

Simple Managed RAP

With Action, Validation and Determination

Developer Helper Guide

Last Updated : 16th Oct 2020

1. Table Definition	1
2. Data Insertion Class Definition	2
3. CDS Interface View	4
4. CDS Consumption View	5
5. Metadata for CDS View	6
6. Behavior Definition	8
7. Behavior implementation	9

Note: This document only consists of Final Code. The course contains many codes in between the explanations.

1. Table Definition

File Name : **zuserinfo_03**

Type : Database Table

=====

```
@EndUserText.label : 'User Data for App'
@AbapCatalog.enhancementCategory : #NOT_EXTENSIBLE
@AbapCatalog.tableCategory : #TRANSPARENT
@AbapCatalog.deliveryClass : #A
@AbapCatalog.dataMaintenance : #RESTRICTED
define table zuserinfo_03 {
  key client      : abap.clnt not null;
  key user_email  : abap.sstring(32) not null;
  first_name      : abap.sstring(32);
  last_name       : abap.sstring(32);
  is_admin        : abap.sstring(3);
  last_changed_at : timestamp;

}
```

2. Data Insertion Class Definition

File Name : **z_data_insert_03**

Type : ABAP Class

=====

CLASS z_data_insert_03 **DEFINITION**

PUBLIC

FINAL

CREATE PUBLIC .

PUBLIC SECTION.

INTERFACES if_oo_adt_classrun.

METHODS add_user_data.

METHODS reset_admin_all_user.

PROTECTED SECTION.

PRIVATE SECTION.

ENDCLASS.

CLASS z_data_insert_03 **IMPLEMENTATION.**

METHOD add_user_data.

DATA: lt_type_use **TYPE STANDARD TABLE OF** zuserinfo_03.

lt_type_use = **VALUE #**(

(user_email = 'aj@demo.com'

first_name = 'DEMO1'

last_name = 'DEMO1')

(user_email = 'vj@demo.com'

first_name = 'DEMO2'

last_name = 'DEMO2')

(user_email = 'Alex@demo.com'

first_name = 'DEMO3'

last_name = 'DEMO3')

).

```
INSERT zuserinfo_03 FROM TABLE @lt_type_use.  
ENDMETHOD.
```

```
METHOD reset_admin_all_user.  
  UPDATE zuserinfo_03 SET IS_ADMIN = 'NO'.  
ENDMETHOD.
```

```
METHOD if_oo_adt_classrun~main.  
*   add_user_data( ).  
    reset_admin_all_user( ).
```

```
ENDMETHOD.  
ENDCLASS.
```

3. CDS Interface View

File Name : **Z_I_USERINFO_M_03**

Type : CDS View

```
=====
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Interface CDS View for User'
define view entity Z_I_USERINFO_M_03 as select from zuserinfo_03
{
    //zuserinfo_03
    key user_email,
    first_name,
    last_name,
    is_admin,

    @Semantics.systemDateTime.lastChangedAt: true
    last_changed_at
}
```

4. CDS Consumption View

File Name : **Z_C_USERINFO_M_03**

Type : CDS View

=====

@AccessControl.authorizationCheck: #NOT_ALLOWED

@Metadata.allowExtensions: true

@EndUserText.label: 'Root View for User'

```
define root view entity Z_C_USERINFO_M_03 as select from Z_I_USERINFO_M_03
{
    //Z_I_USERINFO_M_03
    key user_email as Email,
    first_name,
    last_name,
    is_admin,
    concat_with_space(first_name, last_name, 1) as full_name,

    @Semantics.systemDateTime.lastChangedAt: true
    last_changed_at
}
```

5. Metadata for CDS View

File Name : **Z_C_USERINFO_M_03**

Type : Metadata Extension

```
=====
@Metadata.layer: #CORE
@UI.headerInfo:{

    typeName: 'User Info',
    typeNamePlural: 'Users Info',
    title:{
        value:'full_name',
        label:'Name'
    }
}
annotate view Z_C_USERINFO_M_03
with
{

    @UI.facet: [{ id:'Email', type:#IDENTIFICATION_REFERENCE, label:'User Info Detail' }]

    @UI.lineItem: [{
        position: 10 ,
        importance: #HIGH,
        label: 'Email'
    }]
    @UI.identification: [{ position: 10 ,label: 'Email'}]
    Email;

    @UI.lineItem: [{
        position: 20 ,
        importance: #HIGH,
        label: 'First Name'
    }]
    @UI.identification: [{ position: 10 ,label: 'First Name'}]
    first_name;

    @UI.lineItem: [{
        position: 30 ,
        importance: #HIGH,
```

```
label: 'Last Name'
}]
@UI.identification: [{ position: 10 ,label: 'Last Name'}]
```

```
last_name;
```

```
@UI.lineItem: [{
position: 40 ,
importance: #HIGH,
label: 'Admin'
},
{
type: #FOR_ACTION,
dataAction: 'makeUserAdmin',
label: 'Make Admin'
},
{
type: #FOR_ACTION,
dataAction: 'removeUserAdmin',
label: 'Remove Admin'
}
]
is_admin;
```

```
}
```


6. Behavior Definition

File Name : **zuserinfo_03**

Type : Behavior Definition

=====

managed implementation in class z_c_userinfo_bi_m_03 **unique;**

define behavior for Z_C_USERINFO_M_03 **alias** User_03

persistent table zuserinfo_03

lock master

//authorization master (instance)

etag master last_changed_at

{

//field (readonly) Email, first_name, last_name;

create;

update;

delete;

// mapping for zuserinfo_03 corresponding;

mapping for zuserinfo_03

{

Email = user_email;

first_name = first_name;

last_name = last_name;

is_admin = is_admin;

}

action makeUserAdmin **result [1] \$self;**

action removeUserAdmin **result [1] \$self;**

validation validEmail **on save {field Email;}**

determination assignAdmin **on modify { field Email;}**

}

7. Behavior implementation

File Name : **z_c_usernfo_bi_m_03**

Type : Behavior Definition

Global

=====

```
CLASS z_c_usernfo_bi_m_03 DEFINITION
PUBLIC ABSTRACT FINAL FOR BEHAVIOR OF z_c_userinfo_m_03.
ENDCLASS.
```

```
CLASS z_c_usernfo_bi_m_03 IMPLEMENTATION.
ENDCLASS.
```

Local Class

=====

```
*** use this source file for the definition and implementation of
*** local helper classes, interface definitions and type
*** declarations
```

```
CLASS lhc_user DEFINITION INHERITING FROM cl_abap_behavior_handler.
```

```
PRIVATE SECTION.
```

```
METHODS set_user_admin
```

```
FOR MODIFY IMPORTING keys FOR ACTION
User_03~makeUserAdmin RESULT result.
```

```
METHODS remove_user_admin
```

```
FOR MODIFY IMPORTING keys FOR ACTION
User_03~removeUserAdmin RESULT result.
```

```
METHODS check_valid_email
```

```
FOR VALIDATE ON SAVE importing keys FOR
User_03~validEmail.
```

```
METHODS assign_is_admin
```

```
FOR DETERMINE ON MODIFY importing keys FOR
User_03~assignAdmin.
```

```
ENDCLASS.
```

UI5CN

Learn Simple

www.ui5cn.com

© Copyright UI5 Community Network Pvt. Ltd.

SAP, SAP NetWeaver, SAP HANA, SAP Fiori are registered trademarks of SAP SE in Germany and in several other countries.

CLASS lhc_user IMPLEMENTATION.

METHOD set_user_admin.

LOOP AT keys ASSIGNING FIELD-SYMBOL(<fs_user>).

UPDATE zuserinfo_03 SET IS_ADMIN = 'YES' WHERE user_email =
@<fs_user>-Email.
if sy-subrc eq 0.

APPEND VALUE #(

Email = <fs_user>-Email
%param-Email = <fs_user>-Email
%param-is_admin = 'YES'

) TO result.

endif.

ENDLOOP.

ENDMETHOD.

METHOD remove_user_admin.

LOOP AT keys ASSIGNING FIELD-SYMBOL(<fs_user>).

UPDATE zuserinfo_03 SET IS_ADMIN = 'NO' WHERE user_email =
@<fs_user>-Email.
if sy-subrc eq 0.

APPEND VALUE #(

```
Email      = <fs_user>-Email
%param-Email = <fs_user>-Email
%param-is_admin = 'NO'
```

) TO result.

endif.

ENDLOOP.

ENDMETHOD.

METHOD check_valid_email.

data:

```
lr_regex type ref to cl_abap_regex,
lr_matcher type ref to cl_abap_matcher,
lv_email_to_check type string.
```

create object lr_regex

exporting

```
pattern = '[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}'
```

```
ignore_case = abap_true.
```

LOOP AT keys ASSIGNING FIELD-SYMBOL(<fs_user>).

```
lr_matcher = lr_regex->create_matcher( text = <fs_user>-Email ).
```

if lr_matcher->match() is INITIAL.

"We reach if email is not proper

```
APPEND VALUE #( Email = <fs_user>-Email ) TO failed-user_03.
```

```
APPEND VALUE #( Email = <fs_user>-Email
```

```
%msg = new_message(
```

```
id = 'Z_USER_M_03'
```

```
number = '001'
```

UI5CN

Learn Simple

www.ui5cn.com

© Copyright UI5 Community Network Pvt. Ltd.

SAP, SAP NetWeaver, SAP HANA, SAP Fiori are registered trademarks of SAP SE in Germany and in several other countries.

```

        v1 = <fs_user>-Email
        severity = if_abap_behv_message=>severity-error

    ) ) TO reported-user_03.

else.
    "We reach if email is proper

endif.

ENDLOOP.

ENDMETHOD.

METHOD assign_is_admin.

    LOOP AT keys ASSIGNING FIELD-SYMBOL(<fs_user>).

        IF <fs_user>-Email cs 'admin'.

            MODIFY ENTITY z_c_userinfo_m_03
            UPDATE FIELDS ( is_admin )
            WITH VALUE #( (
                Email = <fs_user>-Email
                is_admin = 'YES'
            ) ).

        ELSE.

            MODIFY ENTITY z_c_userinfo_m_03
            UPDATE FIELDS ( is_admin )
            WITH VALUE #( (
                Email = <fs_user>-Email
                is_admin = 'NO'
            ) ).

        ENDIF.

    ENDIF.

```

ENDLOOP.

ENDMETHOD.

ENDCLASS.

