LOGALI

Teoría

EML – Determinaciones

ABAP RESTful – Arquitectura Cloud

Contenido

1. EML – Determinaciones

La determinación de valores permite asignar valores a campos específicos mediante lógica de programación, incluso si estos campos están en modo de solo lectura. Esto es útil cuando se quiere que los valores se generen automáticamente y no sean introducidos por el usuario.

1.1. Determinación On Save

Durante el proceso de creación de registros, los usuarios pueden introducir datos en campos que permiten la entrada. Sin embargo, para ciertos campos como por ejemplo los que están configurados como solo lectura o campos donde sea necesario actualizar el valor actual después de una operación al momento realizar una operación no se pueden introducir manualmente y se deben determinar mediante programación.

La determinación de valores se define en la clase de comportamiento Behavior Definition. Recordando que se realiza por medio de la sentencia:

determination setComponent **on save { create; }**

Implementación:

La determinación de valores para la operación **ON SAVE** se activa al guardar algún registro y para establecer la lógica pertinente en la clase Behavior Pool asociada, se utilizan métodos con el mismo nombre que la definición, los cuales se generan al momento de crear la clase. Para generar la lógica correspondiente para la declaración del método para agregar la lógica requerida en el Behavior Pool asociado a la entidad raíz, se debe realizar el proceso utilizando la ayuda del framework a través del botón  . Dichas determinaciones se declaran de la siguiente manera:

```
METHODS setComponentOnSave FOR DETERMINE ON SAVE
  IMPORTING keys FOR root_entity_name~setComponentOnSave.
```

Estos métodos cuentan con la tabla **keys,** utilizada para la lectura y modificación de los datos la entidad raíz obtenida de la interfaz de usuario al asignar valores en los campos de la entidad y la estructura **reported** para la personalización de los mensajes de las validaciones.

Ejemplo

```
METHOD setComponentOnSave.
  READ ENTITIES OF root_entity_name IN LOCAL MODE
  ENTITY alias_root_entity_name
  ALL FIELDS
  WITH CORRESPONDING #( keys )
  RESULT DATA(lt_root_entity).

  DELETE lt_root_entity WHERE Component IS NOT INITIAL.
  CHECK lt_root_entity IS NOT INITIAL.

DATA(lv_value) = 'Value' . // Any logic should be implemented here

MODIFY ENTITIES OF root_entity_name IN LOCAL MODE
ENTITY alias_root_entity_name
UPDATE FIELDS ( field field2 … )
  WITH VALUE #(  for ls_key in keys ( %tky = ls_key-%tky
                                      field = lv_value ) ).
ENDMETHOD.
```

1.2. Determinación On Modify

La determinación de valores para la operación **ON MODIFY** se activa al crear algún registro y para establecer la lógica pertinente en la clase Behavior Pool asociada, se utilizan métodos con el mismo nombre que la definición, los cuales se generan al momento de crear la clase. Sin embargo, en caso de agregar nuevas determinaciones donde la clase ya ha sido creada previamente, dichas determinaciones para las operaciones de guardado se deben declarar de la siguiente manera:

```
METHODS setComponentOnModify FOR DETERMINE ON SAVE
  IMPORTING keys FOR root_entity_name~setComponentOnModify.
```

Estos métodos cuentan con la tabla **keys,** utilizada para la lectura y modificación de los datos la entidad raíz obtenida de la interfaz de usuario al asignar valores en los campos de la entidad y la estructura

reported para la personalización de los mensajes de las validaciones. Por lo que la lógica de implementación es prácticamente la misma con la diferencia del momento de su ejecución siendo al momento de crear un registro y no al modificar como el concepto anterior.

1.3. Ejecución con Acción Interna

La acción interna es una característica que permite modularizar la lógica de programación que puede ejecutarse desde diferentes determinaciones, haciéndola reutilizable en múltiples procesos y escenarios dentro de la aplicación. Esta acción no se representa a través de botones en la interfaz de usuario, sino que se ejecuta mediante la lógica de programación (EML).

La determinación de las acciones internas se define en la clase de comportamiento Behavior Definition por medio de la sentencia:

```
internal action internalActionName;
```

Para generar la lógica correspondiente para la declaración del método para agregar la lógica requerida en el Behavior Pool asociado a la entidad raíz, se debe realizar el proceso utilizando la ayuda del framework a través del botón  .

La lógica mantiene el mismo formato al ya explicado anteriormente o en la implementación de las acciones donde es necesario leer la entidad raíz realizar, eliminar los registros duplicados realizar un loop con la lógica para editar los registros consultados para luego utilizar la tabla interna modificada para modificar los registros de la entidad.

Ejemplo:

```
METHOD internalActionName.
* Read root entity entries
  READ ENTITIES OF root_entity_name IN LOCAL MODE
  ENTITY alias_root_entity_name
  ALL FIELDS
  WITH CORRESPONDING #( keys )
  RESULT DATA(lt_root_entity).

* Delete duplicate records
  DELETE lt_root_entity WHERE Component IS NOT INITIAL.
  CHECK lt_root_entity IS NOT INITIAL.
```

```
* Any logic should be implemented here
  loop at lt_root_entity assigning field_symbol_name.
  endloop.

* Modify status in Root Entity
MODIFY ENTITIES OF root_entity_name IN LOCAL MODE
ENTITY alias_root_entity_name
UPDATE FIELDS ( field field2 … )
  WITH CORRESPONDING #( lt_root_entity  ).
ENDMETHOD.
```

1.4. Determinación sobre Elementos

Las acciones internas se pueden ejecutar desde cualquier determinación para las operaciones de creación y actualización de valores. Estas acciones internas permiten modularizar la lógica, haciéndola reutilizable en múltiples partes de la aplicación de esta forma es posible ejecutar procesos basándose en cambios en otros campos que se hayan declarados en la definición de la determinación en el Behavior Definition. Por medio de la sentencia:

```
determination setComponent on save/modify { create; field Component, …; }
```

Como es usual la lógica pertinente a la determinación se declara en la clase Behavior Pool asociada, se utilizan métodos con el mismo nombre que la definición. Para luego incluir la sentencia **MODIFY,** en el método agregando la palabra reservada **execute** para hacer la llamada a la ejecución de la acción interna:

Ejemplo:

```
MODIFY ENTITIES OF root_entity_name IN LOCAL MODE
ENTITY alias_root_entity_name
EXECUTE internalActionName
  FROM CORRESPONDING #( KEYS ).
```