**SAP Cloud Platform ABAP Environment Handbook**

* End to End ABAP Cloud Implementation for S4HC Side-by-side extensibility

**Han Jungwoo(Jungwoo.han@sap.com)**

**Digital Platform**

**SAP Korea**

|  |  |  |  |
| --- | --- | --- | --- |
| **version** | **Update date** | **writer** | **comment** |
| **1.0** | **2019/03/24** | **Jungwoo.han@sap.com** |  |
| **2.0** | **2022/12/11** | **Jungwoo.han@sap.com** | **Good progression** |

Table of ContentS

[Before you start 4](#_Toc122039024)

[Chapter 1: Why sap cloud platform abap environment 5](#_Toc122039025)

[Chapter 2: Step by step implementation of abap cloud 5](#_Toc122039026)

[2.1 Prerequisites 5](#_Toc122039027)

[2.2 RAP Implementation guides(Managed Implementation) 11](#_Toc122039028)

[2.2 RAP Implementation guides(Unmanaged Implementation) 11](#_Toc122039029)

[2.3 Applied CDS Associations 21](#_Toc122039030)

[2.4 Applied CDS Annotations(Designing the User Interface for a Fiori Elements App) 22](#_Toc122039031)

[Chapter 3: Integration with s/4hana : Outbound communication 24](#_Toc122039032)

[3.1 ABAP Cloud와 S/4H간의 interface – S/4HANA RFC호출 Custom Entity 26](#_Toc122039033)

[3.2 A4C에서 S/4HC API에 대한 Select/Insert 26](#_Toc122039034)

[Chapter 5: cds (Core data service) 35](#_Toc122039035)

[4.1 CDS 목적 및 특징 35](#_Toc122039036)

[4.2 CDS Design Principle 36](#_Toc122039037)

[4.3 CDS templates(ABAP CDS Entities) 37](#_Toc122039038)

[4.4 CDS Associations & Path 44](#_Toc122039039)

[4.5 CDS path 45](#_Toc122039040)

[4.6 CDS Annotations 46](#_Toc122039041)

[4.6.1 CDS Annotations – AccessControl Annotations 46](#_Toc122039042)

[4.6.2 CDS Annotations – Consumption Annotation 48](#_Toc122039043)

[4.6.3 CDS Annotations – ObjectModel Annotations 50](#_Toc122039044)

[4.6.4 CDS Annotations – Search Annotations 51](#_Toc122039045)

[4.6.5 CDS Annotations – Semantics Annotations 52](#_Toc122039046)

[4.6.6 CDS Annotations – UI Annotations 53](#_Toc122039047)

[4.7 CDS SQL functions 61](#_Toc122039048)

[Chapter 5: UI Design & Development 64](#_Toc122039049)

[5.1 UI5 Development Guides with Layout Editor(code-less UI development) 64](#_Toc122039050)

[Summary 65](#_Toc122039051)

[Reference link 66](#_Toc122039052)

# Before you start

SAP는 ABAP 개발자분들을 위해 Cloud상에서 ABAP 개발을 수행할 수 있는 SAP CP ABAP Environment를 2018년 출시했으며, S/4HANA의 side-by-side extensibility를 위해 주로 사용될 예정입니다.

이번 Hands-on을 통해SAP ABAP Cloud에서 사용되는 Business Object의 concept과 이를 위한 개발 framework을 study할 수 있습니다.(CDS를 이용한 Business Object 정의, Business Object에 대한 Behavior정의를 위한 Cloud optimized ABAP implementation 그리고 OData 생성을 위한 Service definition & binding 순서로 진행됩니다.)

**Chapter 1: Why ABAP Cloud**

* 1. Concept & Development Methodology
  2. Business Values

**Chapter 2: Step-by-step Implementation of ABAP Cloud**

2.1 데이터 모델 구체화

2.2 RAP Business Object 생성

2.3 RAP Business Object의 Behavior정의 및 구현

2.4 OData생성을 위한 Service definition & binding

2.5 Communicatoin Scenario및 Arrangement를 통해 외부에서 수행될 수 있는 API생성.(Inbound Communication)

2.6 CF환경에서의 연동테스트.

**Chapter 3: Integration with S/4HC (Outbound Communication)**

3.1 S/4H와의 integration설정 및 code수행

**Chapter 4: CDS Annotations**

4.1 Access Control Annotations

4.2 Consumption annotations

4.3 Object Model Annotations

4.4 Search annotations

4.5 Semantics annotations

4.6 UI Annotations

- 대표적인 UI Annotation 과 예시.

**Chapter 5: UI Development (codeless method – UI5 with WebIDE)**

5.1 codeless WebIDE를 활용한 UI development

# Chapter 1: Why sap cloud platform abap environment

ABAP개발자분들이 SAP Cloud Platform상에서 개발할 수 있는 환경을 제공해드리며, 아울러 S4H Cloud side-by-side extension의 플랫폼을 제공해 줍니다.

별도의 ABAP Cloud Architecture와 Business Value등에 관해서는 별도의 첨부 PPT를 참조하시면 됩니다.

.

SAP CP ABAP Cloud Environment는 Eclipse ADT를 통해 접속이 가능하며, RESTful방식의 API를 생성해 표준 프로토콜을 이용해 통신이 가능하며, 또한 상대방의 ODATA를 자유롭게 consume해서 사용 가능합니다.

# Chapter 2: Step by step implementation of abap cloud

우선 기초적인 레벨에서 ABAP Cloud를 개발하는 과정을 설명드리기로 하겠습니다.

2.1 Prerequisites

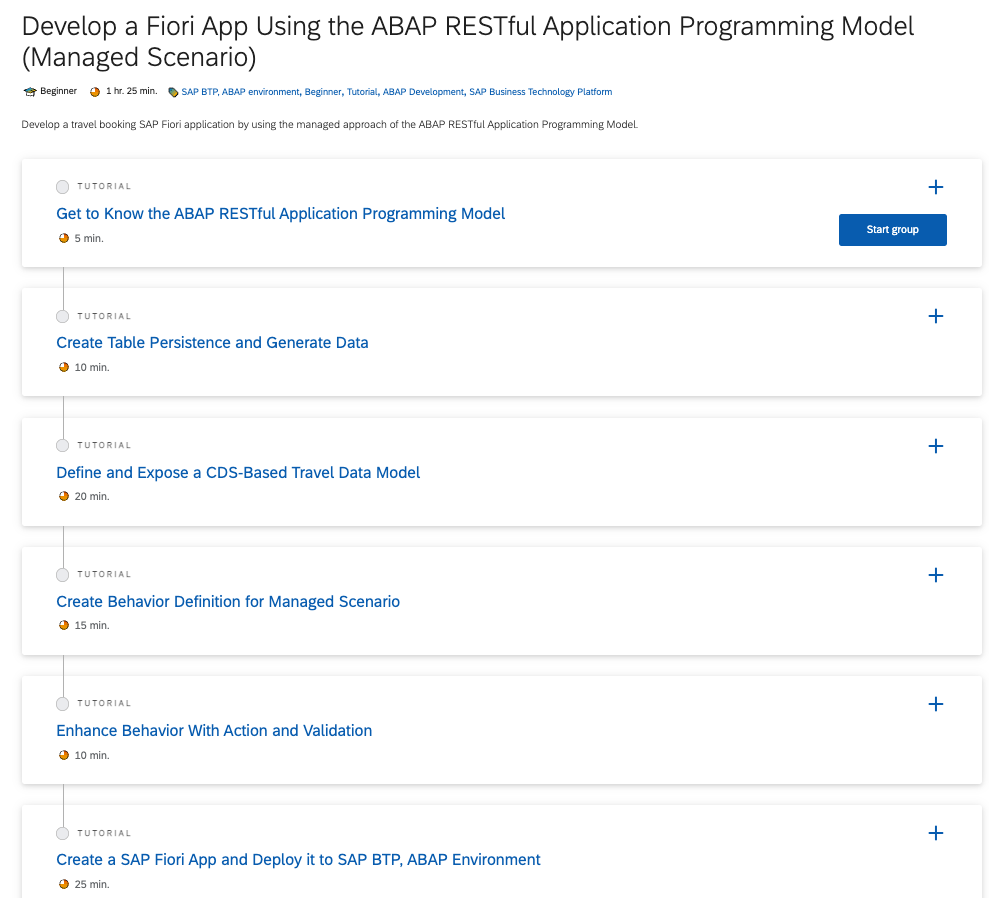
Eclipse Install & Setup.

| Explanation | Screenshot |
| --- | --- |
| Eclipse Installation | <https://developers.sap.com/tutorials/abap-install-adt.html>  위 사이트를 참조해서 Eclipse + ADT를 구성합니다. |
| ABAP Cloud Project 생성 클릭. |  |
| ABAP Service인스턴스에 연결.  Service Key 사용 선택후, Next> |  |
| 첨부파일로 전달된 service key 내용 copy/paste |  |
| “Open Logon Page in Browser” 클릭. |  |
| Connection Settings Overview 확인. |  |
| ABAP Package생성  **Note : 반드시 ZLOCAL\ZROOTPKG밑에 작업할Package 생성.**  **ZSUBPKG##** |  |
| ABAP Package생성 |  |
| ABAP Package생성 |  |
| ABAP Class생성 (테스트) |  |
| ABAP Class생성 (테스트)  **Note : ##을 붙여주세요.** |  |
| ABAP Class생성 (테스트) |  |
| ABAP Class Activation & Run (테스트) | Activataion 하고    out->write( 'Hello Wrold!' ). |
|  | 콘솔에서 결과 확인. |

2.2 RAP Implementation guides(Managed Implementation)

이번 장에서 사용된 implementation guides에 대해 살펴봅니다.

<https://developers.sap.com/group.abap-env-restful-managed.html>



2.2 RAP Implementation guides(Unmanaged Implementation)

이번 장에서 사용된 implementation guides에 대해 살펴봅니다.

| Explanation | Screenshot |
| --- | --- |
| 1. Business Object정의 | Defining Business Object  아래와 같이 본인이 원하는 Business Object을 정의 할 수 있으며, 궁극적으로는 CDS를 통해 Data modeling 을 구현합니다.  아래 예제에서는 Booking이라는 CDS를 정의하고 C/U/D가 발생하게끔 Root CDS로 명명합니다.  Currency와 Country 데이터는 기존에 활용될 수 있는CDS로 정의되어 있습니다. |
|  |  |
| **Note : ##을 붙여주세요.** | 예제) Ztbooking 🡪 ZS0001BK01   |  |  | | --- | --- | |  | 테이블 스크립트  @EndUserText.label : 'booking table'  @AbapCatalog.enhancementCategory : #NOT\_EXTENSIBLE  @AbapCatalog.tableCategory : #TRANSPARENT  @AbapCatalog.deliveryClass : #A  @AbapCatalog.dataMaintenance : #RESTRICTED  define table ztbooking## {  key client : abap.clnt not null;  key booking : abap.int4 not null;  customername : abap.char(50);  numberofpassengers : abap.int2;  emailaddress : abap.char(50);  country : abap.char(50);  dateofbooking : timestampl;  dateoftravel : timestampl;  @Semantics.amount.currencyCode : 'ztbooking.currencycode'  cost : abap.curr(15,2);  currencycode : abap.cuky;  lastchangedat : timestampl;  } | |
| 1. CDS 데이터 모델링   - “Booking” Root CDS정의.  **Note : ##을 붙여주세요.**  Entity Core 정보.  C/U/D를 나타내는 root view.  Association을 통해 entity간의 relationship표현.  UI.facet annotation을 이용해detail page 정의  Position은 화면상에 위치를 나타냄.  (참고로 절대적인 좌표가 아닌 상대적인 위치이며 ordering에 의해 순서가 정해진다)  “CustomerName”으로 검색  Label은 Display될 이름을 표현합니다.  @UI,dataPoint를 이용해 데이터 강조 | |  |  | | --- | --- | |  |  |   @AccessControl.authorizationCheck: #NOT\_REQUIRED  @EndUserText.label: 'cds for business object booking'  @Search.searchable: true  @UI: {  headerInfo: {  typeName: 'Booking',  typeNamePlural: 'Bookings',  title: { type: #STANDARD, value: 'booking' }  }  }  define root view entity Z\_I\_BOOKING\_00  as select from ztbooking## as Booking  association [0..1] to I\_Country as \_Country on $projection.country = \_Country.Country  association [0..1] to I\_Currency as \_Currency on $projection.CurrencyCode = \_Currency.Currency    {  @UI.facet: [  {  id: 'Booking',  purpose: #STANDARD,  type: #IDENTIFICATION\_REFERENCE,  label: 'Booking',  position: 10 }  ]  @UI: {  lineItem: [ { position: 10, importance: #HIGH, label: 'Booking ID @DP' } ],  identification:[ { position: 10, label: 'Booking ID @MP' } ]  }  key booking as Booking,  @UI: {  lineItem: [ { position: 20, label: 'Customer', importance: #HIGH } ],  identification:[ { position: 10, label: 'Customer' } ],  selectionField: [{ position: 10 }]  }  @Search.defaultSearchElement: true  customername as CustomerName,  @UI: {  lineItem: [ { position: 30, label: 'No of Passengers', importance: #HIGH } ],  identification:[ { position: 30, label: 'No of Passengers' } ]  }  numberofpassengers as NumberOfPassengers,  @UI: {  identification:[ { position: 40, label: 'Email' } ]  }  emailaddress as EmailAddress,  @EndUserText.label: 'Country Info'  @EndUserText.quickInfo: 'Country Info quick info'    @UI: { lineItem: [{ position: 110 },  { type: #FOR\_ACTION, dataAction: 'set\_country', label: 'Set to Country(KR)' }],  identification:[{ position: 70 }]}  country,  @UI: {  identification:[ { position: 60, label: 'Booked On' } ]  }  dateofbooking as DateOfBooking,  @UI: { identification:[ { position: 70, label: 'Traveling on' } ] }  dateoftravel as DateOfTravel,  @Semantics.amount.currencyCode: 'CurrencyCode'  @UI.dataPoint : { title: 'Gross Cost' }  cost,  @UI: { identification:[ { position: 90, label: 'Currency' } ] }  // @Semantics.currencyCode: true  currencycode as CurrencyCode,  // @UI: { identification:[ { position: 100, label: 'Last Changed At' } ] }  @UI.dataPoint : { title: 'Last Change Date' }  lastchangedat as LastChangedAt,  //publich associations  \_Country,  \_Currency  } |
| 1. 테스트 데이터 생성   - Class생성 및 실행  **Note : ##을 붙여주세요**  Note :왼쪽의 table이름 (“ztbooking##”)을 앞서 생성한 이름으로 변경. | |  |  | | --- | --- | |  |  |   class ZCL\_GENERATE\_BOOKINGS## definition  public  final  create public .  public section.  INTERFACES if\_oo\_adt\_classrun.  protected section.  private section.  ENDCLASS.  CLASS ZCL\_GENERATE\_BOOKINGS## IMPLEMENTATION.  METHOD if\_oo\_adt\_classrun~main.  DATA:it\_bookings TYPE TABLE OF ztbooking##.  \* read current timestamp  GET TIME STAMP FIELD DATA(zv\_tsl).  \* fill internal table (itab)  it\_bookings = VALUE #(  ( booking = '1' customername = 'Buchholm' numberofpassengers = '3' emailaddress = 'tester1@flight.example.com'  country = 'Germany' dateofbooking ='20180213125959' dateoftravel ='20180213125959' cost = '546' currencycode = 'EUR' lastchangedat = zv\_tsl )  ( booking = '2' customername = 'Jeremias' numberofpassengers = '1' emailaddress = 'tester2@flight.example.com'  country = 'USA' dateofbooking ='20180313125959' dateoftravel ='20180313125959' cost = '1373' currencycode = 'USD' lastchangedat = zv\_tsl )  ).  \* Delete the possible entries in the database table - in case it was already filled  DELETE FROM ztbooking##.  \* insert the new table entries  INSERT ztbooking## FROM TABLE @it\_bookings.  \* check the result  SELECT \* FROM ztbooking## INTO TABLE @it\_bookings.  out->write( sy-dbcnt ).  out->write( 'data inserted successfully!').  ENDMETHOD.  ENDCLASS.  **Class 실행.**    **데이터 조회** |
| 1. CDS로 구성된 Data Model에 대한 transaction발생을 위해 Behavior Definition & Implementation   **Note : ##을 붙여주세요**  **Note : ##을 붙여주세요** | Behavior Definition (강사와 함께 진행)   |  |  | | --- | --- | |  | unmanaged implementation in class zbp\_i\_booking\_00 unique;  //strict ( 1 );  define behavior for Z\_I\_BOOKING\_## alias ZBooking##  lock master  authorization master ( instance )  {  create;  update;  delete;  action set\_country result [1] $self;  } |   Behavior Implementation   |  |  | | --- | --- | |  |  |   CLASS lhc\_ZBooking## DEFINITION INHERITING FROM cl\_abap\_behavior\_handler.  PUBLIC SECTION.  TYPES: BEGIN OF ty\_buffer.  INCLUDE TYPE ztbooking## AS data.  TYPES: flag TYPE c LENGTH 1,  END OF ty\_buffer.  TYPES tt\_bookings TYPE SORTED TABLE OF ty\_buffer WITH UNIQUE KEY booking.  CLASS-DATA mt\_buffer TYPE tt\_bookings.  PRIVATE SECTION.  METHODS get\_instance\_authorizations FOR INSTANCE AUTHORIZATION  IMPORTING keys REQUEST requested\_authorizations FOR ZBooking## RESULT result.  METHODS create FOR MODIFY  IMPORTING entities FOR CREATE ZBooking##.  METHODS update FOR MODIFY  IMPORTING entities FOR UPDATE ZBooking##.  METHODS delete FOR MODIFY  IMPORTING entities FOR DELETE ZBooking##.  METHODS read FOR READ  IMPORTING keys FOR READ ZBooking## RESULT result.  METHODS lock FOR LOCK  IMPORTING keys FOR LOCK ZBooking##.  METHODS set\_country FOR MODIFY  IMPORTING keys FOR ACTION ZBooking##~set\_country RESULT result.  ENDCLASS.  CLASS lhc\_ZBooking## IMPLEMENTATION.  METHOD get\_instance\_authorizations.  ENDMETHOD.  METHOD create.  SELECT SINGLE MAX( booking ) FROM ztbooking## INTO @DATA(lv\_max\_booking).  LOOP AT entities INTO DATA(ls\_create).  " next booking number  ADD 1 TO lv\_max\_booking.  ls\_create-%data-booking = lv\_max\_booking.  GET TIME STAMP FIELD DATA(zv\_tsl).  ls\_create-%data-lastchangedat = zv\_tsl.  "Handle field LastChangedAt  " insert as created into buffer  INSERT VALUE #( flag = 'C' data = CORRESPONDING #( ls\_create-%data ) ) INTO TABLE mt\_buffer.  " tell framework about new key if a content id (%cid) is used.:  IF ls\_create-%cid IS NOT INITIAL.  INSERT VALUE #( %cid = ls\_create-%cid booking = ls\_create-booking ) INTO TABLE mapped-zbooking##.  ENDIF.  ENDLOOP.  ENDMETHOD.  METHOD update.  " handle update  LOOP AT entities INTO DATA(ls\_update).  " check for content id (%cid) handling  IF ls\_update-booking IS INITIAL.  ls\_update-booking = mapped-zbooking##[ %cid = ls\_update-%cid\_ref ]-booking.  ENDIF.  " search in buffer  READ TABLE mt\_buffer WITH KEY booking = ls\_update-booking ASSIGNING FIELD-SYMBOL(<ls\_buffer>).  IF sy-subrc <> 0.  " not yet in buffer, read from table  SELECT SINGLE \* FROM ztbooking## WHERE booking = @ls\_update-booking INTO @DATA(ls\_db).  INSERT VALUE #( flag = 'U' data = ls\_db ) INTO TABLE mt\_buffer ASSIGNING <ls\_buffer>.  ENDIF.  IF ls\_update-%control-customername IS NOT INITIAL..  <ls\_buffer>-customername = ls\_update-customername.  ENDIF.  IF ls\_update-%control-cost IS NOT INITIAL..  <ls\_buffer>-cost = ls\_update-cost.  ENDIF.  IF ls\_update-%control-dateoftravel IS NOT INITIAL..  <ls\_buffer>-dateoftravel = ls\_update-dateoftravel .  ENDIF.  IF ls\_update-%control-currencycode IS NOT INITIAL..  <ls\_buffer>-currencycode = ls\_update-currencycode.  ENDIF.  IF ls\_update-%control-country IS NOT INITIAL..  <ls\_buffer>-country = ls\_update-country.  ENDIF.  GET TIME STAMP FIELD DATA(zv\_tsl2).  <ls\_buffer>-lastchangedat = zv\_tsl2. "handling for field LastChangedAt (for ETag)  ENDLOOP.  ENDMETHOD.  METHOD delete.  " handle delete  LOOP AT entities INTO DATA(ls\_delete).  " check for content id (%cid) handling  IF ls\_delete-booking IS INITIAL.  ls\_delete-booking = mapped-zbooking##[ %cid = ls\_delete-%cid\_ref ]-booking.  ENDIF.  READ TABLE mt\_buffer WITH KEY booking = ls\_delete-booking ASSIGNING field-symbol(<ls\_buffer>).  IF sy-subrc = 0.  " already in buffer, check why  IF <ls\_buffer>-flag = 'C'.  "delete after create => just remove from buffer  DELETE TABLE mt\_buffer WITH TABLE KEY booking = ls\_delete-booking.  ELSE.  <ls\_buffer>-flag = 'D'.  ENDIF.  ELSE.  " not yet in buffer.  INSERT VALUE #( flag = 'D' booking = ls\_delete-booking ) INTO TABLE mt\_buffer.  ENDIF.  ENDLOOP.  ENDMETHOD.  METHOD read.  ENDMETHOD.  METHOD lock.  ENDMETHOD.  METHOD set\_country.  "Modify travel instance  MODIFY ENTITIES OF z\_i\_booking\_## IN LOCAL MODE  ENTITY zbooking##  UPDATE FROM VALUE #( FOR key IN keys ( booking = key-booking  country = 'korea' " Accepted  %control-country = if\_abap\_behv=>mk-on ) )  FAILED failed  REPORTED reported.  "Read changed data for action result  READ ENTITIES OF z\_i\_booking\_## IN LOCAL MODE  ENTITY zbooking##  ALL FIELDS WITH  CORRESPONDING #( keys )  RESULT DATA(lt\_travel).  result = VALUE #( FOR travel IN lt\_travel ( booking = travel-booking  %param = travel  ) ).  ENDMETHOD.  ENDCLASS.  CLASS lsc\_Z\_I\_BOOKING\_## DEFINITION INHERITING FROM cl\_abap\_behavior\_saver.  PROTECTED SECTION.  METHODS finalize REDEFINITION.  METHODS check\_before\_save REDEFINITION.  METHODS save REDEFINITION.  METHODS cleanup REDEFINITION.  METHODS cleanup\_finalize REDEFINITION.  ENDCLASS.  CLASS lsc\_Z\_I\_BOOKING\_## IMPLEMENTATION.  METHOD finalize.  ENDMETHOD.  METHOD check\_before\_save.  ENDMETHOD.  METHOD save.  DATA lt\_data TYPE STANDARD TABLE OF ztbooking##.  " find all rows in buffer with flag = created  lt\_data = VALUE #( FOR row IN lhc\_ZBooking##=>mt\_buffer WHERE ( flag = 'C' ) ( row-data ) ).  IF lt\_data IS NOT INITIAL.  INSERT ztbooking## FROM TABLE @lt\_data.  ENDIF.  " find all rows in buffer with flag = updated  lt\_data = VALUE #( FOR row IN lhc\_ZBooking##=>mt\_buffer WHERE ( flag = 'U' ) ( row-data ) ).  IF lt\_data IS NOT INITIAL.  UPDATE ztbooking## FROM TABLE @lt\_data.  ENDIF.  " find all rows in buffer with flag = deleted  lt\_data = VALUE #( FOR row IN lhc\_ZBooking##=>mt\_buffer WHERE ( flag = 'D' ) ( row-data ) ).  IF lt\_data IS NOT INITIAL.  DELETE ztbooking## FROM TABLE @lt\_data.  ENDIF.  ENDMETHOD.  METHOD cleanup.  ENDMETHOD.  METHOD cleanup\_finalize.  ENDMETHOD.  ENDCLASS. |
| 1. Service Definition과 Binding.   **Note : ##을 붙여주세요** | |  |  | | --- | --- | |  |  |  |  |  | | --- | --- | |  |  |   Note: 최신버전에선 화면이 약간 달라졌습니다. “Activate” 버튼을 클릭하세요 |
| 1. Fiori Element를 이용한 결과 확인. |  |

2.3 Applied CDS Associations

이번 장에서 사용된 Associations에 대해 살펴봅니다.

| Explanation | Screenshot |
| --- | --- |
| 왼쪽과 같이 Association관계를 확인할 수 있음. | 아래 Association[0..1] ztbooking 테이블과target view인 ( I\_Country 혹은 I\_Currency) 가 1:1 혹은 1:0의 관계임을 알 수 있다.  추가로, 1:N을 표현하기 위해서는 **Association[1..\*]** 으로 표기하면 됨.  @AccessControl.authorizationCheck: #NOT\_REQUIRED  @EndUserText.label: 'cds for business object booking'  @Search.searchable: true  @UI: {  headerInfo: {  typeName: 'Booking',  typeNamePlural: 'Bookings',  title: { type: #STANDARD, value: 'booking' }  }  }  **define** **root** **view** **entity** Z\_I\_BOOKING\_##  **as** **select** **from** ztbooking## **as** Booking  **association** **[**0**..**1**]** **to** I\_Country **as** \_Country **on** **$projection.**country **=** \_Country**.**Country  **association** **[**0**..**1**]** **to** I\_Currency **as** \_Currency **on** **$projection.**CurrencyCode **=** \_Currency**.**Currency |
|  |  |

2.4 Applied CDS Annotations(Designing the User Interface for a Fiori Elements App)

| Explanation | Screenshot |
| --- | --- |
| header정보를 표현 | @UI: {  headerInfo: {  typeName: 'Booking',  typeNamePlural: 'Bookings',  title: { type: #STANDARD, value: 'booking' }  }  } |
| Facet을 이용해 상세 화면에 대한 정의를 수행 | @UI.facet: [  {  id: 'Booking',  purpose: #STANDARD,  type: #IDENTIFICATION\_REFERENCE,  label: 'Booking',  position: 10 }  ] |
| UI 화면상에서 Display되는 내용 | @UI: {  lineItem: [ { position: 10, importance: #HIGH, label: 'Booking ID @DP' } ],  identification:[ { position: 10, label: 'Booking ID @MP' } ]  }  key booking as Booking, |
| Action 정의 | @UI: { lineItem: [{ position: 110 },  { type: #FOR\_ACTION, dataAction: 'set\_country', label: 'Set to Country' }],  identification:[{ position: 70 }]}  country**,** |
| UI화면상의 Customer name에 대한 Selection field와 검색기능을 추가. | @UI: {  lineItem: [ { position: 20, label: 'Customer', importance: #HIGH } ],  identification:[ { position: 10, label: 'Customer' } ],  selectionField: [{ position: 10 }]  }  @Search.defaultSearchElement: true  customername as CustomerName, |
| @Semantics.amount.currencyCode 활용 | @Semantics.amount.currencyCode: 'CurrencyCode'  @UI.dataPoint : { title: 'Gross Cost' }  cost, |
| @UI.facet : Object Page | @UI.facet: [  {  id: 'Booking',  purpose: #STANDARD,  type: #IDENTIFICATION\_REFERENCE,  label: 'Booking',  position: 10 }  ] |
| @UI.dataPoint 를 통해 상세화면에서Data 강조. | @UI.dataPoint : { title: 'Last Change Date' }  lastchangedat as LastChangedAt, |
| Expose Entity | 상위 CDS에서 추가적으로 컬럼들을 지정하기 위해 association을 expose함.  CDS Path로 구현될 때 사용되며, 뒷부분의 CDS Path부분을 참조하면 됩니다.  //expose associations  \_Country,  \_Currency |
|  |  |

# Chapter 3: Integration with s/4hana : Outbound communication

- Prerequisites은 다음과 같습니다.

1) ABAP Cloud입장에서 아래 단계를 수행합니다.

아래 그림과 같이 Communication Arrangement(“SAP\_COM\_0276”)를 구성한 후, 이를 이용해 HTTP client와 OAuth인증을 이용해 S/4HANA Cloud시스템 혹은 S/4H On-premsise시스템과 interface를 수행합니다.



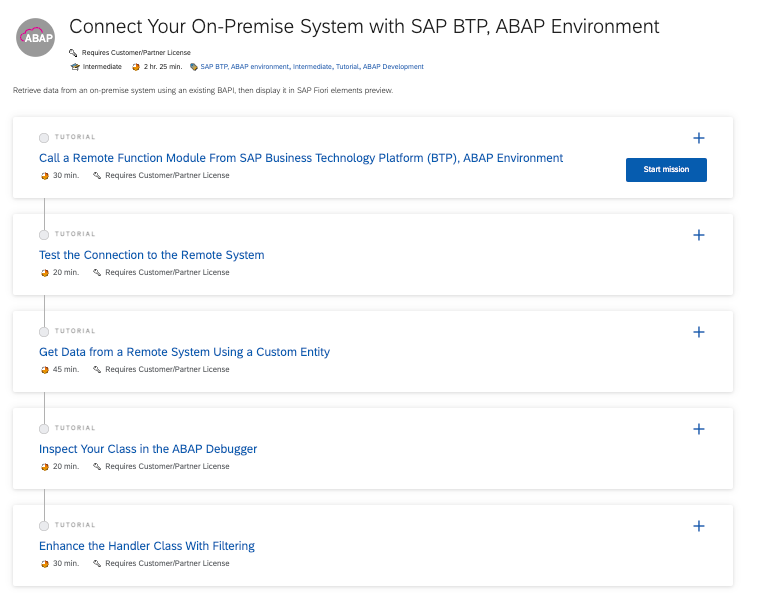
3.1 ABAP Cloud와 S/4H간의 interface – S/4HANA RFC호출 Custom Entity

API Provider : S/4HANA

API Consumer : BTP ABAP

Connect your On-premise System with SAP BTP, ABAP Environment

<https://developers.sap.com/mission.abap-env-connect-onpremise.html>



4.2 CDS Design Principle

| Explanation | Screenshot |
| --- | --- |
| 목적을 명확히. 다목적 보다는 ‘well tailored’ CDS | * S/4내에 들어있는 CDS처럼 Business logic처리를 위함인지? * UI로 부터의 단순한 조회인지 ? * 분석 목적의 리포트인지 ? * Code push down시나리오용인지 ? * A4C의 Business Object용인지 ? * CDS consumer가 기대하는 QoS(Quality of Service)를 명확히.   --아래의 Annotation도 있음.  @ObjectModel.usageType.serviceQuality  @ObjectModel.usgaeType.dataClass  @ObjectModel.usageType.sizeCategory. |
| Aggregation은 가급적 Star-schema모델로 |  |
| 필터 조건 반드시 포함 | Parameter를 이용한 필터 처리.  외부에서 필터 조건 추가시, Projection column list에 컬럼이 포함되어야 함. |

4.3 CDS templates(ABAP CDS Entities)

| Explanation | Screenshot |
| --- | --- |
| Define View  **Note : ##에 본인번호 입력.** | <https://wiki.wdf.sap.corp/wiki/display/CDSDDIC/Test+Examples>  @AbapCatalog.sqlViewName: 'sql\_view\_name'  @AbapCatalog.compiler.compareFilter: true  @AbapCatalog.preserveKey: true  @AccessControl.authorizationCheck: #CHECK  @EndUserText.label: 'cds type test'  define view /DMO/ZCDSTYPE01 as select from data\_source\_name {    }   * Example   @AbapCatalog.sqlViewName: 'ZSVAGENCY##'  @AbapCatalog.compiler.compareFilter: true  @AbapCatalog.preserveKey: true  @AccessControl.authorizationCheck: #CHECK  @EndUserText.label: 'Agency Description'  define view ZCDS\_AGENCY## as select from /dmo/agency {  key agency\_id,  @Semantics.text : true  name,  street,  postal\_code,  city  } |
| Define View with Join  **Note : ##에 본인번호 입력.** | @AbapCatalog.sqlViewName: '${sql\_view\_name}'  @AbapCatalog.compiler.compareFilter: true  @AbapCatalog.preserveKey: true  @AccessControl.authorizationCheck: #CHECK  @EndUserText.label: '${ddl\_source\_description}'  define view ${ddl\_source\_name\_editable} as select from ${data\_source\_name}  left outer join ${joined\_data\_source\_name}  on ${data\_source\_name}.${element\_name} = ${joined\_data\_source\_name}.${joined\_element\_name} {  ${cursor}  }   * Example   @AbapCatalog.sqlViewName: 'ZSVAGYTRV##'  @AbapCatalog.compiler.compareFilter: true  @AbapCatalog.preserveKey: true  @AccessControl.authorizationCheck: #CHECK  @EndUserText.label: 'Agency associated with Travel'  define view ZCDS\_AGYTRV## as select from /dmo/agency as AG  left outer join /dmo/travel as TR  on AG.agency\_id = TR.agency\_id {  key AG.agency\_id as AgencyID,  key TR.travel\_id as TravelID,  AG.name as AgencyName,  AG.phone\_number as PhoneNumber,  TR.customer\_id as CustomerID,    @Semantics.currencyCode: true  TR.currency\_code,  @Semantics.amount.currencyCode: 'currency\_code'  TR.total\_price  } |
| Define View with Assoication  **Note : ##에 본인번호 입력.** | @AbapCatalog.sqlViewName: '${sql\_view\_name}'  @AbapCatalog.compiler.compareFilter: true  @AbapCatalog.preserveKey: true  @AccessControl.authorizationCheck: #CHECK  @EndUserText.label: '${ddl\_source\_description}'  define view ${ddl\_source\_name\_editable} as select from ${data\_source\_name}  association [${1}] to ${target\_data\_source\_name} as ${\_association\_name}  on ${data\_source\_name}.${element\_name} = ${\_association\_name}.${target\_element\_name} {  ${cursor}  ${\_association\_name} // Make association public  }  @AbapCatalog.sqlViewName: 'ZSVAGYTRVA##'  @AbapCatalog.compiler.compareFilter: true  @AbapCatalog.preserveKey: true  @AccessControl.authorizationCheck: #CHECK  @EndUserText.label: 'Extension view test'  define view ZCDS\_AGYTRVA## as select from /dmo/agency as AG  association [0..1] to /dmo/travel as TR  on AG.agency\_id = TR.agency\_id {  key AG.agency\_id as AgencyID,  key TR.travel\_id as TravelID,  AG.name as AgencyName,  AG.phone\_number as PhoneNumber,  TR.customer\_id as CustomerID,    @Semantics.currencyCode: true  TR.currency\_code,  @Semantics.amount.currencyCode: 'currency\_code'  TR.total\_price,    TR //association view  } |
| Define View with To-Parent Association | @AbapCatalog.sqlViewName: '${sql\_view\_name}'  @AbapCatalog.compiler.compareFilter: true  @AbapCatalog.preserveKey: true  @AccessControl.authorizationCheck: #CHECK  @EndUserText.label: '${ddl\_source\_description}'  define view ${ddl\_source\_name\_editable} as select from ${data\_source\_name}  association to parent ${target\_data\_source\_name} as ${\_association\_name}  on ${data\_source\_name}.${element\_name} = ${\_association\_name}.${target\_element\_name} {  ${cursor}  ${\_association\_name} // Make association public  } |
| Define View with Parameters  **Note : ##에 본인번호 입력.** | @AbapCatalog.sqlViewName: '${sql\_view\_name}'  @AbapCatalog.compiler.compareFilter: true  @AbapCatalog.preserveKey: true  @AccessControl.authorizationCheck: #CHECK  @EndUserText.label: '${ddl\_source\_description}'  define view ${ddl\_source\_name\_editable}  with parameters ${parameter\_name} : ${parameter\_type}  as select from ${data\_source\_name} {  ${cursor}  }  @AbapCatalog.sqlViewName: ‘ZSVAGYTRVP##‘  @AbapCatalog.compiler.compareFilter: true  @AbapCatalog.preserveKey: true  @AccessControl.authorizationCheck: #CHECK  @EndUserText.label: 'Agency associated /w Travel /w Params'  define view ZCDS\_AGYTRVP  with parameters p\_agency\_name : /dmo/agency\_name  as select from /DMO/ZVD\_AGYTRV as AG  {  key AG.AgencyID as AgencyID,  key AG.TravelID as TravelID,  AG.AgencyName as AgencyName,  AG.PhoneNumber as PhoneNumber,  AG.CustomerID as CustomerID,  @Semantics.currencyCode: true  AG.currency\_code,  @Semantics.amount.currencyCode: 'currency\_code'  AG.total\_price  }  where AG.AgencyName = $parameters.p\_agency\_name |
| Extend View(X) | @AbapCatalog.sqlViewAppendName: '${sql\_view\_append\_name}'  @EndUserText.label: '${ddl\_source\_description}'  extend view ${view\_name} with ${ddl\_source\_name\_editable} {  ${data\_source\_name}.${element\_name}  }   * Not Allowed. |
| Define Table Function with Parameters  **Note : ##에 본인번호 입력.** | @EndUserText.label: '${ddl\_source\_description}'  define table function ${ddl\_source\_name\_editable}  with parameters ${parameter\_name} : ${parameter\_type}  returns {  ${client\_element\_name} : abap.clnt;  ${element\_name} : ${element\_type};  ${cursor}  }  implemented by method ${class\_name}=>${method\_name};   * AMDP Class 정의 및 구현.   class ZAMDPTEST## definition  public  final  create public .  public section.  Interfaces IF\_AMDP\_MARKER\_HDB.  types:  begin of ty\_agency,  client\_id TYPE mandt,  agid TYPE /dmo/agency\_id,  agname TYPE /dmo/agency\_name,  end of ty\_agency.  types : tt\_agency type standard table of ty\_agency.  **class-methods select\_data**  **for table function /DMO/ZVD\_AGTF.**  protected section.  private section.  ENDCLASS.  CLASS ZAMDPTEST## IMPLEMENTATION.  method select\_data by database function for hdb language sqlscript  options read-only  using /dmo/agency.  return  select '100' as client\_id, agency\_id as agid, name as agname from "/DMO/AGENCY";  endmethod.  ENDCLASS.  @EndUserText.label: 'table function cds view'  define table function ZCDS\_AGTF##  returns {  client\_id : mandt;  agid : /dmo/agency\_id;  agname : /dmo/agency\_name;    }  **implemented by method ZAMDPTEST##=>select\_data;**  @AbapCatalog.sqlViewName: 'ZSVAGTFV##'  @AbapCatalog.compiler.compareFilter: true  @AbapCatalog.preserveKey: true  @AccessControl.authorizationCheck: #CHECK  @EndUserText.label: 'cds view for table function'  define view ZCDS\_AGTFV## as select from /DMO/ZVD\_AGTF {  key agid,  agname  } |
| Define Abstract Entity with Parameters | @EndUserText.label: '${ddl\_source\_description}'  define abstract entity ${ddl\_source\_name\_editable}  with parameters ${parameter\_name} : ${parameter\_type} {  ${element\_name} : ${element\_type};  ${cursor}  }  @EndUserText.label: '가격 재결정 Action Input'  **define** **abstract** **entity** Z\_AE\_REPRICE\_INPUT  **{**  @EndUserText.label: '가격결정일'  @EndUserText.quickInfo: '가격결정일'  RepriceDate **:** *abap***.***dats***;**  **}** |
| Define Parent Child Hierarchy | define hierarchy ${ddl\_source\_name\_editable}  as parent child hierarchy (  source ${data\_source\_name}  child to parent association ${\_association\_name}  start where ${element\_name} = ${value}  siblings order by ${order\_by\_element\_name}  )  {  ${element\_name}  ${cursor}  }  **Define Parent Child Hierarchy – This is for hierarchies, which has nothing to do with**  **RAP composition hierarchies** |
| Define Custom Entity with Parameters  🡺 앞장에서 Hands-on 수행. | @EndUserText.label: '${ddl\_source\_description}'  @QueryImplementedBy: '${class\_name\_editable}'  define custom entity ${ddl\_source\_name\_editable}  with parameters ${parameter\_name} : ${parameter\_type} {  key ${key\_element\_name} : ${key\_element\_type};  ${element\_name} : ${element\_type};  ${cursor}  }  @ObjectModel.query.implementedBy: 'ABAP:ZCLS\_PRODUCT\_VIA\_RFC01'  **define** **root** **custom** **entity** zce\_product\_01  **{**  @UI.facet : [  {  id : 'Product',  purpose : #STANDARD,  type : #IDENTIFICATION\_REFERENCE,  label : 'Product',  position : 10 }  ]  // DDL source code for custom entity for BAPI\_EPM\_PRODUCT\_HEADER  @UI : {  lineItem : [{position: 10, importance: #HIGH}],  identification: [{position: 10}],  selectionField: [{position: 10}]  }  **key** ProductId **:** *abap***.***char***(** 10 **);**  TypeCode **:** *abap***.***char***(** 2 **);**  @UI : {  ….  CDS custom entities allow developers to implement their own data provision using ABAP. Data retrieval is implemented via an ABAP class and performed at runtime. On CDS level, only an interface is defined, while the implementation of the logic is encapsulated in the application server artifact. Custom entities are used if data access cannot be done via means of [SQL](javascript:call_link('abensql_glosry.htm')) and the available CDS feature set and other CDS entity types. For example, custom entities can be used when   * data is stored in non-relational storage, for example in a [BLOB](javascript:call_link('abenblob_glosry.htm')). * data is stored in SAP liveCache and access is done via ABAP APIs. * data cannot be computed by means of CDS due to additional application-server-side logic. * a data model can be defined via CDS views, but the [DCL](javascript:call_link('abendcl_glosry.htm')) feature set is not sufficient to define the necessary authorizations. |

4.4 CDS Associations & Path

| Explanation | Screenshot |
| --- | --- |
| Association은 table혹은 CDS entity간의 relation을 표현하며, left outer join을 일반적으로 나타냄.  **Note : ##에 본인번호 입력.** | Association관계에 있는 target view의 cardinality를 나타낸다.  아래의 예제에서 /dmo/travel은 min..max(1..N)를 갖는다.  결국, /dmo/agency와 /dmo/travel간의 관계가 1:N의 관계임을 알 수 있다.  Optimizer에게 CDS뷰간의 Join cardinality를 예상케 해 주며, 최적의 Plan세울 수 있도록 도와준다. 아울러 개발자에게 뷰간의 관계를 알려주는 역할 도 수행한다.  @AbapCatalog.sqlViewName: '/DMO/VZVDAGT0'  @AbapCatalog.compiler.compareFilter: true  @AbapCatalog.preserveKey: true  @AccessControl.authorizationCheck: #CHECK  @EndUserText.label: 'Extension view test'  define view /DMO/AGYTRV0 as select from /dmo/agency as AG  **association [1..\*] to /dmo/travel as TR**  on AG.agency\_id = TR.agency\_id {  key AG.agency\_id as AgencyID,  key TR.travel\_id as TravelID,  AG.name as AgencyName,  AG.phone\_number as PhoneNumber,  TR.customer\_id as CustomerID,    @Semantics.currencyCode: true  TR.currency\_code,  @Semantics.amount.currencyCode: 'currency\_code'  TR.total\_price,    TR //association view  } |
|  |  |

4.5 CDS path

| Explanation | Screenshot |
| --- | --- |
| Nested CDS views에 정의되지 않은 컬럼들도 Path와 expose기능을 이용해 상위레벨에서 정의해서 상용할 수 있다. 즉 하위레벨의 CDS view를 수정하지 않고도 상위레벨에서 컬럼을 추가할 수 있다. | Nested CDS뷰를 확인 할 수 있다.  @AbapCatalog.sqlViewName: '/DMO/VZVDAGT0'  @AbapCatalog.compiler.compareFilter: true  @AbapCatalog.preserveKey: true  @AccessControl.authorizationCheck: #CHECK  @EndUserText.label: 'Extension view test'  define view /DMO/AGYTRV0 as select from /dmo/agency as AG  association [1..\*] to /dmo/travel as TR  on AG.agency\_id = TR.agency\_id {  key AG.agency\_id as AgencyID,  key TR.travel\_id as TravelID,  AG.name as AgencyName,  AG.phone\_number as PhoneNumber,  TR.customer\_id as CustomerID,    @Semantics.currencyCode: true  TR.currency\_code,  @Semantics.amount.currencyCode: 'currency\_code'  TR.total\_price,    **TR //exposed association view**  }  아래 „\_TRV.TR.begin\_date“ 처럼 association 관계로 맺어진 CDS view에 대해 Path로 표현할 수 있다.  @AbapCatalog.sqlViewName: '/DMO/VAGYTRV0P'  @AbapCatalog.compiler.compareFilter: true  @AbapCatalog.preserveKey: true  @AccessControl.authorizationCheck: #CHECK  @EndUserText.label: 'cds path'  define view /DMO/AGYTRV0P as select from /DMO/AGYTRV0 as \_TRV {  key \_TRV.AgencyID,  key \_TRV.TravelID,//\_TRV  **\_TRV.TR.begin\_date,**  **\_TRV.TR.end\_date**  } |
|  |  |

4.6 CDS Annotations

4.6.1 CDS Annotations – AccessControl Annotations

CDS entity 권한(authorization check)를 정의합니다.

@Scope:[#VIEW, #TABLE\_FUNCTION]

AccessControl.authorizationCheck : String(20) enum { NOT\_REQUIRED; NOT\_ALLOWED; CHECK; PRIVILEGED\_ONLY; } default #CHECK;

| Explanation | Screenshot |
| --- | --- |
| AccessControl Annotations | AccessControl.authorizatinoCheck :#CHECK 으로 되어 있으면, 아래와 같이 Access Control하에서 권한이 적용되어 결과들이 평가된다.  @AbapCatalog.sqlViewName: '/DMO/VZVDAGT0'  @AbapCatalog.compiler.compareFilter: true  @AbapCatalog.preserveKey: true  **@AccessControl.authorizationCheck: #CHECK**  @EndUserText.label: 'Extension view test'  define view /DMO/AGYTRV0 as select from /dmo/agency as AG  association [1..\*] to /dmo/travel as TR  on AG.agency\_id = TR.agency\_id {  key AG.agency\_id as AgencyID,  key TR.travel\_id as TravelID,  AG.name as AgencyName,  AG.phone\_number as PhoneNumber,  TR.customer\_id as CustomerID,    @Semantics.currencyCode: true  TR.currency\_code,  @Semantics.amount.currencyCode: 'currency\_code'  TR.total\_price,    TR //association view  }  별도로 Role을 생성해서 Access Control을 수행하는게 맞으나, 여기에서는 강제로 Column Condition을 설정해서 테스트 하였다.  @EndUserText.label: 'Access Control test'  @MappingRole: true  define role /dmo/zcvd\_Ac01 {  grant  select  on  /DMO/AGYTRV0  where  **AgencyName = 'Sunshne Travel00';**  // -- Example WHERE condition  // -- Two-field mapping to PFCG authorization with filter on read authorization  // ( SalesOrderID, OrgID ) = aspect pfcg\_auth( S\_ACM\_DEMO, SACMTSOID, SACMORGUID, ACTVT = '03' )  // and  // -- Equals-or-initial operator  // ( CustomerCountry ) ?= aspect pfcg\_auth( S\_ACM\_DEMO, SACMCNTRY )  // or  // -- Reference to the logged on users name  // CreatedBy = aspect User  // or  // inherit OtherRole for grant select on Entity  // -- Literal condition  // isPublic = 'X';    } |
|  |  |
|  |  |

4.6.2 CDS Annotations – Consumption Annotation

Domain-specific frameworks contents들을 consume하는 annotation.

@Scope:[#ELEMENT, #PARAMETER]

Annotation Consumption

valueHelpDefinition: array of

{

qualifier: String(120);

entity

{

@Scope:[#VIEW, #ELEMENT, #PARAMETER]

name : String(40);

element : String(40);

};

distinctValues : Boolean default true;

additionalBinding : array of

{

localParameter : ParameterRef;

localElement : ElementRef;

parameter : String(40);

element : String(40);

usage : String(30) enum

{

FILTER;

RESULT;

FILTER\_AND\_RESULT;

};

};

label : String(60);

presentationVariantQualifier : String(120);

};

| Explanation | Screenshot |
| --- | --- |
| Consumption Annotations | @UI: { lineItem: [{ position: 40 }], identification:[{ position: 40 }]}  @Semantics.amount.currencyCode: 'CurrencyCode'  Travel.booking\_fee as BookingFee,  @UI: { lineItem: [{ position: 50 }], identification:[{ position: 50 }]}  @Semantics.amount.currencyCode: 'CurrencyCode'  Travel.total\_price as TotalPrice,  @Consumption.valueHelpDefinition: [{ entity: { name: 'I\_Currency',  element: 'Currency' }}]  @Semantics.currencyCode: true  Travel.currency\_code as CurrencyCode,  아래와 같이 Currency value선택시, Consumption.valueHelpDefinition을 활용해 도움을 받을 수 있다.      The annotation Consumption.valueHelpDefinition is used to define a value help for the annotated element. The value help provider can be a different CDS entity without association. To consume the value help, the value help provider entity must be added to the respective OData service.  You can filter the available value help options by defining an additional binding. In the following example case, only the business partners are displayed that use the same currency code. |

4.6.3 CDS Annotations – ObjectModel Annotations

텍스트 관련된 데이터 모델을 지원합니다.

@Scope:[#ELEMENT]

text

{

element : array of ElementRef;

association : AssociationRef;

control : String(60) enum { NONE; ASSOCIATED\_TEXT\_UI\_HIDDEN; };

}

| Explanation | Screenshot |
| --- | --- |
| ObjectModel Annotations | define root view /DMO/ZVD\_Travel  as select from /dmo/travel as Travel  association [0..1] to I\_Currency as \_Currency on $projection.CurrencyCode = \_Currency.Currency  association [1..1] to /DMO/ZVD\_AGENCY as \_Agency on $projection.agency\_id = \_Agency.agency\_id  …  Agency ID대신 Agency Name으로 대체.  Association 관계에 있는 „\_Agency“ CDS를 참조해서 이름으로 대체 가능.  @UI: { lineItem: [{ position: 30 }], identification:[{ position: 30 }]}  @ObjectModel.text.association: '\_Agency'  Travel.agency\_id,  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Agency CDS view \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  @AbapCatalog.sqlViewName: '/DMO/VDAGENCY'  @AbapCatalog.compiler.compareFilter: true  @AbapCatalog.preserveKey: true  @AccessControl.authorizationCheck: #CHECK  @EndUserText.label: 'Agency Description'  define view /DMO/ZVD\_AGENCY as select from /dmo/agency {  key agency\_id,  @Semantics.text : true  name,  street,  postal\_code,  city  } |
|  | ObjectModel.text.element[ ] 사용시, 해당 언어에 맞게 텍스트가 출력된다.  define view I\_Plant ... {  @ObjectModel.text.element: [’PlantName’]  key Plant,  @Semantics.text: true  PlantName, ...  } |

4.6.4 CDS Annotations – Search Annotations

SAP HANA의 검색기능을 활용하며, fuzziness threshold와 term mapping을 element(column) level에서 정의할 수 있음.

@Scope:[#ENTITY]

Annotation Search

{

searchable : Boolean default true;

};

@Scope:[#ELEMENT]

Annotation Search

{

defaultSearchElement : Boolean default true;

ranking : String(6) enum { HIGH = 'high'; MEDIUM = 'medium'; LOW = 'low'; } default #MEDIUM;

fuzzinessThreshold : Decimal(3,2);

termMappingDictionary : String(128);

termMappingListId : array of String(32);

};

| Explanation | Screenshot |
| --- | --- |
| Search.searchable  Search.defaultSearchElement  Search.fuzzinessThreshold | @Search.searchable: true  …  @Search.defaultSearchElement: true  @Search.fuzzinessThreshold:0.7  @UI: { lineItem: [{ position: 60 }], identification:[{ position: 60 }]}  Travel.description as Memo,  Description element(column)에 대해 Search 할 수 있는 annotation.  추가적으로 fuzzy search 기능까지 추가함. |
| Search.termMappingDictionary  Search.termMappingListID | Specifies the table that holds the term mappings (synonyms) to be considered in the context of a search on this view.Scope: #ElementEvaluation Runtime (Engine): No engine usage right now. Reserved for future usage.  현재 지원되지 않으며, 나중에 지원될 예정. |

4.6.5 CDS Annotations – Semantics Annotations

Data processing과 consumption을 위해 core engine에 의해 사용됨.

@Scope: [#ELEMENT, #PARAMETER]

Annotation Semantics

text : Boolean default true;

language : Boolean default true;

@Scope:[#ELEMENT]

Annotation Semantics

amount

{

currencyCode : ElementRef;

};

quantity

{

unitOfMeasure : ElementRef;

};

currencyCode : Boolean default true;

unitOfMeasure : Boolean default true;

| Explanation | Screenshot |
| --- | --- |
| Semantics.text  Semantics.language | 설정된 언어에 맞게 자동으로 언어가 변경되어 선택된다.  앞선 예제에서 언급되어 추가로 실습할 필요는 없습니다.  @UI: { lineItem: [{ position: 30 }], identification:[{ position: 30 }]}  @ObjectModel.text.association: '\_Agency'  Travel.agency\_id,  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Agency CDS view \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  @AbapCatalog.sqlViewName: '/DMO/VDAGENCY'  @AbapCatalog.compiler.compareFilter: true  @AbapCatalog.preserveKey: true  @AccessControl.authorizationCheck: #CHECK  @EndUserText.label: 'Agency Description'  define view /DMO/ZVD\_AGENCY as select from /dmo/agency {  key agency\_id,  @Semantics.text : true  name,  street,  postal\_code,  city  } |
| Semantics.currencyCode  Semantics.amount.currencyCode  Semantics.unitofMeasure  Semantics.quantity.unitOfMeasure | Currency혹은 Unit 컬럼 타입을 지정할 수 있고, 그에 맞는 Display를 위해 사용되는 annotation.  앞선 예제에서 언급되어 추가로 실습할 필요는 없습니다.  @Semantics.amount.currencyCode: 'CurrencyCode'  Travel.total\_price as TotalPrice,  @Semantics.currencyCode: true  Travel.currency\_code as CurrencyCode, |

4.6.6 CDS Annotations – UI Annotations

UI5기술과는 독립적으로 동작한다고 생각하면 될거 같고, Business Data에 대한 의미적인 해석에 대해서만 간략히 지원한다고 보면 됩니다. 필드네임변경, 필드에 대한 User-defined action(business logic)적용, 상세페이지 조회, 검색, field selection 정도만 지원됩니다.

| Explanation | Screenshot |
| --- | --- |
| UI.facet | @Scope:[#ELEMENT]  facet : array of  {  qualifier : String(120);  id : String(120);  purpose : String(40) enum  {  STANDARD;  HEADER;  QUICK\_VIEW;  QUICK\_CREATE;  FILTER;  } default #STANDARD;  parentId : String(120);  position : DecimalFloat;  exclude : Boolean default true;  isPartOfPreview : Boolean default true;  isSummary : Boolean default true;  isMap : Boolean default true;  importance : String(6) enum  {  HIGH;  MEDIUM;  LOW;  };  @LanguageDependency.maxLength : 40  label : String(60);  type : String(40) enum  {  COLLECTION;  ADDRESS\_REFERENCE;  BADGE\_REFERENCE;  CHART\_REFERENCE;  CONTACT\_REFERENCE;  DATAPOINT\_REFERENCE;  FIELDGROUP\_REFERENCE;  HEADERINFO\_REFERENCE;  IDENTIFICATION\_REFERENCE;  LINEITEM\_REFERENCE;  STATUSINFO\_REFERENCE;  URL\_REFERENCE;  };  targetElement : ElementRef;  targetQualifier : String(120);  url : ElementRef;  };   * Simple Example   UI.facet은 다양한 용도로 활용되며, 아래 예제에서는 Identification\_reference를 통해 상세 화면으로 이동할 때 사용되기도 한다.  @AbapCatalog.sqlViewName: '/DMO/ZVDUIBADGET '  @AbapCatalog.compiler.compareFilter: true  @AbapCatalog.preserveKey: true  @AccessControl.authorizationCheck: #CHECK  @EndUserText.label: 'UI anno Badge Test'  define root view /DMO/ZUIBADGET as select from /dmo/agency {  @UI.facet: [  {  id: 'AgencyID',  purpose: #STANDARD,  type: #IDENTIFICATION\_REFERENCE,  label: 'Agency',  position: 10  }]  @UI: { lineItem:[{ position: 10 }], identification:[{ position: 10 }]}  @UI.masked: true  key agency\_id ,  @UI: { lineItem: [{ position: 20 }], identification:[{ position: 20 }]}    @Semantics.text : true  name ,  @UI: { lineItem: [{ position: 30 }], identification:[{ position: 30 }]}    street ,  postal\_code ,  city  } |
| UI.lineitem | @Scope:[#ELEMENT]  lineItem : array of  {  qualifier : String(120);  position : DecimalFloat;  exclude : Boolean default true;  importance : String(6) enum { HIGH; MEDIUM; LOW; };  type : String(40) enum  {  FOR\_ACTION;  AS\_ADDRESS;  AS\_DATAPOINT;  AS\_CONTACT;  AS\_CHART;  FOR\_INTENT\_BASED\_NAVIGATION;  STANDARD;  WITH\_INTENT\_BASED\_NAVIGATION;  WITH\_NAVIGATION\_PATH;  WITH\_URL;  } default #STANDARD;  @LanguageDependency.maxLength : 40  label : String(60);  iconUrl : String(1024);  criticality : ElementRef;  criticalityRepresentation : String(12) enum  {  WITHOUT\_ICON;  WITH\_ICON;  } default #WITHOUT\_ICON;  dataAction : String(120);  invocationGrouping : String(12) enum { ISOLATED; CHANGE\_SET; } default #ISOLATED;  semanticObjectAction : String(120);  value : ElementRef;  valueQualifier : String(120);  targetElement : ElementRef;  url : ElementRef;  };   * Simple Example   UI Lineitem은 아래와 같이 컬럼의 위치와 Alias 등을 지정할 때 사용되며, 특히 FOR\_ACTION을 통해 C/U/D가 아닌 User-defined function을 수행할 때 사용된다.  이에 대한 자세한 내용은 intermediate level에서 좀더 자세히 언급된다.  < CDS >  @UI: { lineItem: [{ position: 60 }], identification:[{ position: 60 }]}  Travel.description as Memo,  @UI: { lineItem: [{ position: 70 }, { type: #FOR\_ACTION, dataAction: 'set\_status\_booked', label: 'Set to Booked' }], identification:[{ position: 70 }]}  Travel.status as Status |
|
|
|
|
| @UI.Identification | Scope은 @UI.lineitem과 비슷하며, UI.facet과 함께 사용되며, 상세화면에서의 position과 alias그리고 data action을 지정할 수 있다. |
| @UI.selectionField | @Scope:[#ELEMENT]  selectionField : array of  {  qualifier : String(120);  position : DecimalFloat;  exclude : Boolean default true;  element : ElementRef;  };   * Simple Example   특정 컬럼에 대한 ABAP에서의 Selection-list와 같은 기능을 수행할 수 있다.  @UI: {  lineItem: [ { position: 20, label: 'Customer', importance: #HIGH } ],  identification:[ { position: 10, label: 'Customer' } ],  selectionField: [{ position: 10 }]  }  customername as CustomerName, |
| @UI.masked | @Scope:[#ELEMENT]  masked : Boolean default true;   * Simple Example   @AbapCatalog.sqlViewName: '/DMO/ZVDUIBADGET '  @AbapCatalog.compiler.compareFilter: true  @AbapCatalog.preserveKey: true  @AccessControl.authorizationCheck: #CHECK  @EndUserText.label: 'UI anno Badge Test'  define root view /DMO/ZUIBADGET as select from /dmo/agency {  @UI.facet: [  {  id: 'AgencyID',  purpose: #STANDARD,  type: #IDENTIFICATION\_REFERENCE,  label: 'Agency',  position: 10  }]  @UI: { lineItem:[{ position: 10 }], identification:[{ position: 10 }]}  @UI.masked: true  key agency\_id ,  @UI: { lineItem: [{ position: 20 }], identification:[{ position: 20 }]}    @Semantics.text : true  name ,  @UI: { lineItem: [{ position: 30 }], identification:[{ position: 30 }]}    street ,  postal\_code ,  city  } |
| @UI.hidden | @Scope:[#ELEMENT]  hidden : Boolean default true;   * Simple Example   @AbapCatalog.sqlViewName: '/DMO/ZVDUIBADGET '  @AbapCatalog.compiler.compareFilter: true  @AbapCatalog.preserveKey: true  @AccessControl.authorizationCheck: #CHECK  @EndUserText.label: 'UI anno Badge Test'  define root view /DMO/ZUIBADGET as select from /dmo/agency {  @UI.facet: [  {  id: 'AgencyID',  purpose: #STANDARD,  type: #IDENTIFICATION\_REFERENCE,  label: 'Agency',  position: 10  }]  @UI: { lineItem:[{ position: 10 }]}  @UI.hidden: true  key agency\_id ,  @UI: { lineItem: [{ position: 20 }], identification:[{ position: 20 }]}    @Semantics.text : true  name ,  @UI: { lineItem: [{ position: 30 }], identification:[{ position: 30 }]}    street ,  postal\_code ,  city  }  아래와 같이 선택필드에서 hidden처리됨. |
| @UI.dataPoint | @Scope:[#ELEMENT]  dataPoint  {  @LanguageDependency.maxLength : 40  title : String(60);  @LanguageDependency.maxLength : 80  description : String(120);  @LanguageDependency.maxLength : 193  longDescription : String(250);  targetValue : DecimalFloat;  targetValueElement : ElementRef;  forecastValue : ElementRef;  minimumValue : DecimalFloat;  maximumValue : DecimalFloat;  visualization : String(12) enum  {  NUMBER;  BULLET\_CHART;  DONUT;  PROGRESS;  RATING;  };  valueFormat  {  scaleFactor : DecimalFloat;  numberOfFractionalDigits : Integer;  };  referencePeriod  {  @LanguageDependency.maxLength : 80  description : String(120);  start : ElementRef;  end : ElementRef;  };  criticality : ElementRef;  criticalityRepresentation : String(12) enum  {  WITHOUT\_ICON;  WITH\_ICON;  } default #WITHOUT\_ICON;  criticalityCalculation  {  improvementDirection : String(8) enum  {  MINIMIZE;  TARGET;  MAXIMIZE;  };  acceptanceRangeLowValue : DecimalFloat;  acceptanceRangeHighValue : DecimalFloat;  toleranceRangeLowValue : DecimalFloat;  toleranceRangeLowValueElement : ElementRef;  toleranceRangeHighValue : DecimalFloat;  toleranceRangeHighValueElement : ElementRef;  deviationRangeLowValue : DecimalFloat;  deviationRangeLowValueElement : ElementRef;  deviationRangeHighValue : DecimalFloat;  deviationRangeHighValueElement : ElementRef;  constantThresholds: array of  {  aggregationLevel: array of ElementRef;  acceptanceRangeLowValue: DecimalFloat;  acceptanceRangeHighValue: DecimalFloat;  toleranceRangeLowValue: DecimalFloat;  toleranceRangeHighValue: DecimalFloat;  deviationRangeLowValue: DecimalFloat;  deviationRangeHighValue: DecimalFloat;  };  };  trend : ElementRef;  trendCalculation  {  referenceValue : ElementRef;  isRelativeDifference : Boolean default true;  upDifference : DecimalFloat;  upDifferenceElement : ElementRef;  strongUpDifference : DecimalFloat;  strongUpDifferenceElement : ElementRef;  downDifference : DecimalFloat;  downDifferenceElement : ElementRef;  strongDownDifference : DecimalFloat;  strongDownDifferenceElement : ElementRef;  };  responsible : ElementRef;  responsibleName : String(120);  };  @Semantics.amount.currencyCode: 'CurrencyCode'  @UI.dataPoint : { title: 'Gross Cost' }  cost,  @UI.dataPoint : { title: 'Last Change Date' }  lastchangedat as LastChangedAt,  아래와 같이 detail page의 header에 원하는 컬럼을 강조할 수 있습니다. |
|  |  |

4.7 CDS SQL functions

| Explanation | Screenshot |
| --- | --- |
| Example :  substring  case when…else end. currency\_conversion() | @AbapCatalog.sqlViewName: '/DMO/ZVDSQLF01'  @AbapCatalog.compiler.compareFilter: true  @AbapCatalog.preserveKey: true  @AccessControl.authorizationCheck: #CHECK  @EndUserText.label: 'cds sql function test'  define view /DMO/ZVD\_SQL\_FUNCT  with parameters  to\_currency :abap.cuky( 5 ),  exc\_date :abap.dats  as select from /dmo/travel  {  substring(description,1,20) as desc\_substr,  case when status = 'F' then 'AAA' else 'BBB' end as status\_desc,  total\_price,  --currency\_conversion\_test  currency\_conversion( amount => total\_price,  source\_currency => currency\_code,  round => 'X',  target\_currency => :to\_currency,  exchange\_rate\_date => :exc\_date,  error\_handling => 'SET\_TO\_NULL' ) as amount  } |
| 결과 확인 |  |
| 자세한 내용은 Manual 참조 | <https://help.sap.com/doc/abapdocu_751_index_htm/7.51/en-US/abenddic_builtin_functions.htm> |

# Reference link

SAP Cloud Platform - ABAP Development User Guide

<https://help.sap.com/viewer/5371047f1273405bb46725a417f95433/Cloud/en-US/e9bb90f5609d438d889a0b3cbfc7d52c.html?q=SAP%20Cloud%20platform%20-%20ABAP%20Development>

SAP Cloud Platform - ABAP RESTful Programming Model

<https://help.sap.com/viewer/c0d02c4330c34b3abca88bdd57eaccfc/Cloud/en-US/3b77569ca8ee4226bdab4fcebd6f6ea6.html>

SAP Cloud Platform, ABAP Environment: Setup

<https://help.sap.com/doc/461d980dcf484589977271a40050c899/Cloud/en-US/loio5678876d530448e4a85addbedc55db6b.pdf>

SAP Cloud Platform ABAP Environment: Integration with SAP S/4HANA Cloud

<https://help.sap.com/viewer/abap_s4/dbb7d4dcca15409e96f398d979f7a81e.html>

Developer hands-on link

<https://developers.sap.com/tutorials/abap-environment-business-partner.html>

Vital links:

<https://wiki.wdf.sap.corp/wiki/display/fioritech/Alignment+Fiori+Tech+and+RAP>

ABAP Programming Model for SAP Fiori

<https://help.sap.com/viewer/cc0c305d2fab47bd808adcad3ca7ee9d/7.5.9/en-US/38baf2fc3a8e4ed887b29de738296fa9.html>

CDS Tutorial

<https://wiki.wdf.sap.corp/wiki/display/teamsti2/Tutorials>

**Project Link**

GitHub Link for Demo Flight

<https://github.com/SAP/abap-platform-refscen-flight>

Debugging & Tracing

•How to access runtime errors or other errors in ADT using the feed reader is described here:

<https://help.sap.com/doc/saphelp_nw75/7.5.5/en-US/4e/c37b016e391014adc9fffe4e204223/frameset.htm>

•How to change the user that is set in a breakpoint in ABAP in Eclipse is described in this blog.

<https://blogs.sap.com/2018/01/25/abap-in-eclipse-external-debugger-setting/>