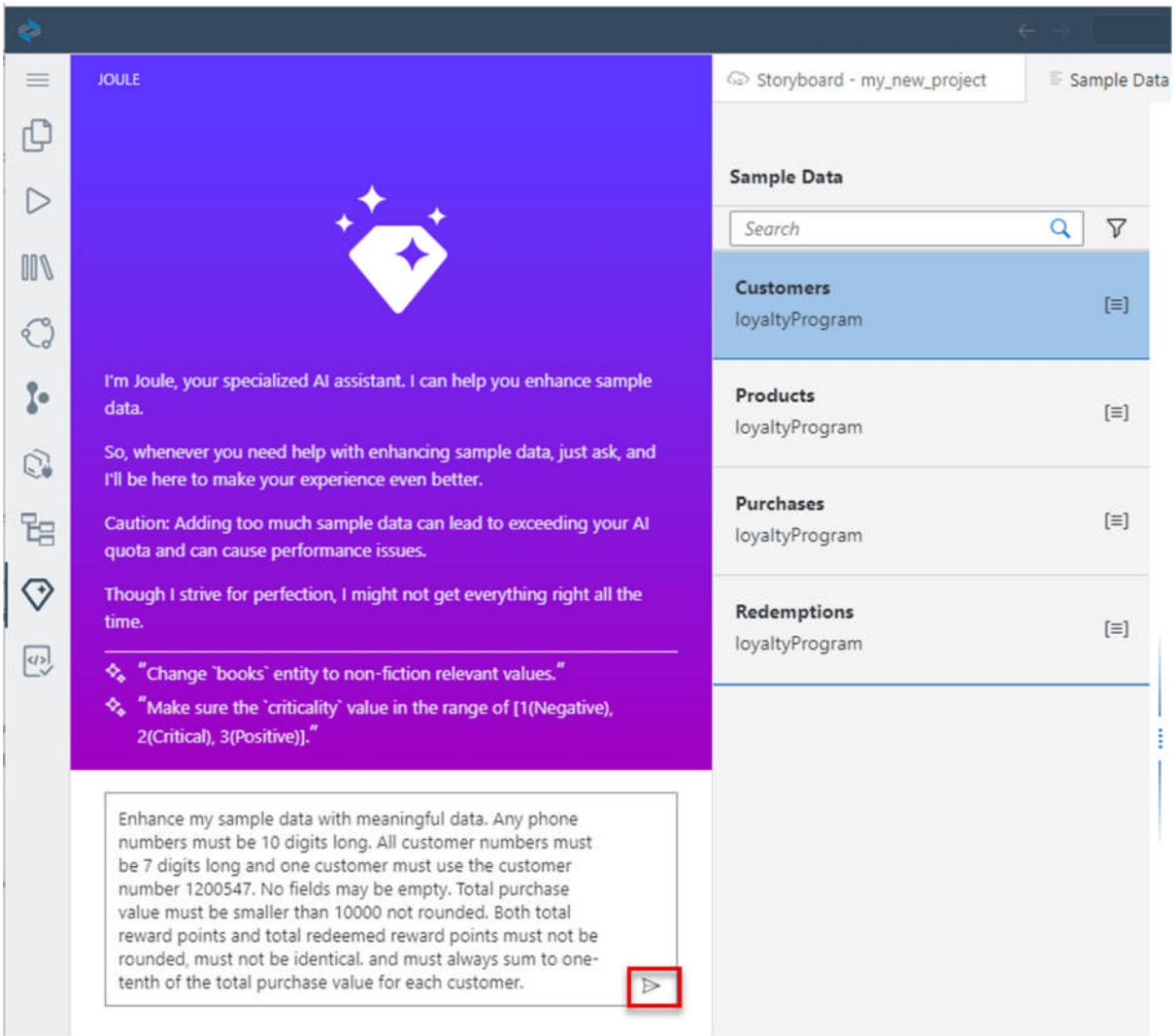


Copy the prompt below:

```
Enhance my sample data with meaningful data. Any phone numbers
must be 10 digits long.
All customer numbers must be 7 digits long and one customer must
use the customer number 1200547.
No fields may be empty.
Total purchase value must be smaller than 10000 not rounded.
Both total reward points and total redeemed reward points must
not be rounded, must not be identical. and must always sum to
one-tenth of the total purchase value for each customer.
```

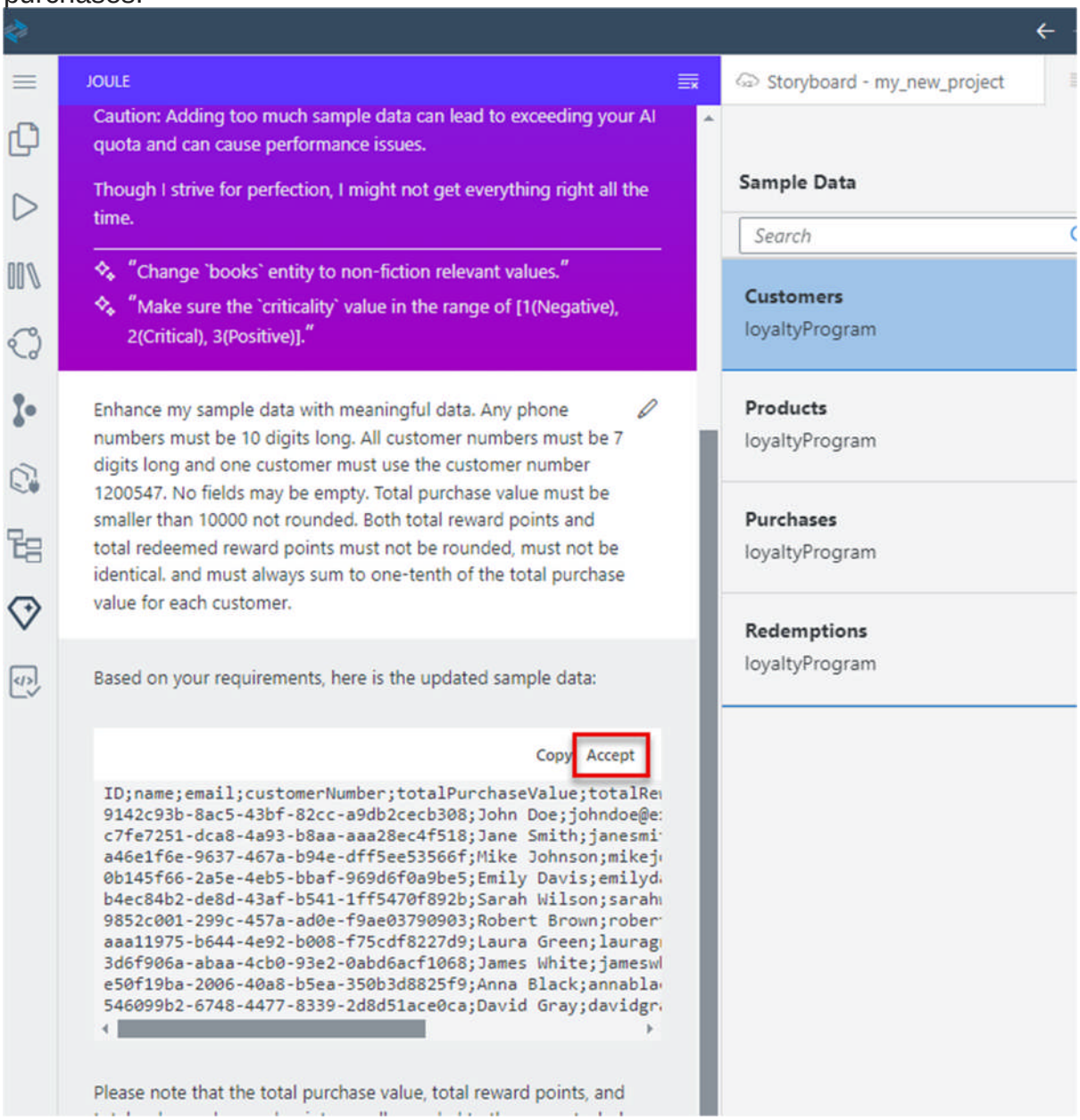
Paste the prompt in the text field, and click the arrow (

) to send the prompt to Joule.



The code is generated and is displayed below your prompt.

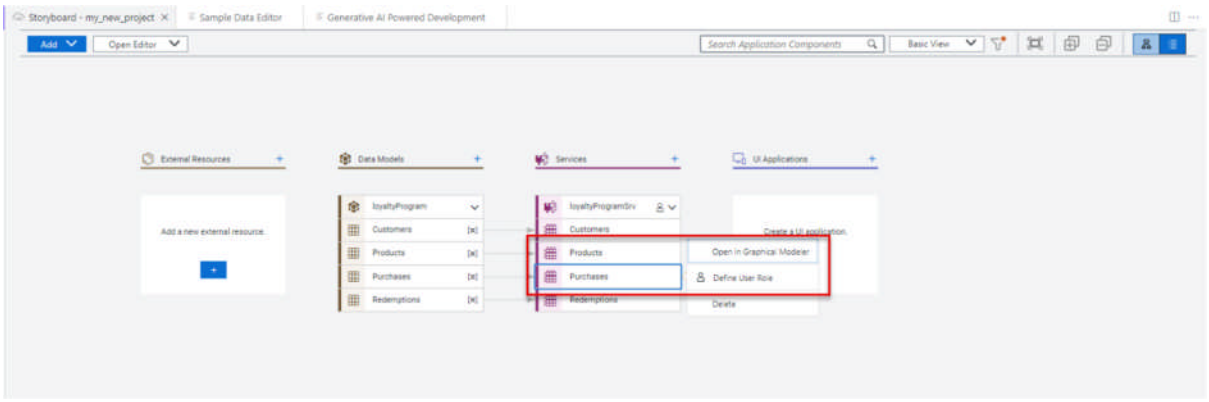
4. Accept the code.This will add the customer names, email addresses, and purchases.



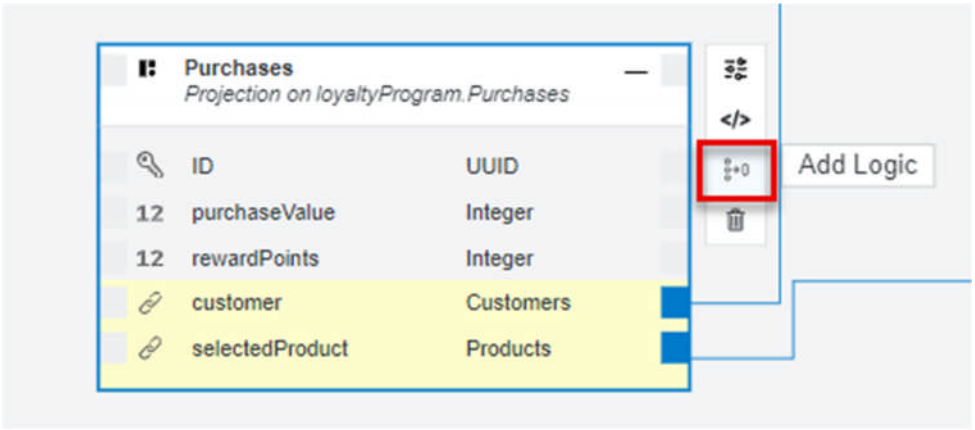
4 Create Application Logic with Joule

We already have created the data model, service, and sample data with Joule. Now we want to create some logic for our service. We would like to calculate the bonus points automatically when a customer makes a purchase. Additionally, we want to provide logic for customers to redeem these bonus points.

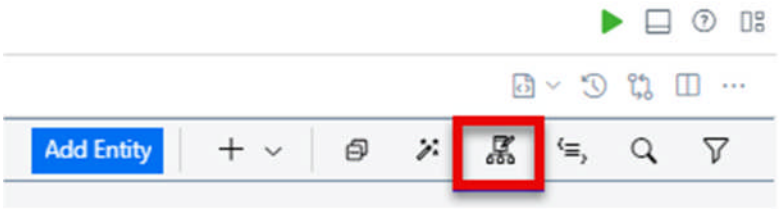
1. In the Storyboard, click on the **Purchases** entity under **Services**, and select **Open in Graphical Modeler**.



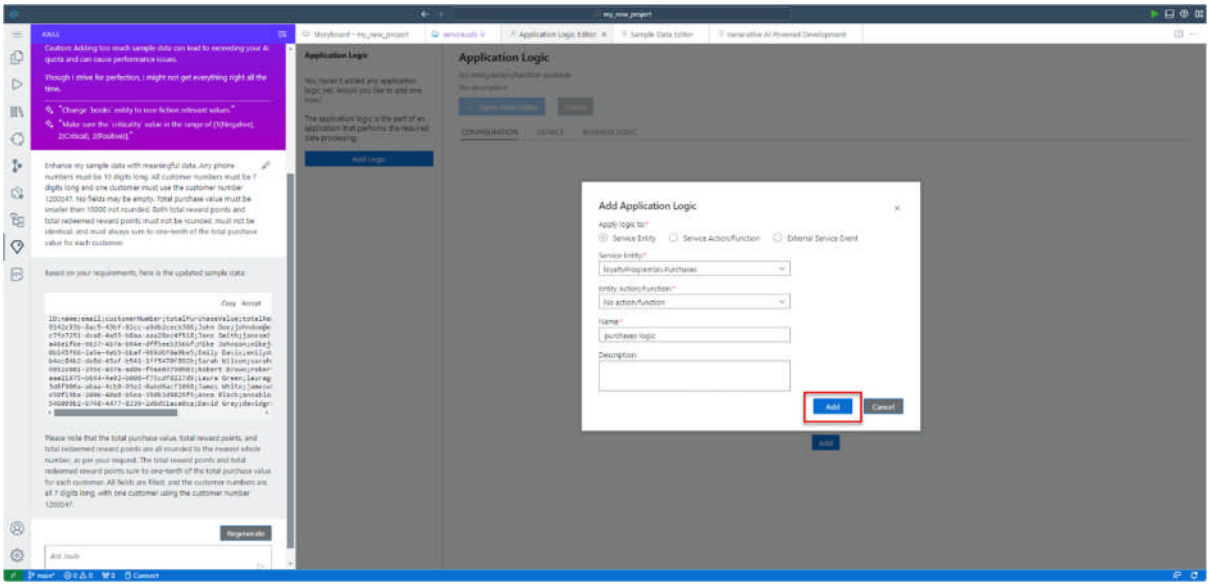
2. Select the **Purchases** entity by clicking on the title. Then, click **Add Logic**.



If you do not see the entity, click the Show All icon.

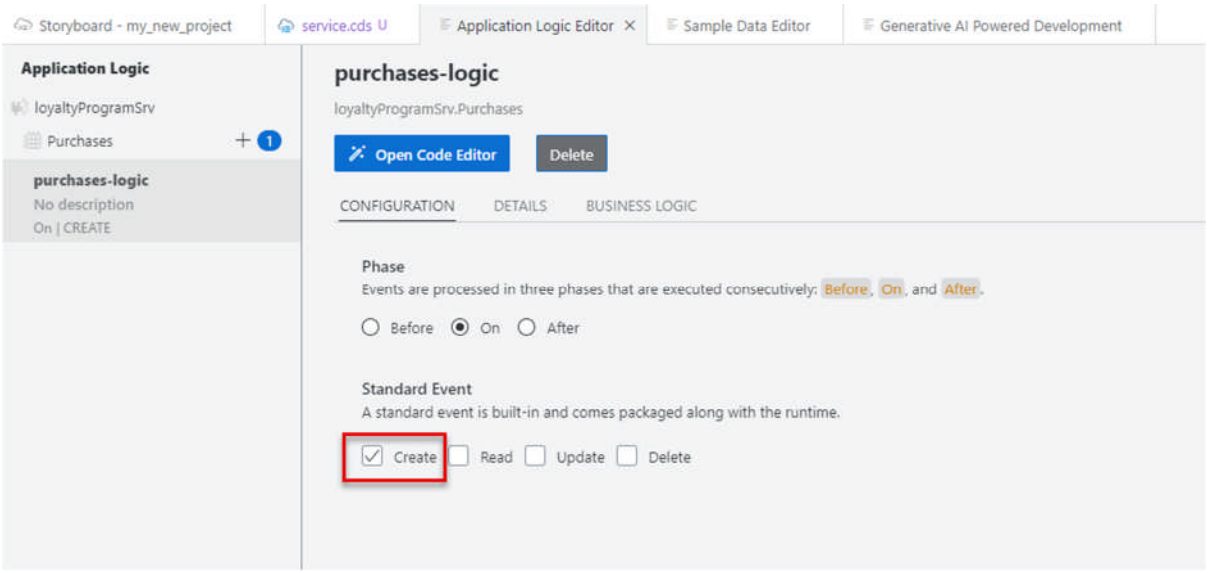


3. In the **Add Application Logic** dialog, leave the default values, and click **Add**.

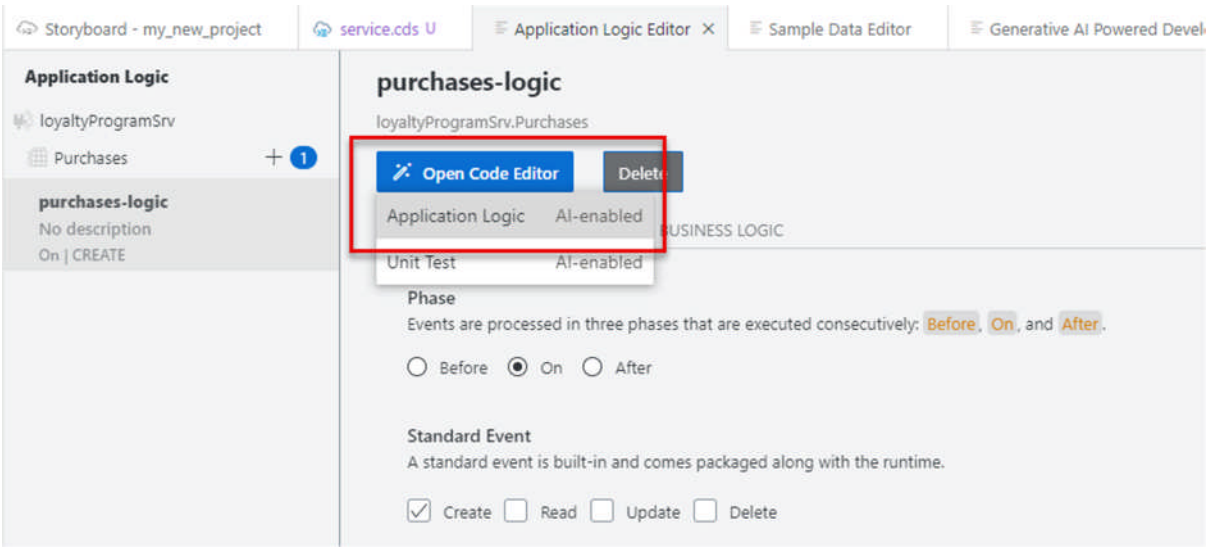


The Application Logic Editor opens.

4. In the **Standard Event** section, select **Create**. That means that this logic will be automatically executed if an OData create operation is requested.



5. Click **Open Code Editor**, and select **Application Logic**. This will open Joule again to allow us to send a prompt to Joule to create the logic for us.

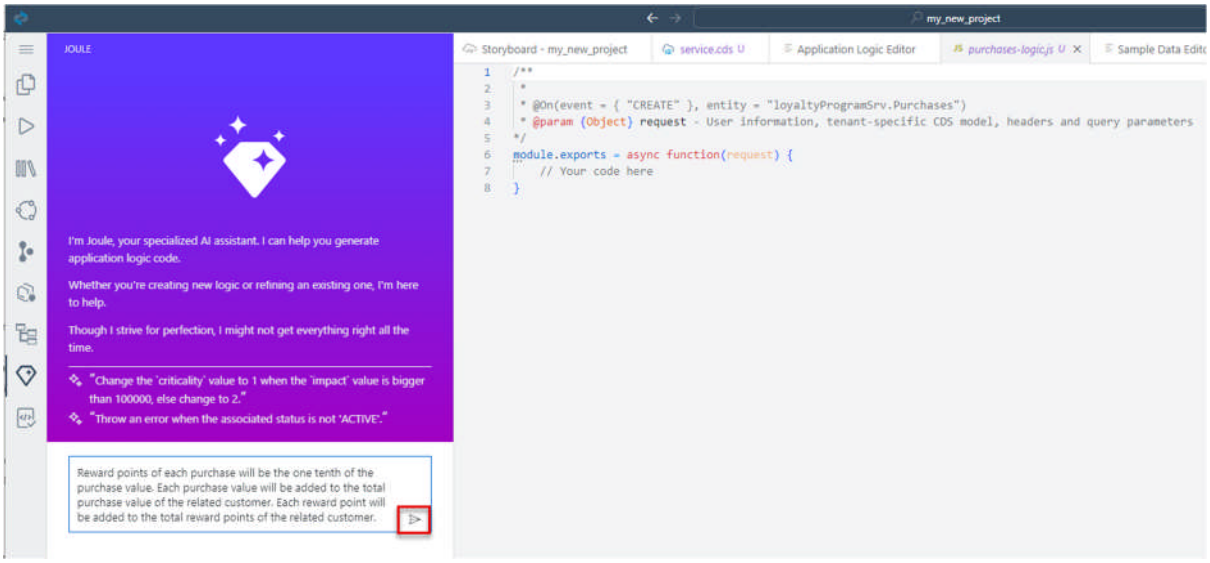


Copy the prompt below:

```
Reward points of each purchase will be the one tenth of
the purchase value.
Each purchase value will be added to the total purchase
value of the related customer.
Each reward point will be added to the total reward
points of the related customer.
```

Paste the prompt in the text field, and click the arrow (

) to send the prompt to Joule.

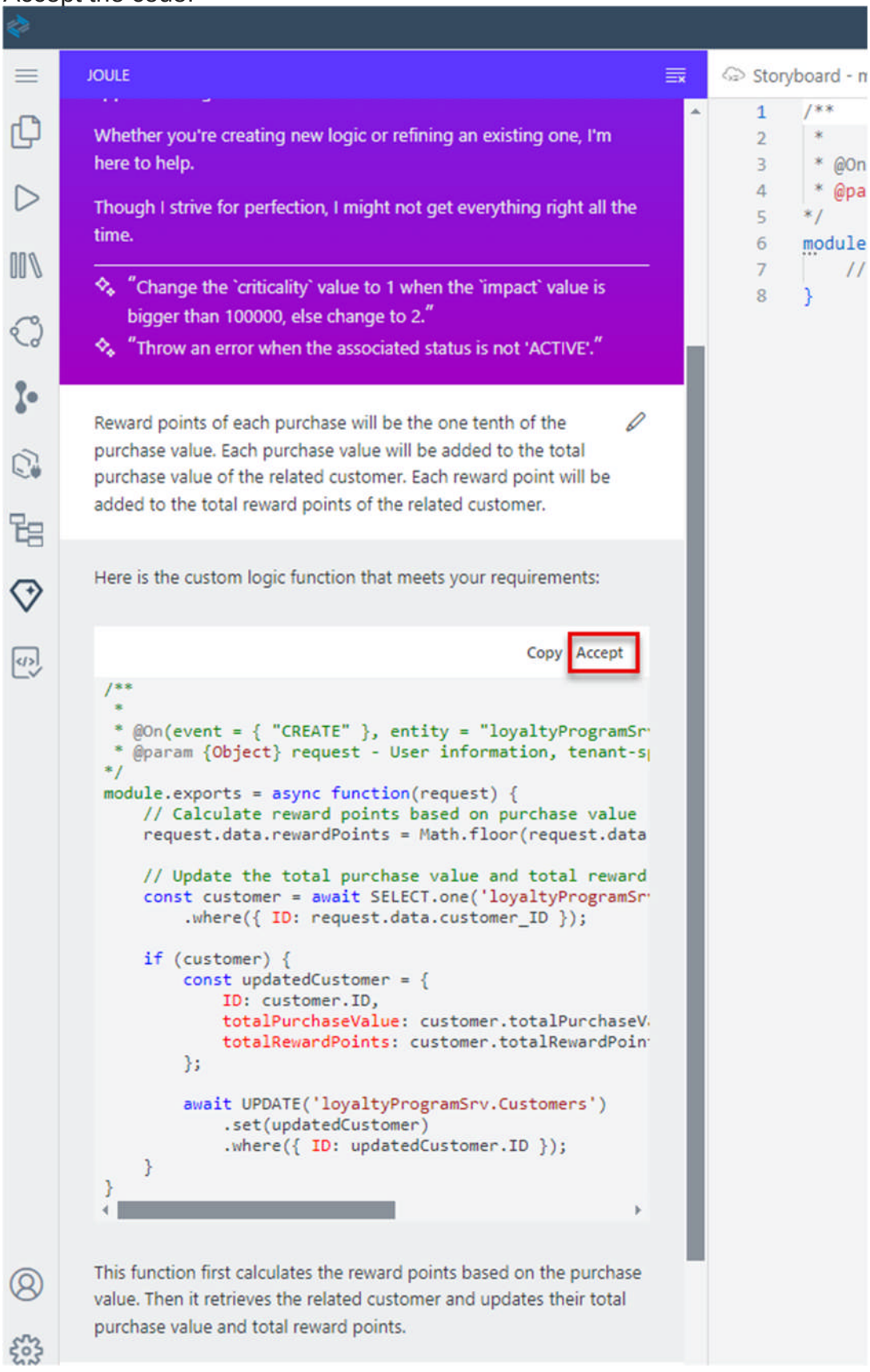


So Joule created code that implements the following logic:

- Check if the customer exists
- Calculate the rewardPoints from the purchase value

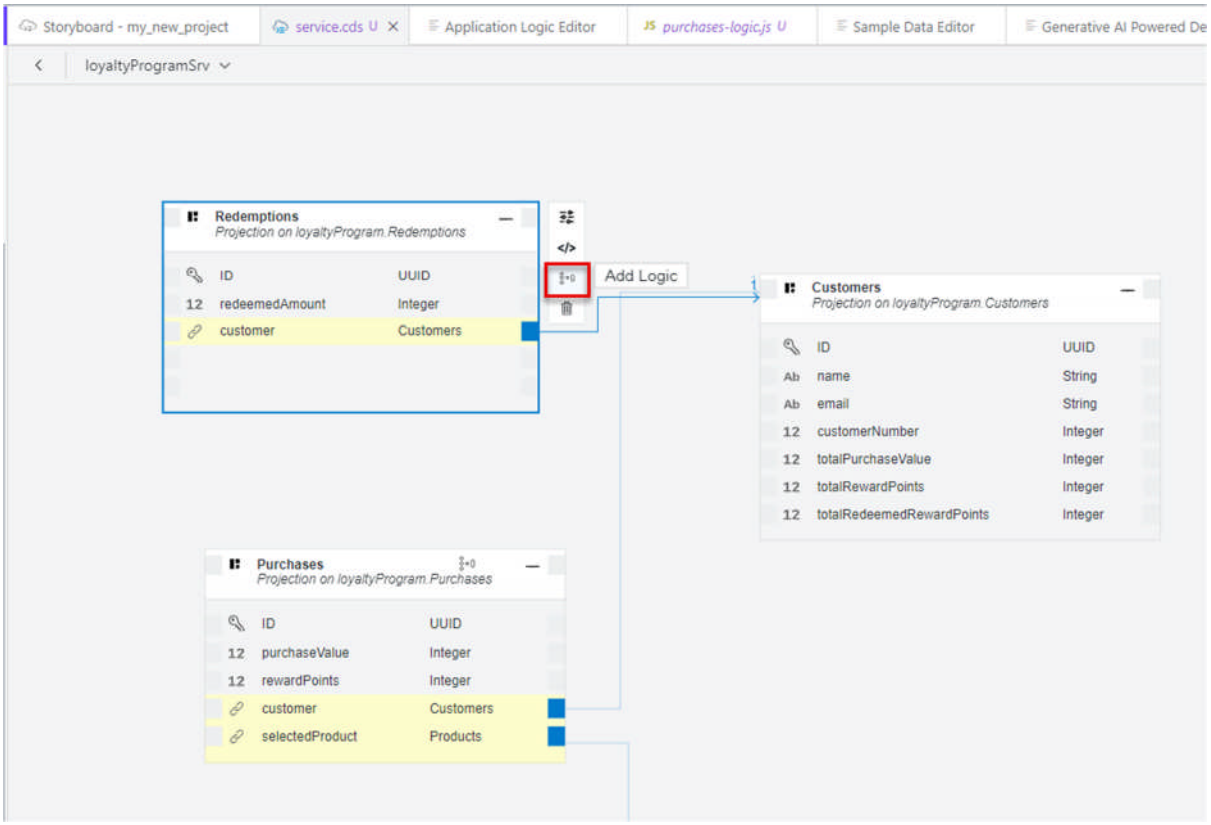
- Update the total purchase value and the total reward points in the customers entity

6. Accept the code.

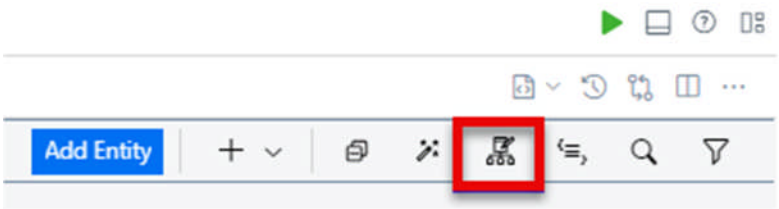


Note: Joule typically generates different code each time for the same prompt. If yours is different to what you can see here, that's fine as long as it does the same job.If there are no obvious errors, just keep working on the exercise. If you aren't sure, you can ask Joule to try again by clicking **Regenerate**.

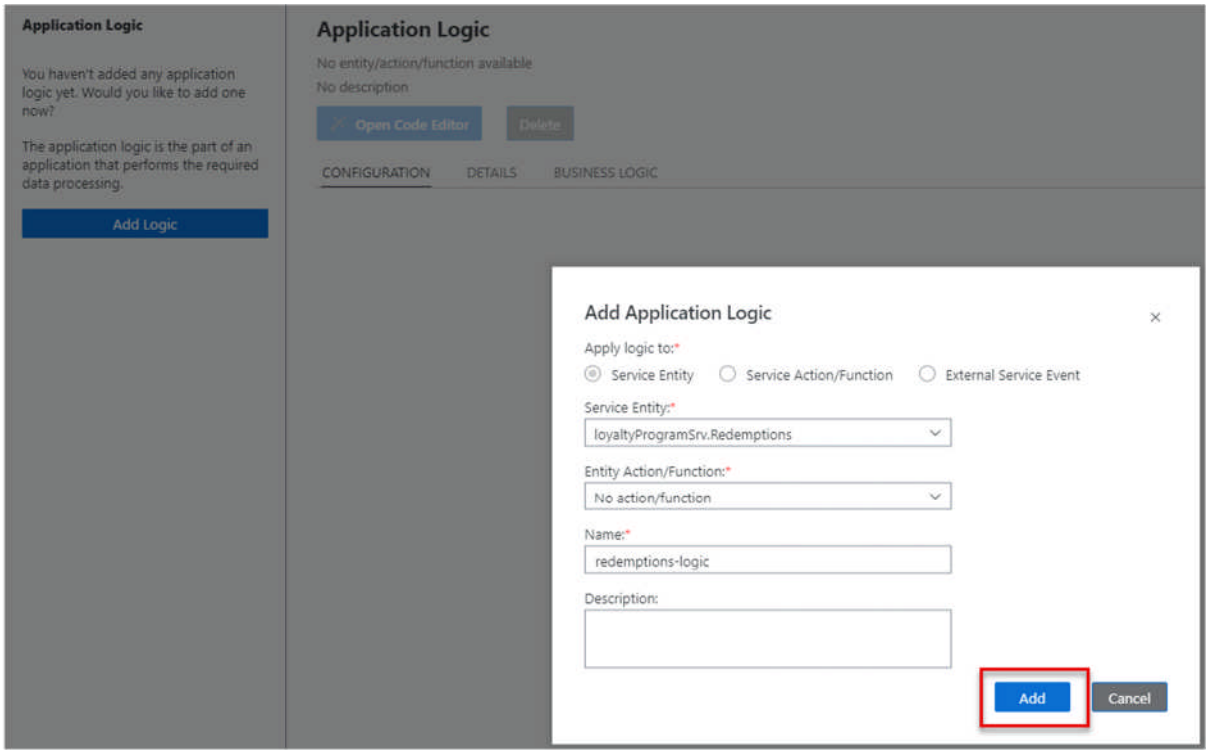
7. Go back to the service.cds tab.
8. Select the **Redemptions** entity by clicking on the title. Then, click **Add Logic**.



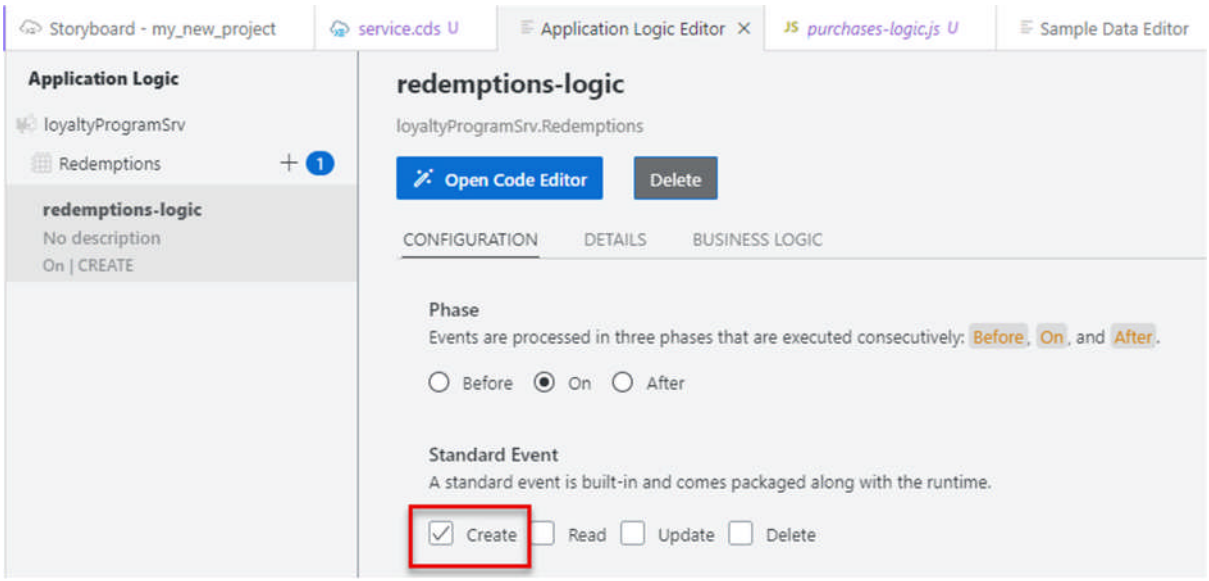
If you do not see the entity, click the Show All icon.



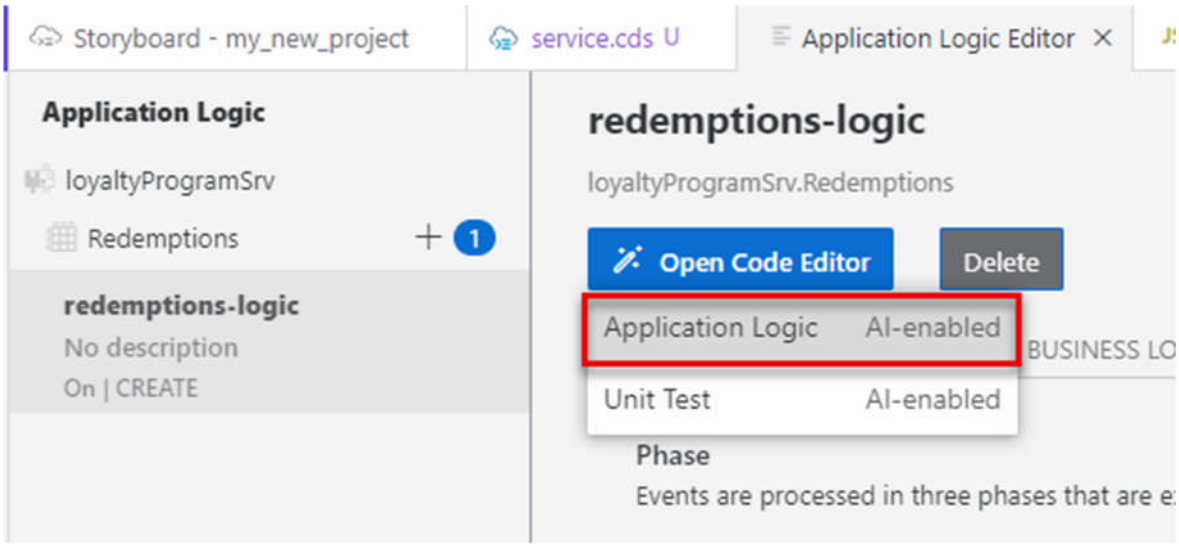
9. In the **Add Application Logic** dialog, leave the default values, and click **Add**.



10. In the **Standard Event** section, select **Create**.



11. Click **Open Code Editor**, and select **Application Logic**. This will open Joule again to allow us to send a prompt to Joule to create the logic for us.

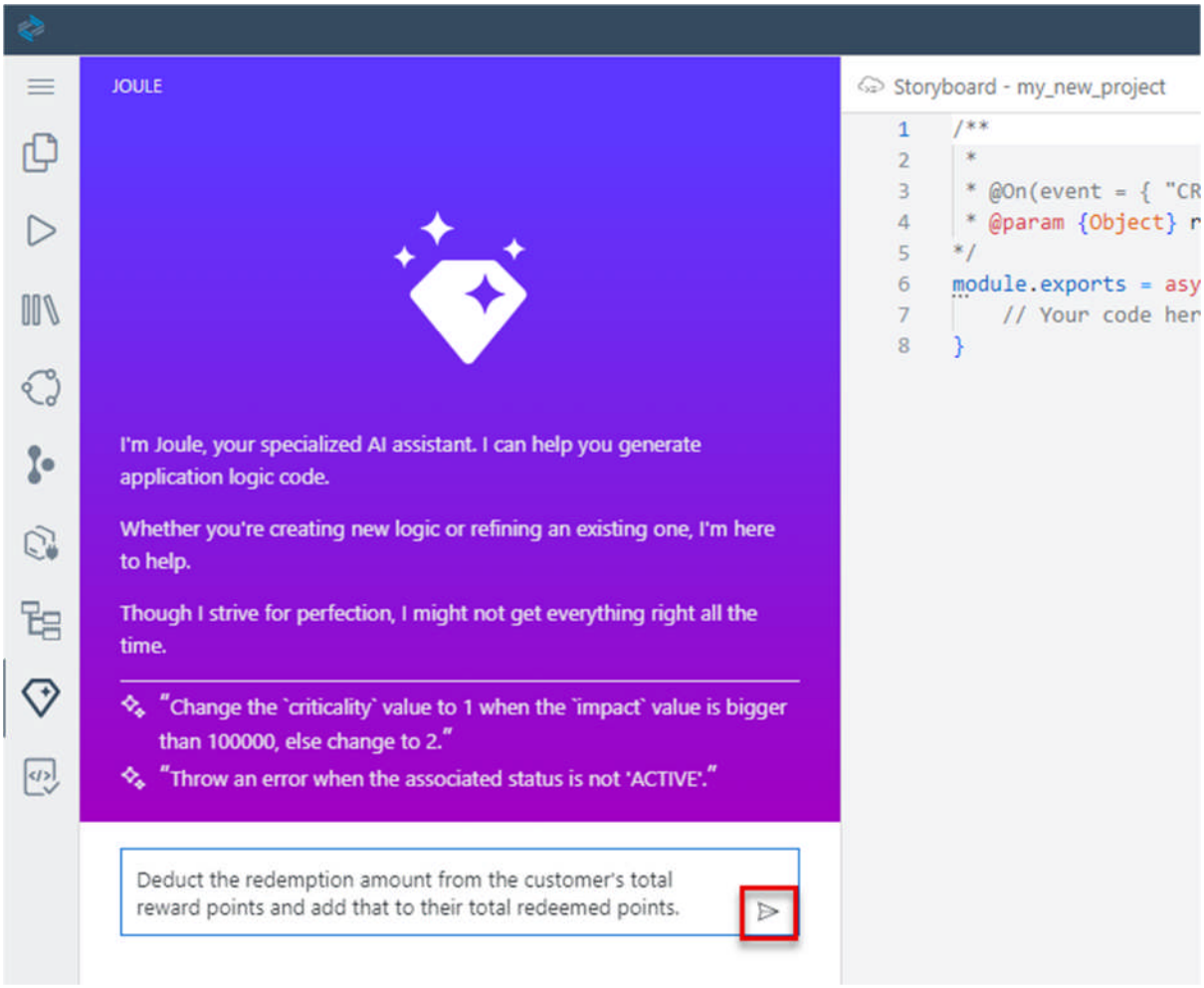


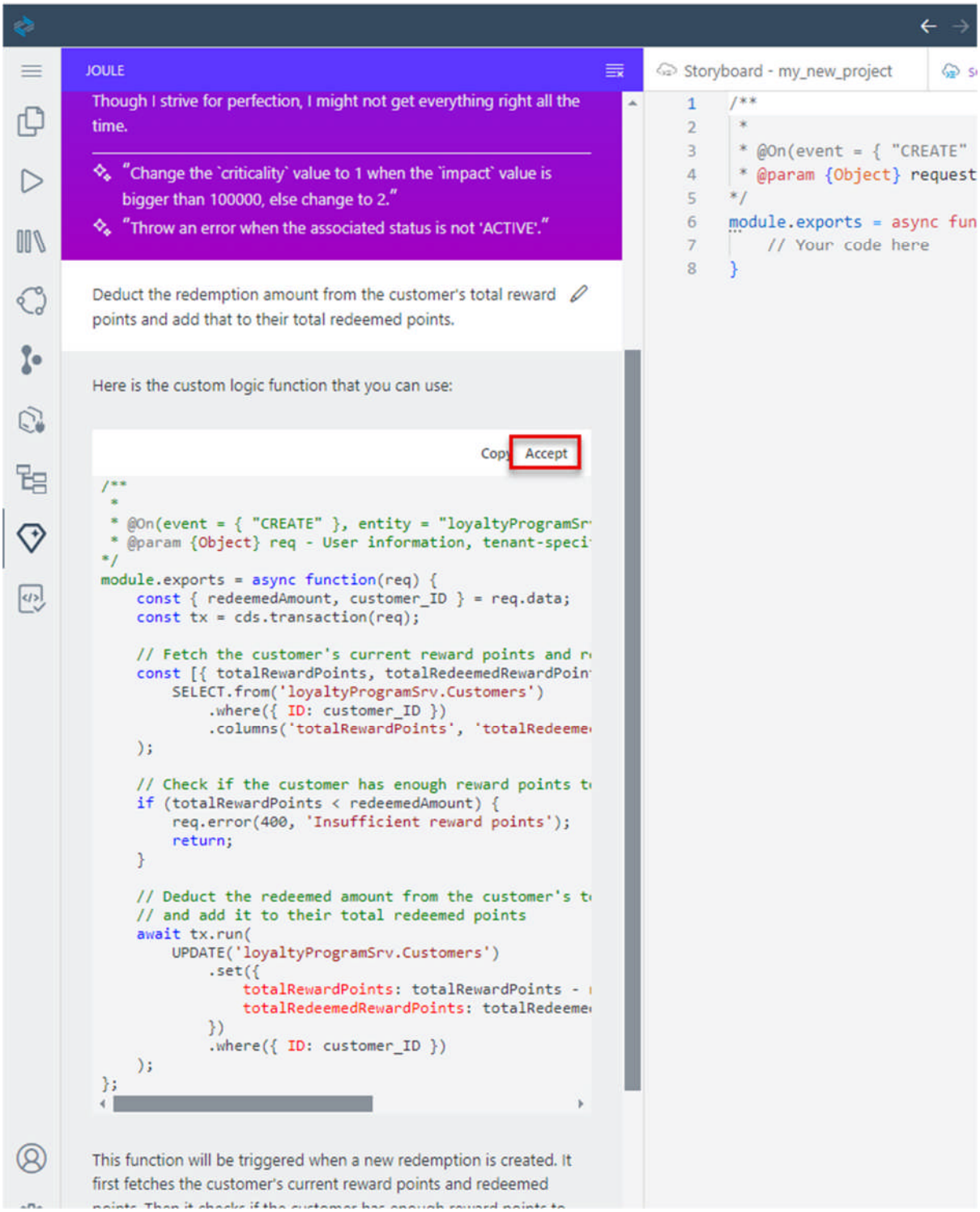
Copy the prompt below::

```
Deduct the redemption amount from the customer's total
reward points and add that to their total redeemed
points.
```

Paste the prompt in the text field, and click the arrow (

) to send the prompt to Joule.


JOULE
I'm Joule, your specialized AI assistant. I can help you generate application logic code.
Whether you're creating new logic or refining an existing one, I'm here to help.
Though I strive for perfection, I might not get everything right all the time.
"Change the `criticality` value to 1 when the `impact` value is bigger than 100000, else change to 2."
"Throw an error when the associated status is not 'ACTIVE'."
Deduct the redemption amount from the customer's total reward points and add that to their total redeemed points.
Storyboard - my_new_project
/**
*
* @On(event = { "CR
* @param {Object} r
*/
module.exports = asy
// Your code her
}

16. 

Have a closer look at the generated code. It even includes some checks to see if the customer has enough points to redeem.

my_new_project

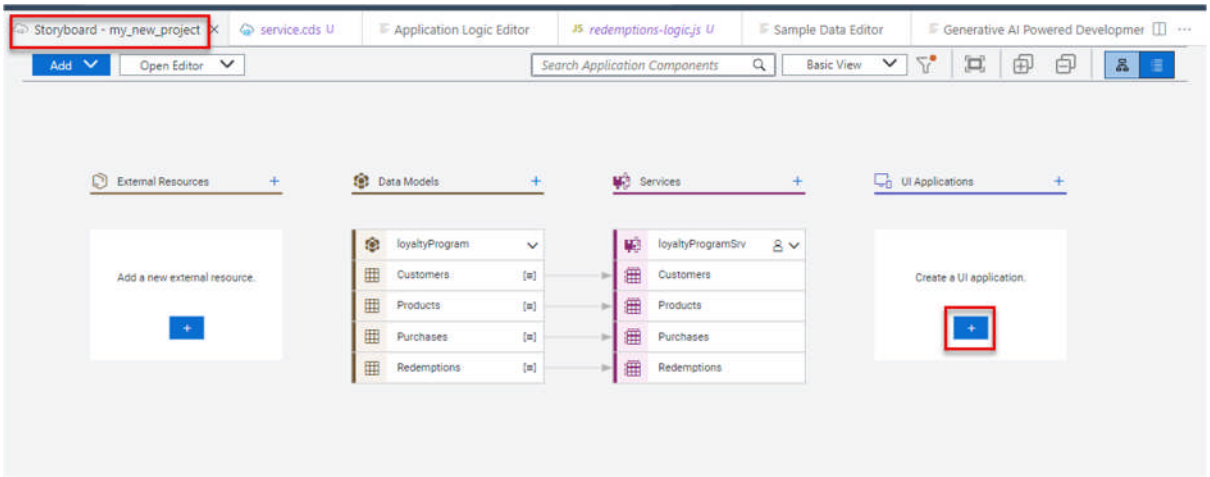
Storyboard - my_new_project | service.cds U | Application Logic Editor | redemptions-logic.js U | Sample Data

```
/**
 *
 * @On(event = { "CREATE" }, entity = "loyaltyProgramSrv.Redemptions")
 * @param {Object} req - User information, tenant-specific CDS model, headers and query parameters
*/
module.exports = async function(req) {
    const { redeemedAmount, customer_ID } = req.data;
    const tx = cds.transaction(req);

    // Fetch the customer's current reward points and redeemed points
    const [{ totalRewardPoints, totalRedeemedRewardPoints }] = await tx.run(
        SELECT.from('loyaltyProgramSrv.Customers')
            .where({ ID: customer_ID })
            .columns('totalRewardPoints', 'totalRedeemedRewardPoints')
    );

    // Check if the customer has enough reward points to redeem
    if (totalRewardPoints < redeemedAmount) {
        req.error(400, 'Insufficient reward points');
        return;
    }

    // Deduct the redeemed amount from the customer's total reward points
    // and add it to their total redeemed points
    await tx.run(
        UPDATE('loyaltyProgramSrv.Customers')
            .set({
                totalRewardPoints: totalRewardPoints - redeemedAmount,
                totalRedeemedRewardPoints: totalRedeemedRewardPoints + redeemedAmount
            })
            .where({ ID: customer_ID })
    );
};
```

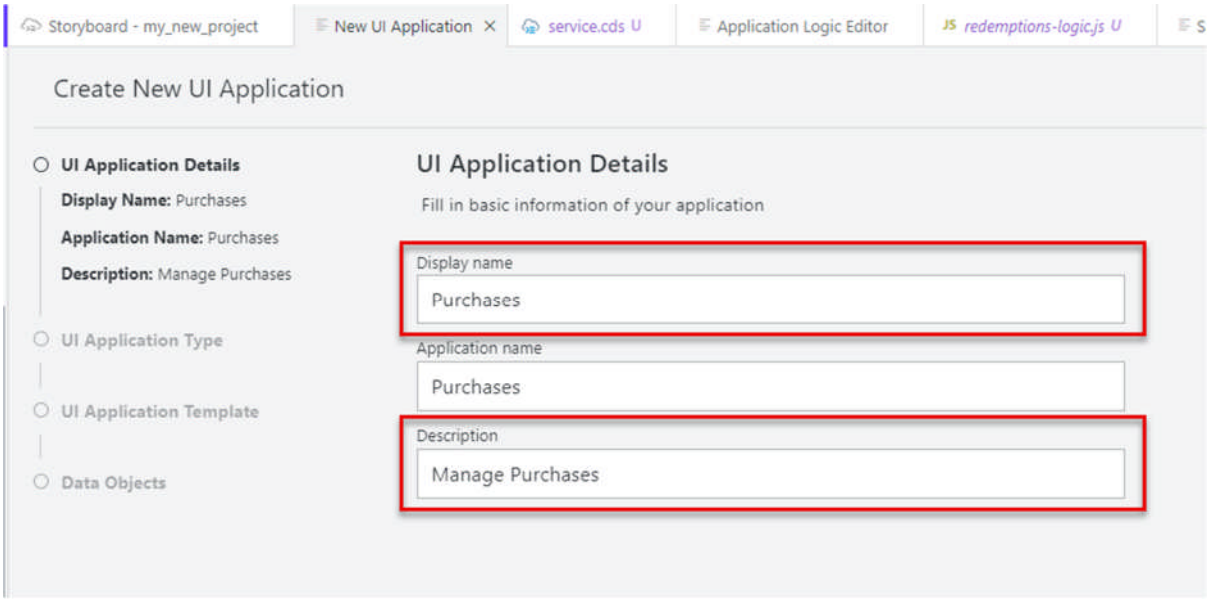
5 Add UI to the Application

To display and test the content we created for the customer loyalty program, we need to create an SAP Fiori elements UI.

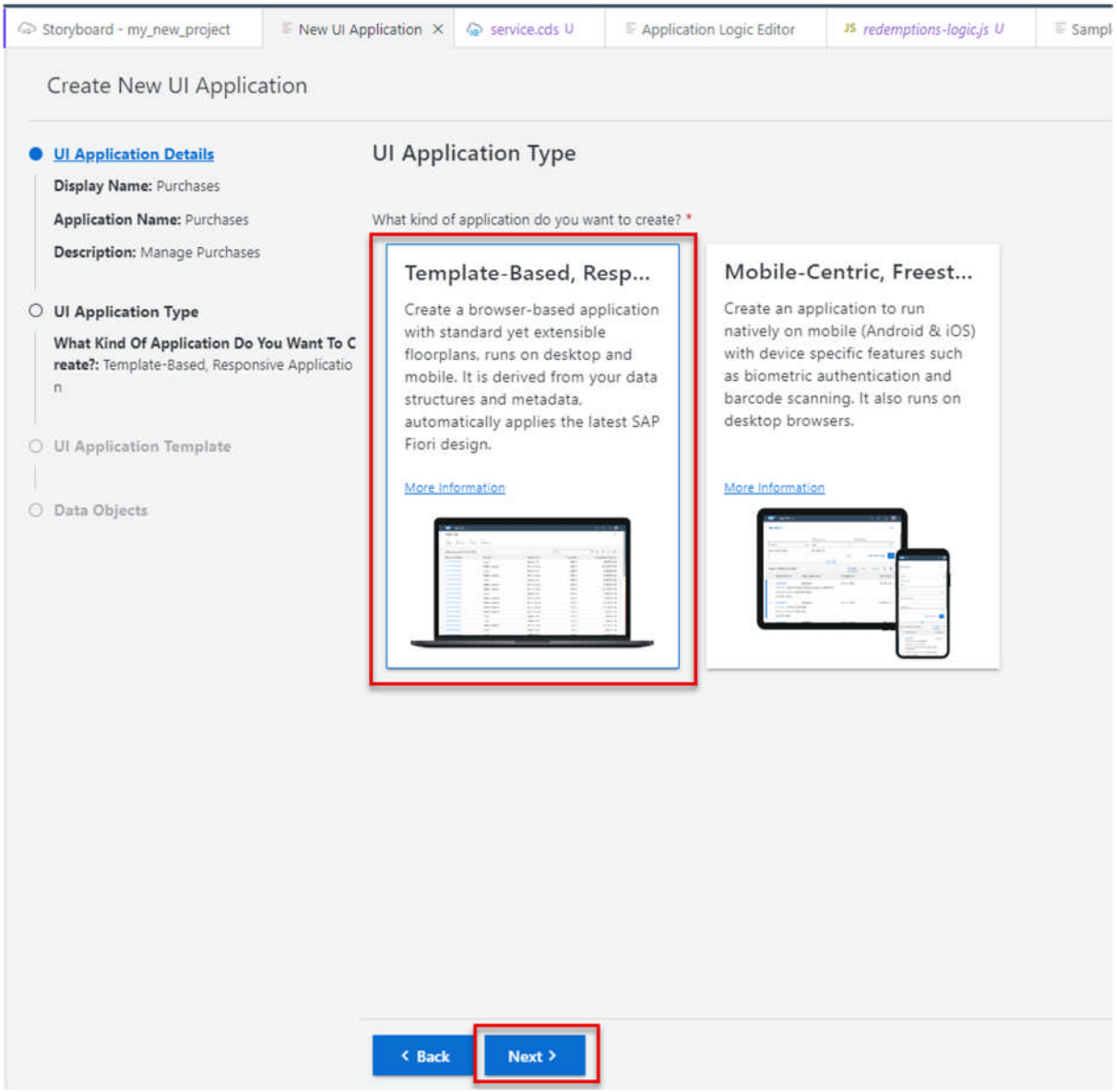
1. Go to back to the Storyboard and add a UI application.



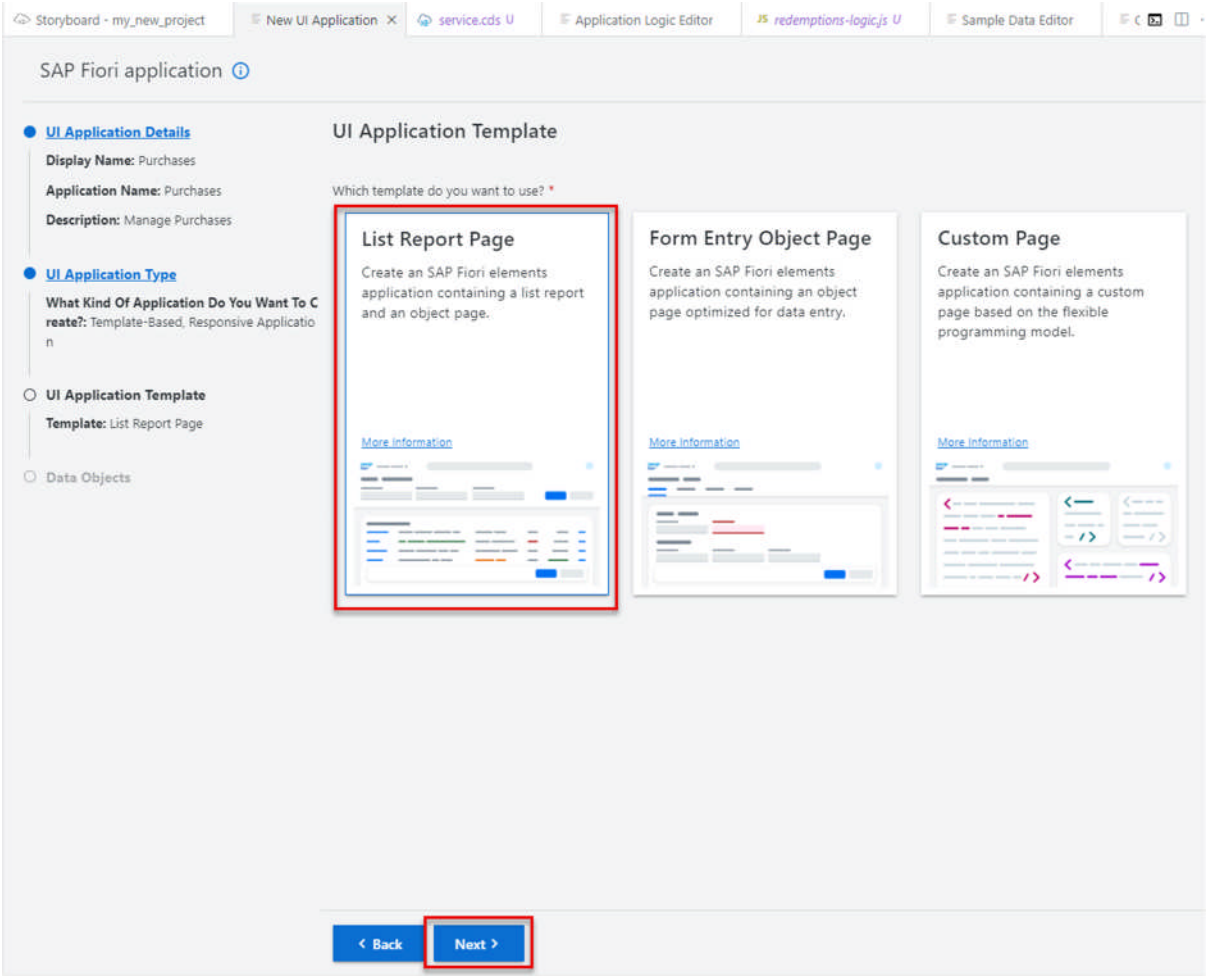
2. We will start with the user interface for the **Purchases** data entity.Set the **Display name** to **Purchases** and the **Description** to **Manage Purchases**, and then click **Next**.



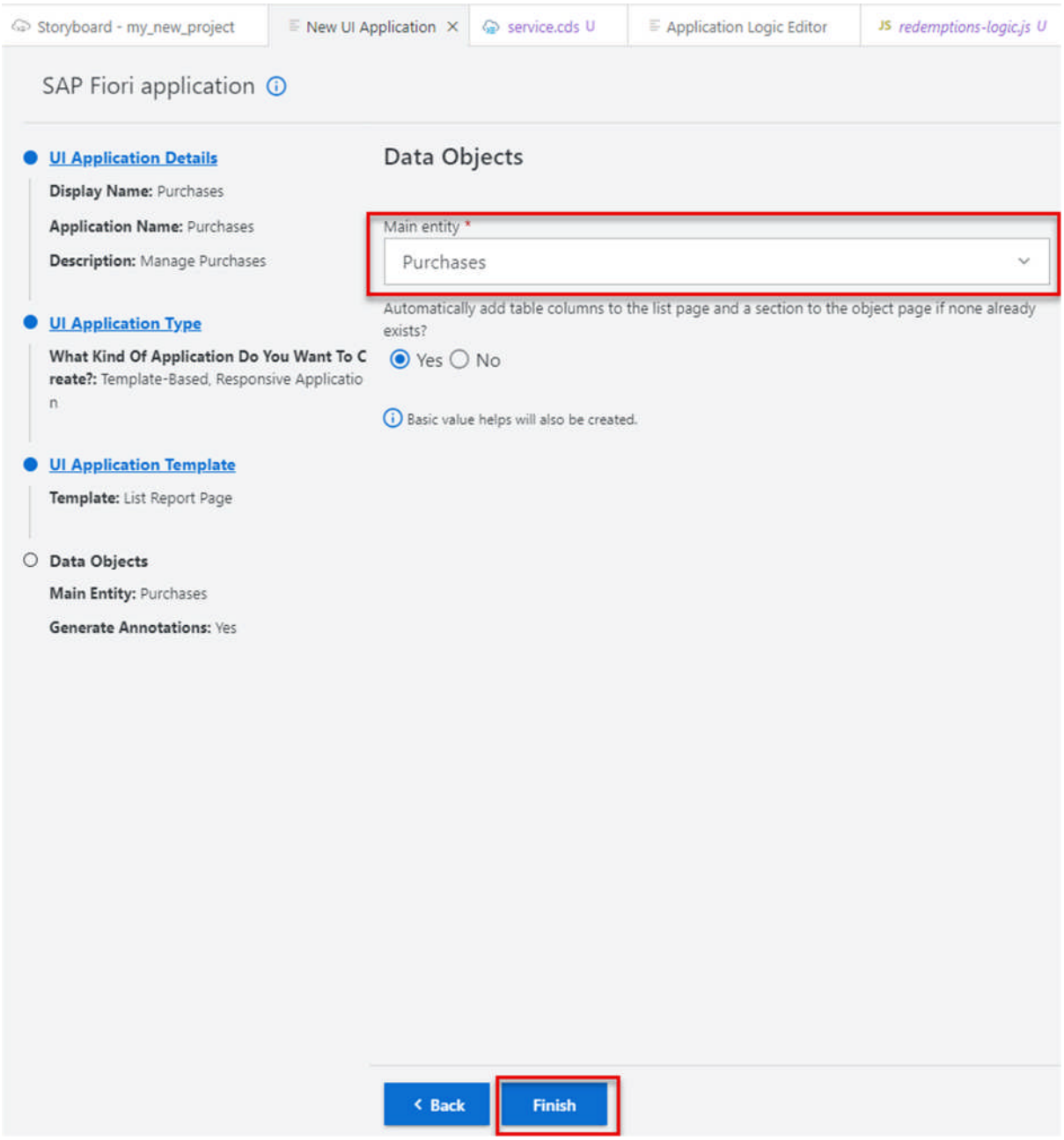
3. We are using the browser, so we will select **Template-Based Responsive Application** as the UI Application type, and click **Next**.


Storyboard - my_new_project
New UI Application
service.cds U
Application Logic Editor
redemptions-logic.js U
Sampl
Create New UI Application
UI Application Details
Display Name: Purchases
Application Name: Purchases
Description: Manage Purchases
UI Application Type
What Kind Of Application Do You Want To Create?: Template-Based, Responsive Application
UI Application Template
Data Objects
UI Application Type
What kind of application do you want to create? *
Template-Based, Resp...
Create a browser-based application with standard yet extensible floorplans, runs on desktop and mobile. It is derived from your data structures and metadata, automatically applies the latest SAP Fiori design.
More Information
Mobile-Centric, Freest...
Create an application to run natively on mobile (Android & iOS) with device specific features such as biometric authentication and barcode scanning. It also runs on desktop browsers.
More Information
Back
Next

4. Select **List Report Page** as the UI application template, and click **Next**.



5. Select **Purchases** as the **Main entity**, and click **Finish**. The page will be created now.



It might take a few moments for the UI to be created because the dependencies need to be installed.

6. Repeat steps 2 through 5 to create additional UI apps for the **Customers** and the **Redemptions** entities.

   Customer:

   - Display name: **Customers**

- Description: **Manage Customers**
- UI Application type: **Template-Based Responsive Application**
- UI Application Template: **List Report Page**
- Main Entity: **Customers**

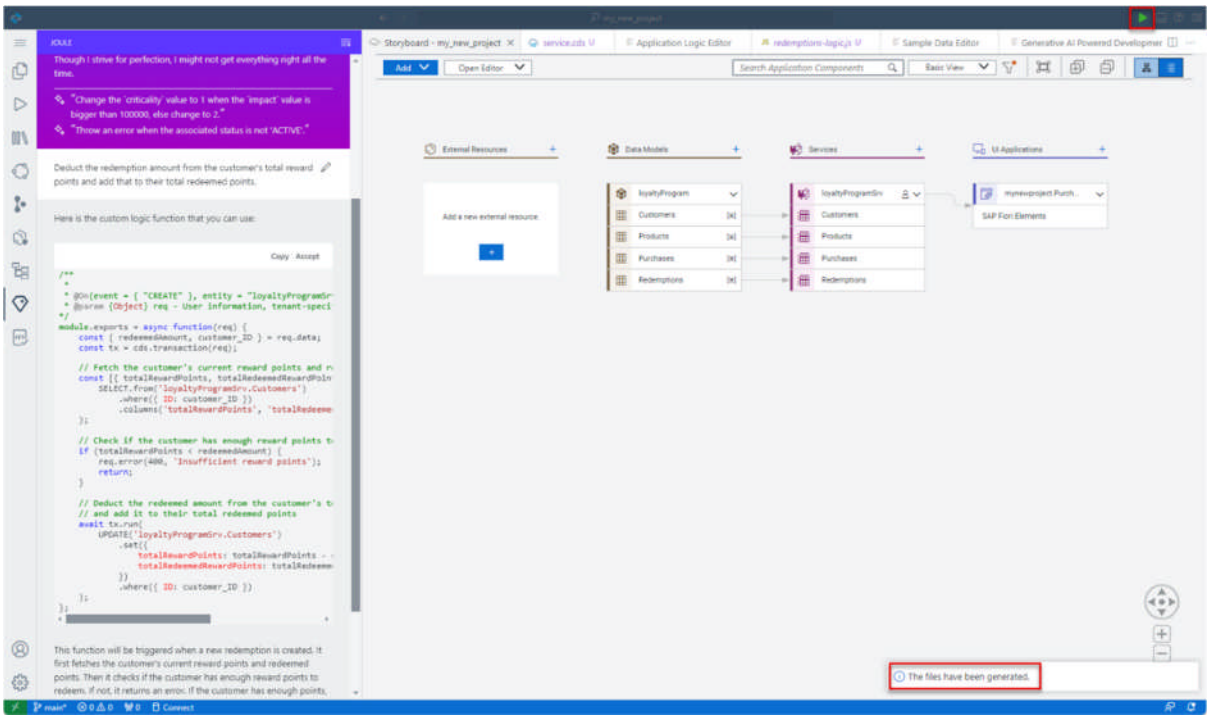
Redemptions:

- Display name: **Redemptions**
- Description: **Manage Redemptions**
- UI Application type: **Template-Based Responsive Application**
- UI Application Template: **List Report Page**
- Main Entity: **Redemptions**

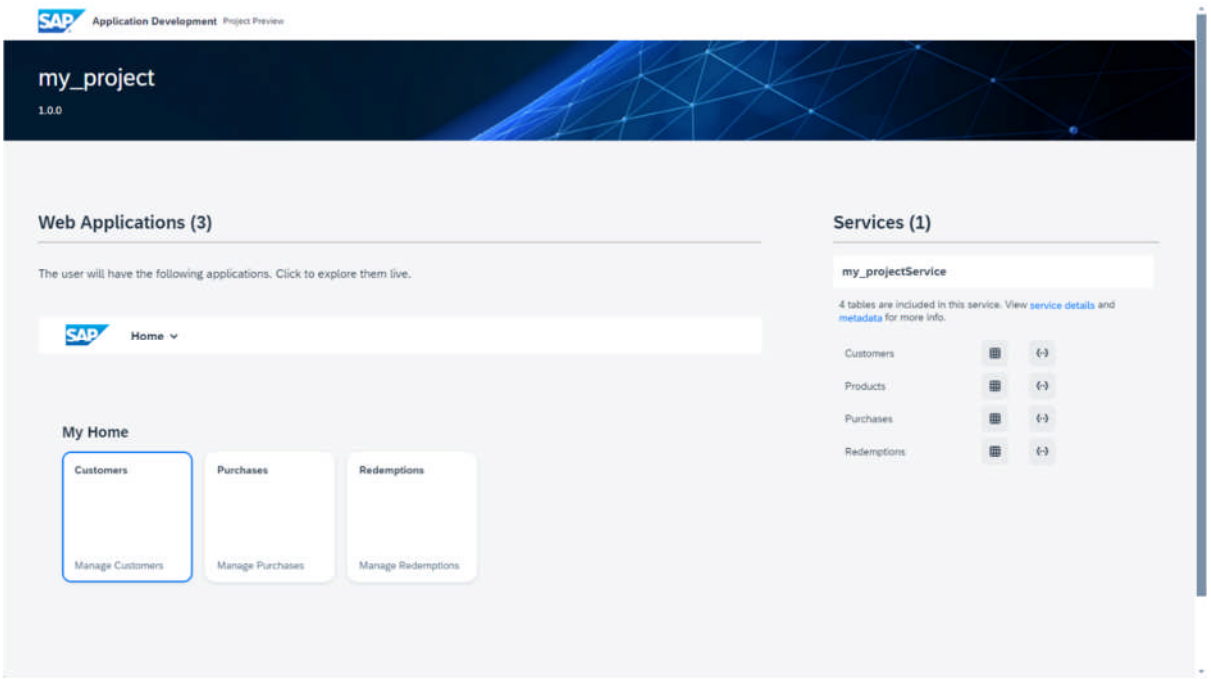
And that's it! You've created an application.

7. To preview your application, once the files have been generated, go to the upper-right corner, and click

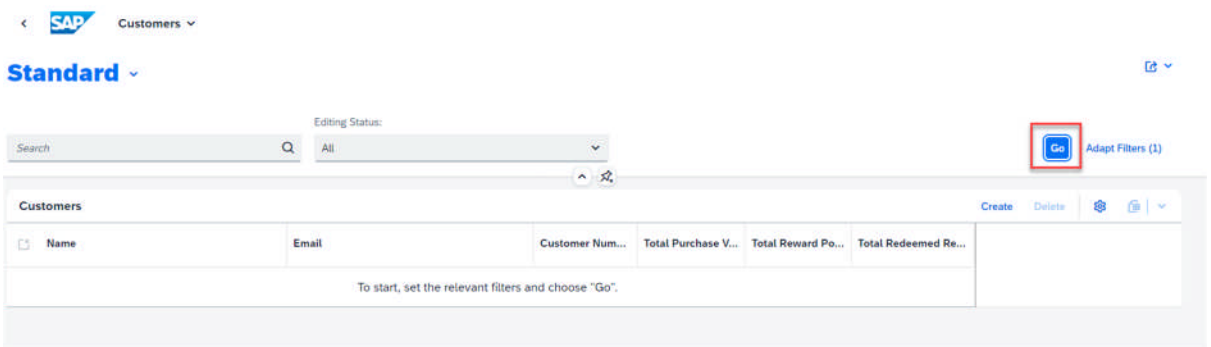
(Run and Debug).



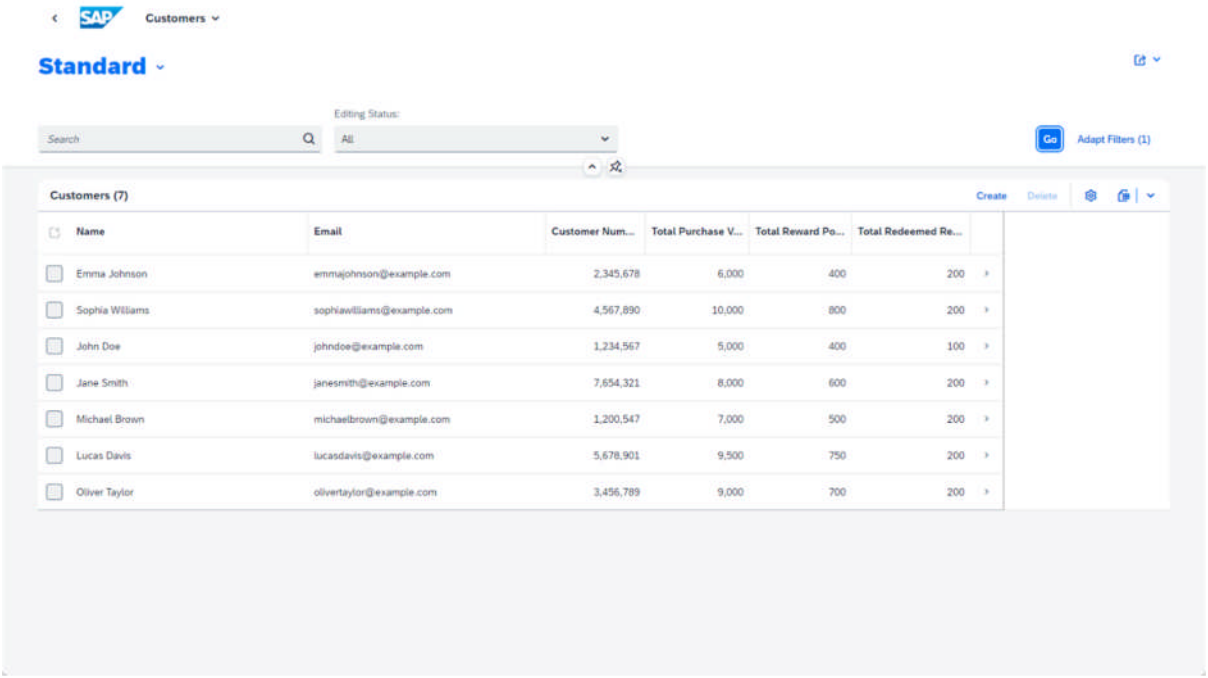
The application's preview is displayed.


Application Development Project Preview
my_project
1.0.0
Web Applications (3)
The user will have the following applications. Click to explore them live.
Home
My Home
Customers
Manage Customers
Purchases
Manage Purchases
Redemptions
Manage Redemptions
Services (1)
my_projectService
4 tables are included in this service. View service details and metadata for more info.
Customers
Products
Purchases
Redemptions

8. Click **Go**.


Customers
Standard
Editing Status:
Search
All
Go
Adapt Filters (1)
Customers
Create
Delete
Name
Email
Customer Num...
Total Purchase V...
Total Reward Po...
Total Redeemed Re...
To start, set the relevant filters and choose "Go".

The customer information is displayed.



9. From the dropdown list at the top of the page, select **Home** to go back and preview the other applications.