



AIと共創！ SAP S/4HANA Cloud拡張開発の最前線

株式会社K Y O S O

宇田川莉央

蝦名亮汰

登壇者について



宇田川 莉央

所属

株式会社K Y O S O

ビジネスソリューション事業部ビジネスソリューショングループ

経歴



蝦名 亮汰

2023



株式会社K Y O S O 新卒入社

2024



SAPプロジェクトに参画し、SAP Fiori アプリケーションを中心に
SAP Business Technology Platform (以下、SAP BTP) での開発業
務に携わっている

2025



SAP BTPでの開発業務に加え、In-Appでの開発（技術検証）にも従事



事業内容について

SAPサービス導入を中心とした、お客様のニーズに応える製品・サービスの開発や構築、維持・運用

業務実績例

2018



SAP BTPビジネス開始（ SAP TechDays 2018 ） ほか

...

2022



Y社（小売業）配達管理システム開発 SAP BTP/SAP HANA/SAP Fiori（Side-by-Side）
G社（製造業）発注システム保守 SAP BTP/SAP HANA/SAP Fiori（Side-by-Side）
M社（素材）組織変更シュミレーション開発
SAP SuccessFactors連携 SAP BTP/SAP Fiori（Side-by-Side）ほか

2023



Y社（小売業）配達管理システム開発 SAP BTP/SAP HANA/SAP Fiori（Side-by-Side）
G社（製造業）発注システム保守 SAP BTP/SAP HANA/SAP Fiori（Side-by-Side）
G社（製造業）物流システム保守 SAP BTP/SAP HANA/SAP Fiori（Side-by-Side）ほか

2024



B社（SIer） SAP BTP/SAP HANA/SAP Fiori PoC開発（Side-by-Side）
T社（印刷業）システム部門向け業務支援システム開発
SAP BTP/SAP HANA/SAP Fiori（Side-by-Side）ほか





1. Jouleの活用がもたらす効果
2. Joule機能のデモ
3. 総括

こんなお困りごとはありませんか？



ABAP開発の経験が浅く、複雑な実装に自信がない。

ABAP開発の経験はあるが新しい記法の学習コストが高く、
実務レベルへの到達に時間がかかる。



UI5は自由度が高い反面、複雑な要件が発生した場合の開発コストが気になる。

Jouleによる開発効率向上に興味があるが、
具体的にどう開発プロセスに組み込めば良いか分からない。



開発者の学習コストや開発工数を削減し、もっと生産性を上げたい。
In-App開発とSide-by-Side開発両方の技術と知見を深め、
最適な提案を実現できる体制を作りたい。

こんなお困りごとはありませんか？



ABAP開発の経験が浅く、複雑な実装に自信がない。

ABAP開発の経験はあるが新しい記法の学習コストが高く、
実務レベルへの到達に時間がかかる。



その課題、Jouleとともに解決しませんか？

Jouleによる開発効率向上に興味があるが、
具体的にどう開発プロセスに組み込めば良いか分からない。



開発者の学習コストや開発工数を削減し、もっと生産性を上げたい。
In-App開発とSide-by-Side開発両方の技術と知見を深め、
最適な提案を実現できる体制を作りたい。

本セッションの目的



「ABAP」の実績がない

In-App開発は経験がない

Side-by-Side開発を
効率化したい



**「Joule」を活用することで、
「In-App・Side-by-Side開発の障壁」を迅速に乗り越え、
ビジネスの選択肢を拡大することができる**

ということを、我々の経験をもとに共有させていただきます

Jouleとは？

SAP社が提供する生成AIデジタルアシスタント

自然言語で表現された要求をもとにユーザをサポートし、ビジネスユーザーの生産性を高めるための支援を提供する

Joule for Developers :

- ✓ Joule 経由で提供される SAP Build および ABAP 向けの組み込み型 AI 機能のコレクション
- ✓ 膨大なABAPコード（39万クラス以上）、CDS（3000万行以上）、SAPドキュメント、ブログ等で学習済み
- ✓ 開発におけるコード生成、既存コードの説明（Explain）、デバッグを強力にサポート
- ✓ 自然言語での対話を通じて、データモデルの定義、サービスロジック、開発を劇的に効率化

利用環境：

- ① SAP S/4HANA Cloud
ABAP development tools for Eclipse
- ② SAP BTP
SAP Build Code

ABAP RESTful application programming modelとは？

SAP HANAに最適化されたODataサービスを効率的に開発するためのアーキテクチャの定義

Generate機能：

- ✓ データベーステーブルに基づいて、ビジネスオブジェクトの土台を高速で自動生成する機能
- ✓ ABAP開発の経験が浅くても、簡単なCRUD操作が出来るAPIとSAP Fioriの画面を作成可能

Generate機能とJouleを組み合わせることで効率的に開発が可能



Jouleの活用がもたらすメリット

①開発者へのメリット



開発効率の劇的向上

Jouleによるコード生成で、
定型的な作業から解放される。



学習コストの最小化

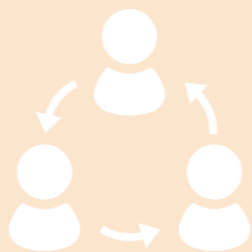
Jouleの支援により、短期間で
In-App開発やSide-by-Side開
発の戦力になることが可能。

ドキュメント検索、
参照時間を約40%※
削減！

※今回のデモアプリ作成における値

Jouleの活用がもたらすメリット

② マネジメント層へのメリット



人材育成の加速

In-App/Side-by-Side
開発者の早期成長を促し、社内の人材リソースを最大化できる。



品質の向上と標準化

属人化を防ぎ、品質が安定する。



開発コストと期間の削減

Jouleの活用による開発効率の向上は、そのままコストと納期に直結する。



ビジネスの選択肢拡大 顧客への貢献

In-App/Side-by-Side
開発の両方の選択肢を持って最適なソリューションを提案・提供できるようになる。

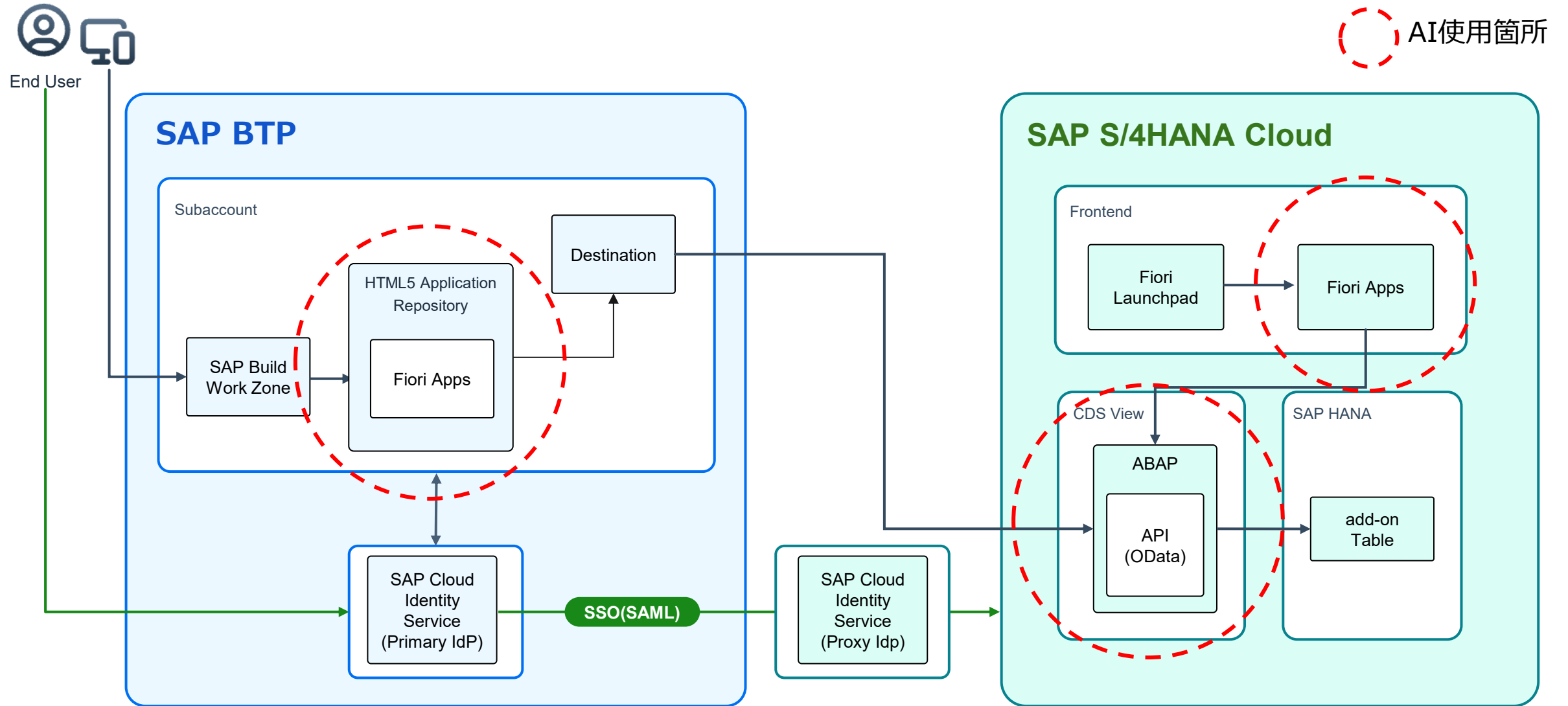
開発工数約30%※削減！

※今回のデモアプリ作成における値

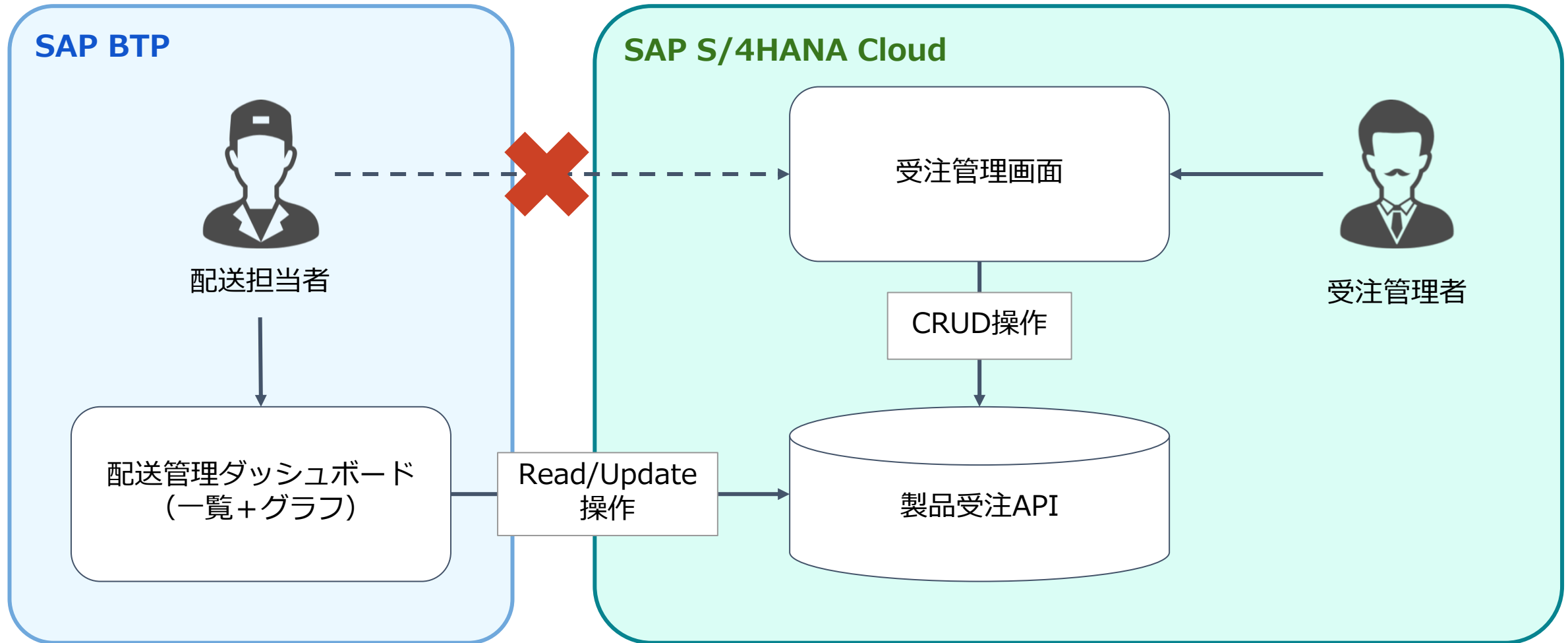


デモ紹介

ユースケースの紹介



ユースケースの紹介



ユースケースの紹介

SAP S/4HANA Cloud : 受注管理画面

Partner Demo DevelopmentD8Q/080

≡

<

Order Management (Demo) ▾

🔍

🔊

?

🔔

ru

Standard* ▾

🔗 ▾

Editing Status:

All ▾

OrderID:

ProductID:

ProductName:

Price:

Cost:

StatusID:

ArealD:

Quantity:

OrderAmount:

GrossProfit:

Description:

OrderedAt:

Created By:

Created On:

Changed By:

Changed On:

Go

Adapt Filters

⬆️⬇️⬅️

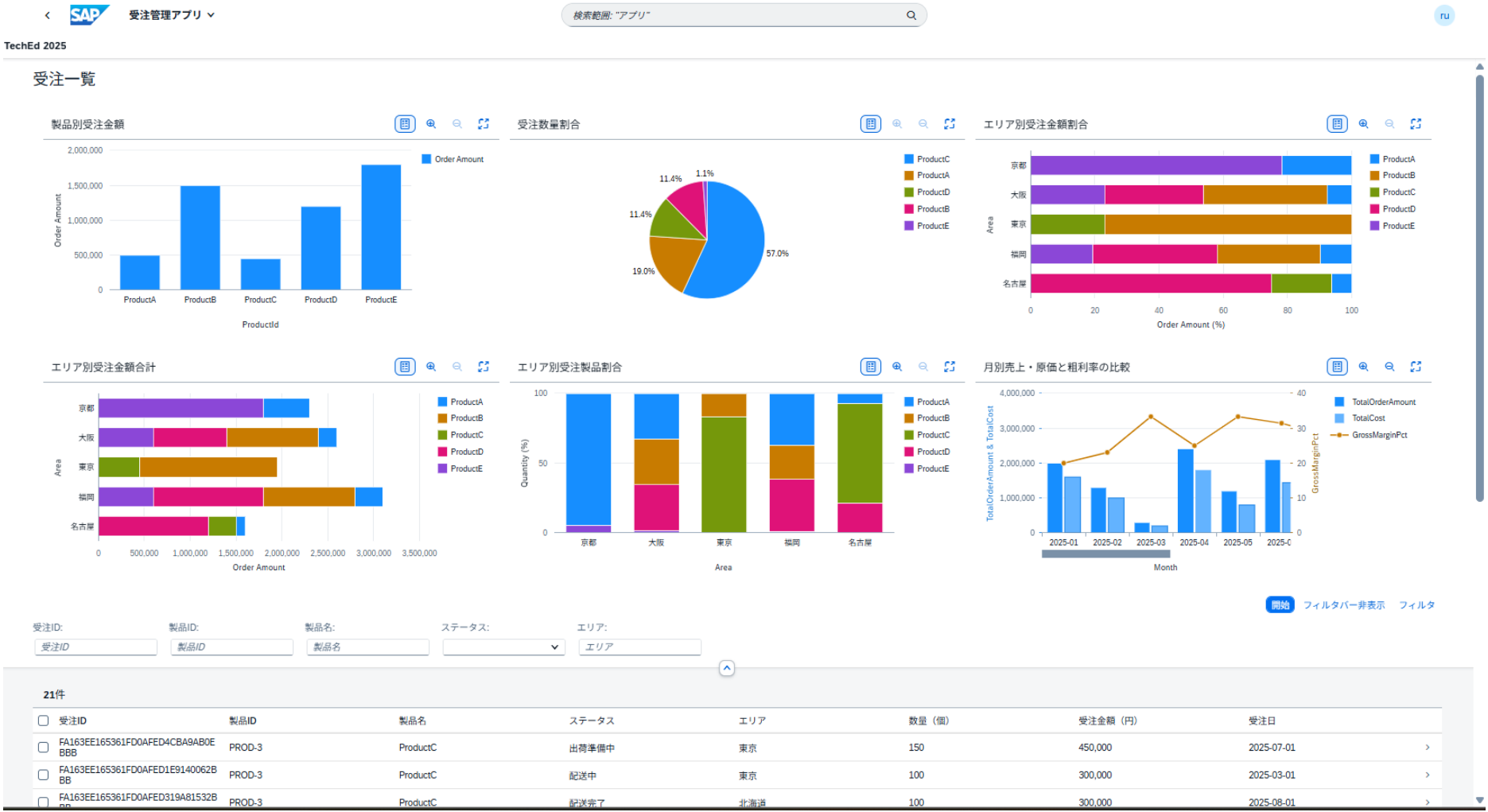
⬆️⬇️⬅️

CreateDelete🔗📄🔍⚙️🔒🔑⋮

<input type="checkbox"/>	OrderID	ProductID	ProductName	Price	Cost	ArealD	Quantity	OrderedAt	
<input type="checkbox"/>	FA163EE165361FD0AFED179D8DF2EBBB	PROD-1	ProductA	10,000	8,000	AREA-1	50	01/01/2025	>
<input type="checkbox"/>	FA163EE165361FD0AFED1AC110B1ABBB	PROD-2	ProductB	50,000	40,000	AREA-2	20	02/01/2025	>
<input type="checkbox"/>	FA163EE165361FD0AFED1E9140062BBB	PROD-3	ProductC	3,000	2,000	AREA-3	100	03/01/2025	>
<input type="checkbox"/>	FA163EE165361FD0AFED24B620F3EBBB	PROD-4	ProductD	40,000	30,000	AREA-4	30	04/01/2025	>
<input type="checkbox"/>	FA163EE165361FD0AFED2817691B2BBB	PROD-5	ProductE	600,000	400,000	AREA-5	1	05/01/2025	>
<input type="checkbox"/>	FA163EE165361FD0AFED2D153D226BBB	PROD-5	ProductE	600,000	400,000	AREA-1	3	06/01/2025	>
<input type="checkbox"/>	FA163EE165361FD0AFED2F638DC3ABBB	PROD-4	ProductD	40,000	30,000	AREA-2	20	07/01/2025	>
<input type="checkbox"/>	FA163EE165361FD0AFED319A81532BBB	PROD-3	ProductC	3,000	2,000	AREA-4	100	08/01/2025	>
<input type="checkbox"/>	FA163EE165361FD0AFED34095B568BBB	PROD-2	ProductB	50,000	40,000	AREA-3	30	08/01/2025	>
<input type="checkbox"/>	FA163EE165361FD0AFED37ED75CDABBB	PROD-2	ProductB	50,000	40,000	AREA-5	20	09/01/2025	>
<input type="checkbox"/>	FA163EE165361FD0AFED3A866879ABBB	PROD-1	ProductA	10,000	8,000	AREA-5	30	10/01/2025	>
<input type="checkbox"/>	FA163EE165361FD0AFED3DDD6FDE2BBB	PROD-1	ProductA	10,000	8,000	AREA-2	20	10/01/2025	>
<input type="checkbox"/>	FA163EE165361FD0AFED4048B10F6BBB	PROD-2	ProductB	50,000	40,000	AREA-3	30	01/01/2025	>
<input type="checkbox"/>	FA163EE165361FD0AFED432C6A5AABBB	PROD-3	ProductC	3,000	2,000	AREA-4	100	02/01/2025	>
<input type="checkbox"/>	FA163EE165361FD0AFED46057CB86BBB	PROD-4	ProductD	40,000	30,000	AREA-5	30	04/01/2025	>
<input type="checkbox"/>	FA163EE165361FD0AFED48227B3FABBB	PROD-5	ProductE	600,000	400,000	AREA-2	1	05/01/2025	>
<input type="checkbox"/>	FA163EE165361FD0AFED4A6B862AABBB	PROD-1	ProductA	10,000	8,000	AREA-1	30	06/01/2025	>
<input type="checkbox"/>	FA163EE165361FD0AFED4CBA9A80EBBB	PROD-3	ProductC	3,000	2,000	AREA-3	150	07/01/2025	>
<input type="checkbox"/>	FA163EE165361FD0AFF474D2C08AEBBB	PROD-1	ProductA	10,000	8,000	AREA-2	10	08/01/2025	>
<input type="checkbox"/>	FA163EE165361FD0B18B5A65D5886BBD	PROD-1	ProductA	10,000	8,000	AREA-4	10	11/11/2025	>
<input type="checkbox"/>	FA163EE165361FD0B2BA5D8804C1ABBD	PROD-1	ProductA	10,000	8,000	AREA-1	10	11/25/2025	>

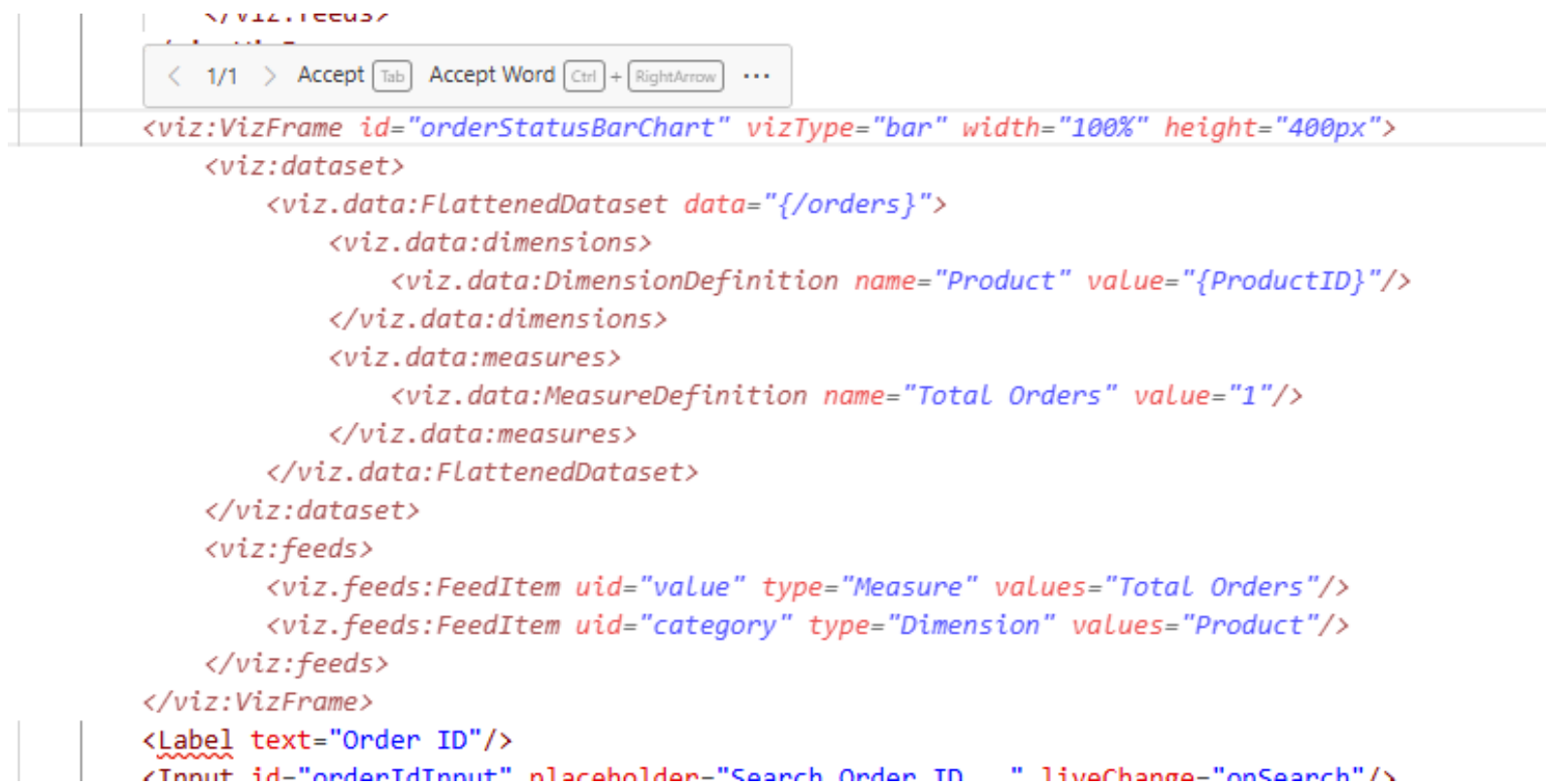
ユースケースの紹介

SAP BTP : 配送管理ダッシュボード



ゴーストテキストについて

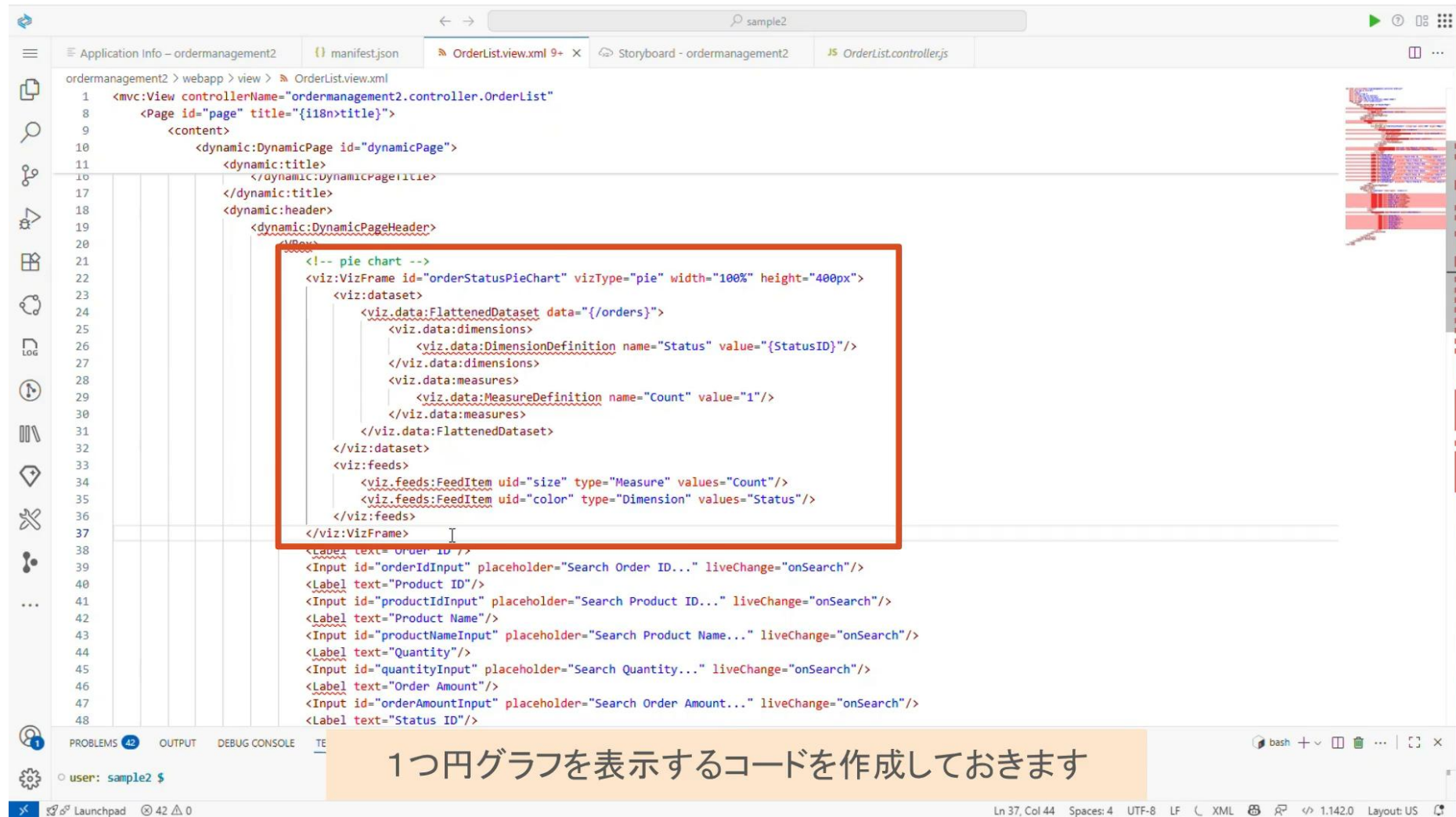
Jouleによるリアルタイムの予測・自動補完機能



```
</viz:VizFrame>
<viz:VizFrame id="orderStatusBarChart" vizType="bar" width="100%" height="400px">
  <viz:dataset>
    <viz.data:FlattenedDataset data="{/orders}">
      <viz.data:dimensions>
        <viz.data:DimensionDefinition name="Product" value="{ProductID}"/>
      </viz.data:dimensions>
      <viz.data:measures>
        <viz.data:MeasureDefinition name="Total Orders" value="1"/>
      </viz.data:measures>
    </viz.data:FlattenedDataset>
  </viz:dataset>
  <viz:feeds>
    <viz.feeds:FeedItem uid="value" type="Measure" values="Total Orders"/>
    <viz.feeds:FeedItem uid="category" type="Dimension" values="Product"/>
  </viz:feeds>
</viz:VizFrame>
<Label text="Order ID"/>
<Input id="orderIdInput" placeholder="Search Order ID" liveChange="onSearch"/>
```

ゴーストテキストについて

SAP Build Codeでの利用例

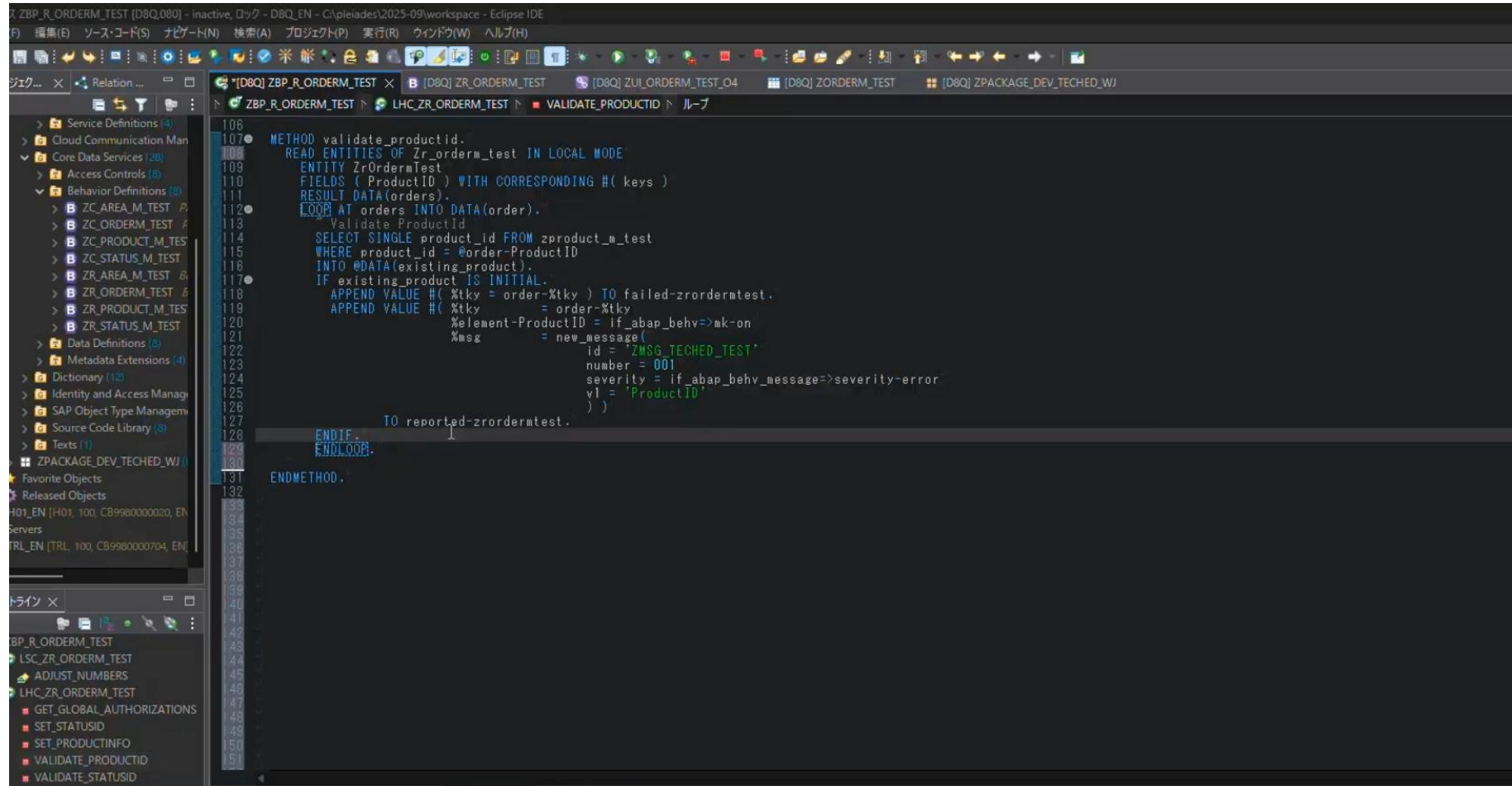


```
1 <mvc:View controllerName="ordermanagement2.controller.OrderList"
8   <Page id="page" title="{i18n>title}">
9     <content>
10       <dynamic:DynamicPage id="dynamicPage">
11         <dynamic:title>
12           <dynamic:DynamicPageTitle>
13             </dynamic:title>
14           </dynamic:title>
15         <dynamic:header>
16           <dynamic:DynamicPageHeader>
17             </dynamic:header>
18           </dynamic:header>
19         <content>
20           <!-- pie chart -->
21           <viz:VizFrame id="orderStatusPieChart" vizType="pie" width="100%" height="400px">
22             <viz:dataset>
23               <viz.data:FlattenedDataset data="{/orders}">
24                 <viz.data:dimensions>
25                   <viz.data:DimensionDefinition name="Status" value="{StatusID}" />
26                 </viz.data:dimensions>
27                 <viz.data:measures>
28                   <viz.data:MeasureDefinition name="Count" value="1" />
29                 </viz.data:measures>
30               </viz.data:FlattenedDataset>
31             </viz:dataset>
32             <viz:feeds>
33               <viz.feeds:FeedItem uid="size" type="Measure" values="Count" />
34               <viz.feeds:FeedItem uid="color" type="Dimension" values="Status" />
35             </viz:feeds>
36           </viz:VizFrame>
37           <Label text="Order ID" />
38           <Input id="orderIdInput" placeholder="Search Order ID..." liveChange="onSearch" />
39           <Label text="Product ID" />
40           <Input id="productIdInput" placeholder="Search Product ID..." liveChange="onSearch" />
41           <Label text="Product Name" />
42           <Input id="productNameInput" placeholder="Search Product Name..." liveChange="onSearch" />
43           <Label text="Quantity" />
44           <Input id="quantityInput" placeholder="Search Quantity..." liveChange="onSearch" />
45           <Label text="Order Amount" />
46           <Input id="orderAmountInput" placeholder="Search Order Amount..." liveChange="onSearch" />
47           <Label text="Status ID" />
48         </content>
49       </dynamic:DynamicPage>
50     </content>
51   </Page>
52 </mvc:View>
```

1つ円グラフを表示するコードを作成しておきます

ゴーストテキストについて

ABAP development tools for Eclipseでの利用例



```
106 METHOD validate_productid.  
107 READ ENTITIES OF Zr_orderm_test IN LOCAL MODE  
108 ENTITY ZrOrdermTest  
109 FIELDS ( ProductID ) WITH CORRESPONDING #( keys )  
110 RESULT DATA(orders).  
111 LOOP AT orders INTO DATA(order).  
112   Validate ProductId  
113   SELECT SINGLE product_id FROM zproduct_m_test  
114   WHERE product_id = @order-ProductID  
115   INTO @DATA(existing_product).  
116   IF existing_product IS INITIAL.  
117     APPEND VALUE #( %tky = order-%tky ) TO failed-zrordermtest.  
118     APPEND VALUE #( %tky = order-%tky  
119                     %element-ProductID = if_abap_behv=>mk-on  
120                     %msg = new_message(  
121                       id = 'ZMSG_TECHED_TEST'  
122                       number = 001  
123                       severity = if_abap_behv_message=>severity-error  
124                       v1 = 'ProductID'  
125                     ) )  
126   TO reported-zrordermtest.  
127 ENDIF.  
128 ENDOLOOP.  
129  
130 ENDMETHOD.  
131  
132
```



① 繰り返し作業の効率化


前後の文脈を読み取ってコードが提案されるため一貫性を保つために活用




② 構文の使い方のヒントにする



正しい構文を理解するために活用

既存のコードや複雑な処理を、Jouleが自然な言葉で説明してくれる（翻訳）機能

 You

 Joule ui5




FILES USED

 OrderList.controller.js  OrderList.view.xml

Certainly! The `onSearch` method in the `OrderList.controller.js` file is responsible for filtering the items displayed in the order table based on user input from several search fields. Here's a detailed explanation of how it works:

- 1. Retrieve Input Values:** The method starts by retrieving the values entered by the user in various input fields. These fields correspond to different attributes of an order, such as Order ID, Product ID, Product Name, Quantity, Order Amount, Status ID, Area ID, and Ordered At. Each input field is accessed using the `byId` method, which fetches the UI element by its ID.
- 2. Get Table Binding:** The method then retrieves the binding of the items aggregation of the order table using `getBinding("items")`. This binding represents the data model that the table is displaying.
- 3. Create Filters:** An array named `filters` is initialized to store the filter objects. For each input field, if the user has entered a value, a new filter is created using the `Filter` class. The filter checks if the corresponding property in the data model contains the input value. The `FilterOperator.Contains` is used to perform a "contains" search, meaning it will match any item where the property includes the input value.
- 4. Apply Filters:** Once all relevant filters are created based on the user's input, they are applied to the table's binding using the `filter` method. This updates the table to only display items that match the filter criteria.

Overall, the `onSearch` method dynamically filters the table's data based on user input, allowing for a more interactive and responsive user experience. Users can search for specific orders by entering criteria in the input fields, and the table will update to show only the matching results.

説明機能について

SAP Build Codeでの利用例

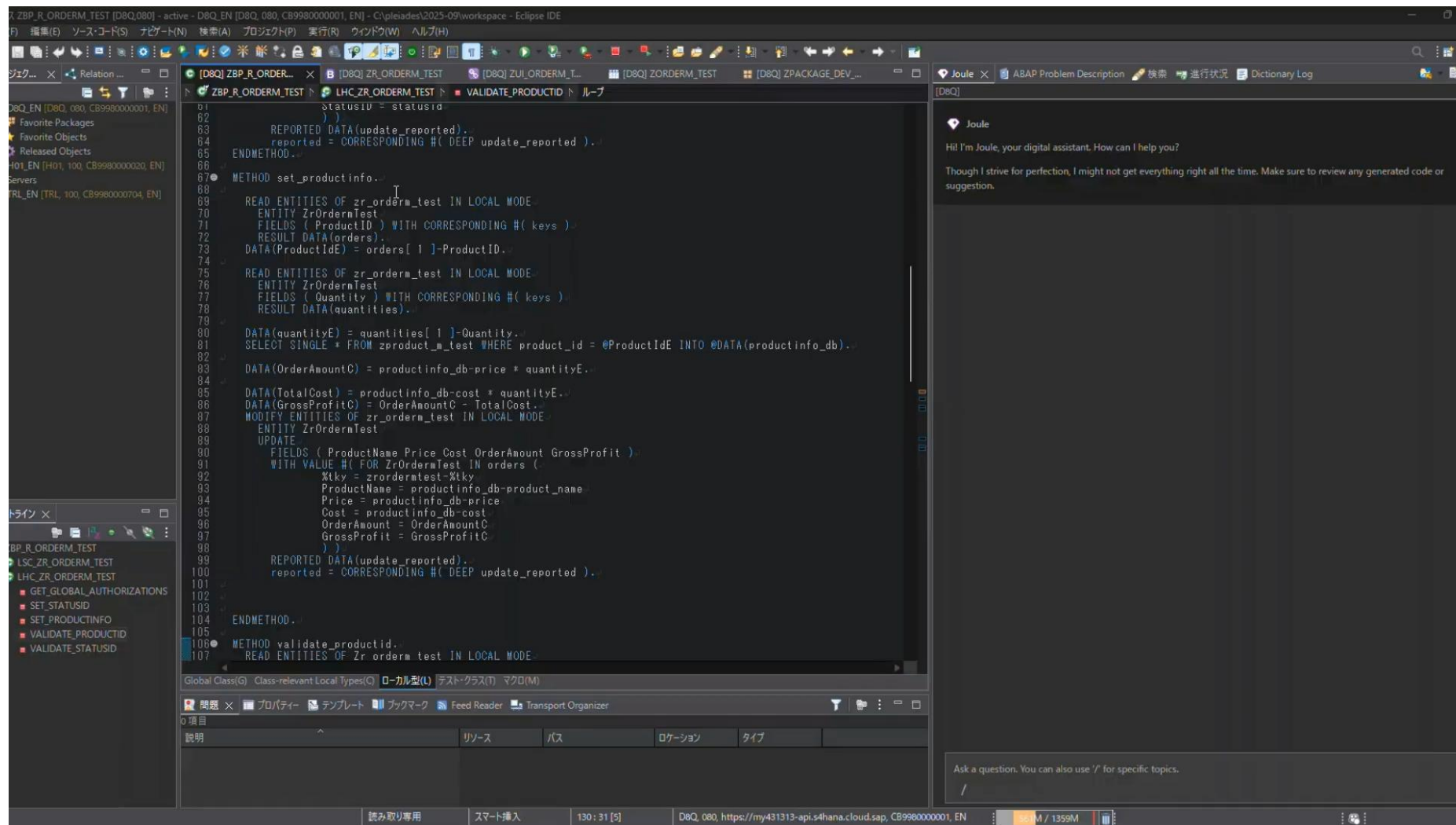
The screenshot displays the SAP Build Code IDE interface. On the left, the 'Joule' assistant is active, showing a chat window with the text: 'Hi! I'm Joule, your digital assistant. How can I help you? In your prompt, enter: / at the beginning for commands # to reference a file from your context. Though I strive for perfection, I might not get everything right all the time.' Below the chat, a list of commands is visible, including '/code-search Search the repository', '/fiori-gen-cap-ui Generate SAP Fiori Elements UI from Storyboard', and others. A red box highlights the input field with the placeholder text 'Type "/" for Joule's commands'. The main editor area shows a JavaScript file named 'OrderList.controller.js' with the following code:

```
ordermanagement2 > webapp > controller > JS OrderList.controller.js > sap.ui.define() callback > onSearch
6 ], (Controller, UIComponent, Filter, FilterOperator) => {
13
14 * onSearch(event) {
15     const orderIdQuery = this.byId("orderIdInput").getValue();
16     const productIdQuery = this.byId("productIdInput").getValue();
17     const productNameQuery = this.byId("productNameInput").getValue();
18     const quantityQuery = this.byId("quantityInput").getValue();
19     const orderAmountQuery = this.byId("orderAmountInput").getValue();
20     const statusIdQuery = this.byId("statusIdInput").getValue();
21     const areaIdQuery = this.byId("areaIdInput").getValue();
22     const orderedAtQuery = this.byId("orderedAtInput").getValue();
23
24     const table = this.byId("orderTable");
25     const binding = table.getBinding("items");
26     const filters = [];
27
28     if (orderIdQuery) {
29         filters.push(new Filter("OrderID", FilterOperator.Contains, orderIdQuery));
30     }
31     if (productIdQuery) {
32         filters.push(new Filter("ProductID", FilterOperator.Contains, productIdQuery));
33     }
34     if (productNameQuery) {
35         filters.push(new Filter("ProductName", FilterOperator.Contains, productNameQuery));
36     }
37     if (quantityQuery) {
38         filters.push(new Filter("Quantity", FilterOperator.Contains, quantityQuery));
39     }
40     if (orderAmountQuery) {
41         filters.push(new Filter("OrderAmount", FilterOperator.Contains, orderAmountQuery));
42     }
43     if (statusIdQuery) {
44         filters.push(new Filter("StatusID", FilterOperator.Contains, statusIdQuery));
45     }
46     if (areaIdQuery) {
47
48
```

An orange callout box at the bottom right of the code editor contains the text: '"/ui5コマンドを使用して「○○を説明してください」というプロンプトを投げます'.

説明機能について

ABAP development tools for Eclipseでの利用例





① 未経験者の学習をサポート

ABAP/SAPUI5のコードの理解や新構文の学習を促進するために活用



② チーム内の連携

チーム内の認識のズレの防止やチームメンバー間のコードレビューに活用

その他のJoule利用時※のポイント



①英語でプロンプトを作成

日本語よりも英語でプロンプトを作成することで、精度の高い回答を取得



②プロンプトの標準フォーマットの作成・活用

どのメンバーが使用しても一定の品質を維持したコード生成を実現

※今回のデモアプリ作成におけるポイント

- ✓ Jouleは開発者の能力を増幅させ、開発効率を最大化する「パートナー」
- ✓ AIの活用により、顧客のビジネス価値を高める「本来の設計・実装」という、より創造的な業務に集中できる

皆さんもJouleを利用してみませんか？

私たちは「Joule」を活用することで、ビジネス拡大を目指していきます！

今後のSAPシステム開発について



Join the global
dev community

- SAP TechEd Berlin2025に参加
 - ✓ SAP独自のAIモデル “SAP-RPT-1”
 - ✓ 流行りのエージェントによるアプリ開発をSAPでも可能にするMCPサーバーの提供開始
 - ✓ カスタムAIエージェントを開発可能なJoule Studio がリリース
 - ✓ ABAP LLMモデルがGenerative AI Hubで提供開始
- etc.

技術発信について

SAP BTPのSide-by-Side開発を中心とした技術記事を投稿しています。
今回のセッションに関連する記事も投稿予定のため、「**KYOSO 受託開発サービス**」で
検索をお願いいたします。



<https://www.solkyoso.biz/>

KY@SO