



INTERNAL – SAP AND PARTNER USE ONLY



## SAP S/4HANA Cloud, Public edition ABAP 拡張ブート キャンプ

### 演習 3: Side-by-Side 拡張のためのカスタム API の開発

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

© 2022 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/trademark](http://www.sap.com/trademark) for additional trademark information and notices.



# 目次

イントロダクション.....	3
演習のスコープ.....	3
前提条件 .....	4
ヒント .....	4
ステップ 1: API の作成.....	5
ステップ 1.1: 受信用 API の作成 .....	5
ステップ 1.2: 受信用 API の公開 .....	9
ステップ 2: API の取込み .....	13
ステップ 2.1: Service Consumption Model の作成 .....	13
ステップ 2.2: HTTP Service の作成.....	18
ステップ 2.3: HTTP Service の公開.....	25
ステップ 2.4: API のテスト .....	28
ご参考: HTTP Service 実装クラス”ZCL_SHOP_API_READ_JXX” コードサンプル .....	29

## イントロダクション

このハンズオンワークショップでは、S/4HANA Cloud ABAP 環境システム (S4HC) で(On-Stack)開発者拡張を構築する方法について説明します。

開発者拡張により、S/4HANA Cloud システムでカスタム API を作成できるようになりました。これらは受信 API と送信 API の両方があり、外部システムとの通信を確立することができます。開発者は Eclipse 版 ABAP 開発ツール (ADT) を使用して、このような API を実装することが可能です。

ただし外部システムと受信/送信通信する場合は、管理者が対応する通信管理を実行して、目的の API を使用できるようにする必要があることに注意してください。

カスタム受信 API を開発する場合、OData、プレーン HTTP、または RFC 通信を使用するオプションがあります。

このようなカスタム API は、SAP Business Technology Platform などからリモートで呼び出すことができます。Side-by-Side 拡張シナリオでは、S/4HANA Cloud のカスタム API を SAP BTP 上の ABAP クラウドシステムを呼び出すことが可能になります。

このチュートリアルでは、XX が表示されるすべての場所で、自分に割り当てられた番号 (00 など) を使用します。その際は ADT の Find/Replace 機能(メニューの“Edit”→“Find/Replace”) を使用して置き換えて下さい。

## 演習のスコープ

演習 3 では、S/4HANA Cloud システムでカスタム受信 OData API を登録します。演習 2 で作成したビジネスオブジェクト("online\_shop")を再利用し、その上に Web API タイプの OData サービスバインディングを登録します。さらにカスタム通信シナリオを作成して API を公開します。

実際のビジネスケースでは、開発テナント (クライアント 080) で開発が実行され、通信管理はカスタマイジングテナント (クライアント 100) で実行されます。つまりより完全なビジネスコンフィグレーションとデータが含まれているカスタマイジングテナントからカスタム API が呼び出されることになります。ですが本演習では、技術習得の目的のために開発テナントですべてのアクティビティを実行します。

演習では SAP BTP ABAP Environment から API を呼び出します。この環境上で API のメタデータを使用してサービス利用モデルを作成することができます。これは ABAP コーディングから直接呼び出すことができる API となります。簡単な例として、(SAP BTP ABAP Environment 上の)プレーンな HTTP サービスから API を呼び出します。このサービスでは、ハンドラクラスにカスタムコーディングが含まれます。

この HTTP サービスは送信通信の一種であるため、通信管理も必要です。そのため、別のカスタム通信シナリオを作成して、送信サービスを公開します。

注記:

サービス利用モデルでは、API の仕様が分析され、API のモデル化に必要な ABAP 開発オブジェクトが生成されます。これにより、ABAP コーディングを使用した API の簡単な利用が可能になります。ADT のサービス利用モデルオブジェクトページには、API でサポートされるさまざまな操作用のコードスニペットがいくつか用意されています。これらは、独自の実装のスタートポイントとして使用することができます。

## 前提条件

- 事前に SAP ABAP Developer Tools (ADT) のバージョン 3.16 以降を参加者の利用するコンピューター上にインストールいただく必要があります。
- 参加者は SAP Standard および ABAP RESTful Application Programming Model (RAP) のコンセプトをご理解いただいていること。
- “演習 2: RAP による簡易オンラインショップアプリの開発”が完了していること。

## ヒント

コード完了機能 (ショートカット: Ctrl+Space) または本ドキュメントに記載されたコードスニペットを使用することでコーディングを素早く行うことができるようになります。またはショートカット Ctrl+Shift+A を使用して、オブジェクトを簡単に開くことができます。本ワークショップの受講にあたっては、これらの機能をご活用ください。

※本セッションは、2/7,8 に Global(APJ)開催の “SAP S/4HANA Cloud, public edition Developer Extensibility - Hands on Bootcamp”の中で、S/4HC プロジェクト導入関係者全員が拡張開発部分の基礎知識習得を目的に日本向けに再構成したものです。

## ステップ 1: API の作成

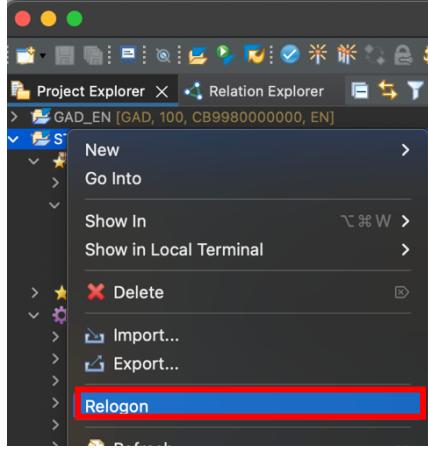
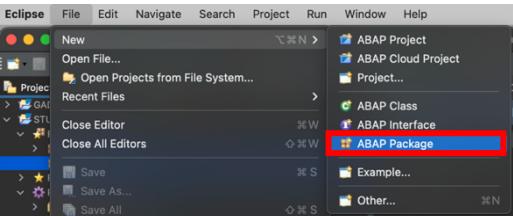
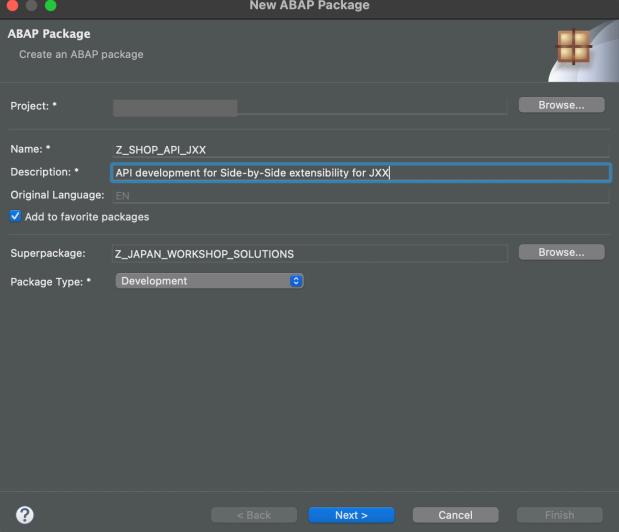
SAP S/4HANA Cloud 上で送信用の OData API を作成して公開します。

### ステップ 1.1: 受信用 API の作成

Web API タイプの OData サービスバインディングを作成して、さらに通信管理に必要な開発オブジェクトを作成します。

#### 手順

#### スクリーンショット/コード

<p>ADT を起動し、ABAP Project を選択して S/4HANA Cloud ヘアクセスします。 (右のスクリーンショットでは ABAP Cloud Project を選択して“Relogon”から再度 S/4HANA Cloud 環境にログオンしています。) *ADT の起動から S/4HANA Cloud へのログオンに関しては演習 1 ステップ 2: SAP ABAP Development Tool (ADT)へのアクセスをご参照ください。</p>	
<p>ADT のメニューから“File” → “New” → “ABAP Package”を選択します。</p>	
<p>ABAP Package を以下のように設定します。</p> <ul style="list-style-type: none"><li>● Name: Z_SHOP_API_JXX</li><li>● Description: API development for Side-by-Side extensibility for JXX</li><li>● Add to Favorite packages : チェックを付けてください。</li><li>● Superpackage: Z_JAPAN_WORKSHOP_SOLUTIONS</li></ul> <p>注意:XXの部分については、<u>全てご自身に割り振られた番号に変更して下さい。</u></p> <p>2 回“Next”ボタンを押してください。</p>	

次ページに続く

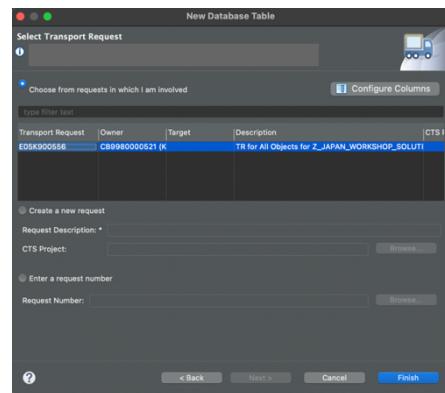
## 手順

## スクリーンショット/コード

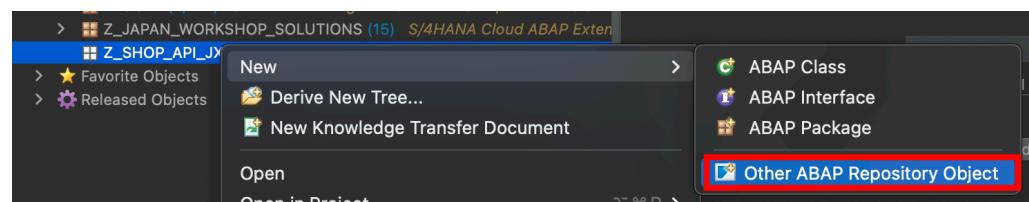
Select Transport Request の画面では以下のように設定してください。

- Choose from requests in which I am involved:  
チェックしてください。
- Requests List:  
Package 作成時の Transport Request を選択して下さい。

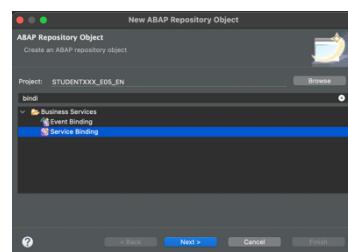
"Finish"ボタンを押すと ABAP Package が作成されます。



ABAP Package  
"Z\_SHOP\_API\_JXX"を選択し、右クリックして"New"→"Other ABAP Repository Object"を選択します。  
注意:XX もしくは xx の部分については、全てご自身に割り振られた番号になります。



New ABAP Repository Object のポップアップ画面から"Service Binding"を検索の上、選択して"Next"ボタンを押して下さい。

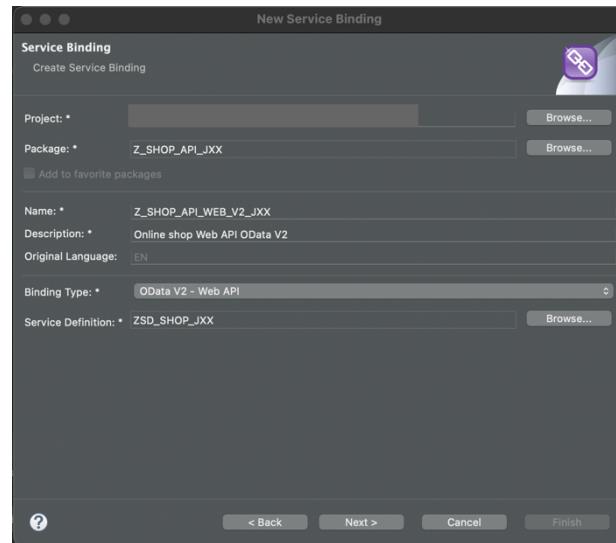


Service Binding を以下のように設定して下さい。

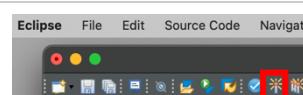
- Name:  
Z\_SHOP\_API\_WEB\_V2\_JXX
- Description: Online shop  
Web API OData V2
- Binding Type: OData V2-Web API
- Service Definition:  
ZSD\_SHOP\_JXX (演習 2 でご自身が作成したもの)

注意:XX もしくは xx の部分については、全てご自身に割り振られた番号になります。

"Next"ボタンを押して下さい。



Service Binding を"Activate"して下さい。

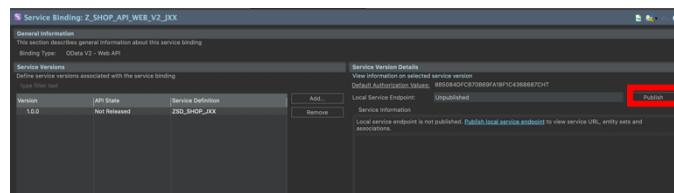


次ページに続く

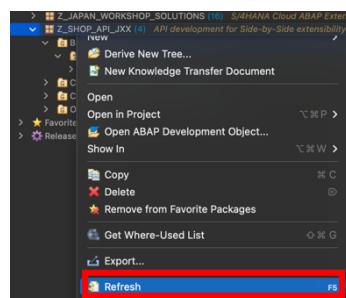
## 手順

## スクリーンショット/コード

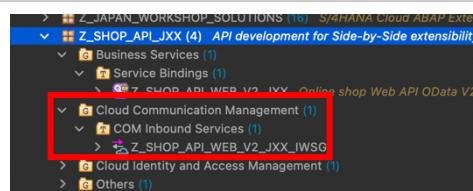
Service Binding 上の“Publish”ボタンを押して、サービスを S/4HANA にパブリッシュして下さい。



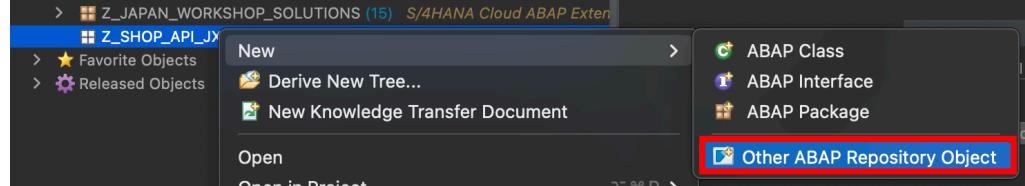
パブリッシュが完了した後に Package を選択して“Refresh”ボタンを押して下さい。  
ご注意: パブリッシュには少し時間がかかります。



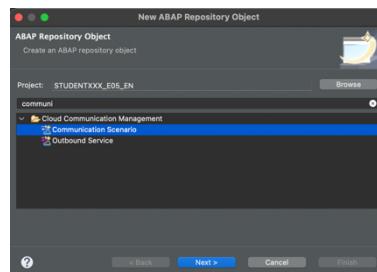
Cloud Communication Management フォルダの下に COM Inbound Service オブジェクトが作成されていることを確認して下さい。



ABAP Package “Z\_SHOP\_API\_JXX”を選択し、右クリックして“New”→“Other ABAP Repository Object”を選択します。  
注意:XX もしくは xx の部分については、全てご自身に割り振られた番号になります。



New ABAP Repository Object のポップアップ画面から“Communication Scenario”を検索の上、選択して“Next”ボタンを押して下さい。



次ページに続く

## 手順

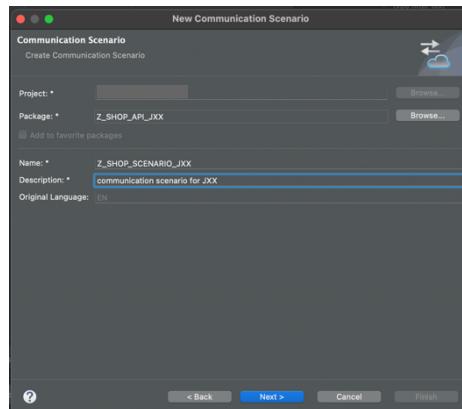
## スクリーンショット/コード

Communication Scenario を以下のように設定して下さい。

- Name: Z\_SHOP\_SCENARIO\_JXX
- Description: Communication Scenario for JXX

注意:XX もしくは xx の部分については、全てご自身に割り振られた番号になります。

“Next”ボタンを押して下さい。

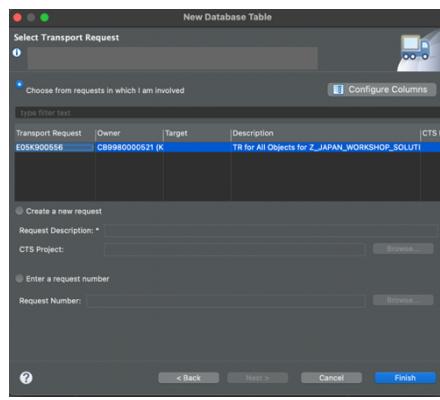


Select Transport Request の画面では以下のように設定してください。

- Choose from requests in which I am involved: チェックしてください。
- Requests List: Package 作成時の Transport Request を選択して下さい。

“Finish”ボタンを押すと

Communication Scenario が作成されます。



Communication Scenario の”Overview”タブを開いて以下のように設定して下さい。

- Allowed Instances: One instance per client

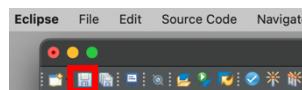


Communication Scenario の”Inbound”タブを開いて以下のように設定して下さい。

- Basic Authorization: チェックしてください。
- Inbound Services: “Add”ボタンを押して先に作成した COM Inbound Service “Z\_SHOP\_API\_WEB\_V2\_JXX”を設定してください。



Communication Scenario を“Save”して下さい。



次ページに続く

## 手順

Communication Scenario の“Overview”タブを開いて“Locally Publish”ボタンを押してください。

Communication Scenario を S/4HANA Cloud にパブリッシュが開始されます。  
ご注意: パブリッシュには少し時間がかかります。

## スクリーンショット/コード



## ステップ 1.2: 受信用 API の公開

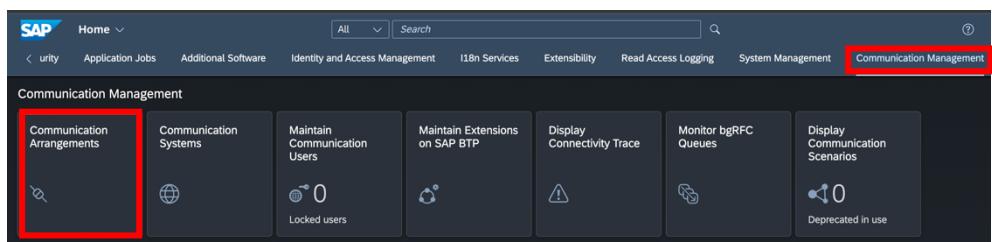
S/4HANA Cloud の管理者として、通信管理に必要な設定して受信用 API を公開します。

## 手順

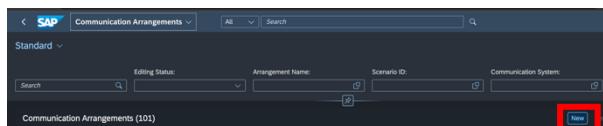
以下の URL から SAP S/4HANA Cloud にアクセスして下さい。  
<https://my400788.lab.s4hana.cloud.sap>

“Communication Management”→ “Communication Arrangement”アプリを開いてください。

## スクリーンショット/コード



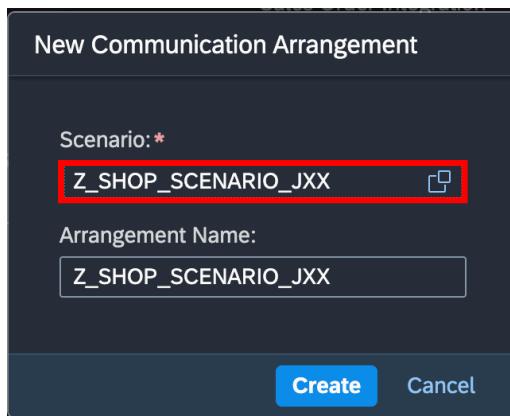
“New”ボタンを押してください。



New Communication Arrangement のポップアップ画面の“Scenario”では、先ほどパブリッシュした Communication Scenario “Z\_SHOP\_SCENARIO\_JXX”を選択してください。

注意 1:XX もしくは xx の部分については、全てご自身に割り振られた番号になります。

注意 2: ご自分で作成されたオブジェクト以外は変更しないようお願いします。



次ページに続く

## 手順

## スクリーンショット/コード

Communication Scenario  
“Z\_SHOP\_SCENARIO\_JXX”の設定画面になります。

“Common Data”にある項目  
“Communication System”的右横にある“New”ボタンを押してください。

New Communication System のポップアップ画面で以下のように設定してください。

- System ID:  
**Z\_SHOP\_COMSYS\_JXX**
- System Name:  
(System ID を入力すると自動的に同じ名称が入力されます。)

注意:XX もしくは xx の部分については、全てご自身に割り振られた番号になります。

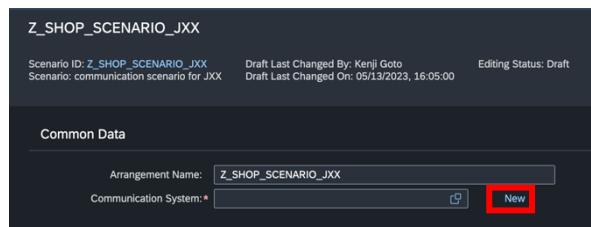
“Create”ボタンを押します。

Communication System の設定画面に遷移します。

“Technical Data”タブで以下のように設定してください。

- Inbound Only: チェックしてください。(表示されていたエラーが消えます。)

“User for Inbound Communication”タブに遷移して、“+”ボタンを押してUser を登録します。

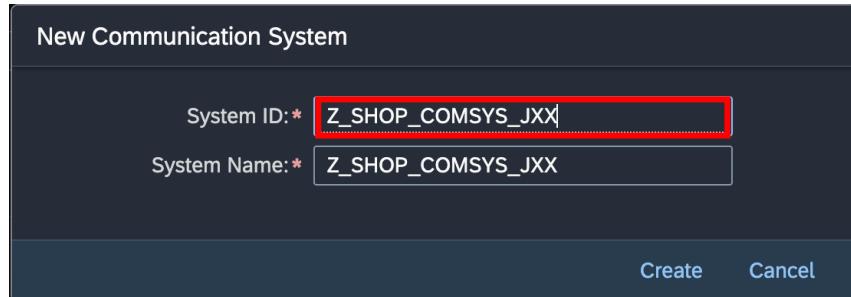


Z\_SHOP\_SCENARIO\_JXX

Scenario ID: Z\_SHOP\_SCENARIO\_JXX Scenario: communication scenario for JXX Draft Last Changed By: Kenji Goto Draft Last Changed On: 05/13/2023, 16:05:00 Editing Status: Draft

Common Data

Arrangement Name: Z\_SHOP\_SCENARIO\_JXX Communication System: \* [dropdown] [New]

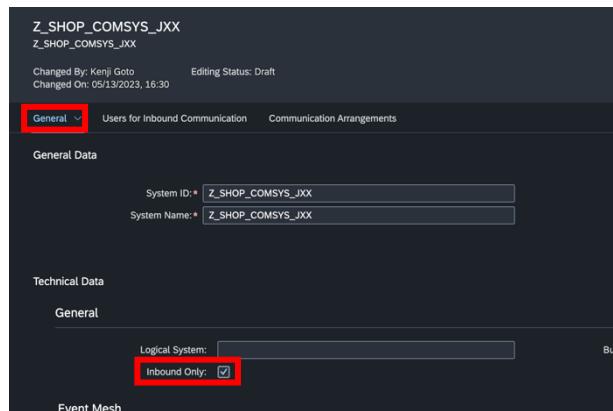


New Communication System

System ID: \* **Z\_SHOP\_COMSYS\_JXX**

System Name: \* Z\_SHOP\_COMSYS\_JXX

Create Cancel



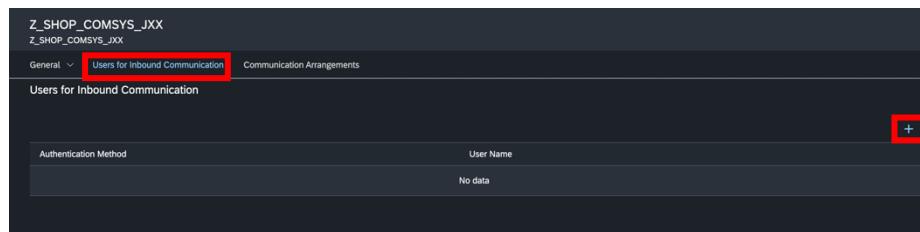
Z\_SHOP\_COMSYS\_JXX

Changed By: Kenji Goto Editing Status: Draft  
Changed On: 05/13/2023, 16:30

General

System ID: \* Z\_SHOP\_COMSYS\_JXX  
System Name: \* Z\_SHOP\_COMSYS\_JXX

Logical System: [input field]  
Inbound Only:



Z\_SHOP\_COMSYS\_JXX

General Users for Inbound Communication Communication Arrangements

Users for Inbound Communication

Authentication Method User Name  
No data

[+]

次ページに続く

## 手順

## スクリーンショット/コード

New Inbound Communication User のポップアップ画面で”New User”ボタンを押してください。



Create Communication User の画面では以下のように設定してください。

- User Name:  
**Z\_SHOP\_COMUSER\_JXX**
- Description:  
**Z\_SHOP\_COMUSER\_JXX**
- Password: “Propose Password”  
ボタンを押してパスワードを生成してください。

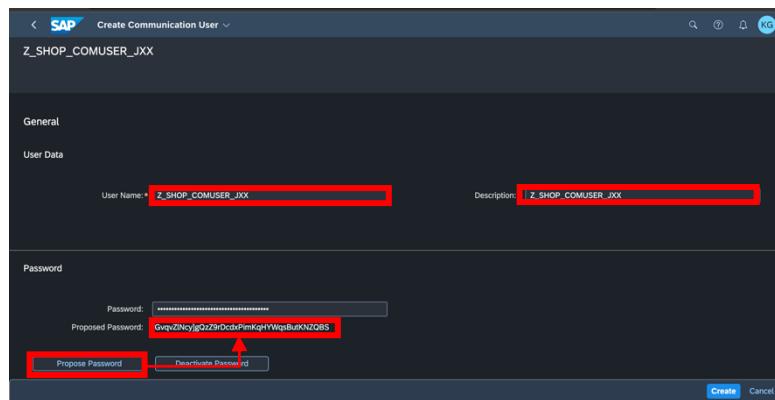
注意 1:後ほど使用するのでユーザーと生成された Proposed Password (必ずコピーしてメモ帳などに保存しておいてください)。

注意 2:XX もしくは xx の部分については、全てご自身に割り振られた番号になります。

”Create”ボタンを押してください。

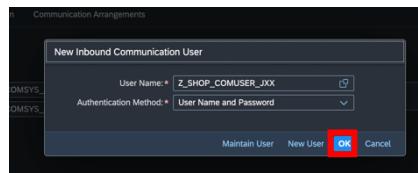
New Inbound Communication User のポップアップ画面に戻り、User Name が設定されました。

”OK”ボタンを押してください。



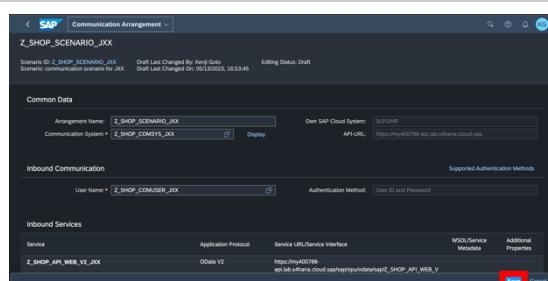
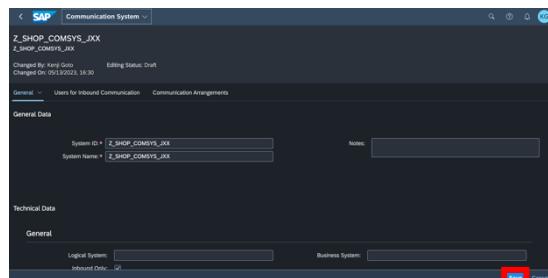
Communication System の設定画面に戻ります。

”Save”ボタンを押します。



Communication Scenario の設定画面に戻ります。

”Save”ボタンを押して、Communication Scenario をアクティベートしてください。



次ページに続く

## 手順

## スクリーンショット/コード

アクティベートされた  
Communication Scenario  
“Z\_SHOP\_SCENARIO\_JXX”的  
Inbound Service”  
Z\_SHOP\_API\_WEB\_V2\_JXX のサー  
ビスマタデータをダウンロードしま  
す。

注意: XX もしくは xx の部分について  
は、全てご自身に割り振られた番号に  
なります。

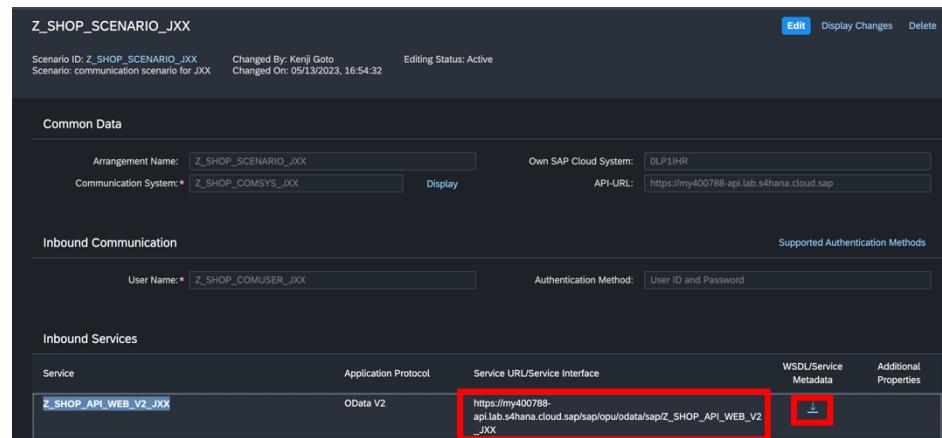
“WSDL/Service Metadata”にあるダ  
ウンロードボタンを押して、ダウント  
ロードされた XML 形式のファイルを保  
存してください。

もしくは “Service URL/Service  
Interface”にある URL をコピーして以  
下の変更を加えてください。

- 文字列 “-api”を削除してください。
- 最後に “/\$metadata”を追加してく  
ださい。

修正した URL を Web ブラウザに貼り  
付けてメタデータを取得してください。

メタデータを “Save As”で保存してく  
ださい。



The screenshot shows the SAP Fiori Launchpad with the Z\_SHOP\_SCENARIO\_JXX scenario selected. In the 'Inbound Services' section, there is a table with one row. The 'Service URL/Service Interface' column displays the URL [https://my400788-api.lab.s4hana.cloud.sap/opu/odata/sap/Z\\_SHOP\\_API\\_WEB\\_V2\\_JXX](https://my400788-api.lab.s4hana.cloud.sap/opu/odata/sap/Z_SHOP_API_WEB_V2_JXX), which is highlighted with a red box.

メタデータを Web ブラウザで表示した場合。

```
This XML file does not appear to have any style information associated with it. The document tree is shown below.

<?xml version="1.0" encoding="UTF-8"?>
<edmx:Edmx xmlns:edmx="http://schemas.microsoft.com/ado/2007/06/edmx" xmlns="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
  xmlns:sap="http://www.sap.com/Protocol/SAPData" Version="1.0">
  <edmx:Reference Uri="https://my400788.lab.s4hana.cloud.sap/api/43/opu/odata/v1/fnd/catalogservice;v=2/Vocabularies(TechnicalName='ZFIWIWBEP#JFVOC_AGGREGATION',Version='0001',SAP_Origin='')/Value"/>
    <edmx:Include Namespace="Org.OData.Agggregation.V1" Alias="Aggregation"/>
  </edmx:Reference>
  <edmx:Reference Uri="https://my400788.lab.s4hana.cloud.sap/api/43/opu/odata/v1/fnd/catalogservice;v=2/Vocabularies(TechnicalName='ZFIWIWBEP#JFVOC_ANALYTICS',Version='0001',SAP_Origin='')/Value"/>
    <edmx:Include Namespace="Com.sap.vocabularies.Analytics.v1" Alias="Analytics"/>
  </edmx:Reference>
  <edmx:Reference Uri="https://my400788.lab.s4hana.cloud.sap/api/43/opu/odata/v1/fnd/catalogservice;v=2/Vocabularies(TechnicalName='ZFIWIWBEP#JFVOC_CAPABILITIES',Version='0001',SAP_Origin='')/Value"/>
    <edmx:Include Namespace="Org.OData.Capabilities.V1" Alias="Capabilities"/>
  </edmx:Reference>
  <edmx:Reference Uri="https://my400788.lab.s4hana.cloud.sap/api/43/opu/odata/v1/fnd/catalogservice;v=2/Vocabularies(TechnicalName='ZFIWIWBEP#JFVOC_CODELIST',Version='0001',SAP_Origin='')/Value"/>
    <edmx:Include Namespace="Com.sap.vocabularies.CodeList.V1" Alias="CodeList"/>
  </edmx:Reference>
  <edmx:Reference Uri="https://my400788.lab.s4hana.cloud.sap/api/43/opu/odata/v1/fnd/catalogservice;v=2/Vocabularies(TechnicalName='ZFIWIWBEP#JFVOC_COMMON',Version='0001',SAP_Origin='')/Value"/>
    <edmx:Include Namespace="Com.sap.vocabularies.Common.v1" Alias="Common"/>
  </edmx:Reference>
  <edmx:Reference Uri="https://my400788.lab.s4hana.cloud.sap/api/43/opu/odata/v1/fnd/catalogservice;v=2/Vocabularies(TechnicalName='ZFIWIWBEP#JFVOC_COMMUNICATION',Version='0001',SAP_Origin='')/Value"/>
    <edmx:Include Namespace="Com.sap.vocabularies.Communication.v1" Alias="Communication"/>
  </edmx:Reference>
  <edmx:Reference Uri="https://my400788.lab.s4hana.cloud.sap/api/43/opu/odata/v1/fnd/catalogservice;v=2/Vocabularies(TechnicalName='ZFIWIWBEP#JFVOC_MEASURES',Version='0001',SAP_Origin='')/Value"/>
    <edmx:Include Namespace="Org.OData.Measures.V1" Alias="Measure"/>
  </edmx:Reference>
  <edmx:Reference Uri="https://my400788.lab.s4hana.cloud.sap/api/43/opu/odata/v1/fnd/catalogservice;v=2/Vocabularies(TechnicalName='ZFIWIWBEP#JFVOC_PERSONALDATA',Version='0001',SAP_Origin='')/Value"/>
    <edmx:Include Namespace="Com.sap.vocabularies.PersonalData.v1" Alias="PersonalData"/>
  </edmx:Reference>
  <edmx:Reference Uri="https://my400788.lab.s4hana.cloud.sap/api/43/opu/odata/v1/fnd/catalogservice;v=2/Vocabularies(TechnicalName='ZFIWIWBEP#JFVOC_UI',Version='0001',SAP_Origin='')/Value"/>
    <edmx:Include Namespace="Com.sap.vocabularies.UI.v1" Alias="UI"/>
  </edmx:Reference>
  <edmx:Reference Uri="https://my400788.lab.s4hana.cloud.sap/api/43/opu/odata/v1/fnd/catalogservice;v=2/Vocabularies(TechnicalName='ZFIWIWBEP#JFVOC_VALIDATION',Version='0001',SAP_Origin='')/Value"/>
    <edmx:Include Namespace="Org.OData.Validation.V1" Alias="Validation"/>
  </edmx:Reference>
  <edmx:DataService m:DataServiceVersion="1.0">
    <entityContainer name="online_shop_type" sap:content-version="1">
      <entityType Name="online_shop_type" sap:label="shop projection" sap:content-version="1">
        <annotation url="http://docs.oasis-open.org/odata/ns/edm" Term="Core.SchemaVersion" String="1.0.0"/>
        <entityType Name="online_shop_type" sap:label="shop projection" sap:content-version="1">
          <annotation url="http://docs.oasis-open.org/odata/ns/edm" Term="Core.SchemaVersion" String="1.0.0"/>
          <Property Name="Order_Duid" />
          <Property Name="Order_Duid" Type="Edm.Guid" Nullable="false" sap:label="UUID" sap:quickinfo="16 byte UUID in 16 Bytes (Raw Format)" sap:createable="false" sap:updatable="false" sap:deletable="false" />
          <Property Name="Order_Id" Type="Edm.String" MaxLength="10" sap:createable="false" sap:updatable="false" />
          <Property Name="Ordereditid" Type="Edm.String" MaxLength="10" />
          <Property Name="Orderkey" Type="Edm.String" MaxLength="10" sap:display-format="Uppercase" sap:label="Order Key" />
          <Property Name="Createidate" Type="Edm.DateTime" Precision="0" sap:display-format="Date" sap:createable="false" sap:updatable="false" />
          <Property Name="Createkey" Type="Edm.String" MaxLength="10" sap:display-format="Uppercase" sap:label="Cost Center" />
          <Property Name="Purchaserem" Type="Edm.String" MaxLength="256" sap:createable="false" sap:updatable="false" />
          <entityContainer Name="cds_xsd_shop_ixs_Entities" m:isDefaultEntityContainer="true" sap:message-scoped-supported="true" sap:is-supported-formats="atom json xlink" />
          <entitySet Name="online_shop" EntityTypeName="cds_xsd_shop_ixs_online_shoptype" sap:searchable="true" sap:content-version="1" />
        </entityType>
      </entityContainer>
    </entityContainer>
  </edmx:DataService>
```

次ページに続く

## ステップ 2: API の取り込み

作成したカスタム OData API を SAP BTP, ABAP Environment へ取り込みます。

### ステップ 2.1: Service Consumption Model の作成

SAP BTP, ABAP Environment で開発者として Service Consumption Model を作成して外部の OData API を取り込みます。

手順	スクリーンショット/コード
ADT の ABAP Perspective にてメニューから "File" → "New" → "ABAP Cloud Project" を選択します。	
以下のように選択してください。 ● "SAP BTP ABAP Environment" および "Use a Service Key"	
"Next" ボタンを押してください。	
リンクから JSON ファイルをダウンロードして、それぞれの PC に保存してください。  System Connection Using a Service Key の画面で "Import" ボタンを押して、先ほど保存した JSON ファイルを選択してアップロードしてください。	
"Next" ボタンを押してください。	

次ページに続く

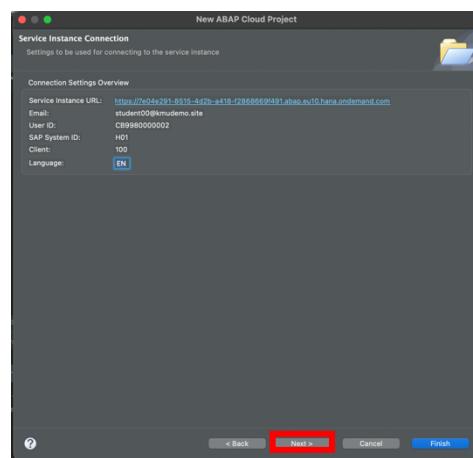
## 手順

## スクリーンショット/コード

次画面で"Open Logon Page in Browser"を選択します。  
ブラウザーが起動し、利用する Identity Provider を選択する画面が表示されます。  
そこ  
で"httpsarrktsezy.account400.ondemand"  
を選択してください。  
SAP BTP ABAP Environmentへのログオン  
画面が表示されます。  
事前に共有済みのユーザーID およびパスワードを利用してログオンをしてください。



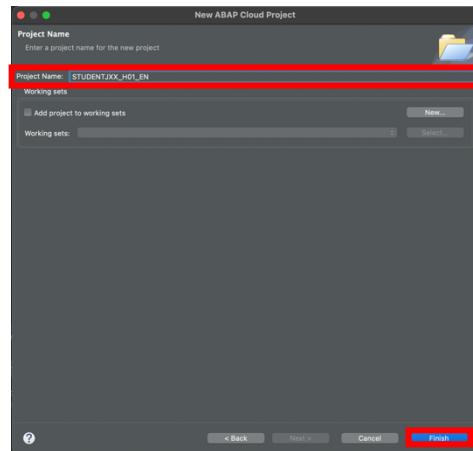
一度ログオンが成功した場合、ADT 上のボップアップ画面が自動的に Service Instance Connection の画面に遷移します。ここで"Next"ボタンを押します。



"ADT Project Name" は[システム ID]\_[ログオン言語]がデフォルト値としてセットされています。  
分かりやすくするために、プリフィックスとして"STUDENTJXX"を付けてください。  
例: STUDENTJXX\_[システム ID]\_[ログオン言語]

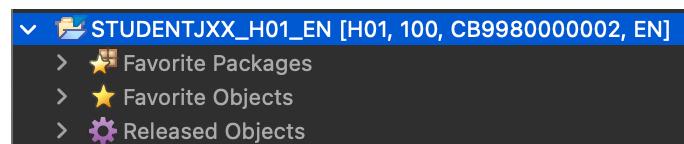
注意:XXの部分については、全てご自身に割り振られた番号に変更して下さい。

最後に"Finish"ボタンを押してください。



プロジェクトが追加されると Project Explorer に表示されます。

作成したプロジェクトをダブルクリックして展開してください。

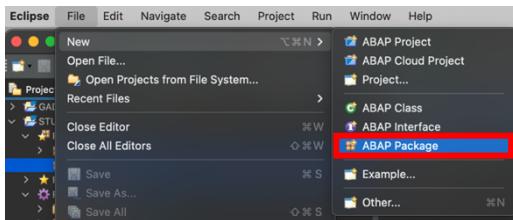


次ページに続く

## 手順

## スクリーンショット/コード

ADT のメニューから“File”→“New”→“ABAP Package”を選択します。

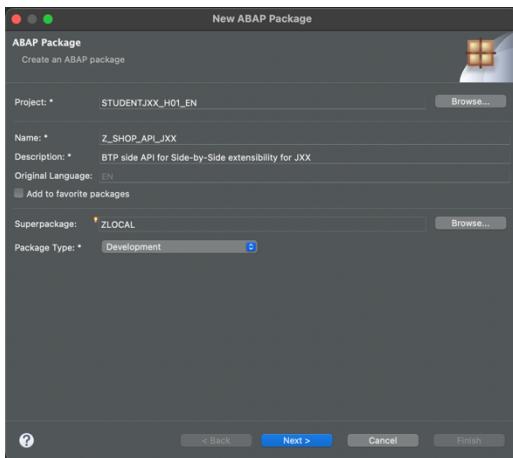


ABAP Package を以下のように設定します。

- Name: Z\_SHOP\_API\_JXX
- Description: BTP side API for Side-by-Side extensibility for JXX
- Add to Favorite packages :  
チェックを付けてください。
- Superpackage: ZLOCAL

注意:XX の部分については、全てご自身に割り振られた番号に変更して下さい。

2回“Next”ボタンを押してください。

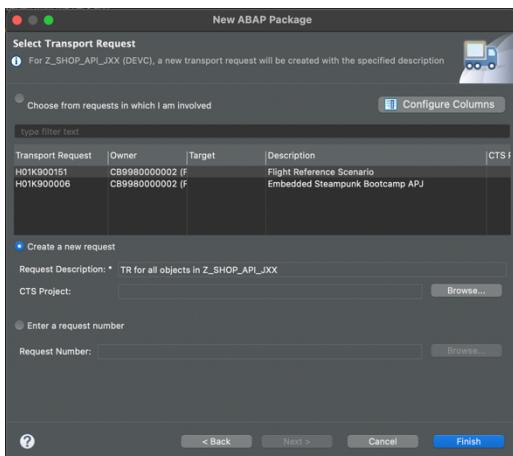


Select Transport Request の画面では以下のように設定してください。

- Create a new request :  
チェックしてください。
- Request Description:  
TR for All Objects for Z\_SHOP\_API\_JXX

注意:XX の部分については、全てご自身に割り振られた番号に変更して下さい。

“Finish”ボタンを押すと ABAP Package が作成されます。



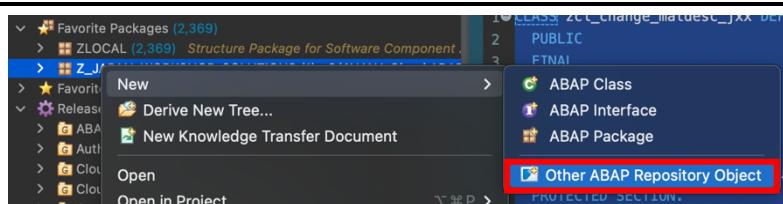
次ページに続く

## 手順

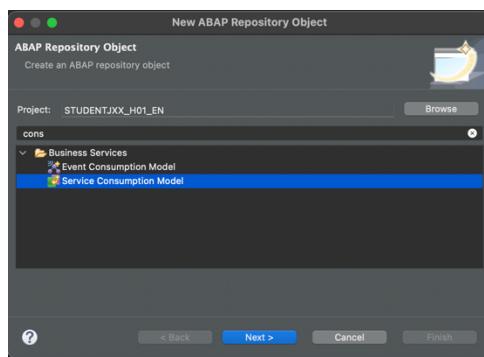
## スクリーンショット/コード

ABAP Package “Z\_SHOP\_API\_JXX”を選択し、右クリックして“New”→“Other ABAP Repository Object”を選択します。

注意:XX もしくは xx の部分については、全てご自身に割り振られた番号になります。



New ABAP Repository Object のポップアップ画面から“Service Consumption Model”を検索の上、選択して“Next”ボタンを押してください。

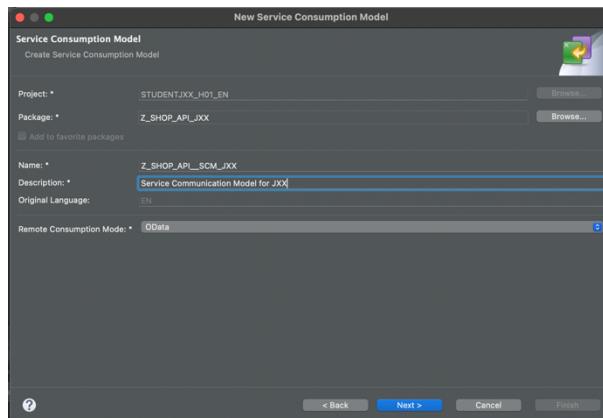


Service Consumption Model を以下のように設定します。

- Name:  
`Z_SHOP_API_SCM_JXX`
- Description:  
Service Communication Model for JXX
- Remote Consumption Model: OData

注意:XX の部分については、全てご自身に割り振られた番号に変更して下さい。

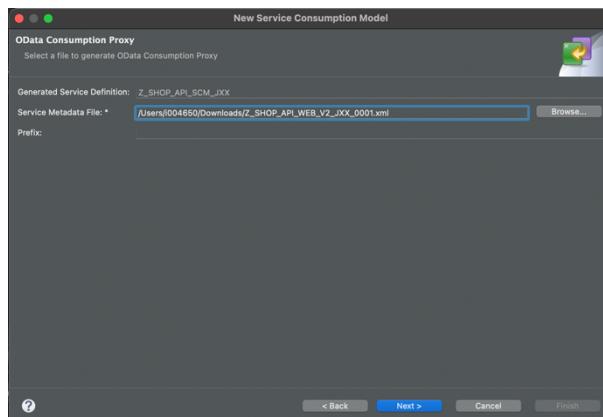
“Next”ボタンを押します。



OData Consumption Proxy の画面で以下の設定をしてください。

- Service Metadata File:  
ステップ 1.2 受信用 API の公開の最後で S/4HANA Cloud で作成してダウンロードした Inbound Service”  
`Z_SHOP_API_WEB_V2_JXX`”のサービスメタデータのファイルパスを指定します。

“Next”ボタンを押してください。



次ページに続く

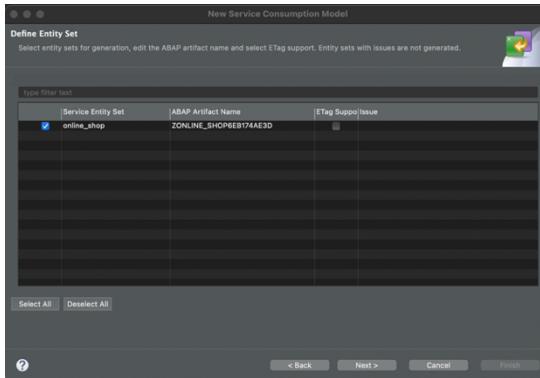
## 手順

## スクリーンショット/コード

Define Entity Set の画面では以下のように設定してください。

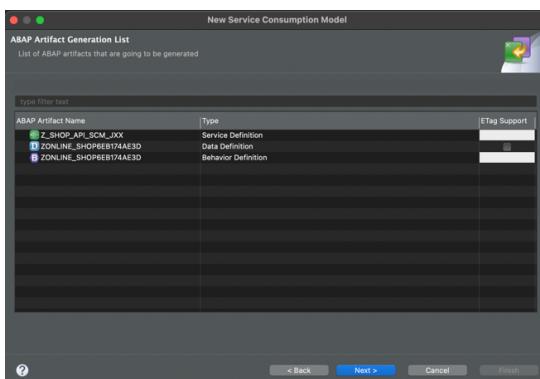
- Service Entity Set の online\_shop の左側のチェックボックス: チェックしてください。

"Next"ボタンを押してください。



ABAP Artifact Generation List の画面では、Service Consumption Model の作成のために自動的に生成されるオブジェクトのリストが表示されます。

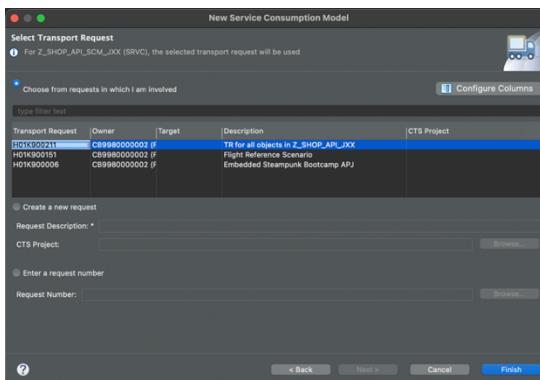
"Next"ボタンを押してください。



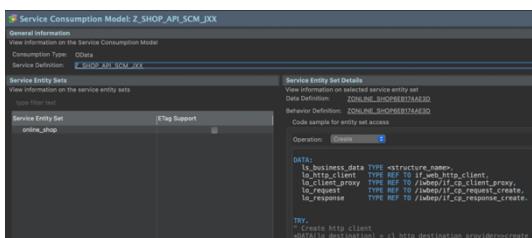
Select Transport Request の画面では以下のように設定してください。

- Choose from requests in which I am involved:  
チェックしてください。
- Requests List:  
Package 作成時の Transport Request を選択して下さい。

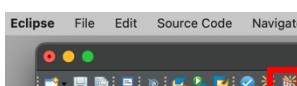
"Finish"ボタンを押すと Service Consumption Service および関連オブジェクトが作成されます。



Service Consumption Model がメタデータから自動生成されます。

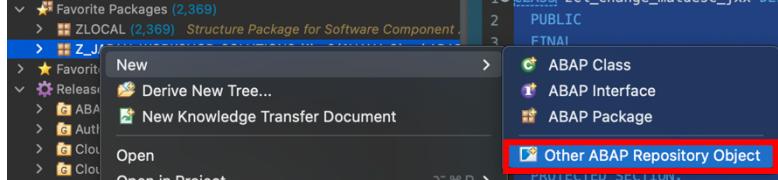
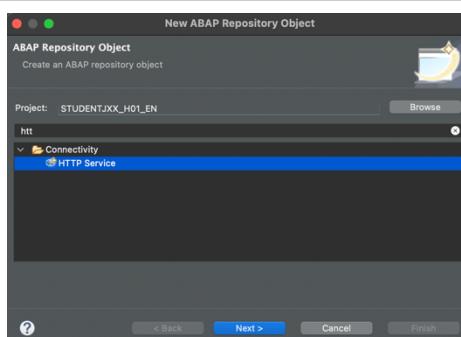
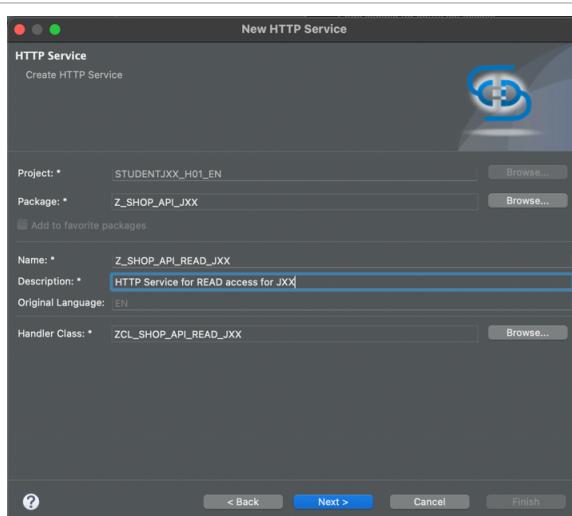


全てのオブジェクトに対して"Activate inactive ABAP development objects"して下さい。



## ステップ 2.2: HTTP Service の作成

SAP BTP, ABAP Environment で開発者として先のステップで作成した Service Consumption Model Service を利用する REST HTTP サービスを実装します。

手順	スクリーンショット/コード
ABAP Package “Z_SHOP_API_JXX”を選択し、右クリックして“New”→“Other ABAP Repository Object”を選択します。  注意:XX もしくは xx の部分については、全てご自身に割り振られた番号になります。	
New ABAP Repository Object のポップアップ画面から“HTTP Service”を検索の上、選択して“Next”ボタンを押してください。	
HTTP Service を以下のように設定してください。  ● Name: Z_SHOP_API_READ_JXX ● Description: HTTP Service for READ Access for JXX ● Handler Class: ZCL_SHOP_API_READ_JXX  注意:XX もしくは xx の部分については、全てご自身に割り振られた番号になります。	

次ページに続く

## 手順

## スクリーンショット/コード

Select Transport Request の画面では以下のように設定してください。

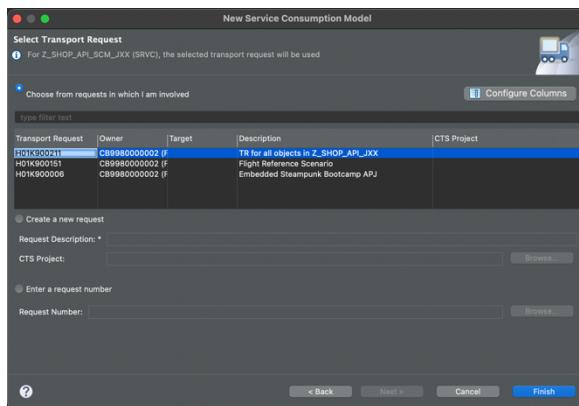
- Choose from requests in which I am involved:

チェックしてください。

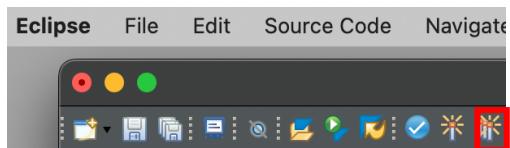
- Requests List:

Package 作成時の Transport Request を選択して下さい。

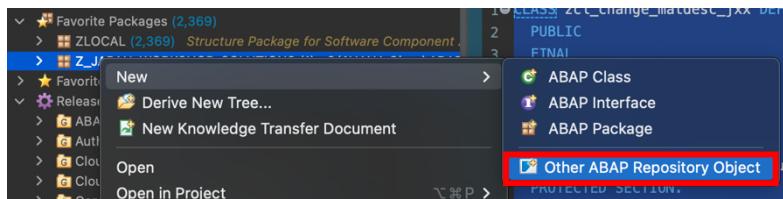
"Finish"ボタンを押すと HTTP Service が作成されます。



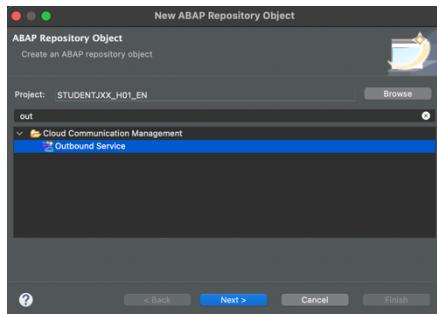
HTTP Service を "Save"して、"Activate"して下さい。



ABAP Package "Z\_SHOP\_API\_JXX"を選択し、右クリックして"New"→"Other ABAP Repository Object"を選択します。  
注意:XX もしくは xx の部分については、全てご自身に割り振られた番号になります。



New ABAP Repository Object のポップアップ画面から"Outbound Service"を検索の上、選択して"Next"ボタンを押して下さい。



Outbound Service を以下のように設定してください。

- Name:

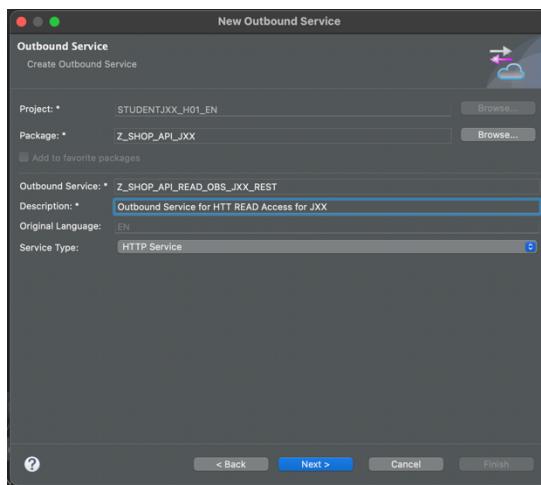
Z\_SHOP\_API\_READ\_OBS\_JXX\_REST

- Description: Outbound Service for HTTP READ Access for JXX

- Service Type: HTTP Service

注意:XX もしくは xx の部分については、全てご自身に割り振られた番号になります。

"Next"ボタンを押してください。



次ページに続く

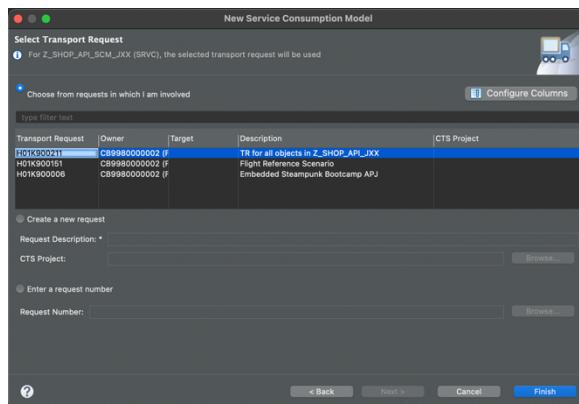
## 手順

## スクリーンショット/コード

Select Transport Request の画面では以下のように設定してください。

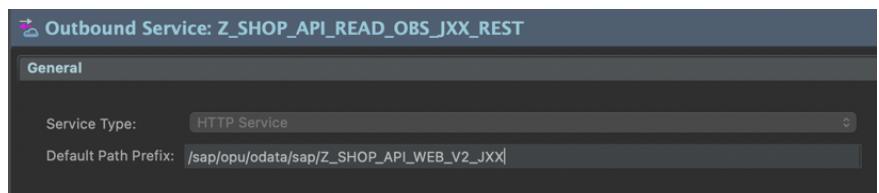
- Choose from requests in which I am involved:  
チェックしてください。
- Requests List:  
Package 作成時の Transport Request を選択して下さい。

"Finish"ボタンを押すと Outbound Service が作成されます。

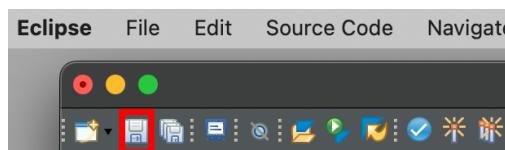


作成された Outbound Service を以下のように設定してください。

- Default Path Prefix:  
`/sap/opu/odata/sap/Z_SHOP_API_WEB_V2_JXX"`  
(こちらはステップ 1.2 受信用 API の公開の最後で S/4HANA Cloud で作成してダウンロードした Inbound Service "Z\_SHOP\_API\_WEB\_V2\_JXX" の Path Prefix を指定します。

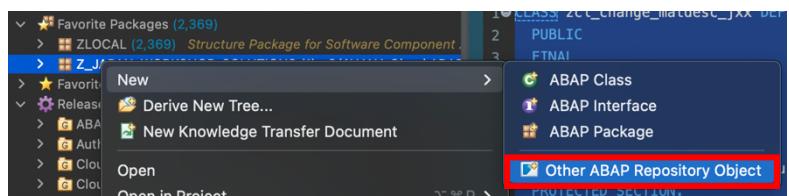


Outbound Service を "Save"して下さい。

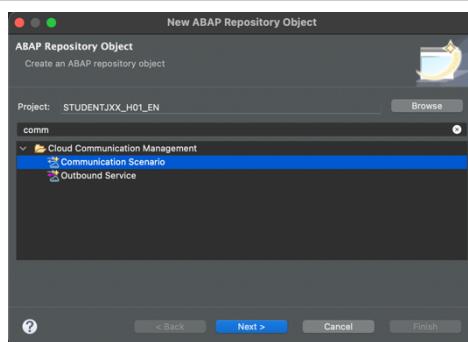


ABAP Package "Z\_SHOP\_API\_JXX"を選択し、右クリックして"New"→"Other ABAP Repository Object"を選択します。

注意:XX もしくは xx の部分については、全てご自身に割り振られた番号になります。



New ABAP Repository Object のポップアップ画面から"Communication Scenario"を検索の上、選択して "Next"ボタンを押して下さい。



次ページに続く

## 手順

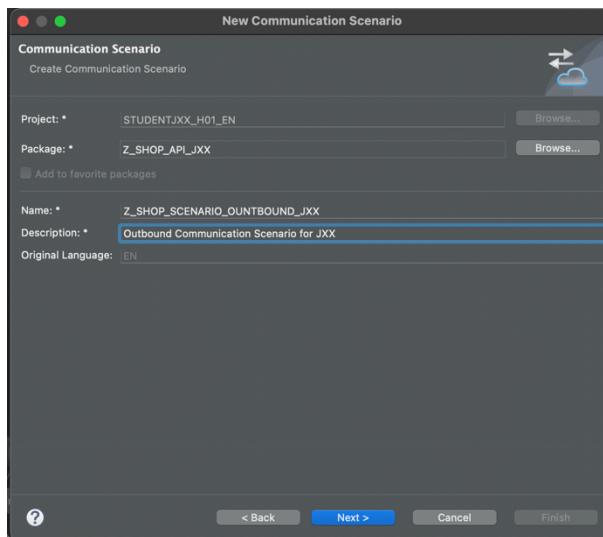
## スクリーンショット/コード

Communication Scenario を以下のように設定してください。

- Name:  
Z\_SHOP\_SCENARIO\_OUNBOUND\_JXX
- Description: Outbound Communication Scenario for JXX

注意:XX もしくは xx の部分については、全てご自身に割り振られた番号になります。

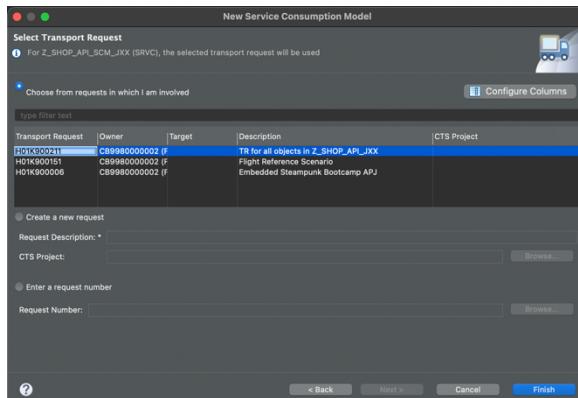
"Next"ボタンを押します。



Select Transport Request の画面では以下のように設定してください。

- Choose from requests in which I am involved:  
チェックしてください。
- Requests List:  
Package 作成時の Transport Request を選択して下さい。

"Finish"ボタンを押すと Communication Scenario が作成されます。



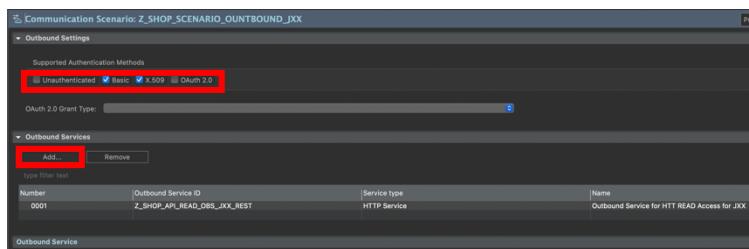
Communication Scenario の"Overview" タブを開いて以下のように設定して下さい。

- Allowed Instances:  
One instance per client

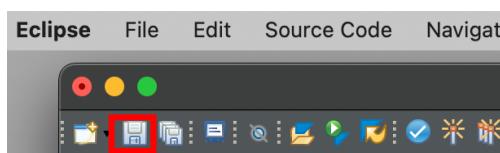


Communication Scenario の"Outbound" タブを開いて以下のように設定して下さい。

- Basic Authorization:  
チェックしてください。
- Inbound Services: "Add"ボタンを押して先に作成した COM Inbound Service"  
Z\_SHOP\_API\_WEB\_V2\_JXX"を設定してください。



Communication Scenario を"Save"して下さい。

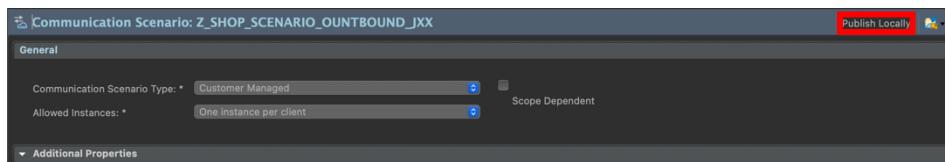


次ページに続く

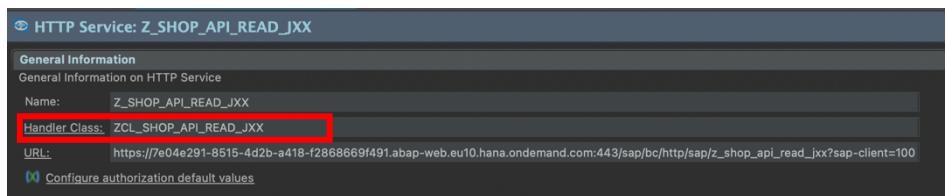
## 手順

## スクリーンショット/コード

Communication Scenario の "Overview" タブを開いて "Locally Publish" ボタンを押してください。  
Communication Scenario を SAP BTP ABAP Environment にパブリッシュが開始されます。  
ご注意: パブリッシュには少し時間がかかります。



HTTP Service のロジックを実装します。  
HTTP Service "Z\_SHOP\_API\_READ\_JXX" の General パートにある Handler Class をクリックして ABAP Class "ZCL\_SHOP\_API\_READ\_JXX" を開いてください。  
注意: XX もしくは xx の部分については、全てご自身に割り振られた番号になります。



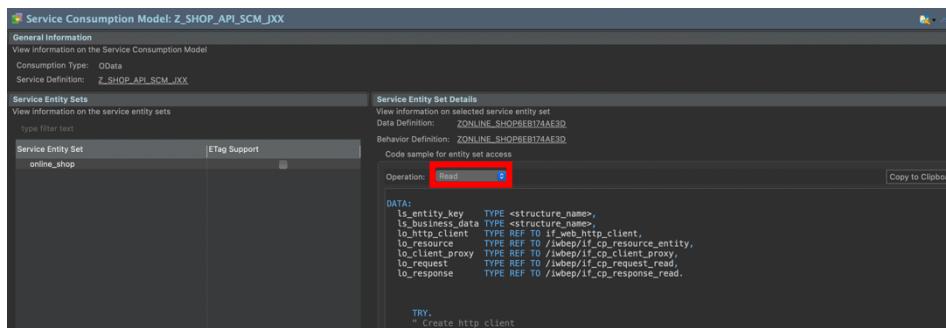
ABAP Class "ZCL\_SHOP\_API\_READ\_JXX" が開かれます。メソッド "IF\_HTTP\_SERVICE\_EXTENSION ~HANDLE\_REQUEST" の部分を実装します。  
注意: XX もしくは xx の部分については、全てご自身に割り振られた番号になります。  
(画面は分かりやすいように method から endmethod の間を広げています。)

```

1@class ZCL_SHOP_API_READ_JXX > IF_HTTP_SERVICE_EXTENSION-HANDLE_REQUEST
2  public
3    create public .
4
5    public section.
6
7    interfaces IF_HTTP_SERVICE_EXTENSION .
8    protected section.
9    private section.
10   ENDOCLASS.
11
12
13
14@CLASS ZCL_SHOP_API_READ_JXX IMPLEMENTATION.
15
16
17@ METHOD IF_HTTP_SERVICE_EXTENSION-HANDLE_REQUEST.
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40 ENDMETHOD.
ENDCLASS.

```

実装コードのスニペットは、Service Consumption Model "Z\_SHOP\_API\_SCM\_JXX" の "READ" オペレーションのものを利用します。  
こちらのコードを "Copy to Clipboard" ボタンを押して全てコピーしてください。  
注意: XX もしくは xx の部分については、全てご自身に割り振られた番号になります。



コピーしてコードを ABAP Class "ZCL\_SHOP\_API\_READ\_JXX" の開かれます。メソッド "IF\_HTTP\_SERVICE\_EXTENSION ~HANDLE\_REQUEST" (method から endmethod の間に)に貼り付けてください。

```

16
17@ METHOD if_http_service_extension-handle_request.
18
19
20
21
22
23
24
25
26
27
28
29
30
31@ TRY.
32  " Create http client
33 *DATA(l0_destination) = cl_http_destination_provider->create_by_comm_arrangement(

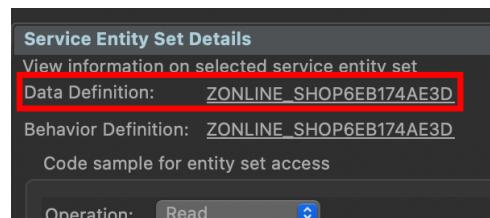
```

次ページに続く

## 手順

## スクリーンショット/コード

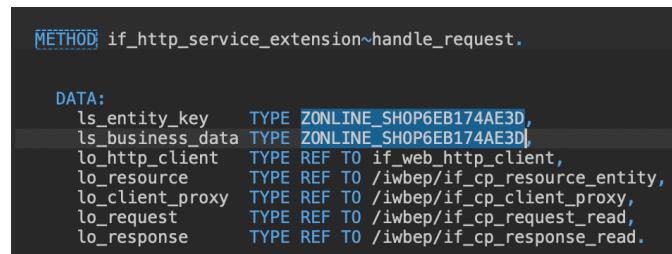
データ宣言パートにある<Structure\_name>を CDS View (Data Definition)に置き換えます。これは Consumption Service を作成した際に同時に作られたオブジェクトになります。  
こちらの名称は Consumption Model "Z\_SHOP\_API\_SCM\_JXX" の "Overview" タブの Service Entity Set Details に記載されています。  
注意:XX もしくは xx の部分については、全てご自身に割り振られた番号になります。



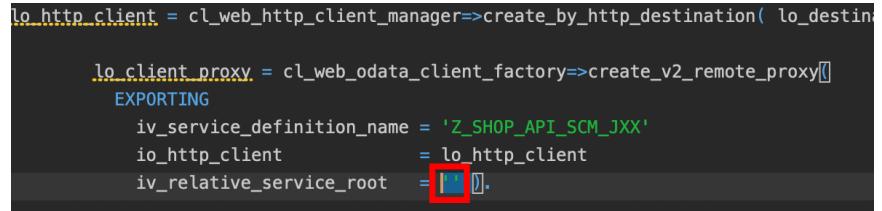
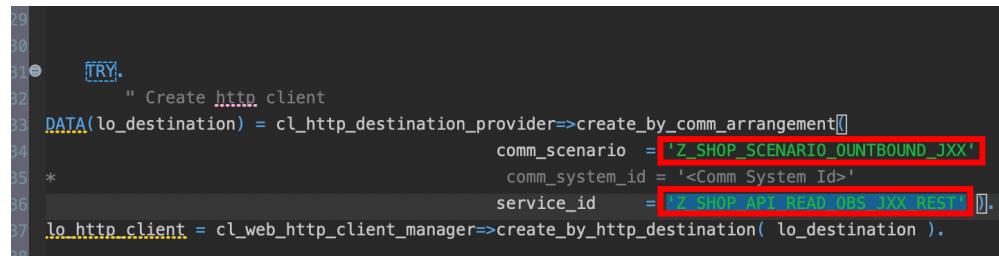
TRY パートの下にある  
DATA(lo\_destination) で始まる部分および  
lo\_http\_client で始まる部分のコメントアウトを解除してください。  
(ただし Comm\_system\_id = の部分はそのままコメントアウトしたままで問題ありません。) その上でパラメータを以下のように設定してください。

- Comm\_scenario: (以前作成した Communication Scenario 名である)  
Z\_SHOP\_SCENARIO\_OUNTBOND\_JXX
- Comm\_system\_id:  
コメントアウトしたままで問題ありません。
- Service\_id: (以前作成した Outbound Service 名である)  
Z\_SHOP\_API\_READ\_OBS\_JXX\_REST

注意:XX もしくは xx の部分については、全てご自身に割り振られた番号になります。



lo\_client\_proxy でリモートプロキシーを生成するためのパラメータ  
iv\_relative\_service\_root のパラメータはデフォルトで<service\_root>となっていますが、こちらを削除してからの値を渡すようにしてください。(右図参照)  
注意: こちらはすでに Outbound Service オブジェクトないで設定済みなのでこちらでは指定する必要はありません。



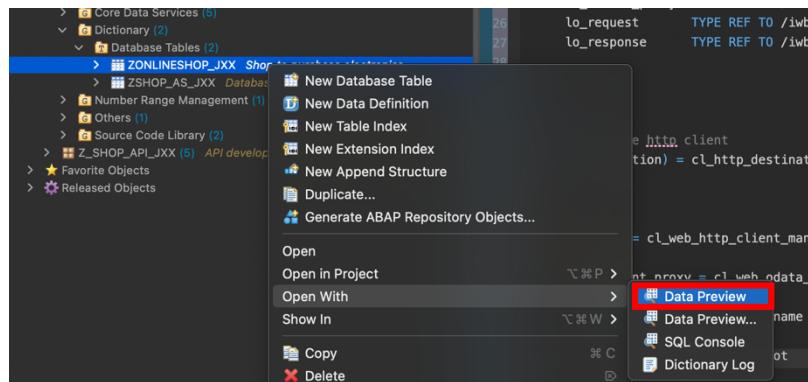
次ページに続く

## 手順

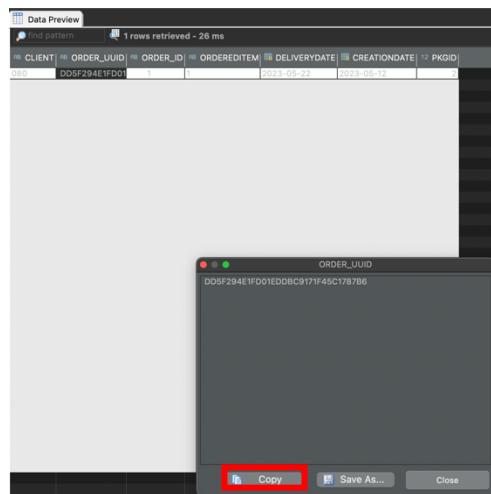
## スクリーンショット/コード

一度、SAP S/4HANA Cloud の ABAP Project にアクセスして最初に作成した Database Table “ZONLINESHOP\_JXX”を選択して右クリックして“Open with”→“Data Preview”を選択してください。

注意:XX もしくは xx の部分については、全てご自身に割り振られた番号になります。



演習 2 の最後でアプリケーションテストの際に作られたデータが表示されます。その中から“ORDER\_UUID”の内、何か適当なものを一つダブルクリックします。ORDER\_UUID を表示するポップアップ画面が表示されますので、“Copy”ボタンを押してUUID をコピーしてください。その後“Close”ボタンを押してポップアップを閉じてください。



再度、SAP BTP, ABAP Environment の ABAP Cloud Project に戻り、HTTP Service の実装クラスである “ZCL\_SHOP\_API\_READ\_JXX”を開いてください。

ls\_entity\_key = VALUE #( の部分の値を割くほどコピーした ORDER\_UUID の値と置き換えてください。

```
" Set entity key
ls_entity_key = VALUE #(
    order_uuid = [REDACTED]).
```

```
" Navigate to the resource
lo_resource = lo_client_proxy->create_resource_for_entity_set('ONLINE_SHOP')->navigate_with_key(ls_entity_key).
```

```
" Execute the request and retrieve the business data
lo_response = lo_resource->create_request_for_read()->execute().
lo_response->get_business_data(IMPORTING es_business_data = ls_business_data).
```

```
CATCH /iwbep/cx_cp_remote INTO DATA(lx_remote).
" Handle remote Exception
".
```

データ取得を実行した後にレスポンスの中から表示するデータ(ここでは OrderID と Ordered Item)を String として取り出します。

右のコードを lo\_response->get\_business\_data( IMPORTING ~のコードの後(且つ CATCH 文の前に)にコピーしてペースとしてください。

```
DATA: result TYPE string.
      result = | OrderID: { ls_business_data-Order_Id }, Ordered Item: {
        ls_business_data-Ordereditem } |.
      response->set_text(result).
```

次ページに続く

## 手順

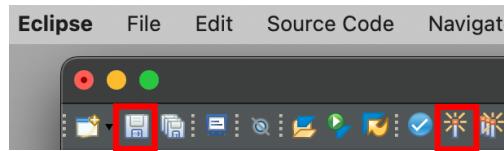
## スクリーンショット/コード

TRY の CATCH 文の実装をします。  
先ほど追加したコードの下にある  
CATCH 文の部分を全て削除して右  
のコードに置き換えてください

```
[CATCH] /iwbep/cx_cp_remote INTO DATA(lx_remote).
    " Handle remote Exception
    " It contains details about the problems of your http(s) connection
    response->set_text( | Remote Error: { lx_remote->get_longtext( ) } | ).
[CATCH] /iwbep/cx_gateway INTO DATA(lx_gateway).
    " Handle Exception
    response->set_text( | Gateway Error: { lx_gateway->get_longtext( ) } | ).
[CATCH] cx_http_dest_provider_error INTO DATA(lx_destination).
    "handle exception
    response->set_text( | Destination Error: { lx_destination->get_longtext( ) } | ).
[CATCH] cx_web_http_client_error INTO DATA(lx_http).
    "handle exception
    response->set_text( | HTTP Error: { lx_http->get_longtext( ) } | ).
```

最終的なコードは[こちら](#)のようになります。(もしくは資料末のコードご参照ください。)

HTTP Service の実装クラスを“Save”して、“Activate”して下さい。



次ページに続く

## ステップ 2.3: HTTP Service の公開

SAP BTP, ABAP Environment で管理者として外部の OData API を利用するための Outbound 通信管理を行います。

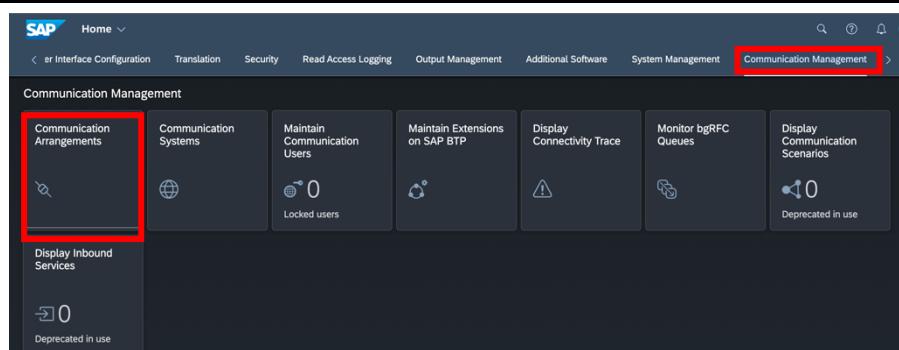
## 手順

## スクリーンショット/コード

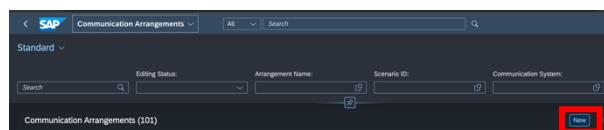
以下の URL から SAP BTP、ABAP Environment にアクセスして下さい。

<https://7e04e291-8515-4d2b-a418-f2868669f491.abap-web.eu10.hana.ondemand.com/ui>

“Communication Management”→  
“Communication Arrangement”アブ  
リを開いてください。



“New”ボタンを押してください。



次ページに続く

## 手順

## スクリーンショット/コード

New Communication Arrangement のポップアップ画面の“Scenario”では、先ほど  
パブリッシュした Communication Scenario  
“Z\_SHOP\_SCENARIO\_OUTBOUND\_JXX”  
を選択してください。

注意 1:**XX** もしくは **xx** の部分について  
は、全てご自身に割り振られた番号になります。

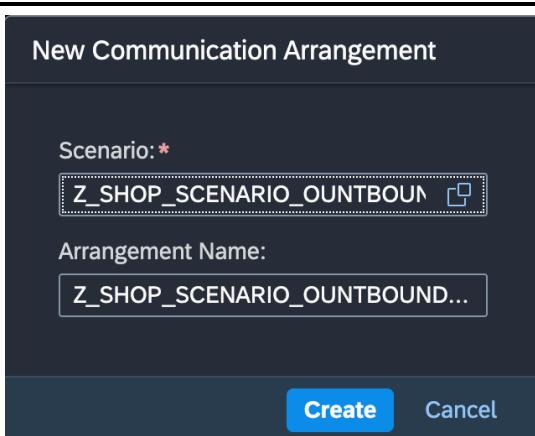
注意 2: ご自分で作成されたオブジェクト以外は変更しないようにお願いします。

New Communication Arrangement

Scenario: \*

Arrangement Name:

**Create** **Cancel**



Communication Scenario  
“Z\_SHOP\_SCENARIO\_OUTBOUND\_JXX”  
の設定画面になります。

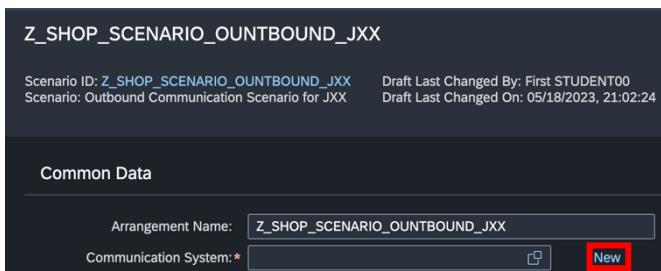
“Common Data”にある項目  
“Communication System”的右横にある“New”ボタンを押してください。

Z\_SHOP\_SCENARIO\_OUTBOUND\_JXX

Scenario ID: Z\_SHOP\_SCENARIO\_OUTBOUND\_JXX Draft Last Changed By: First STUDENT00  
Scenario: Outbound Communication Scenario for JXX Draft Last Changed On: 05/18/2023, 21:02:24

Common Data

Arrangement Name:   
Communication System: \*



New Communication System のポップアップ画面で以下のように設定してください。

- System ID: Z\_SHOP\_COMSYS\_JXX
- System Name:  
(System ID を入力すると自動的に同じ名称が入力されます。)

注意:**XX** もしくは **xx** の部分については、全てご自身に割り振られた番号になります。

“Create”ボタンを押します。

New Communication System

System ID: \*   
System Name: \*

**Create** **Cancel**



Communication System の設定画面に遷移します。

“Technical Data”タブで以下のように設定してください。

- Host Name:  
my400788.lab.s4hana.cloud.sap
- Port: 443 (デフォルト値)

Z\_SHOP\_COMSYS\_JXX

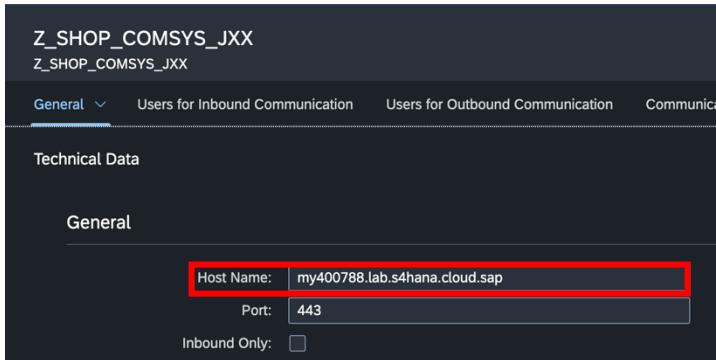
Z\_SHOP\_COMSYS\_JXX

General

Technical Data

General

Host Name:   
Port:   
Inbound Only:

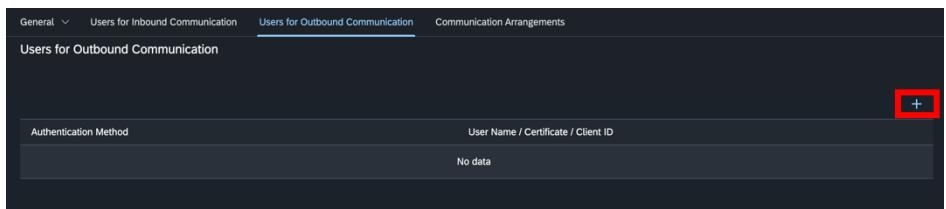


次ページに続く

## 手順

## スクリーンショット/コード

"User for outbound Communication"タブに遷移して、"+"ボタンを押して User を登録します。



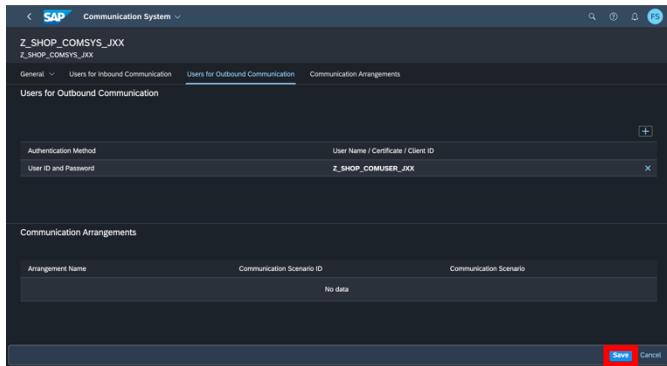
New Outbound User の登録画面で以下のように設定してください。

- Authentication Method:  
User Name and Password を選択
- User Name:  
<SAP S/4HANA Cloud で登録した  
Inbound User Name>
- Password:  
<SAP S/4HANA Cloud で登録した  
Inbound User Password>

"Create"を押してください。

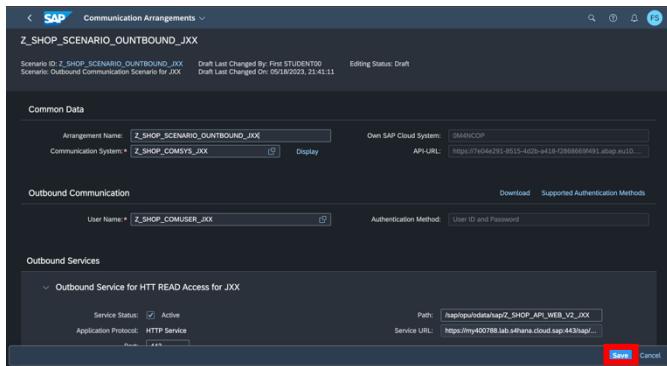
Communication System の設定画面に戻ります。

"Save"ボタンを押します。



Communication Scenario の設定画面に戻ります。

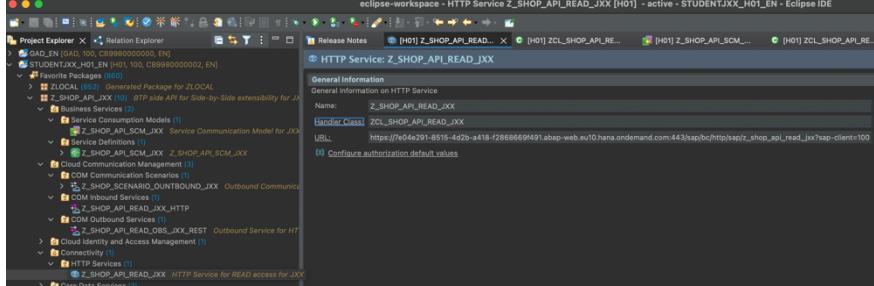
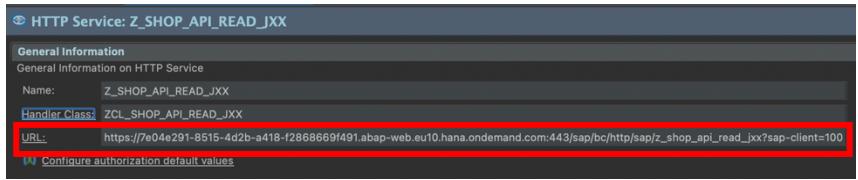
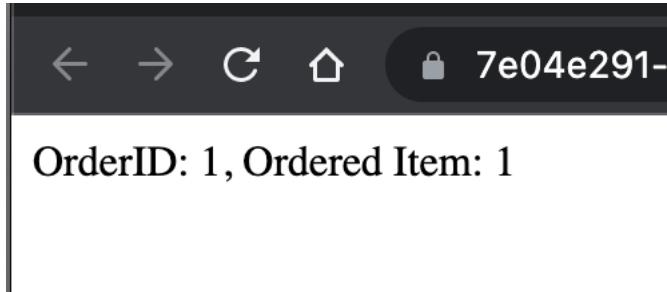
"Save"ボタンを押して、Communication Scenario をアクティベートしてください。



次ページに続く

## ステップ 2.4: API のテスト

SAP BTP, ABAP Environment で開発者として HTTP サービスを提供して、ビジネスユーザーとしてカスタムの HTTP サービスを呼び出します。

手順	スクリーンショット/コード
ADT から SAP BTP, ABAP Environment の ABAP Cloud Project にある作成した HTTP Service を開きます。	
"General Information"のある URL アドレスはエンドポイントになります。こちらの URL をコピーして Web ブラウザーで開いてください。もし URL のリンクをクリックすると直接 Web ブラウザーを開くことができます。	
購買依頼のデータが表示されます。	

演習 2 は終了です。お疲れ様でした。

ご参考情報:

- ユーザにカスタム HTTP サービスの権限が自動的に付与されます。これは、ユーザに割り当てられている開発者ビジネスロールにより、このようなすべてのカスタムサービスに対する権限が自動的に付与されるためです。他のビジネスユーザに対してその使用を有効にするには、サービスに必要な ID およびアクセス管理を実行する必要があります。これは、この演習問題の範囲外です。
- Eclipse でブレークポイントを設定して (行番号マーカーの左側をダブルクリック)、リモートコールをデバッグすることができます。if\_http\_service\_extension~handle\_request メソッドの最初に 1 つを設定し、ステップバイステップで実行することをお奨めします。
- この演習では、SAP BTP ABAP 環境インスタンスで有用な ABAP 機能 (サービス利用モデルおよびカスタム HTTP サービス) を使用しました。S/4HANA Cloud では同じテクノロジースタックが共有されるため、これらの同じ機能を S/4HANA Cloud 開発者拡張でも使用することができます。

## ご参考: HTTP Service 実装クラス“ZCL\_SHOP\_API\_READ\_JXX”コードサンプル

注意:XX もしくは xx の部分については、全てご自身に割り振られた番号になります。

```
CLASS zcl_shop_api_read_jxx DEFINITION
  PUBLIC
    CREATE PUBLIC .

  PUBLIC SECTION.

    INTERFACES if_http_service_extension .
  PROTECTED SECTION.
  PRIVATE SECTION.
ENDCLASS.

CLASS zcl_shop_api_read_jxx IMPLEMENTATION.

  METHOD if_http_service_extension~handle_request.

    DATA:
      ls_entity_key      TYPE zonline_shop6eb174ae3d,
      ls_business_data   TYPE zonline_shop6eb174ae3d,
      lo_http_client     TYPE REF TO if_web_http_client,
      lo_resource        TYPE REF TO /iwbep/if_cp_resource_entity,
      lo_client_proxy    TYPE REF TO /iwbep/if_cp_client_proxy,
      lo_request         TYPE REF TO /iwbep/if_cp_request_read,
      lo_response        TYPE REF TO /iwbep/if_cp_response_read.

    TRY.
      " Create http client
      DATA(lo_destination) = cl_http_destination_provider=>create_by_comm_arrangement(
        comm_scenario      = 'Z_SHOP_SCENARIO_OUNBOUND_JXX'
      *           comm_system_id = '<Comm System Id>'
      *           service_id     = 'Z_SHOP_API_READ_OBS_JXX_REST' ).
      lo_http_client = cl_web_http_client_manager=>create_by_http_destination( lo_destination ).

      lo_client_proxy = cl_web_odata_client_factory=>create_v2_remote_proxy(
        EXPORTING
          iv_service_definition_name = 'Z_SHOP_API_SCM_JXX'
          io_http_client            = lo_http_client
          iv_relative_service_root   = '' ).

      *lo_client_proxy = /iwbep/cl_cp_factory_alv5=>create_v2_remote_proxy(
      * EXPORTING
      *   is_proxy_model_key       = value #( repository_id      = /iwbep/if_cp_registry_types=>gcs_repository_id-srvd
      *                                     proxy_model_id     = 'Z_SHOP_API_SCM_JXX'
      *                                     proxy_model_version = '0001' )
      *   io_http_client           = lo_http_client
      *   iv_relative_service_root = '<service_root>' ).

      " Set entity key
      ls_entity_key = VALUE #(
        order_uuid = 'DD5F294E1FD01EDDBC9171F45C1787B6' ).

      " Navigate to the resource
      lo_resource = lo_client_proxy->create_resource_for_entity_set( 'ONLINE_SHOP' )->navigate_with_key( ls_entity_key ).

      " Execute the request and retrieve the business data
      lo_response = lo_resource->create_request_for_read( )->execute( ).
      lo_response->get_business_data( IMPORTING es_business_data = ls_business_data ).

      DATA: result TYPE string.
      result = | OrderID: { ls_business_data-Order_Id }, Ordered Item: {
        ls_business_data-Ordereditem } |.
      response->set_text( result ).

      CATCH /iwbep/cx_cp_remote INTO DATA(lx_remote).
      " Handle remote Exception
      " It contains details about the problems of your http(s) connection
      response->set_text( | Remote_Error: { lx_remote->get_longtext( ) } | ).
      CATCH /iwbep/cx_gateway INTO DATA(lx_gateway).
      " Handle Exception
      response->set_text( | Gateway_Error: { lx_gateway->get_longtext( ) } | ).
      CATCH cx_http_dest_provider_error INTO DATA(lx_destination).
      "handle exception
      response->set_text( | Destination_Error: { lx_destination->get_longtext( ) } | ).
      CATCH cx_web_http_client_error INTO DATA(lx_http).
      "handle exception
      response->set_text( | HTTP_Error: { lx_http->get_longtext( ) } | ).

    ENDTRY.

  ENDMETHOD.
ENDCLASS.
```

SAP S/4HANA Public Cloud ABAP 拡張ブートキャンプ演習 3 [www.sap.com](http://www.sap.com).

© 2023 SAP SE or an SAP affiliate company. All rights reserved. See Legal Notice on [www.sap.com/legal-notice](https://www.sap.com/legal-notice) for use terms, disclaimers, disclosures, or restrictions related to SAP Materials for general audiences.

