LOGALI

Teoría

Elementos Virtuales

ABAP RESTful – Arquitectura Cloud

Contenido

1. Elementos Virtuales

1.1. Elemento virtual – Creación

Los elementos virtuales en aplicaciones RAP permiten la integración de datos externos y la implementación de lógica avanzada para cálculos y transformaciones que no pueden realizarse directamente en la base de datos. Estos elementos se determinan en la capa ABAP utilizando el Service Adaptation Description Language (SADL). Utilizando vistas de proyección y clases ABAP, se pueden definir y calcular estos elementos de manera eficiente, asegurando que los datos se presenten de manera coherente y precisa en la interfaz de usuario.

Creación de los elementos virtuales:

Los elementos virtuales se crean en las vistas de proyección CDS de consumo que están vinculadas en la entidad raíz por medio del tipo proyección **transactional_query.** Recordando que este tipo de entidad mantiene la siguiente estructura:

```
@EndUserText.label: 'CDS Interface Projection'
@AccessControl.authorizationCheck: #NOT_REQUIRED
@Metadata.ignorePropagatedAnnotations: true
define root view entity entity_name
 provider contract transactional_query
 as projection on data_source_name
{
// components from database_source_name with alias
 components,
 // audit components
@Semantics.systemDateTime.localInstanceLastChangedAt: true
local_last_changed_at as LocalLastChangedAt,
@Semantics.systemDateTime.lastChangedAt: true
last_changed_at      as LastChangedAt,
 // Publication of associations
 _AssociationName // Make association public
 }
```

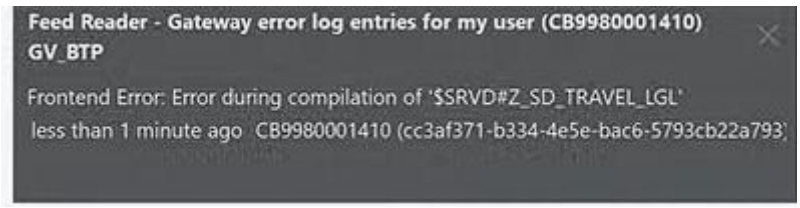
Los elementos visuales se agregan como campos adicionales donde es necesario especificar el elemento de datos asociado por medio de un elemento de datos y su valor en la interfaz de usuario se determina al agregar lógica por medio de una clase abap utilizando el Service Adaptation Description Language (SADL). Para definir un elemento virtual a un campo se debe añadir lo siguiente:

```
virtual VirtualElementName  : data_element_name,
```

Para especificar la clase abap asociada es necesario especificarla por medio de la anotación **@ObjectModel** con la propiedad **virtualElementCalculatedBy**, además se puede incluir una descripción del campo. Como se muestra a continuación:

```
@EndUserText.label: 'VAT Included'
@Semantics.amount.currencyCode : 'CurrencyCode'
@ObjectModel.virtualElementCalculatedBy: 'ABAP:ZCL_SADL_CLASS_NAME'
virtual ComponentName : data_element_name,
```

En el caso de que no se haya especificado una clase abap asociada por medio de la anotación **@ObjectModel,** entonces se mostrará un error similar en la interfaz de usuario:



1.2. SADL Exit – Información de cálculo

La Implementación de los Elementos Virtuales en SAP con SADL, permiten exponer los elementos que no existen directamente en la base de datos y que requieren cálculos o información externa, se utiliza el SADL junto con las Core Data Services (CDS). En este contexto, se define un elemento virtual en la vista de proyección de

tipo **transactional_query** y se vincula a la clase ABAP que implementa su cálculo.

El proceso inicia con la creación manual de la clase que se va a utilizar para determinar el valor de los elementos virtuales, para luego implementar la interfaz **if_sadl_exit_calc_element_read,** e implementar los métodos de la interfaz para continuar con el proceso.

Ejemplo:

```
CLASS ZCL_SADL_CLASS_NAME DEFINITION
 PUBLIC
 FINAL
 CREATE PUBLIC .
 PUBLIC SECTION.
 interfaces if_sadl_exit_calc_element_read.
 PROTECTED SECTION.
 PRIVATE SECTION.
ENDCLASS.
CLASS zcl_virt_emem_sadl_price_437 IMPLEMENTATION.
 METHOD if_sadl_exit_calc_element_read~calculate.
 ENDMETHOD.
 METHOD if_sadl_exit_calc_element_read~get_calculation_info.
 ENDMETHOD.
ENDCLASS.
```

Implementación del método get_calculation_info:

Este método permite proporcionar la lista de elementos necesarios antes de calcular el valor de un elemento virtual en una vista de proyección CDS y esta es ejecutada antes de la recuperación de datos original. Dicho método presenta los siguientes parámetros necesarios para su configuración:

- **it_requested_calc_elements:** Esta tabla interna contiene los elementos de cálculo solicitados. Se utiliza para almacenar temporalmente los elementos que se necesitan para el cálculo del elemento virtual.

- **iv_entity:** Este parámetro almacena el nombre de la entidad CDS de la que se están obteniendo los datos. Es importante para identificar la fuente de los datos y asegurarse de que se están utilizando los datos correctos para el cálculo.
- **et_requested_orig_elements:** Esta tabla interna contiene los elementos que serán utilizados para la lógica. A diferencia de it_requested_calc_elements, se utiliza para almacenar los elementos originales que se necesitan para el cálculo.

Por lo general, se utiliza el parámetro **iv_entity** para identificar la entidad CDS con la que se va a trabajar, en este caso, la vista de proyección. Luego, se realiza una iteración en la tabla **it_requested_calc_elements** para identificar los elementos virtuales a utilizar e insertar los nombres de los campos de la entidad que se emplearán en la lógica en la tabla ordenada **et_requested_orig_elements**.

Ejemplo:

```
METHOD if_sadl_exit_calc_element_read~get_calculation_info.
  CASE iv_entity.
    WHEN 'ZDD_C_PROJ_VIEW_NAME' .
      LOOP AT it_requested_calc_elements INTO DATA(ls_requested_calc_elements).
        IF ls_requested_calc_elements EQ 'VirtualElementName '.
          INSERT CONV #( 'COMPONENT' ) INTO et_requested_orig_elements.
        ENDIF.
      ENDLOOP .
  ENDCASE.
ENDMETHOD.
```

1.3. SADL Exit – Lógica de cálculo

La implementación de la Lógica del Elemento Virtual se realiza en la capa ABAP utilizando el SADL, permitiendo cálculos y transformaciones de datos que no se pueden realizar directamente en la base de datos. Este proceso se realiza por medio del método **calculate** de la interfaz **if_sadl_exit_calc_element_read.**

Implementación del método calculate:

El método calculate permite la Implementación para campos calculados de una vista de proyección CDS dada y se ejecuta después de la recuperación de datos originales, proporcionando de esa forma valores calculados basados en valores dados de elementos originales. Además se puede realizar el cálculo ya sea una sola fila (entidad), una página o una tabla completa. Dicho método presenta los siguientes parámetros necesarios para la implementación de cálculos para los elementos virtuales:

- **it_original_data:** Esta tabla interna contiene los datos originales, al menos para los elementos originales solicitados. Es esencial para realizar los cálculos necesarios.
- **it_requested_calc_elements:** Esta tabla interna contiene los elementos de cálculo solicitados. Es transiente, lo que significa que solo existe durante la ejecución del método y no se almacena permanentemente.
- **ct_calculated_data:** Esta tabla interna contiene los campos calculados con una correspondencia 1:1 con los datos originales por índice. Aquí se devuelven los valores calculados.

Por lo general, es necesario crear una tabla interna del tipo de la vista de proyección para asignarle los valores donde corresponda de la tabla interna **it_original_data**, con los registros de la entidad raíz asociada a la proyección.

Luego, se iteran los registros para aplicar la lógica necesaria para el cálculo de los valores de los elementos virtuales. Finalmente, se devuelven los registros modificados de la tabla interna a la tabla **ct_calculated_data** para actualizar los elementos virtuales.

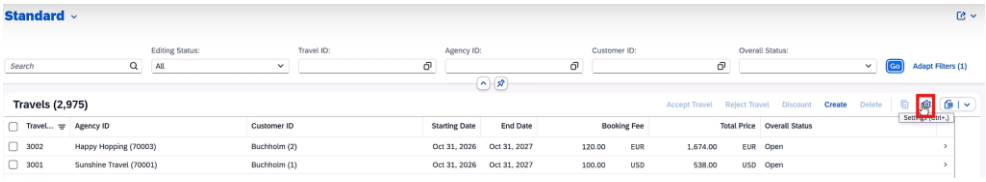
Ejemplo:

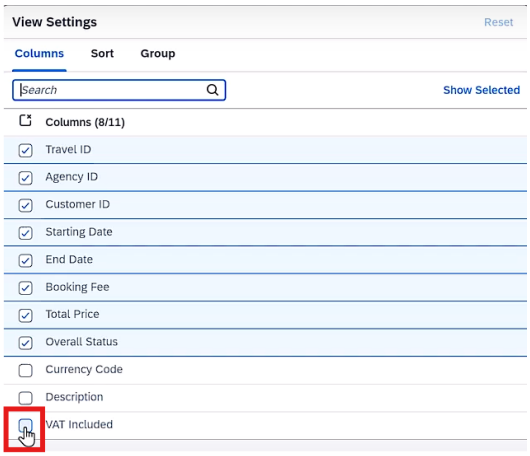
```
METHOD if_sadl_exit_calc_element_read~calculate.
* Declaration of necessary tables
  DATA lt_original_data TYPE STANDARD TABLE OF ZDD_C_PROJ_VIEW_NAME WITH
DEFAULT KEY.

  lt_original_data = CORRESPONDING #( it_original_data ) .

* Iterate through projection records to set values to Virtual Elements
  LOOP AT lt_original_data ASSIGNING FIELD-SYMBOL(<fs_original_data>).
* Logic to calculate values for Virtual elements
     <fs_original_data>-VirtualElementName  = <fs_original_data>-Component
  ENDLOOP.

* Update Virtual Elements
  ct_calculated_data = CORRESPONDING #( lt_original_data ).
ENDMETHOD.
```

1.4. Habilitación en la Interfaz de Usuario

Para que los elementos virtuales se puedan visualizar y utilizar en la interfaz de usuario, es necesario habilitarlo a través de la configuración de la interfaz y, en su caso, permitir que el usuario final también puede habilitarlo manualmente mediante la opción de configuración.

Por defecto los elementos virtuales no se muestran en la interfaz de usuario. Por lo que utilizando las configuraciones de dicha interfaz de usuario de SAP Fiori Elements, con la opción de configuración ⚙ (settings) se pueden mostrar una lista de los campos o elementos disponibles para su visualización en pantalla. Incluidos los ocultos a través de la anotación **@UI.hidden: true** y los elementos virtuales.





Es por medio de las Metadata Extensions que se puede establecer la configuración para agregar de manera automática los elementos virtuales que se encuentran ocultos. Para esto es necesario utilizar las anotaciones **@UI.lineItem** y **@UI.identification**, para establecer una posición, importancia y descripción para estos nuevos campos.

Ejemplo:

```
  @UI: {lineItem: [{ position: 51,
                     importance: #MEDIUM,
                     label: 'Virtual Element Name'}],
    identification: [{ position: 51  } ]}
VirtualElementName;
```