

SAP BTP Onpremise Extension Tutorial_221126

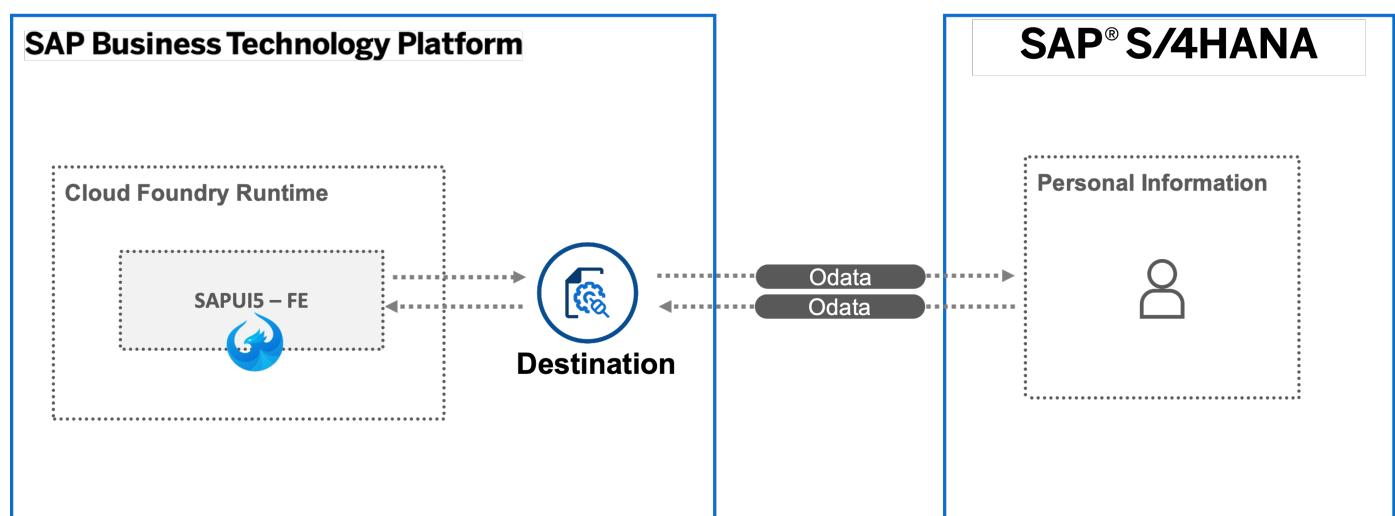
이 튜토리얼은 SAP BTP에서 SAP Onpremise OData를 통한 개발 방법을 설명합니다.

해당 프로젝트는 BTP Tutorial의 mta-hkmc를 기반으로 합니다.

1. Architecture Overview

다음은 애플리케이션의 전체 아키텍처를 나타내는 다이어그램입니다.

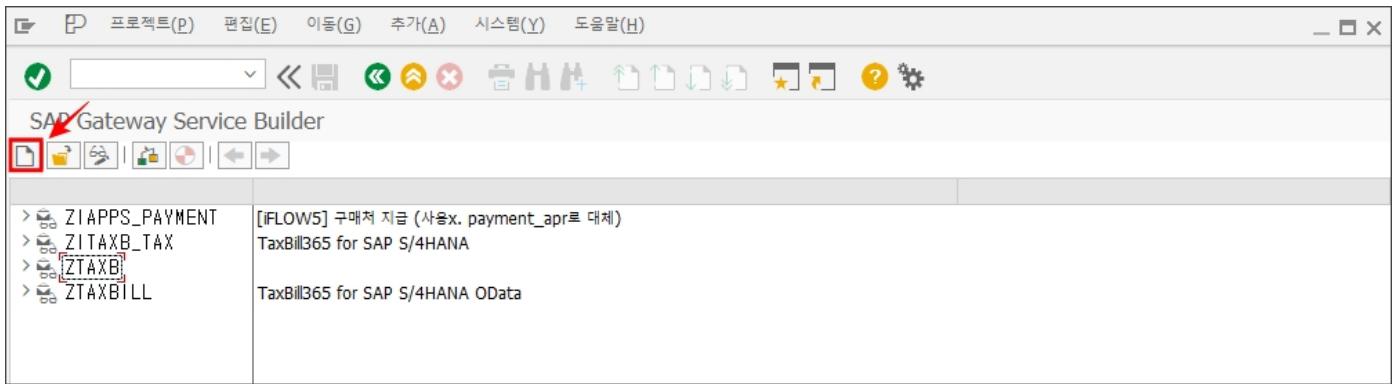
Destination 사용하여 Onpremise에서 OData API를 통해 정보를 읽고 직원의 엔터티를 이용하는 Application입니다.



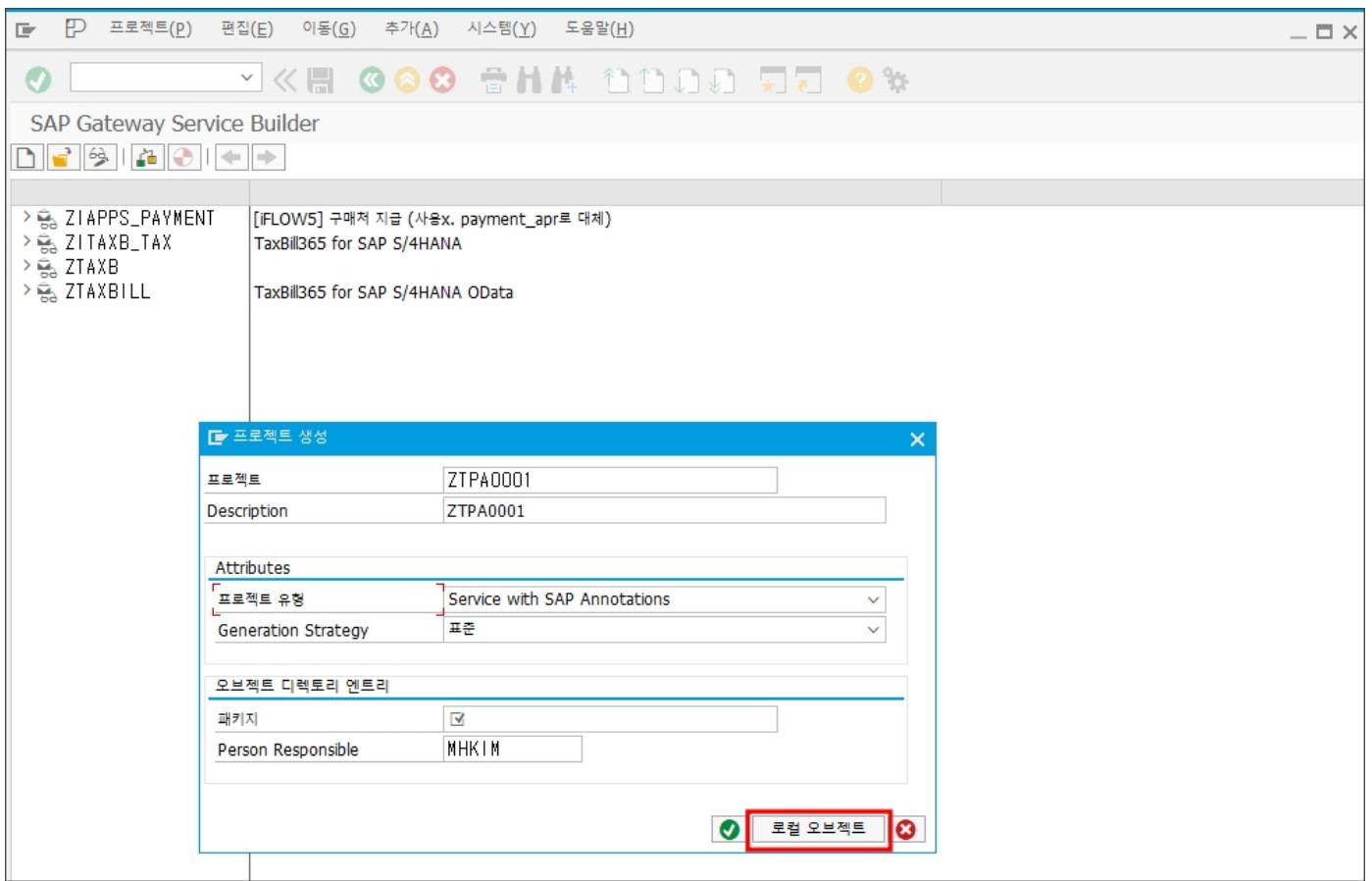
2. Onpremise OData 생성

1. SAPGUI > SE11 > TABLE > PA0001
2. HR 마스터 테이블을 참조해 OData를 생성하겠습니다.

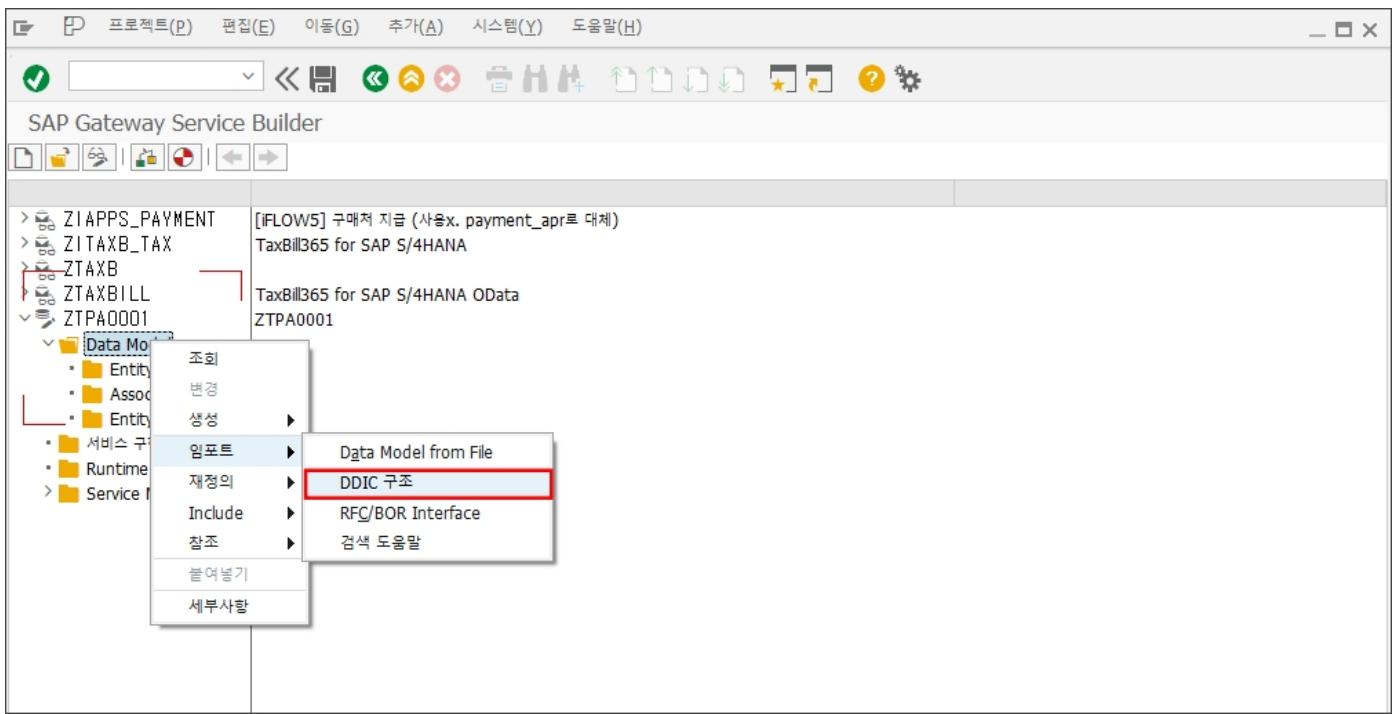
3. Tcode: SEGW > 신규 OData 서비스 생성



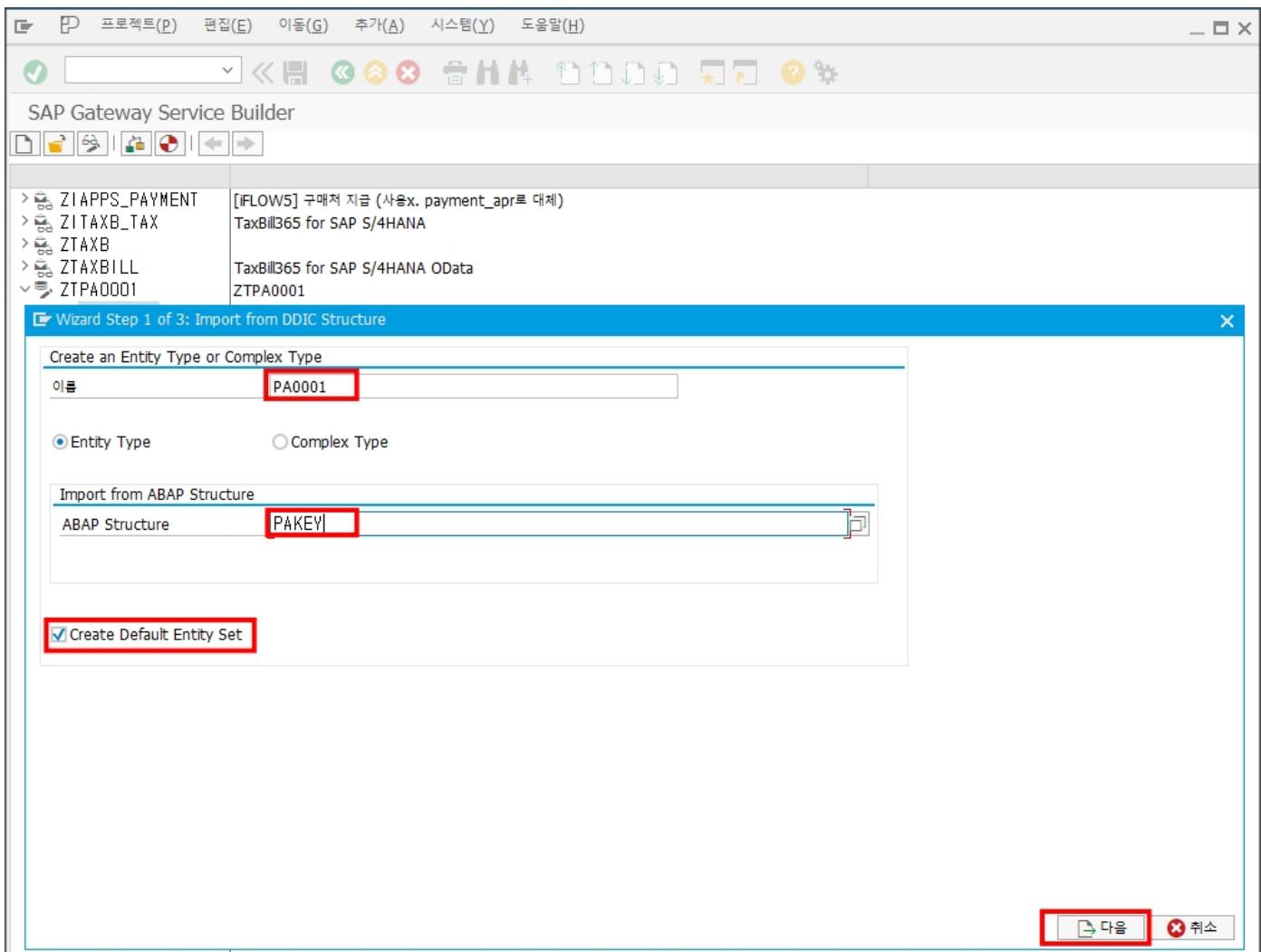
4. 로컬오브젝트 생성



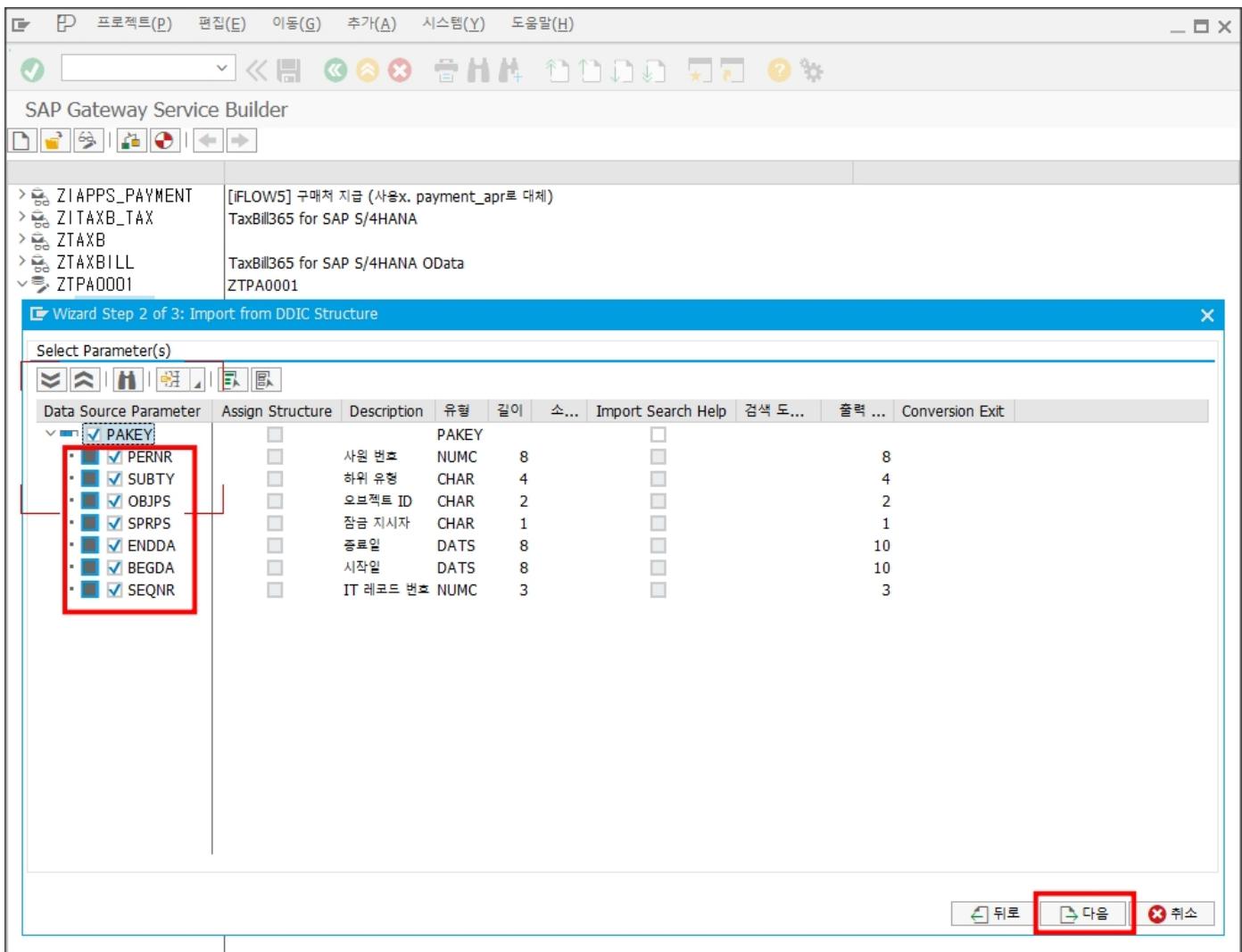
5. 생성한 서비스 > Data Model > Import > DDIC Structure 선택



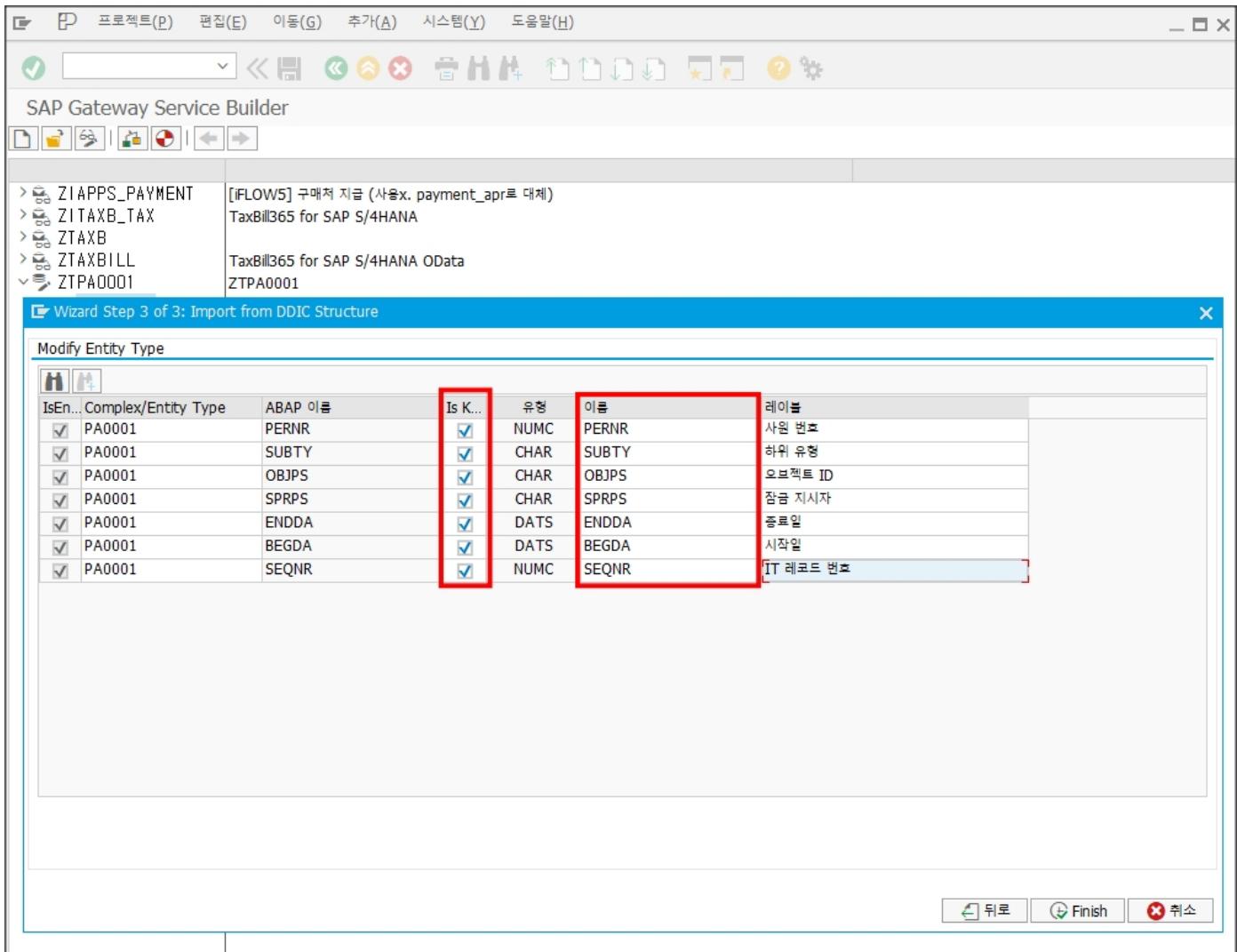
6. ENTITY명 PA0001 > Next



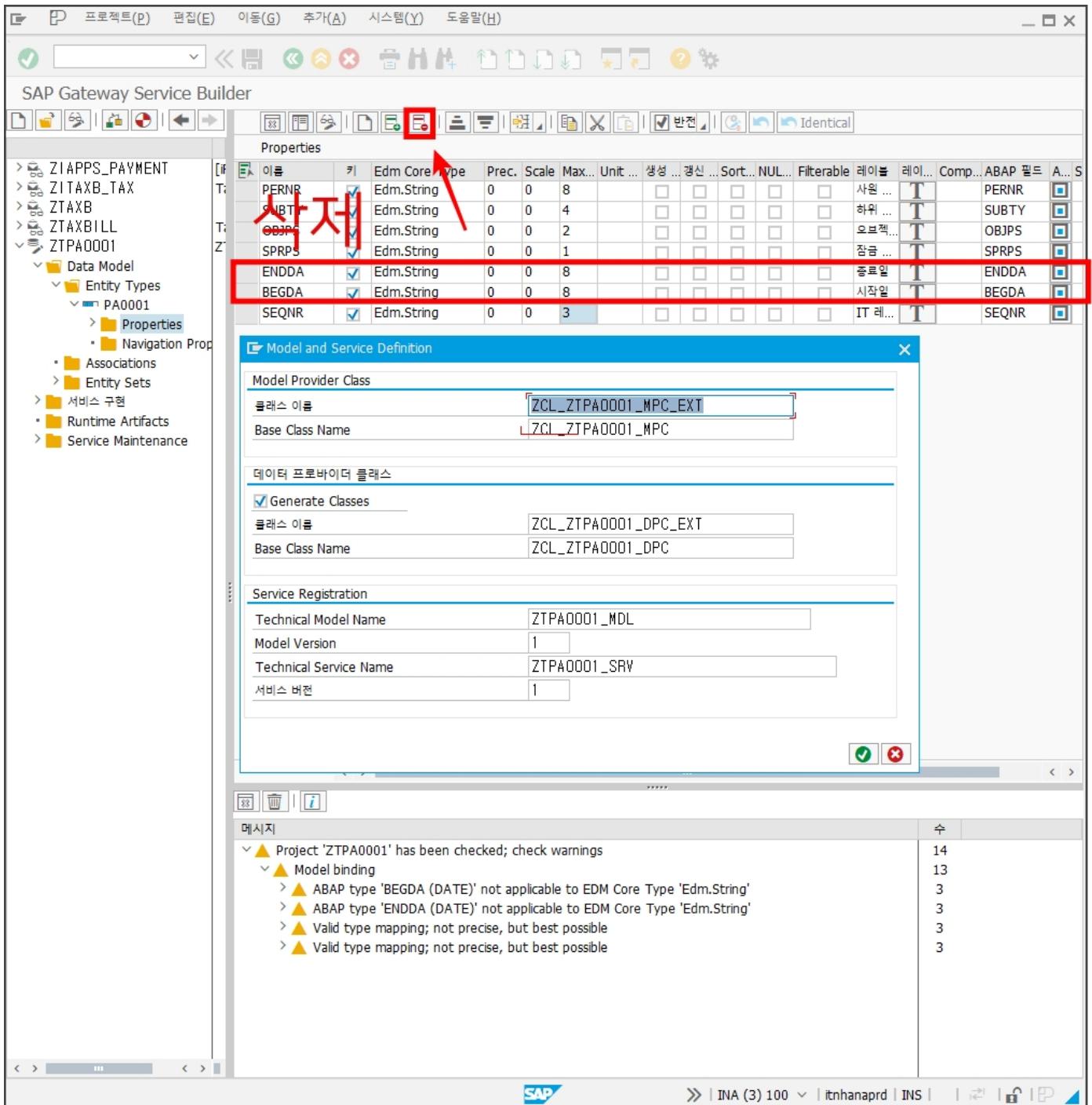
7. Structure에서 모든 필드를 선택 > Next



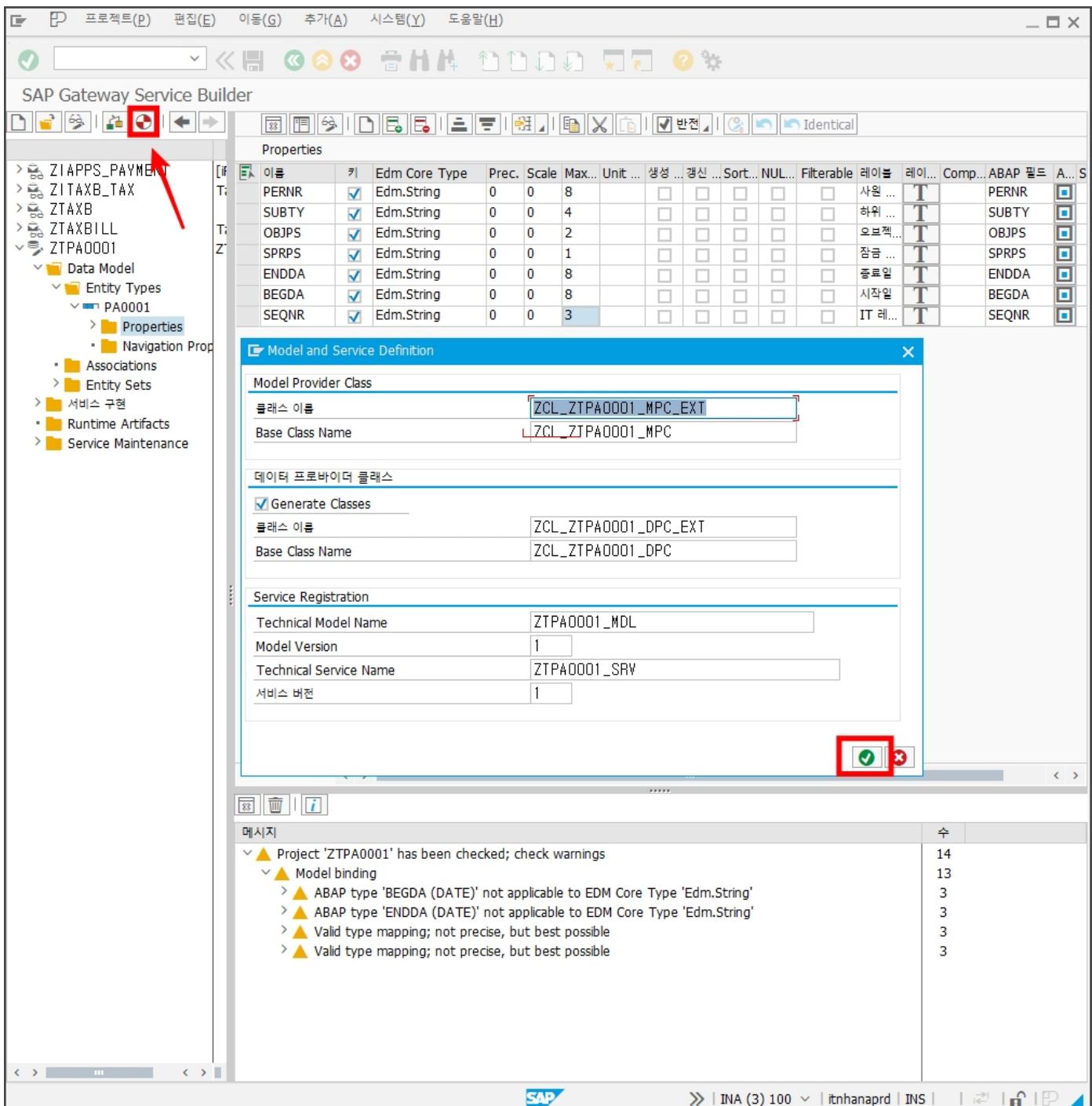
8. 모든 필드 Key선택, Label은 ABAP NAME과 같이 대문자로 통일



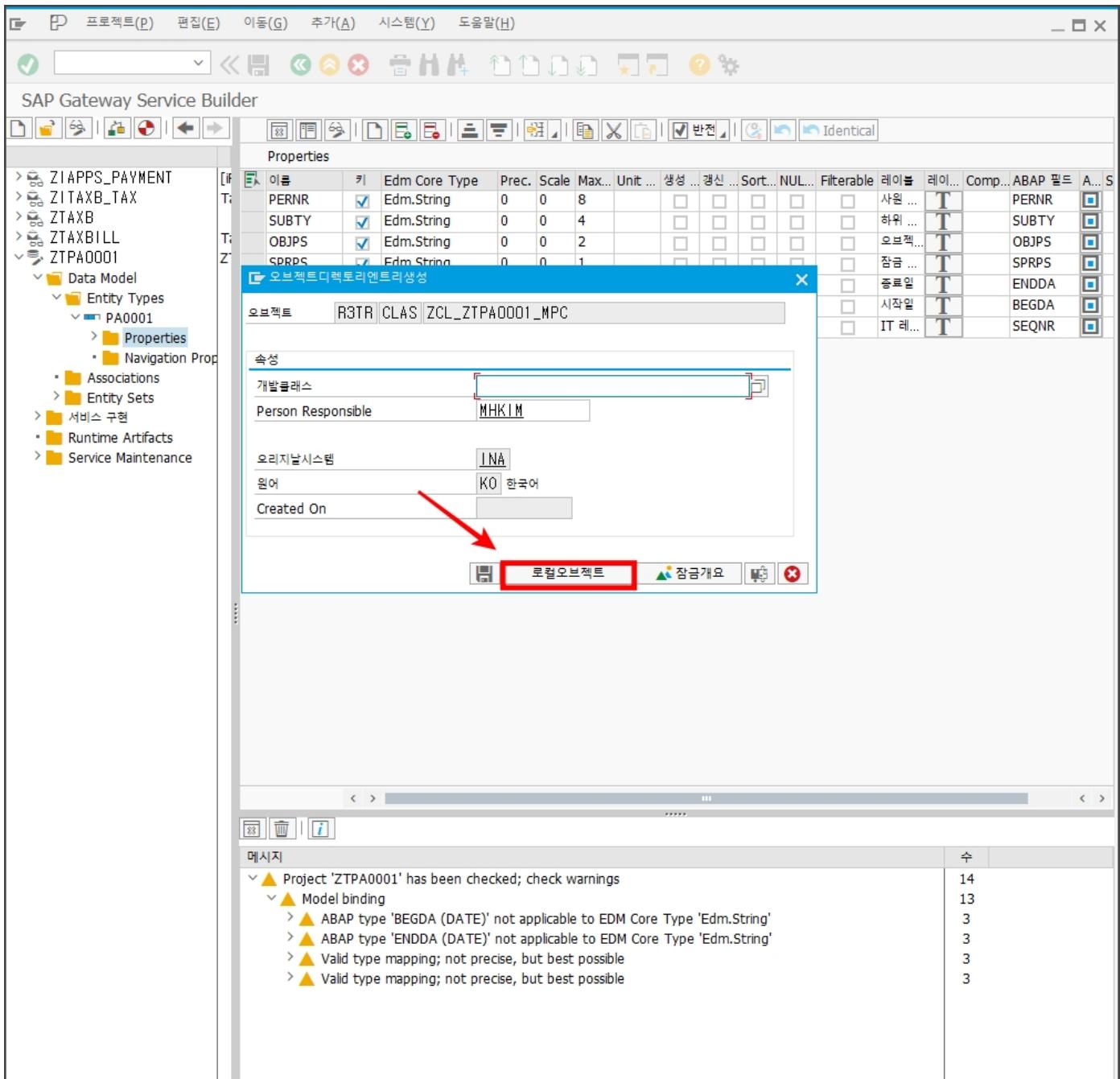
9. ENDDA, BEGDA 필드를 삭제



10. Generate > Confirm 선택



11. 로컬오브젝트 생성



12. 이제 서비스

SAP IDE for Java EE

클래스작성기: 클래스 ZCL_ZTPA0001_DPC_EXT 변경

Repository Browser

ZCL_ZTPA0001_DPC_EXT

Object Name

- ZCL_ZTPA0001_DPC_EXT
 - Superclasses
 - 속성
 - 방법
 - Types
 - Text Elements

Class/Interface: ZCL_ZTPA0001_DPC_EXT

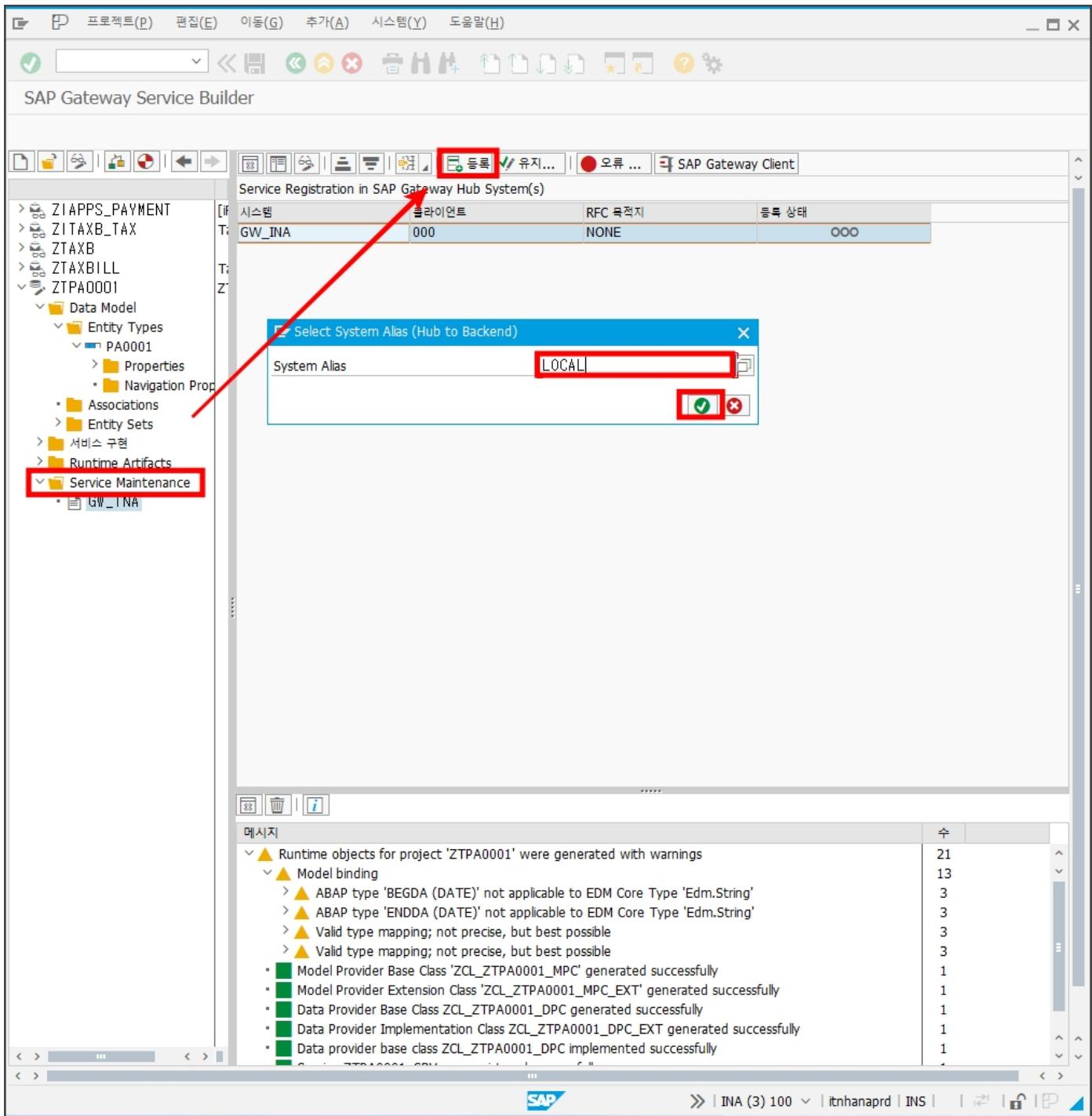
방법

| | 레벨 | 가시성 | 설명 |
|--|-----------------|-----------|---------|
| /IWBEPE/IF_MGW_APPL_SRV_RUNTIME~CHANGESET_PROCESS | Instance Method | Public | Process |
| /IWBEPE/IF_MGW_APPL_SRV_RUNTIME~CREATE_DEEP_ENTITY | Instance Method | Public | Execute |
| /IWBEPE/IF_MGW_APPL_SRV_RUNTIME~CREATE_ENTITY | Instance Method | Public | Execute |
| /IWBEPE/IF_MGW_APPL_SRV_RUNTIME~CREATE_STREAM | Instance Method | Public | Execute |
| /IWBEPE/IF_MGW_APPL_SRV_RUNTIME~DELETE_ENTITY | Instance Method | Public | Execute |
| /IWBEPE/IF_MGW_APPL_SRV_RUNTIME~DELETE_STREAM | Instance Method | Public | Execute |
| /IWBEPE/IF_MGW_APPL_SRV_RUNTIME~EXECUTE_ACTION | Instance Method | Public | execute |
| /IWBEPE/IF_MGW_APPL_SRV_RUNTIME~GET_ENTITY | Instance Method | Public | Execute |
| /IWBEPE/IF_MGW_APPL_SRV_RUNTIME~GET_ENTITYSET | Instance Method | Public | Execute |
| /IWBEPE/IF_MGW_APPL_SRV_RUNTIME~GET_ENTITYSET_DELTA | Instance Method | Public | Execute |
| /IWBEPE/IF_MGW_APPL_SRV_RUNTIME~GET_EXPANDED_ENTITY | Instance Method | Public | Execute |
| /IWBEPE/IF_MGW_APPL_SRV_RUNTIME~GET_EXPANDED_ENTITYSET | Instance Method | Public | Execute |
| /IWBEPE/IF_MGW_APPL_SRV_RUNTIME~GET_IS_CONDITIONAL_IM... | Instance Method | Public | Get fl |
| /IWBEPE/IF_MGW_APPL_SRV_RUNTIME~GET_IS_CONDITIONAL_FO... | Instance Method | Public | Get fl |
| /IWBEPE/IF_MGW_APPL_SRV_RUNTIME~GET_STREAM | Instance Method | Public | Execute |
| /IWBEPE/IF_MGW_APPL_SRV_RUNTIME~PATCH_ENTITY | Instance Method | Public | Execute |
| /IWBEPE/IF_MGW_APPL_SRV_RUNTIME~UPDATE_ENTITY | Instance Method | Public | Execute |
| /IWBEPE/IF_MGW_APPL_SRV_RUNTIME~UPDATE_STREAM | Instance Method | Public | Execute |
| /IWBEPE/IF_SB_DPC_COMM_SERVICES~GET_GENERATION_STRATE... | Instance Method | Public | Get ge |
| /IWBEPE/IF_SB_DPC_COMM_SERVICES~RFC_EXCEPTION_HANDLING | Instance Method | Public | RFC ca |
| /IWBEPE/IF_SB_DPC_COMM_SERVICES~RFC_SAVE_LOG | Instance Method | Public | Save t |
| /IWBEPE/IF_SB_DPC_COMM_SERVICES~SET_INJECTION | Instance Method | Public | SET D |
| /IWBEPE/IF_SB_DPC_COMM_SERVICES~LOG_MESSAGE | Instance Method | Public | Log m |
| /IWBEPE/IF_SB_DPC_COMM_SERVICES~COMMIT_WORK | Instance Method | Public | Call RF |
| CHECK_SUBSCRIPTION_AUTHORITY | Instance Method | Protected | |
| PA0001SET_CREATE_ENTITY | Instance Method | Protected | Relate |
| PA0001SET_DELETE_ENTITY | Instance Method | Protected | Relate |
| PA0001SET_GET_ENTITY | Instance Method | Protected | Relate |
| PA0001SET_GET_ENTITYSET | Instance Method | Protected | Relate |
| PA0001SET_UPDATE_ENTITY | Instance Method | Protected | Relate |

SAP

» | INA (3) 100 | itnhanapr | INS |

13. 이제 서비스



14. GET_ENTITYSET 작성해 PA0001 테이블의 데이터를 OData를 통해 노출시킵니다.

The screenshot shows the SAP ABAP Development Workbench. The title bar reads "클래스작성기: 클래스 ZCL_ZTPA0001_DPC_EXT 변경". The left pane is the "Repository Browser" showing the object structure under "ZCL_ZTPA0001_DPC_EXT". The right pane displays the ABAP code for the "PA0001SET_GET_ENTITYSET" method:

```
1 METHOD pa0001set_get_entityset.
2
3   DATA: lt_pa0001 TYPE zcl_ztpa0001_mpc_ext->tt_pa0001.
4
5   SELECT * INTO CORRESPONDING FIELDS OF TABLE lt_pa0001
6     FROM pa0001.
7
8   MOVE-CORRESPONDING lt_pa0001 TO et_entityset.
9
10  CLEAR: lt_pa0001.
11
12
13
14  ENDMETHOD.
```

15. Tcode: /IWFND/MAINT_SERVICE > Service Catalog 목록에서 서비스 하나를 선택 > 브라우저 열기 선택

The screenshot shows the SAP Service Studio interface. At the top, there's a toolbar with various icons like save, open, and search. Below it is a menu bar with Korean labels: 서비스 유지보수(M), 편집(E), 이동(G), 시스템(Y), 도움말(H). The main window has a title bar "서비스 활성화 및 유지보수". Below that is a toolbar with buttons for catalog, OAuth, Soft State, etc. The main area is divided into two tabs: "서비스 카탈로그" and "ICF 노드".

서비스 카탈로그 tab content:

| 유형 | 기술 서비스 이름 | 버전 | 서비스 내역 | 외부 서비스 이름 | 네임스페이스 OAuth... | 소프트 |
|-----|----------------------------------|----|---|--------------------------------|-----------------|-------------------------------------|
| BEP | ZABAP REPOSITORY_SRV | 1 | SAPUI5 ABAP Repository services via oData | ABAP REPOSITORY_SRV | /UIS/ | <input type="checkbox"/> |
| BEP | ZACCDDET | 1 | | ACCDDET | | <input type="checkbox"/> |
| BEP | ZACM_APPLWC | 1 | ACM Application Workcenter | ACM_APPLWC | | <input type="checkbox"/> |
| BEP | ZACM_CNTRCTAQ | 1 | ACM Contract Approval Queue | ACM_CNTRCTAQ | | <input type="checkbox"/> |
| BEP | ZACM_CNTRCTWC | 1 | ACM Contract Work Center | ACM_CNTRCTWC | | <input type="checkbox"/> |
| BEP | ZACM_MNTNCONTR | 1 | ACM Maintain Contract | ACM_MNTNCONTR | | <input type="checkbox"/> |
| BEP | ZACM_MNTNPRCNG | 1 | ACM Maintain Pricing | ACM_MNTNPRCNG | | <input type="checkbox"/> |
| BEP | ZACM_PRCNGWC | 1 | ACM Pricing Workcenter | ACM_PRCNGWC | | <input type="checkbox"/> |
| BEP | ZACM_STTLAQ | 1 | ACM Settlement Approval Queue | ACM_STTLAQ | | <input type="checkbox"/> |
| BEP | ZACM_STTLWC | 1 | ACM Settlement Work Center | ACM_STTLWC | | <input type="checkbox"/> |
| BEP | ZADAPTATION_PERS | 1 | UI BE Adaptation Service: Personalization | ADAPTATION_PERS | /UI2/ | <input type="checkbox"/> |
| BEP | ADT | 1 | Gateway service for ADT | ADT_SRV | | <input checked="" type="checkbox"/> |
| BEP | ZANA_PA1_PF_SRV | 1 | Predictive Analytics Integrator (PAI) - PF Integrati... | ANA_PA1_PF_SRV | | <input type="checkbox"/> |
| BEP | ZANA_PA1_PS_SRV | 1 | Predictive Analytics Integrator (PAI) - PS Review | ANA_PA1_PS_SRV | | <input type="checkbox"/> |
| BEP | ZANA_PA1_REPOSITORY_SRV | 1 | Predictive Analytics Integrator (PAI) Repository | ANA_PA1_REPOSITORY_SRV | | <input type="checkbox"/> |
| BEP | ZANA_QUERY_DESIGNER | 1 | Analytical Query Designer | ANA_QUERY_DESIGNER | | <input type="checkbox"/> |
| BEP | ZANA_QUERY_DESIGNER_MTD_EXT_S... | 1 | Custom analytical queries metadata extension | ANA_QUERY_DESIGNER_MTD_EXT_SRV | | <input type="checkbox"/> |
| BEP | ZANA_UMM_REPOSITORY | 1 | Advanced Analytics UMM Repository | ANA_UMM_REPOSITORY | | <input type="checkbox"/> |
| BEP | ZAPI_CV_ATTACHMENT_SRV | 1 | Gateway: API for Attachment Service | API_CV_ATTACHMENT_SRV | | <input type="checkbox"/> |
| BEP | ZAPI_DISPUTECASE | 1 | Dispute Case API | API_DISPUTECASE | | <input type="checkbox"/> |
| BEP | ZAPI_DISPUTECASEASCOORDTR | 1 | Dispute Case As Coordinator | API_DISPUTECASEASCOORDTR | | <input type="checkbox"/> |
| BEP | ZAPI_DISPUTECASEASPERSONRESP | 1 | Dispute Case As Person Responsible | API_DISPUTECASEASPERSONRESP | | <input type="checkbox"/> |
| BEP | ZAPI_DISPUTECASEASPROCESSOR | 1 | Dispute Case As Processor | API_DISPUTECASEASPROCESSOR | | <input type="checkbox"/> |
| BEP | ZAPI_DISTRIBUTIONCHANNEL_SRV | 1 | Distribution Channel - Read | API_DISTRIBUTIONCHANNEL_SRV | | <input type="checkbox"/> |

ICF 노드 tab content:

| 상태 | ICF 노드 | 세션 시간 초과 소프트 스테йт | 내역 |
|----|--------|-------------------|---------------|
| OO | ODATA | 00:00:00 | Standard Mode |

시스템 별칭 tab content:

| SAP System Alias | Description | Default |
|------------------|-------------|---------|
| INA (4) 100 | | |

16. 브라우저에서 SAP Gateway의 Host를 확인

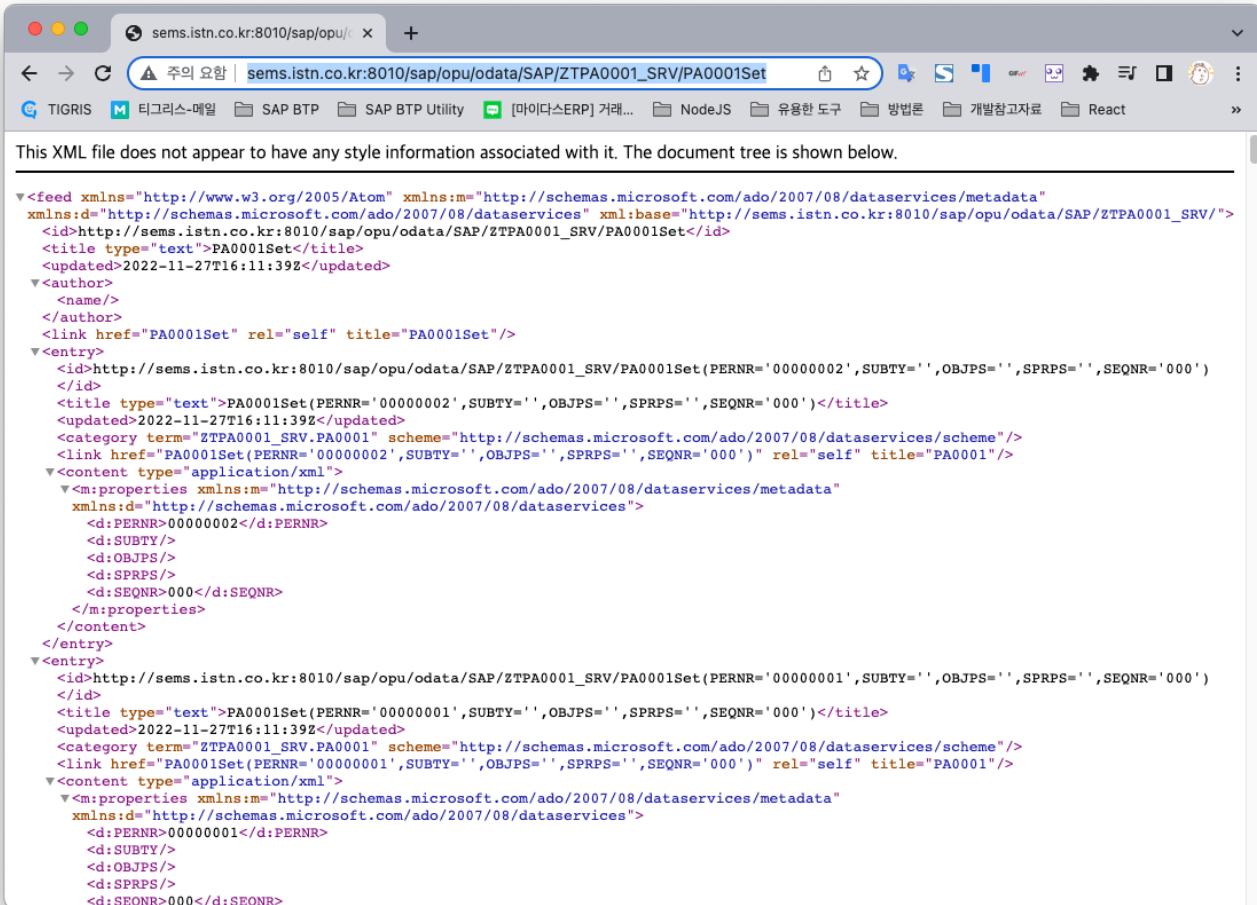
SAP BTP에서 OData를 호출시 Destination에 등록하여 사용합니다.

itnhanaprd.istn.co.kr:8010 해당 서버는 예시이므로 실제 호출 주소는 사용하시는 서버의 아래의 방법을 통해 Host를 직접 확인해야 합니다.

The screenshot shows the SAP Service Activation and Maintenance (SAM) interface. At the top, there's a toolbar with various icons. Below it is a navigation bar with links like '카탈로그 새로 ...', 'OAuth', 'Soft State', '처리 ...', and '전송에 ...'. The main area has tabs for '서비스 카탈로그' and '서비스 활성화 및 유지보수'. A sub-tab 'HR table for employee informat...' is selected. In the center, there's a browser-like window showing a URL: [itnhanaprd.istn.co.kr:8010/sap/opu/odata/sap/ACM_CNTRCTAQ/?\\$format=xml](itnhanaprd.istn.co.kr:8010/sap/opu/odata/sap/ACM_CNTRCTAQ/?$format=xml). The URL bar is highlighted with a red box. Below the browser window, there's a list of services and a table of API mappings. On the left, a sidebar lists various BEP services. On the right, a table maps BEP services to APIs, such as 'ZABAP_REPOSE' to 'API_DISPUTECASEASCOORDTR'. At the bottom, there are two panes: 'ICF 노드' and '시스템 별칭'. The 'ICF 노드' pane shows 'ODATA' is active. The '시스템 별칭' pane shows a table with columns 'SAP System Alias', 'Description', and 'Default'.

17. 실제 OData를 브라우저를 통해 호출한 모습입니다. 이제 이 OData를 SAP BTP Application을 만들어보겠습니다.

| | |
|--------------|---|
| 전체 주소 | http://sems.istn.co.kr:8010/sap/opu/odata/SAP/ZTPA0001_SRV/PA0001Set |
| HOST | http://sems.istn.co.kr:8010 |
| ODATA URI | /sap/opu/odata/SAP/ |
| Service Name | ZTPA0001_SRV |
| Entity Name | PA0001Set |



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

```

<feed xmlns="http://www.w3.org/2005/Atom" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
      xmlns:ds="http://schemas.microsoft.com/ado/2007/08/dataservices" xml:base="http://sems.istn.co.kr:8010/sap/opu/odata/SAP/ZTPA0001_SRV/PA0001Set">
  <id>http://sems.istn.co.kr:8010/sap/opu/odata/SAP/ZTPA0001_SRV/PA0001Set</id>
  <title type="text">PA0001Set</title>
  <updated>2022-11-27T16:11:39Z</updated>
  <author>
    <name/>
  </author>
  <link href="PA0001Set" rel="self" title="PA0001Set"/>
<entry>
  <id>http://sems.istn.co.kr:8010/sap/opu/odata/SAP/ZTPA0001_SRV/PA0001Set(PERNR='00000002',SUBTY='',OBJPS='',SPRPS='',SEQNR='000')</id>
  <title type="text">PA0001Set(PERNR='00000002',SUBTY='',OBJPS='',SPRPS='',SEQNR='000')</title>
  <updated>2022-11-27T16:11:39Z</updated>
  <category term="ZTPA0001_SRV.PA0001" scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
  <link href="PA0001Set(PERNR='00000002',SUBTY='',OBJPS='',SPRPS='',SEQNR='000')" rel="self" title="PA0001"/>
  <content type="application/xml">
    <m:properties xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
                  xmlns:ds="http://schemas.microsoft.com/ado/2007/08/dataservices">
      <d:PERNR>00000002</d:PERNR>
      <d:SUBTY/>
      <d:OBJPS/>
      <d:SPRPS/>
      <d:SEQNR>000</d:SEQNR>
    </m:properties>
  </content>
</entry>
<entry>
  <id>http://sems.istn.co.kr:8010/sap/opu/odata/SAP/ZTPA0001_SRV/PA0001Set(PERNR='00000001',SUBTY='',OBJPS='',SPRPS='',SEQNR='000')</id>
  <title type="text">PA0001Set(PERNR='00000001',SUBTY='',OBJPS='',SPRPS='',SEQNR='000')</title>
  <updated>2022-11-27T16:11:39Z</updated>
  <category term="ZTPA0001_SRV.PA0001" scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
  <link href="PA0001Set(PERNR='00000001',SUBTY='',OBJPS='',SPRPS='',SEQNR='000')" rel="self" title="PA0001"/>
  <content type="application/xml">
    <m:properties xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
                  xmlns:ds="http://schemas.microsoft.com/ado/2007/08/dataservices">
      <d:PERNR>00000001</d:PERNR>
      <d:SUBTY/>
      <d:OBJPS/>
      <d:SPRPS/>
      <d:SEQNR>000</d:SEQNR>
    </m:properties>
  </content>
</entry>

```

3. Application Configuration

이 튜토리얼은 이전에 생성한 `mta-hkmc` 를 기반으로 합니다.

만약 프로젝트가 생성되어 있지 않다면, 이전 튜토리얼을 먼저 진행하시기 바랍니다.

앱을 실행하기 전에 우선적으로 설정해줘야하는 내역을 설명합니다.

1) Approuter Configuration

4. 이제 이전 튜토리얼에서 생성해둔 mta-hkmc 프로젝트로 이동하여 Approuter Destination을 작성합니다.

SAP Approuter

Application이 여러 개의 서로 다른 앱(マイクロサービス)으로 구성된 경우 Application Router는 해당 Application에 대한 단일 진입 점(Single entry point)을 제공하는 데 Nodejs 기반의 라이브러리입니다.

SAP Approuter를 사용할 수 있는 환경

- SAP CF – Cloud Foundry
- SAP XSA – XS Advanced (On Premise)
- Local environment

주요기능

- Dispatching of requests to other microservices
- Authentication
- Authorization check
- Complete integration with Destination service
- Complete integration with HTML5 Application repository
- Complete integration with Business Services

5. /mta-hkmc/mta-hkmc-approuter 경로에서 **default-env.json** 파일을 생성하여 destination을 작성합니다. url을 호출시 name으로 대체하여 url을 호출합니다.

default-env.json

```
{  
  "destinations": [  
    {  
      "url": "https://sandbox.api.sap.com",  
      "name": "sfsf_sandbox"  
    },  
    {  
      "url": "http://sems.istn.co.kr:8010/",  
      "name": "onpremise"  
    }  
}
```

6. /mta-hkmc/mta-hkmc-approuter 경로에서 **xs-app.json**과 **xs-app-local.json** 파일을 수정합니다.

xs-app.json

```
{  
  "welcomeFile": "comhkmctrialztmp001/index.html",  
  "authenticationMethod": "none",  
  "routes": [  
    {  
      "source": "^/onpremise/(.*)$",  
      "target": "$1",  
      "destination": "onpremise",  
      "csrfProtection": false  
    },  
    {  
      "source": "^/onpremise/(.*)$",  
      "target": "$1",  
      "destination": "onpremise",  
      "csrfProtection": false  
    }  
  ]  
}
```

xs-app-local.json

```
{  
  "welcomeFile": "comhkmctrialztmp001/index.html",  
  "authenticationMethod": "none",  
  "routes": [  
    {  
      "source": "^/comhkmctrialztmp001/(.*)$",  
      "target": "$1",  
      "localDir": "../ztmp_001/webapp"  
    },  
    {  
      "source": "^/sfsf_sandbox/(.*)$",  
      "target": "$1",  
      "destination": "sfsf_sandbox",  
      "csrfProtection": false  
    },  
    {  
      "source": "^/onpremise/(.*)$",  
      "target": "$1",  
      "destination": "onpremise",  
      "csrfProtection": false  
    }  
  ]  
}
```

```
        "csrfProtection": false
    }
]
}
```

4. SAPUI5 Application 작성

이제 OData API를 호출해 SAPUI5 Table Component에 데이터를 바인딩해서 출력해보겠습니다.

/mta-hkmc/ztmp_001/webapp/controller 경로에서 main.controller.js 파일을 수정합니다.

main.controller.js

```
sap.ui.define([
    "sap/ui/core/mvc/Controller",
    "sap/ui/model/json/JSONModel",
],
[
    /**
     * @param {typeof sap.ui.core.mvc.Controller} Controller
     */
    function (Controller, JSONModel) {
        "use strict";

        return Controller.extend("com.hkmc.trial.ztmp001.controller.Main", {
            OnInit: function () {
                this._getEmployeeInfo();
                this._getUsersInfo();
            },
            _getEmployeeInfo: function() {
                var view = this.getView();
                var options = {
                    url: "/sfsf_sandbox/successfactors/odata/v2/PerEmergencyContacts",
                    method: "get",
                    headers: {
                        "APIKey": "jzS6mZfacFXArH27IpN03CYzDHF1oq0",
                        "Accept": "application/json"
                    }
                }
                $.ajax(options).then(function(res) {
                    view.setModel(new JSONModel(res.d.results), "PerEmergencyContacts");
                });
            },
        });
    }
]);
```

```

_getUsersInfo: function() {
    var view = this.getView();
    var textAuthorization = "<sapid>:<password>"; // 접속 ID/PW를 입력
    console.log(textAuthorization);
    var encodeAuthorization = btoa(textAuthorization);
    console.log(encodeAuthorization);
    var options = {
        url: "/onpremise/sap/opu/odata/SAP/ZTPA0001_SRV/PA0001Set",
        method: "get",
        headers: {
            "Authorization": "Basic " + encodeAuthorization,
            "Accept": "application/json"
        }
    };
    $.ajax(options).then(function(res) {
        view.setModel(new JSONModel(res.d.results), "Users");
    });
}
);
});

```

ajax 호출을 통해 API 호출합니다. URL 호출에 Destination에서 설정한 name으로 호출시 Destination에 설정된 url을 호출하게 됩니다.

Onpremise에서 OData를 통해 데이터를 호출시 BasicAuth를 통해 OData에 접근합니다.

`_getUsersInfo` 는 Onpremise에서 OData를 호출해 view에 모델을 바인딩하는 함수입니다.

/mta-hkmc/ztmp_001/webapp/view 경로에서 main.view.xml 파일을 수정합니다.

main.view.xml

```

<mvc:View
    controllerName="com.hkmc.trial.ztmp001.controller.Main"
    xmlns:mvc="sap.ui.core.mvc"
    xmlns:ui="sap.ui.table"
    displayBlock="true"
    xmlns="sap.m" >
    <Page title="employee">
        <content>
            <ui:Table
                id="id-tableTxwa"
                visibleRowCount="10"

```

```

rows="{
    path: 'PerEmergencyContacts>/',
    sorter: [],
    templateShareable : false
}" selectionMode="MultiToggle" ariaLabelledBy="title">
<ui:extension>
    <OverflowToolbar>
        <Text text="Employee"/>
    </OverflowToolbar>
</ui:extension>
<ui:columns>
    <ui:Column hAlign="Center" autoResizable="true">
        <Label text="personIdExternal" />
        <ui:template>
            <Text text="{PerEmergencyContacts>personIdExternal}"/>
        </ui:template>
    </ui:Column>
    <ui:Column hAlign="Center" autoResizable="true">
        <Label text="name" />
        <ui:template>
            <Text text="{PerEmergencyContacts>name}"/>
        </ui:template>
    </ui:Column>
    <ui:Column hAlign="Center" autoResizable="true">
        <Label text="phone" />
        <ui:template>
            <Text text="{PerEmergencyContacts>phone}"/>
        </ui:template>
    </ui:Column>
    <ui:Column hAlign="Center" autoResizable="true">
        <Label text="email" />
        <ui:template>
            <Text text="{PerEmergencyContacts>email}"/>
        </ui:template>
    </ui:Column>
</ui:columns>
</ui:Table>

<ui:Table
    id="id-tableUsers"
    visibleRowCount="10"
    rows="{
        path: 'Users>/',
        sorter: [],
        templateShareable : false
}" selectionMode="MultiToggle" ariaLabelledBy="title">
<ui:extension>

```

```

<OverflowToolbar>
    <Text text="User"/>
</OverflowToolbar>
</ui:extension>
<ui:columns>
    <ui:Column hAlign="Center" autoResizable="true">
        <Label text="PERNR" />
        <ui:template>
            <Text text="{Users>PERNR}" />
        </ui:template>
    </ui:Column>
    <ui:Column hAlign="Center" autoResizable="true">
        <Label text="SUBTY" />
        <ui:template>
            <Text text="{Users>SUBTY}" />
        </ui:template>
    </ui:Column>
    <ui:Column hAlign="Center" autoResizable="true">
        <Label text="OBJPS" />
        <ui:template>
            <Text text="{Users>OBJPS}" />
        </ui:template>
    </ui:Column>
    <ui:Column hAlign="Center" autoResizable="true">
        <Label text="SPRPS" />
        <ui:template>
            <Text text="{Users>SPRPS}" />
        </ui:template>
    </ui:Column>
    <ui:Column hAlign="Center" autoResizable="true">
        <Label text="SEQNR" />
        <ui:template>
            <Text text="{Users>SEQNR}" />
        </ui:template>
    </ui:Column>
</ui:columns>
</ui:Table>
</content>
</Page>
</mvc:View>

```

최종 결과

| employee | | | | |
|--------------------------|------------------|----------------|-------------------|--------------------------|
| Employee | | | | |
| | personIdExternal | name | phone | email |
| <input type="checkbox"/> | 108713 | Seph Ruiz | 714-455-9876 | |
| <input type="checkbox"/> | 108733 | Jessica Coburn | 678-456-3322 | jessica.coburn@email.com |
| <input type="checkbox"/> | 101010 | Roger Crow | 44 5678 09876 | r.crow@email.com |
| <input type="checkbox"/> | 102004 | Hannah Jones | 33 02 76 87 76 88 | h.jones@email.com |
| <input type="checkbox"/> | 102036 | Melissa Gape | 33 87 98765 | m.gape@email.com |
| <input type="checkbox"/> | 103018 | Wolfgang Dorn | 49 7768 7653245 | w.dorn@email.com |
| <input type="checkbox"/> | 103093 | Pallavi Singal | 89096543 | p.singal@email.com |
| <input type="checkbox"/> | 103095 | Pramod Pandian | 879086543 | p.pandian@email.com |
| <input type="checkbox"/> | 108732 | Wendy Heesch | 760-887-9876 | wheesch@email.com |
| <input type="checkbox"/> | 104044 | Vitoria | 555-8798 | |
| User | | | | |
| | PERNR | SUBTY | OBJPS | SPRPS |
| <input type="checkbox"/> | 00000002 | | | 000 |
| <input type="checkbox"/> | 00000001 | | | 000 |
| <input type="checkbox"/> | 00000001 | | | 000 |
| <input type="checkbox"/> | 00000024 | | | 000 |
| <input type="checkbox"/> | 00000025 | | | 000 |
| <input type="checkbox"/> | 00000026 | | | 000 |
| <input type="checkbox"/> | 00000003 | | | 000 |
| <input type="checkbox"/> | 00000004 | | | 000 |
| <input type="checkbox"/> | 00000005 | | | 000 |
| <input type="checkbox"/> | 00000006 | | | 000 |

여기까지 SAP SFSF API를 통한 Side-by-Side Extension에 대한 기본적인 개발 방법에 알아봤습니다.

해당 상세 소스는 아래의 git를 통해 확인하실 수 있습니다.

<https://github.com/rendez0829/mta-hkmc-onpremise>