



## Teoría

### **ABAP Cloud Developer Extensibility – Campos**

SAP S/4HANA Cloud – Modelo de extensibilidad Clean Core





## Contenido

<b>1. ABAP Cloud Developer Extensibility – Campos</b>	<b>3</b>
<b>1.1. Requerimiento – Extensión RAP con Nuevos Campos</b>	<b>3</b>
<b>1.2. Extensión Persistencia Activa</b>	<b>8</b>
<b>1.3. Extensión Query – Persistencia Draft</b>	<b>15</b>
<b>1.4. Extensión entidad CDS Root</b>	<b>21</b>
<b>1.5. Extensión entidad CDS Projection</b>	<b>26</b>
<b>1.6. Extensión Metadata Extensions</b>	<b>29</b>



# 1. ABAP Cloud Developer Extensibility – Campos

## 1.1. Requerimiento – Extensión RAP con Nuevos Campos

El requerimiento empresarial en cuestión solicita la ampliación de una aplicación estándar RAP (Rapid Application Programming) mediante la incorporación de nuevos campos. Este escenario se divide en dos posibilidades principales:

- Los campos ya existen en la capa de persistencia pero no están contemplados en el modelado CDS (Core Data Services).
- Los campos no existen en la capa de persistencia y es necesario ampliarla.

Antes de cualquier proyecto de extensión, es crucial realizar un análisis para determinar todos los objetos afectados por la ampliación. Esto asegura que las modificaciones necesarias se lleven a cabo correctamente. Es importante observar si los campos solicitados ya están presentes en la capa de persistencia.

La aplicación a analizar es un listado con pantalla de selección y múltiples columnas, permitiendo la navegación a una pantalla de detalles.

Standard ▾

Editing Status: Customer Text: Agency Name: City: Country/Region Key: Go Adapt Filters (1)

Search: Q

All ▾

Agencies (50)

<input type="checkbox"/> Customer...	Agency Name	Slogan	Rating	Address	Contact Data
<input type="checkbox"/> 70002	modified agency name	Cool and smooth! Fly with us! We are the leader in business!	★★★★★	Berliner Allee 11 40890 Düsseldorf Germany (DE)	+49 2102 69555 info@flyhigh.sap http://www.flyhigh.sap
<input type="checkbox"/> 70003	Happy Hopping	Happy Hopping is the leader in business!	★★★★★	Cablnstr. 36 13467 Berlin Germany (DE)	+49 30 8853 0 info@haphop.sap http://www.haphop.sap
<input type="checkbox"/> 70004	Pink Panther	We are the best!	★★★★★	Auf der Schanz 54 65936 Frankfurt Germany (DE)	+49 69 467653 0 info@pinkpanther.sap http://www.pinkpanther.sap
<input type="checkbox"/> 70005	Your Choice	Better with us!	★★★★★	Gustav-Jung-Str. 425 90455 Nuernberg Germany (DE)	+49 9256 4548 0 info@yc.sap http://www.yc.sap
<input type="checkbox"/> 70006	Bella Italia	Cool and save!	★★★★★	Via Marconi 123 00139 Roma Italy (IT)	+39 6 893546721 info@tours.it/Adventure/ http://www.tours.it/Adventure/

Esta funcionalidad incluye cabecera y campos que pueden ser modificados, eliminados o actualizados a través de la aplicación. Los nuevos campos que se incorporen deben soportar todas estas operaciones.



Package: ZABAP\_CL

General Data

Application Component:  Browse...

Responsible:

Default ABAP Language Version: **ABAP for Cloud Development**

Package Properties

Superpackage:  Browse...

Package Type:

☐ Adding further objects not possible

☐ Encapsulated

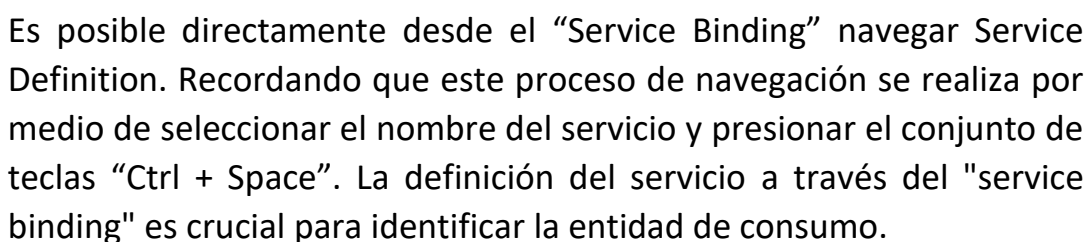
Transport Properties

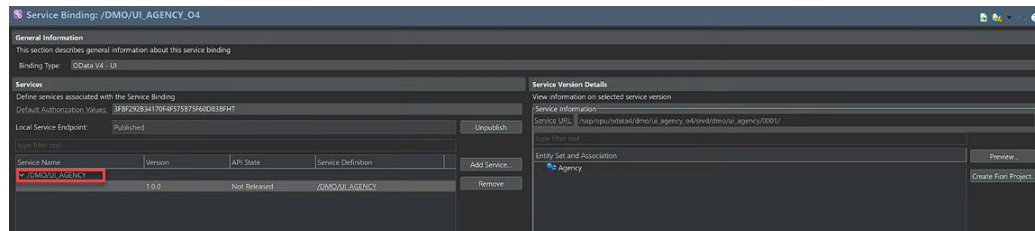
Transport Layer:  Browse...

Software Component:  Browse...

☒ Record objects changes in transport requests

Se analizará el Service Binding para descubrir la definición del servicio.





Donde se expone la entidad de consumo.

```
1 • @EndUserText.label: 'Agency'
2 @AbapCatalog.extensibility.extensible: true
3 @ObjectModel.leadingEntity.name: '/DMO/C_AgencyTP'
4 define service /DMO/UI_AGENCY
5 {
6   expose /DMO/C_AgencyTP as Agency;
7 }
8
```

Además, se analizan las fuentes de datos y las asociaciones necesarias para conectar con la capa de persistencia.

```
@Search.searchable: true
@Metadata.allowExtensions: true
@ObjectModel.semanticKey: ['AgencyID']

@AbapCatalog.extensibility: {
  extensible: true,
  dataSources: ['Agency'],
  elementSuffix: 'ZAG',
  quota: {
    maximumFields: 500,
    maximumBytes: 50000
  }, allowNewCompositions: true
}

define root view entity /DMO/C_AgencyTP
provider contract transactional_query
as projection on /DMO/R_AgencyTP as Agency
```



En la entidad de consumo, la entidad raíz y otras entidades asociadas se debe validar que tienen la anotación: **@AbapCatalog.extensibility**: Esta anotación define las opciones de extensibilidad para una vista CDS. Por medio de las siguientes propiedades:

- **extensible: true**: Indica que la vista puede ser extendida.
- **elementSuffix: 'Z...'**: Especifica un sufijo para los nuevos elementos añadidos a la vista.
- **allowNewDatasources: false**: Indica que no se pueden añadir nuevas fuentes de datos a la vista.
- **dataSources: ['AliasSourceData']**: Lista las fuentes de datos permitidas, en este caso, 'Agency'.
- **quota**: Define los límites de la vista.
  - **maximumFields**: En este caso estableciendo un máximo de 500 campos.
  - **maximumBytes**: En este caso estableciendo un máximo de 50000 bytes.

```
define root view entity /DMO/C_AgencyTP
provider contract transactional_query
as projection on /DMO/R_AgencyTP as Agency
```

```
•@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Agency'
@Search.searchable: true

@AbapCatalog.extensibility: {
  extensible: true,
  elementSuffix: 'ZAG',
  allowNewDatasources: false,
  dataSources: ['_Extension'],
  quota: {
    maximumFields: 500,
    maximumBytes: 50000
  },
  allowNewCompositions: true
}

define root view entity /DMO/R_AgencyTP
as select from /DMO/I_Agency as Agency
association [0..1] to I_Country as _Country on $projection.CountryCode = _Country.Country
association [1] to /DMO/E_Agency as [Extension on $projection.AgencyID = _Extension.AgencyId
```

```
define root view entity /DMO/R_AgencyTP
as select from /DMO/I_Agency as Agency
association [0..1] to I_Country as _Country on $projection.CountryCode = _Country.Country
association [1] to /DMO/E_Agency as [Extension on $projection.AgencyID = _Extension.AgencyId
```



```
@AbapCatalog.viewEnhancementCategory: [#PROJECTION_LIST]
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Agency - Extension Include View'

@AbapCatalog.extensibility: {
  extensible: true,
  elementSuffix: 'ZAG',
  allowNewDatasources: false,
  dataSources: ['Agency'],
  quota: {
    maximumFields: 500,
    maximumBytes: 50000
  }
}

define view entity /DMO/E_Agency
  as select from /dmo/agency as Agency
```

En la capa de persistencia en la base de datos asociada, se pueden realizar extensiones de diferentes tipos.

```
@EndUserText.label : 'Flight Reference Scenario: Managing Agencies'
@AbapCatalog.enhancement.category : #EXTENSIBLE_ANY
@AbapCatalog.tableCategory : #TRANSPARENT
@AbapCatalog.deliveryClass : #A
@AbapCatalog.dataMaintenance : #RESTRICTED
define table /dmo/agency {

  key client          : abap.clnt not null;
  key agency_id       : /dmo/agency_id not null;
  name                : /dmo/agency_name;
  street              : /dmo/street;
  postal_code         : /dmo/postal_code;
  city                : /dmo/city;
  country_code        : land1;
  phone_number        : /dmo/phone_number;
  email_address       : /dmo/email_address;
  web_address         : /dmo/web_address;
  attachment          : /dmo/attachment;
  mime_type           : /dmo/mime_type;
  filename            : /dmo/filename;
  local_created_by    : abp_creation_user;
  local_created_at    : abp_creation_tstmpl;
```

Como carácter, numérico, o cualquier otro tipo permisible.

```
@AbapCatalog.enhancement.category : #EXTENSIBLE ANY
@AbapCatalog.tableCategory : #TRANSPARENT
@AbapCatalog.deliveryClass : #A
@AbapCatalog.dataMaintenance : #RESTRICTED
```

#NOT\_EXTENSIBLE - annotation  
 #EXTENSIBLE\_CHARACTER - annotation  
 #EXTENSIBLE\_CHARACTER\_NUMERIC - annotation  
 #EXTENSIBLE\_ANY - annotation

Por medio de la anotación **@AbapCatalog.enhancement.category** en ABAP define cómo se puede extender una vista CDS (Core Data Services). Con diferentes opciones de configuración:





- **@NOT\_EXTENSIBLE:** Indica que la estructura o tabla de base de datos no puede ser extendida.
- **@EXTENSIBLE\_CHARACTER:** Todos los componentes de la estructura o tabla de base de datos y sus extensiones deben ser de tipo carácter y planos.
- **@EXTENSIBLE\_CHARACTER\_NUMERIC:** Todos los componentes de la estructura o tabla de base de datos y sus extensiones deben ser planos, y pueden ser de tipo carácter o numérico.
- **@EXTENSIBLE\_ANY:** Todos los componentes de la estructura o tabla de base de datos y sus extensiones pueden tener cualquier tipo de dato.

El proceso para implementar la ampliación de una aplicación RAP estándar con nuevos campos es complejo y requiere una planificación meticulosa. Es fundamental realizar un análisis previo exhaustivo para asegurar que todas las capas y entidades involucradas permiten la extensión requerida. Solo después de confirmar esto se puede proceder con el desarrollo y la creación de los artefactos necesarios.

## 1.2. Extensión Persistencia Activa

La extensión de la persistencia activa se refiere al proceso de agregar nuevas funcionalidades a una aplicación estándar RAP (Restful Application Programming), extendiendo sus componentes de base de datos, como tablas y estructuras, para incorporar nuevos campos que satisfacen requisitos empresariales específicos. Este proceso implica modificar la capa de persistencia de la aplicación, asegurando que los nuevos campos sean reflejados adecuadamente en la interfaz de usuario y se integren con la lógica de negocio existente.

Para el caso de una necesidad empresarial donde se requiera agregar un único campo nuevo, aunque el procedimiento puede ser implementado para múltiples campos.





Standard

Editing Status: Customer Text: Agency Name: City: Country/Region Key: Go Adapt Filters (1)

Agencies (50)

Customer...	Agency Name	Slogan	Rating	Address	Contact Data
70002	modified agency name	Cool and smooth! Fly with us! We are the leader in business!	★★★★★	Berliner Allee 11 40880 Duesseldorf Germany (DE)	+49 2102 69555 info@flyhigh.sap http://www.flyhigh.sap
70003 Draft	Happy Hopping	Happy Hopping is the leader in business!	★★★★★	Calvinstr. 36 13467 Berlin Germany (DE)	+49 30-8853-0 info@haphop.sap http://www.haphop.sap
70004	Pink Panther	We are the best!	★★★★★	Auf der Schanz 54 65936 Frankfurt Germany (DE)	+49 69-467653-0 info@pinkpanther.sap http://www.pinkpanther.sap
70005	Your Choice	Better with us!	★★★★★	Gustav-Jung-Str. 425 90455 Nuernberg Germany (DE)	+49 9256-4548-0 info@yc.sap http://www.yc.sap

## Navegación a la Capa de Persistencia:

En primer lugar es necesario identificar cual es la capa de persistencia, es decir, la tabla de base de datos donde se realizará la extensión. En donde desde el editor de código Eclipse, al acceder al service binding es posible navegar directamente al Service Definition.

Service Binding: /DMO/UI\_AGENCY\_O4

General Information

This section describes general information about this service binding

Binding Type: OData V4 - UI

Services

Define services associated with the Service Binding

Default Authorization Values: 3FBF292B34170F4F575B75F60D83BFHT

Local Service Endpoint: Published Unpublish

type filter text

Service Name	Version	API State	Service Definition
▼ /DMO/UI_AGENCY	1.0.0	Not Released	/DMO/UI_AGENCY

Add Service... Remove

La herramienta direccionará la vista al archivo Service binding, al hacer el mismo procedimiento en el servicio expuesto.

```
1 • @EndUserText.label: 'Agency'
2 @AbapCatalog.extensibility.extensible: true
3 @ObjectModel.leadingEntity.name: '/DMO/C_AgencyTP'
4 define service /DMO/UI_AGENCY
5 {
6   expose /DMO/C_AgencyTP as Agency;
7 }
```

Es posible navegar a la entidad de consumo. Y desde allí a la entidad raíz por medio de la proyección.



La entidad de consumo o también conocida como proyección de contrato transaccional Query, define una vista que expone solo las columnas relevantes para una interfaz transaccional específica en el modelo de programación RAP (Restful ABAP Programming) de SAP. Y es responsable de presentar la información en la capa del consumidor, permitiendo que los datos sean accesibles a través de diversos protocolos asociados con el servicio. La proyección de entidades se basa en una entidad raíz (Root Entity), y se configura para facilitar la interacción y el consumo de datos en las aplicaciones.

```
define root view entity /DMO/C_AgencyTP
  provider contract transactional_query
  as projection on /DMO/R_AgencyTP as Agency
```

La entidad raíz es una estructura coherente que integra anotaciones, composiciones, y asociaciones necesarias para la gestión eficiente de los datos y facilita su interacción tanto para los usuarios finales como para los endpoints API. Esta entidad se crea a partir de una entidad de persistencia y actúa como el nodo principal en la jerarquía de datos y es el punto de entrada para todas las relaciones de negocio en una aplicación RAP.

Posteriormente, para navegar a la entidad correspondiente con la fuente de datos, si la entidad raíz posee varias asociaciones, se puede identificar la fuente de datos que se está utilizando mediante la anotación **@AbapCatalog.extensibility.dataSources**.

```
@AbapCatalog.extensibility: {
  extensible: true,
  elementSuffix: 'ZAG',
  allowNewDatasources: false,
  dataSources: ['Extension'],
  quota: {
    maximumFields: 500,
    maximumBytes: 50000
  },
  allowNewCompositions: true
}

define root view entity /DMO/R_AgencyTP
  as select from /DMO/I_Agency as Agency
  association [0..1] to I_Country as _Country on $projection.CountryCode = _Country.Country
  association [1] to /DMO/E_Agency as Extension on $projection.AgencyID = _Extension.AgencyID
```

Esta anotación permite identificar el nombre del alias utilizado por la fuente de datos, facilitando la navegación hacia ella, y permitiendo la posibilidad de la creación de vistas de extensión. Para la ampliación de campos nuevos personalizados.



```
define view entity /DMO/E_Agency
  as select from /dmo/agency as Agency
{
  key agency_id as AgencyId
}
```

### Añadir un Campo Personalizado en la capa de persistencia:

Analizando la tabla de persistencia, se encuentra la anotación **@AbapCatalog.enhancement.category** con la opción **EXTENSIBLE\_ANY**, que permite agregar nuevos campos.

```
@EndUserText.label : 'Flight Reference Scenario: Managing Agencies'
@AbapCatalog.enhancement.category : #EXTENSIBLE_ANY
@AbapCatalog.tableCategory : #TRANSPARENT
@AbapCatalog.deliveryClass : #A
@AbapCatalog.dataMaintenance : #RESTRICTED
define table /dmo/agency {

  key client      : abap.clnt not null;
  key agency_id   : /dmo/agency_id not null;
  name            : /dmo/agency_name;
  street          : /dmo/street;
  postal_code     : /dmo/postal_code;
  city           : /dmo/city;
  country_code    : land1;
  phone_number    : /dmo/phone_number;
  email_address   : /dmo/email_address;
  web_address     : /dmo/web_address;
  attachment      : /dmo/attachment;
  mime_type       : /dmo/mime_type;
  filename        : /dmo/filename;
  local_created_by : abp_creation_user;
  local_created_at : abp_creation_tstmpl;
```

Además, se observa un include donde se pueden realizar las extensiones.

```
local_last_changed_by : abp_locinst_lastchange_user;
local_last_changed_at : abp_locinst_lastchange_tstmpl;
last_changed at       : abp_lastchange_tstmpl;
include /dmo/s_ext_incl_agency;
```

Dentro del include se deben respetar ciertas características, como el sufijo a utilizar y los límites en el número de campos y la memoria, al momento de incluir nuevos campos.

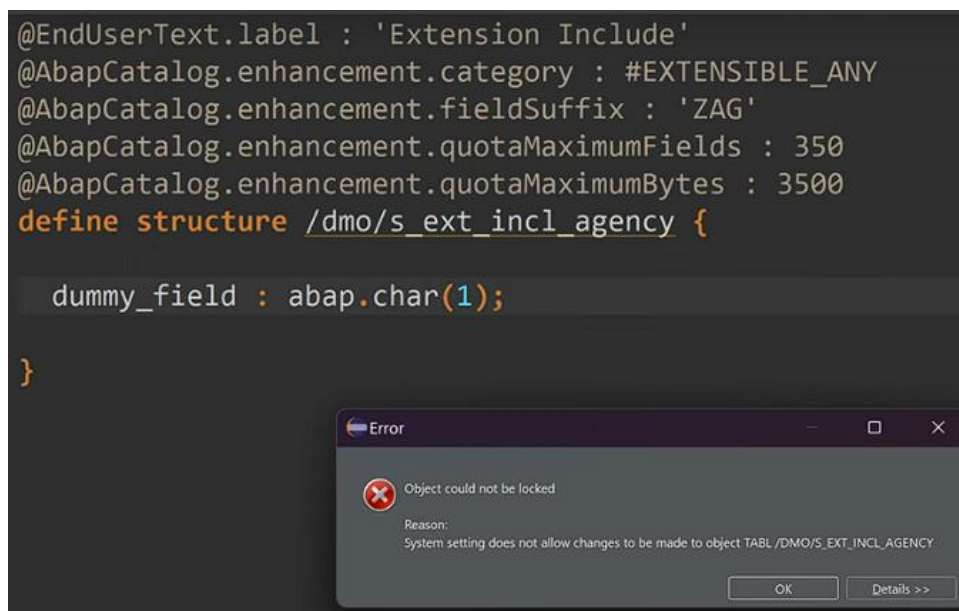


```
@EndUserText.label : 'Extension Include'
@AbapCatalog.enhancement.category : #EXTENSIBLE_ANY
@AbapCatalog.enhancement.fieldSuffix : 'ZAG'
@AbapCatalog.enhancement.quotaMaximumFields : 350
@AbapCatalog.enhancement.quotaMaximumBytes : 3500
define structure /dmo/s_ext_incl_agency {

    dummy_field : abap.char(1);

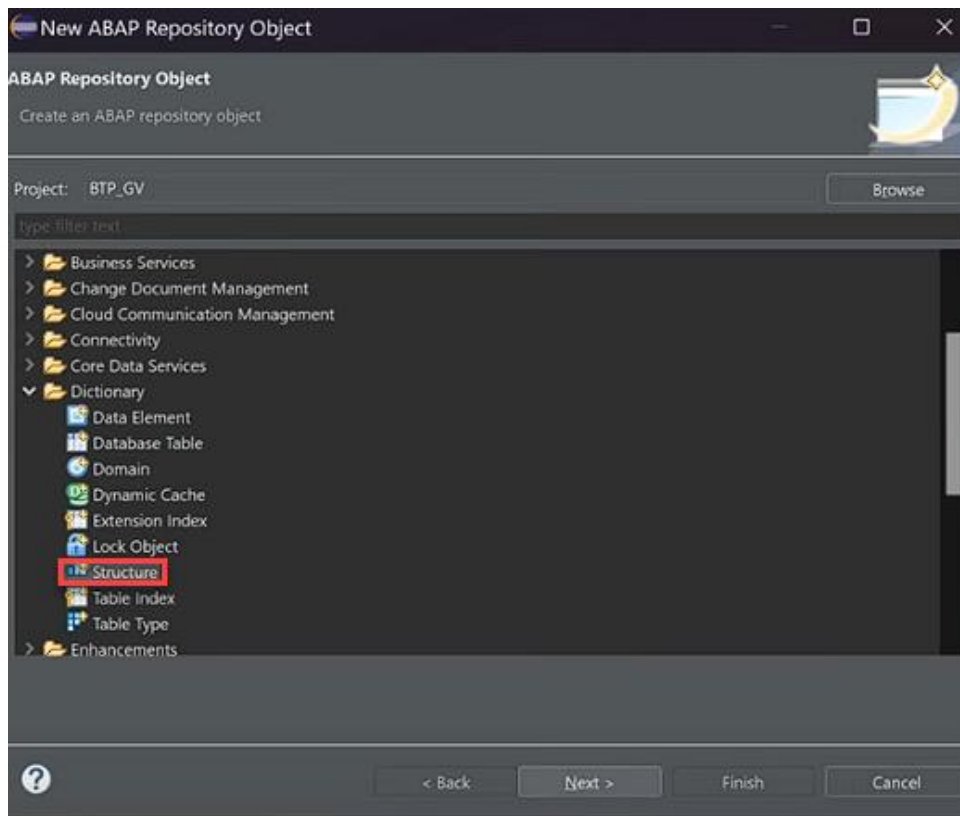
}
```

Debido a que es una estructura estándar SAP, no se puede editar de forma normal.



Pero es posible crear una estructura Z que pueda extender a dicha estructura estándar. Es recomendable crear un paquete que permitirá agregar los objetos que se generarán durante la resolución del requerimiento.





Al crear la nueva estructura tendrá por defecto el siguiente código en su definición.

```
@EndUserText.label : 'Agency Extension - Structure Persistence'
@AbapCatalog.enhancement.category : #NOT_EXTENSIBLE
define structure zst_e_agency {
    component_to_be_changed : abap.string(0);
}
```

Luego es necesario cambiar la definición de la estructura por “EXTEND TYPE” y luego colocar el nombre de la estructura estándar a extender y especificar el nombre de la estructura Z, con la sentencia “with”.

```
@EndUserText.label : 'Agency Extension - Structure Persistence'
@AbapCatalog.enhancement.category : #NOT_EXTENSIBLE
EXTEND TYPE /dmo/s_ext_incl_agency with zst_e_agency {
    component_to_be_changed : abap.string(0);
}
```

Es importante terminar cada nuevo campo creado con el sufijo definido en la estructura estándar a extender en este caso “ZAG”. y se



recomienda agregar el prefijo “ZZ” a cada nuevo elemento para su identificación como campo Z o personalizado.

```
@EndUserText.label : 'Agency Extension - Structure Persistence'
@AbapCatalog.enhancement.category : #NOT_EXTENSIBLE
EXTEND TYPE /dmo/s_ext_incl_agency with zst_e_agency {
  zzagtypezag : abap.string(0);
}
```

En el caso de que no se agregue el sufijo se impedirá con la definición del campo por medio de un error.

Además estos campos añadidos no permiten el tipo de datos estándar **abap.dataType**. En su lugar es necesario colocar un elemento de datos para establecer el tipo de datos.

Este sería un ejemplo de un elemento de datos creado donde su tipo de datos y longitud ha sido establecido por un dominio.

Data Element: ZDE_AG_TYPE	
<b>Data Type Information</b> Specify the data type of the data element	
Category:	Domain
Type Name:	ZDE_AG_TYPE <span>Browse...</span>
Data Type:	CHAR
Length:	2
<b>Field Labels</b> Provide field labels and set maximum lengths	
Short:	Agency Typ
Medium:	Agency Type
Long:	Agency Type
Heading:	Agency Type

Otro aspecto importante en la definición de los nombres de los campos de extensión es evitar caracteres especiales, como el guión bajo (“\_”), y respetar las restricciones de prefijos y sufijos mencionadas anteriormente. Esto se debe a que no es posible agregar caracteres especiales en los nombres de los campos de una tabla Draft. Dado que es posible habilitar nuevos campos en la capa Draft mediante la misma estructura de extensión, es crucial considerar esta limitación para evitar errores futuros.

```
zzag_typezag : abap.string(0);
```

De esta manera se podrá activar el nuevo campo personalizado y este proceso puede ser realizado para agregar múltiples campos en la tabla de persistencia dentro de la misma estructura Z.



```
@EndUserText.label : 'Agency Extension - Structure Persistence'
@AbapCatalog.enhancement.category : #NOT_EXTENSIBLE
extend type /dmo/s_ext_incl_agency with zst_e_agency {
    zzagttypezag : zde_ag_type;
}
```

Por medio de la herramienta “Data Preview” en la definición de la CDS de persistencia se puede identificar el nuevo campo personalizado agregado.

### 1.3. Extensión Query – Persistencia Draft

La extensión de la persistencia Draft se refiere a la ampliación de la capa de persistencia de una aplicación RAP (Restful Application Programming) para incluir nuevos campos, teniendo en cuenta la característica Draft. Esta característica permite que la aplicación gestione borradores de los datos antes de su almacenamiento final, lo que requiere una configuración y mapeo adecuados de los nuevos campos tanto en la tabla de persistencia activa como en la tabla de tipo Draft.

**Navegación a la capa Draft:**





Desde Eclipse con ADT, se evalúa la definición de comportamiento (Behavior Definition) de la aplicación, observando que la aplicación cuenta con la característica Draft con la sentencia **with draft** en el encabezado de la definición. Habilitando el uso de tablas de tipo Draft y se especifica por medio de la sentencia **draft table**.

```
1 managed implementation in class /dmo/bp_r_agencytp unique;  
2 strict ( 2 );  
3 with draft;  
4 extensible  
5 {  
6   with determinations on modify;  
7   with determinations on save;  
8   with validations on save;  
9   with additional save;  
10 }  
11  
12 define behavior for /DMO/R_AgencyTP alias /DMO/Agency  
13 persistent table /dmo/agency  
14 draft table /dmo/agency_d query /DMO/R_AgencyDraft  
15 lock master  
16 total etag LastChangedAt  
17 authorization master ( global )  
18 etag master LocalLastChangedAt  
19 late numbering  
20 extensible  
21 {  
22   create;  
23   update;  
24   delete;
```

Al navegar a la tabla del tipo draft. Se observa que al igual que en la capa de persistencia activa la anotación **@AbapCatalog.enhancement.category** con la opción **#EXTENSIBLE\_ANY**, indicando que se pueden agregar nuevos campos.



```
@EndUserText.label : 'Draft table for entity /DMO/R_AGENCY'
@AbapCatalog.enhancement.category : #EXTENSIBLE_ANY
@AbapCatalog.tableCategory : #TRANSPARENT
@AbapCatalog.deliveryClass : #A
@AbapCatalog.dataMaintenance : #RESTRICTED
define table /dmo/agency_d {

  key mandt      : mandt not null;
  key agencyid   : /dmo/agency_id not null;
  key draftuuid  : sdraft_uuid not null;
  name          : /dmo/agency_name;
  street        : /dmo/street;
  postalcode    : /dmo/postal_code;
  city         : /dmo/city;
  countrycode   : land1;
  phonenumber   : /dmo/phone_number;
  emailaddress  : /dmo/email_address;
  webaddress    : /dmo/web_address;
  attachment    : /dmo/attachment;
  mimetype      : /dmo/mime_type;
  filename      : /dmo/filename;
  localcreatedby : abp_creation_user;
  localcreatedat : abp_creation_tstmpl;
  locallastchangedby : abp_locinst_lastchange_user;
  locallastchangedat : abp_locinst_lastchange_tstmpl;
  lastchangedat  : abp_lastchange_tstmpl;
  "%admin"      : include sych_bdl_draft_admin_inc;
  include /dmo/s_ext_incl_agency;
```

El procedimiento es el mismo para agregar nuevos campos en la capa de persistencia activa. En donde es por medio de la estructura especificada por medio de la instrucción **include**. Se puede extender para agregar nuevos campos. En este caso al ser la misma estructura no es necesario realizar alguna otra acción. Recordando que, en los nombres de los campos de la estructura de extensión, se deben evitar caracteres especiales como el guión bajo (“\_”) y respetar las restricciones de prefijos y sufijos.

```
@EndUserText.label : 'Extension Include'
@AbapCatalog.enhancement.category : #EXTENSIBLE_ANY
@AbapCatalog.enhancement.fieldSuffix : 'ZAG'
@AbapCatalog.enhancement.quotaMaximumFields : 350
@AbapCatalog.enhancement.quotaMaximumBytes : 3500
define structure /dmo/s_ext_incl_agency {

  dummy_field : abap.char(1);

}
```



### Añadir un Campo Personalizado:

Además se debe asegurar de que el nuevo campo también esté incluido en la consulta de la entidad para que se mapee correctamente desde la tabla de tipo Draft y esté disponible en la definición del comportamiento. En otras palabras, debemos actualizar la entidad para que el nuevo campo agregado en la tabla Draft sea accesible y utilizado en las consultas correspondientes.

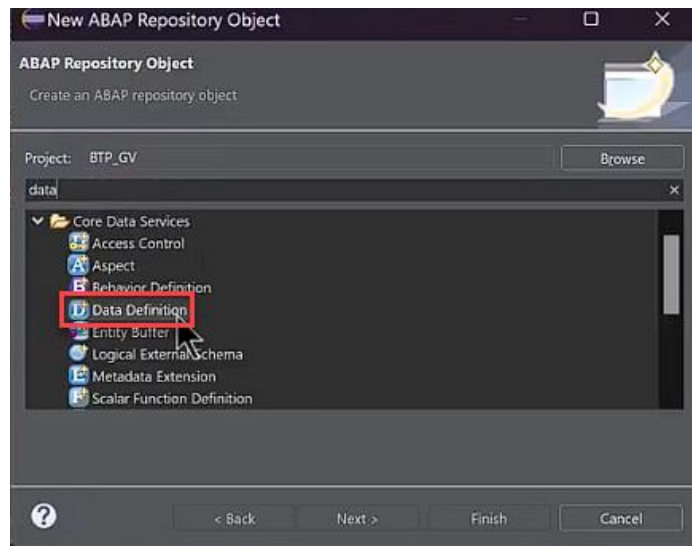
```
define behavior for /DMO/R_AgencyTP alias /DMO/Agency
persistent table /dmo/agency
draft table /dmo/agency_d query /DMO/R_AgencyDraft
```

Al navegar a esa entidad por medio del archivo Behavior Definition es necesario validar que dicha entidad cuente con las anotaciones de extensibilidad para que pueda permitir el proceso de ampliación por medio de vistas de extensión.

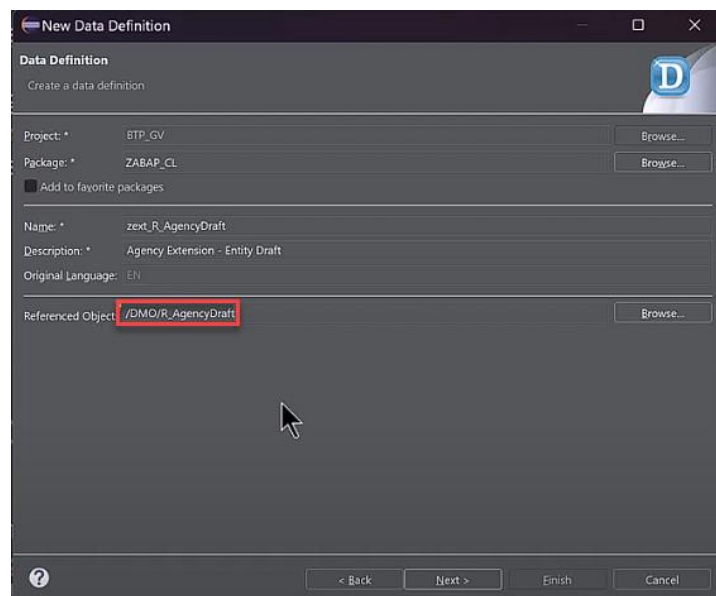
```
@AbapCatalog.extensibility: {
  extensible: true,
  dataSources: ['Agency'],
  elementSuffix: 'ZAG',
  quota: {
    maximumFields: 500,
    maximumBytes: 50000
  }
}

define view entity /DMO/R_AgencyDraft
as select from /dmo/agency_d as Agency
```

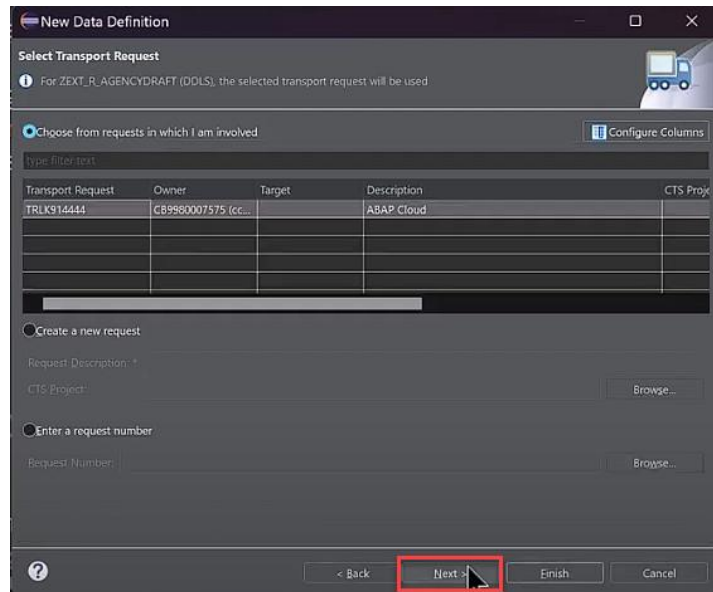
Para crear una vista de extensión es necesario crear un objeto “Data Definition”.



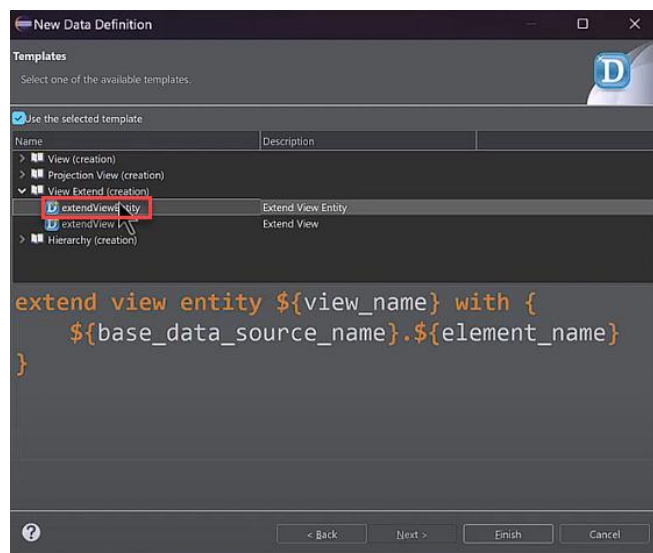
Y referenciar el objeto a ampliar, aunque esta especificación se puede realizar posteriormente a la definición del objeto



Luego es necesario utilizar una plantilla para que en la definición del objeto cuente con los elementos necesarios para la ampliación. Dicha plantilla puede ser seleccionada al momento de asignar una orden de transporte al presionar el botón “Next”.



Y seleccionar la plantilla **extendViewEntity** que se encuentra en la carpeta **View Extend (creation)**.



```

extend view entity /DMO/R_AgencyDraft with {
    base_data_source_name.element_name
}
    
```

Para agregar un campo se debe colocar el nombre de la fuente de datos o el alias si se encuentra este especificado por medio de la anotación de extensibilidad **@AbapCatalog.extensibility.dataSources**. Posteriormente el campo que fue agregado en la estructura de ampliación.

```

extend view entity /DMO/R_AgencyDraft with {
    Agency.zzag
}
    
```

**zzagtypezzag - /dmo/agency\_d as Agency (column)**





```
extend view entity /DMO/R_AgencyDraft with {  
    Agency.zzagtypezag  
}
```

Por último se activa la nueva definición.

Es necesario extender tanto la tabla de persistencia activa como la tabla Draft para evitar errores en la definición del comportamiento y el service binding. La falta del campo en las capas de modelado y proyección puede generar errores que se detectan en el service binding.

#### 1.4. Extensión entidad CDS Root

La extensión de la entidad CDS Root es el proceso de agregar nuevos campos a la entidad raíz de una aplicación RAP (Restful Application Programming) utilizando Core Data Services (CDS). Este proceso asegura que los nuevos campos añadidos en la tabla de persistencia activa y en la tabla de tipo Draft se propaguen correctamente a través de todas las capas del modelo de datos, garantizando su disponibilidad y funcionalidad en la aplicación.

##### Navegación a la entidad raíz:

Continuando con la ampliación de los nuevos campos en la parte del modelado con entidades CDS. Se ha ampliado la tabla de persistencia activa de la aplicación RAP y la tabla de tipo Draft según la definición del comportamiento. Por lo que también es necesario realizar la ampliación en la entidad raíz. Para acceder o navegar a la entidad raíz existen varias opciones entre ellas:

- **Acceder por medio de la entidad de consumo:** Desde el service binding, se navega a la definición del servicio, luego a la entidad de consumo y finalmente a la entidad de tipo Root. Donde se especifica el nombre de la entidad raíz por medio de la sentencia **as projection on**.

```
define root view entity /DMO/C_AgencyTP  
    provider contract transactional_query  
    as projection on /DMO/R_AgencyTP as Agency
```



- **Acceder por medio del Behavior Definition:** También es posible navegar a la entidad raíz por medio del archivo behavior definition. Donde se especifica el nombre de la entidad raíz por medio de la sentencia **define behavior for**.

```

1 managed implementation in class /dmo/bp_r_agencytp unique;
2 strict ( 2 );
3 with draft;
4 extensible
5 {
6   with determinations on modify;
7   with determinations on save;
8   with validations on save;
9   with additional save;
10 }
11
12 define behavior for /DMO/R_AgencyTP alias /DMO/Agency
13 persistent table /dmo/agency
14 draft table /dmo/agency_d query /DMO/R_AgencyDraft
15 lock master
16 total etag LastChangedAt
17 authorization master ( global )
18 etag master LocalLastChangedAt
19 late numbering
20 extensible
21 {
22   create;
23   update;
24   delete;

```

La entidad de tipo Root contiene la anotación **@AbapCatalog.extensibility**. Por lo que permite la creación de vistas de extensión para ampliar los campos definidos por defecto.

```

@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Agency'
@Search.searchable: true

@AbapCatalog.extensibility: {
  extensible: true,
  elementSuffix: 'ZAG',
  allowNewDatasources: false,
  dataSources: ['_Extension'],
  quota: {
    maximumFields: 500,
    maximumBytes: 50000
  },
  allowNewCompositions: true
}

define root view entity /DMO/R_AgencyTP
as select from /DMO/I_Agency as Agency
association [0..1] to I_Country as _Country on $projection.CountryCode = _Country.Country
association [1] to /DMO/E_Agency as _Extension on $projection.AgencyID = _Extension.AgencyId

```

De igual forma que la entidad de la asociación definida como fuente de datos en este caso “\_Extension”. Especificada en la anotación **@AbapCatalog.extensibility.dataSources**.





```
@AbapCatalog.viewEnhancementCategory: [#PROJECTION_LIST]
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Agency - Extension Include View'

@AbapCatalog.extensibility: {
  extensible: true,
  elementSuffix: 'ZAG',
  allowNewDatasources: false,
  dataSources: ['Agency'],
  quota: {
    maximumFields: 500,
    maximumBytes: 50000
  }
}

define view entity /DMO/E_Agency
  as select from /dmo/agency as Agency
```

Con la diferencia que la fuente de datos principal que forma parte de la entidad raíz no permite agregar nuevos campos.

```
define root view entity /DMO/R_AgencyTP
  as select from /DMO/I_Agency as Agency
  association [0..1] to I_Country as _Country on $projection.CountryCode = _Country.Country
  association [1] to /DMO/E_Agency as _Extension on $projection.AgencyID = _Extension.AgencyId
```

Debido a que no cuenta con las anotaciones de extensibilidad.

```
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Agency View - CDS Data Model'
@Search.searchable: true

define view entity /DMO/I_Agency
  as select from /dmo/agency as Agency

  association [0..1] to I_Country as _Country on $projection.CountryCode = _Country.Country
```

Por lo tanto, **es necesario crear dos vistas de extensión**: una para la entidad de la asociación que se utiliza como fuente de datos especificada en las anotaciones de la entidad raíz y otra para la propia entidad raíz, que sí permiten la extensibilidad.

### Añadir un Campo Personalizado en la fuente de datos:

Se crea un nuevo objeto Data Definition referenciando la entidad de la asociación.



The screenshot shows the 'New Data Definition' dialog box. The 'Referenced Object' field is highlighted with a red box and contains the text '/DMO/E\_Agency'. Other fields include Project (BTP\_GV), Package (ZABAP\_CL), Name (zext\_E\_Agency), and Description (Agency Extension - Entity Association).

Recordando seleccionar la plantilla **extendViewEntity** que se encuentra en la carpeta **View Extend (creation)**.

The screenshot shows the 'New Data Definition' dialog box with the 'Templates' section selected. The 'extendViewEntity' template is highlighted with a red box. The dialog also shows a preview of the ABAP code for the selected template.

Y por medio del alias de la entidad de la asociación.

```
@AbapCatalog.extensibility: {
  extensible: true,
  elementSuffix: 'ZAG',
  allowNewDatasources: false,
  datasources: ['Agency'],
  quota: {
    maximumFields: 500,
    maximumBytes: 50000
  }
}

define view entity /DMO/E_Agency
as select from /dmo/agency as Agency
```

Colocar el nuevo campo necesario para propagar primero en la entidad de la asociación.



```
extend view entity /DMO/E_Agency with {
  Agency.zzagtypezag
}
```

### Añadir un Campo Personalizado en la entidad raíz:

De igual forma, se crea un nuevo objeto Data Definition referenciando a la entidad raíz. Y seleccionando la plantilla **extendViewEntity** que se encuentra en la carpeta **View Extend (creation)**.

The screenshot shows the 'New Data Definition' window. The 'Referenced Object' field is highlighted with a red box and contains the value '/DMO/R\_AgencyTP'. Other fields include Project (BTP\_GV), Package (ZABAP\_CL), Name (zext\_R\_AgencyTP), Description (Agency Extension - Entity Root), and Original Language (EN). Navigation buttons at the bottom include Back, Next, Finish, and Cancel.

Y por medio del alias de la fuente de datos especificada en su definición por medio de las anotaciones de extensibilidad en la entidad raíz.

Para luego colocar el nombre del campo definido en la capa de persistencia propagado por medio de la entidad de la asociación.

```
extend view entity /DMO/R_AgencyTP with {
  _Extension.zzag
}

zzagtypezag - /dmo/e_agency as _Extension (column)
```

```
extend view entity /DMO/R_AgencyTP with {
  _Extension.zzagtypezag
}
```

Después de la activación de la extensión, es posible verificar en la entidad raíz que el nuevo campo agregado aparece correctamente en la selección.

La extensión ha permitido agregar elementos nuevos en la capa Root del modelado con entidades CDS.

La extensión de la entidad CDS Projection es el proceso de agregar nuevos campos a una aplicación RAP (Restful Application Programming) estándar en la capa de proyección o de consumo. Este proceso implica extender las entidades de datos para incluir los nuevos campos en todas las capas del modelo, garantizando su disponibilidad y funcionalidad en la interfaz de usuario y en las operaciones CRUD (Create, Read, Update, Delete).

### Navegación a la Capa de Proyección o de Consumo:

26



```
@Search.searchable: true
@Metadata.allowExtensions: true
@ObjectModel.semanticKey: ['AgencyID']

@AbapCatalog.extensibility: {
  extensible: true,
  dataSources: ['Agency'],
  elementSuffix: 'ZAG',
  quota: {
    maximumFields: 500,
    maximumBytes: 50000
  }, allowNewCompositions: true
}

define root view entity /DMO/C_AgencyTP
  provider contract transactional_query
  as projection on /DMO/R_AgencyTP as Agency
```

### Añadir un Campo Personalizado:

Como se ha realizado con anterioridad en puntos anteriores para poder ampliar la entidad de consumo se debe crear una vista de extensión creado a partir de un nuevo objeto Data Definition referenciando la entidad de consumo. Recordando seleccionar la plantilla **extendViewEntity** que se encuentra en la carpeta **View Extend (creation)**.

The screenshot shows the 'New Data Definition' dialog box. The 'Data Definition' section is active. The 'Project' is 'BTP\_GV' and the 'Package' is 'ZABAP\_CL'. The 'Name' is 'zext\_C\_AgencyTP' and the 'Description' is 'Agency Extension - Entity Projection'. The 'Original Language' is 'EN'. The 'Referenced Object' field is highlighted with a red box and contains the value '/DMO/C\_AgencyTP'. The 'Add to favorite packages' checkbox is unchecked. The 'Browse...' button is visible next to the 'Referenced Object' field. The bottom of the dialog has buttons for '< Back', 'Next >', 'Finish', and 'Cancel'.

De igual manera para agregar el campo personalizado propagado de capas anteriores primero, se coloca la fuente de datos utilizando el alias especificado en la anotación **@AbapCatalog.extensibility**, con la propiedad **dataSource**, permitiendo el acceso al nuevo campo que ha sido propagado desde la capa de persistencia.



```
extend view entity /DMO/C_AgencyTP with {
  Agency.zzagtypezag
}
```

Se activa la nueva definición, asegurando que el campo esté disponible en la entidad de consumo.

Agency	No Name	Schedule	Country	Country	PhoneNo	EMailAdd	Website	Rating	Address	City	Country/Region Key	Agency Type
70001	No Name	Schedule	DE	Germany	+49 121 77719	info@no-name.de	http://www.no-name.de	0.000000	4.7	Coal and smooth	We are the leader in business!	Agency
70002	modified agency name	Coal and smooth	DE	Germany	+49 121 77719	info@no-name.de	http://www.no-name.de	0.000000	4.7	Coal and smooth	We are the leader in business!	Agency
70003	Happy Hopping	Happy Hopping	DE	Germany	+49 30 8653 0	info@haghop.de	http://www.haghop.de	0.000000	4.7	Happy Hopping	Happy Hopping is the leader in business!	Agency
70004	Pink Panther	We are the best!	DE	Germany	+49 69 467853 0	info@pinkpanther.de	http://www.pinkpanther.de	0.000000	4.7	Pink Panther	We are the best!	Agency

En la interfaz el campo personalizado agregado se encuentra oculto, pero puede habilitarse en cualquier momento. Para realizar eso es necesario presionar el botón .

Agency Name	Slogan	Rating	Address	Contact Data
70002 modified agency name	Coal and smooth	★★★★★	Berliner Allee 11 40880 Düsseldorf Germany (DE)	+49 2102 69955 info@flyhigh.de http://www.flyhigh.de
70003 Happy Hopping	Happy Hopping is the leader in business!	★★★★★	Calvinstr. 36 13467 Berlin	+49 30 8653 0 info@haghop.de http://www.haghop.de
70004 Pink Panther	We are the best!	★★★★★	Auf der Schanz 54 65836 Frankfurt Germany (DE)	+49 69 467853 0 info@pinkpanther.de http://www.pinkpanther.de

El campo se puede mostrar en las columnas del listado, pero no aparece en el proceso de edición de los registros.

View Settings

Columns Sort Group

Search Show Selected

☒ Columns (715)

☒ Customer Text

☒ Agency Name

☒ Slogan

☒ Rating

☒ Address

☒ Contact Data

☒ **Agency Type**

☐ Attachment

☐ City

☐ Country/Region Key

☐ E-Mail Address

☐ Phone No.

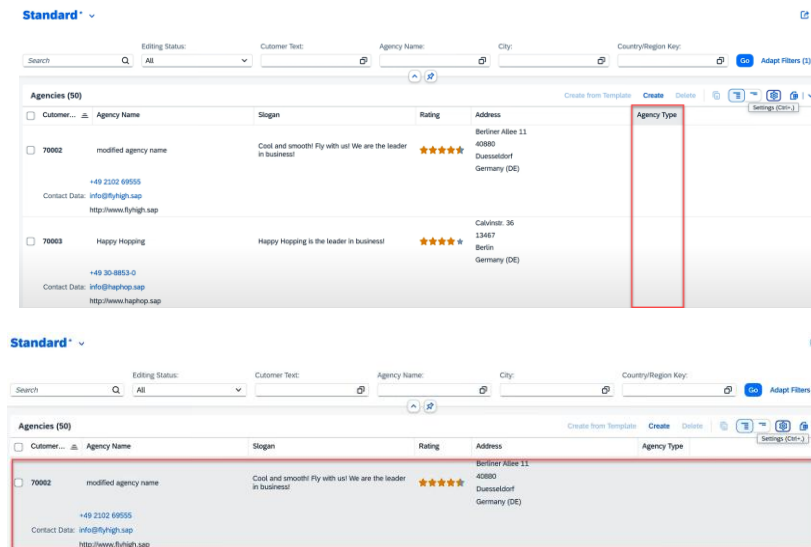
☐ Postal Code

☐ Street

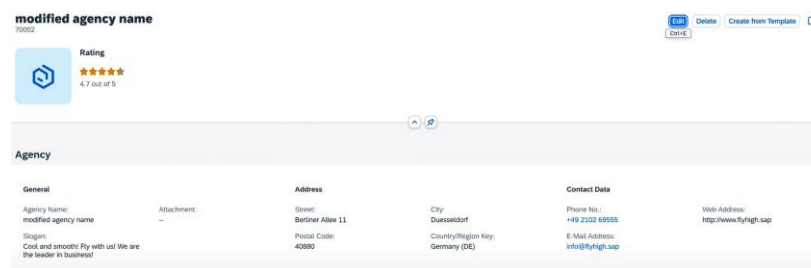
☐ Web Address

OK Cancel





Aunque el campo se ha expuesto en la entidad de consumo, no es posible editarlo fácilmente en la interfaz de usuario debido a la falta de anotaciones semánticas.



Después de la extensión en la capa de proyección o consumo, el campo está totalmente disponible para operaciones CRUD (Create, Read, Update, Delete).

## 1.6. Extensión Metadata Extensions

La extensión Metadata Extensions es el proceso de agregar nuevos campos o elementos en la interfaz de usuario de una aplicación RAP (Restful Application Programming) estándar. Esto se realiza creando un metadata extensions, que permita definir cómo se presentarán y gestionarán estos nuevos campos en diversas vistas, como la pantalla de selección, las columnas del listado y los detalles de navegación.

### Navegación al Metadata Extensions:

Desde la entidad de consumo se debe validar que cuente con la anotación **@Metadata.allowExtensions** con el valor true, lo que indica






que es posible agregar o personalizar aspectos en la interfaz de usuario sobre la nueva columna.

```

1 • @EndUserText.label: 'Agency'
2 @AccessControl.authorizationCheck: #NOT_REQUIRED
3
4 @Search.searchable: true
5 @Metadata.allowExtensions: true
6 @ObjectModel.semanticKey: ['AgencyID']
7
8 @AbapCatalog.extensibility: {
9   extensible: true,
10  dataSources: ['Agency'],
11  elementSuffix: 'ZAG',
12  quota: {
13    maximumFields: 500,
14    maximumBytes: 50000
15  }, allowNewCompositions: true
16 }
17
18 define root view entity /DMO/C_AgencyTP
19   provider contract transactional_query
20   as projection on /DMO/R_AgencyTP as Agency

```

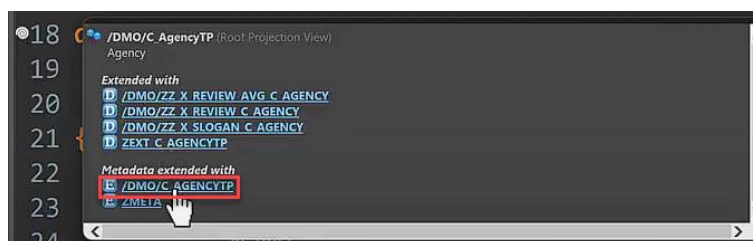
Luego es necesario seleccionar , ubicado en la línea donde se encuentra la sentencia **define root view entity**, para desplegar el menú de relaciones de la entidad de consumo.

```

1 • @EndUserText.label: 'Agency'
2 @AccessControl.authorizationCheck: #NOT_REQUIRED
3
4 @Search.searchable: true
5 @Metadata.allowExtensions: true
6 @ObjectModel.semanticKey: ['AgencyID']
7
8 @AbapCatalog.extensibility: {
9   extensible: true,
10  dataSources: ['Agency'],
11  elementSuffix: 'ZAG',
12  quota: {
13    maximumFields: 500,
14    maximumBytes: 50000
15  }, allowNewCompositions: true
16 }
17
18 define root view entity /DMO/C_AgencyTP
19   provider contract transactional_query
20   as projection on /DMO/R_AgencyTP as Agency

```

Y seleccionar en el apartado “Metadata extended with” el archivo perteneciente al Metadata Extension, vinculado a la entidad de consumo.





Se puede evidenciar que por defecto la capa de la definición del Metadata Extension es **#CORE**. Aunque es necesario que el archivo a extender se encuentre en una capa superior como **#CUSTOMER** para que pueda ser relevante y utilizada en la interfaz de usuario.

```
@Metadata.layer: #CORE
@UI: { headerInfo: {
  typeName: 'Agency',
  typeNamePlural: 'Agencies',
  title: { type: #STANDARD, value: 'Name' },
  imageUrl: 'Attachment',
  description: { type: #STANDARD, value: 'AgencyID' }
},
presentationVariant: [{
  sortOrder: [{ by: 'AgencyID', direction: #ASC }],
  visualizations: [{type: #AS_LINEITEM}]
}]
}
annotate view /DMO/C_AgencyTP with
```

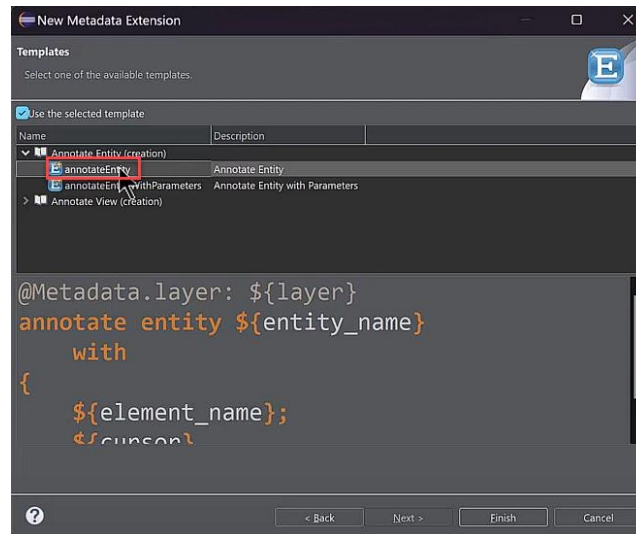
Aunque es posible colocar las anotaciones necesarias para la personalización de la interfaz de usuario directamente en la vista de extensión de la entidad de consumo.

```
extend view entity /DMO/C_AgencyTP with {
  Agency.zzagtypezag
}
```

Es recomendable realizarlo por medio de un nuevo Metadata extension extendida de la entidad de consumo. Es recomendable colocar el sufijo “ZME” como nomenclatura para identificar el archivo como Metadata extensión.



Donde es necesario seleccionar la plantilla **annotateEntity** que se encuentra en la carpeta **Annotate Entity (creation)**.



Luego especificar en la anotación **@Metadata.layer**, el valor **#CUSTOMER**.

```
@Metadata.layer: #CUSTOMER|
annotate entity /DMO/C_AgencyTP
with
{
  element_name:
}
```

Se agregan las anotaciones las anotaciones para modificar la interfaz de usuario por medio de la anotación **@UI**, utilizando las propiedades **lineItem** y **fieldGroup**. Estas anotaciones pueden ser extraídas del Metadata Extension original. Tomando en cuenta que la recomendación es que las posiciones se coloquen en múltiplos de 10 este comportamiento es realizado por el estándar. De esta manera, cuando se requiera crear un Metadatos de extension para agregar características adicionales a la interfaz de usuario se puedan agregar posiciones intermedias como 11 ó 21.

```
@Metadata.layer: #CUSTOMER
annotate entity /DMO/C_AgencyTP with
{
  @UI: { lineItem: [ { position: 21, importance: #HIGH } ],
        fieldGroup: [ { position: 21, qualifier: 'General_FG' } ] }
  zzagtypezag;
}
```



Se activa el Metadata extension y se refresca la aplicación en la interfaz de usuario para observar los nuevos cambios.

Customer	Agency Name	Agency Type	Slogan	Rating	Address
70002	modified agency name		Cool and smooth! Fly with us! We are the leader in business!	★★★★★	Berliner Allee 11 40880 Quasendorf Germany (DE) Calvinstr. 36 13467 Berlin Germany (DE)
70003	Happy Hopping		Happy Hopping is the leader in business!	★★★★★	Auf der Schand 54 80336 Frankfurt Germany (DE)
70004	Pink Panther		We are the best!	★★★★★	Gutten-Jung-Str. 425 90405 Nürnberg Germany (DE)
70005	Your Choice		Better with us!	★★★★★	Via Marcant 123 00128 Berlin Germany (DE)
70006	Bella Bella		Cool and save!	★★★★★	Roma Italy (IT)

Recordando que el nombre del campo Agency Type que aparece en la interfaz de usuario, proviene del elemento de datos.

Data Element: ZDE_AG_TYPE		Field Labels
Data Type Information Specify the data type of the data element		Provide field labels and set maximum lengths
Category:	Domain	Short: Agency Type
Type Name:	ZDO_AG_TYPE	Medium: <b>Agency Type</b>
Data Type:	CHAR	Long: Agency Type
Length:	2	Heading: Agency Type