Exercise 10M – Enhance Behavior with Action and Validation.

How to enhance behavior implementation

SAP Partner Workshop	10 min

Description

In this exercise, you'll learn how:

- How to enhance behavior definition

Prerequisites

- You have completed the steps mentioned until Exercise 10L

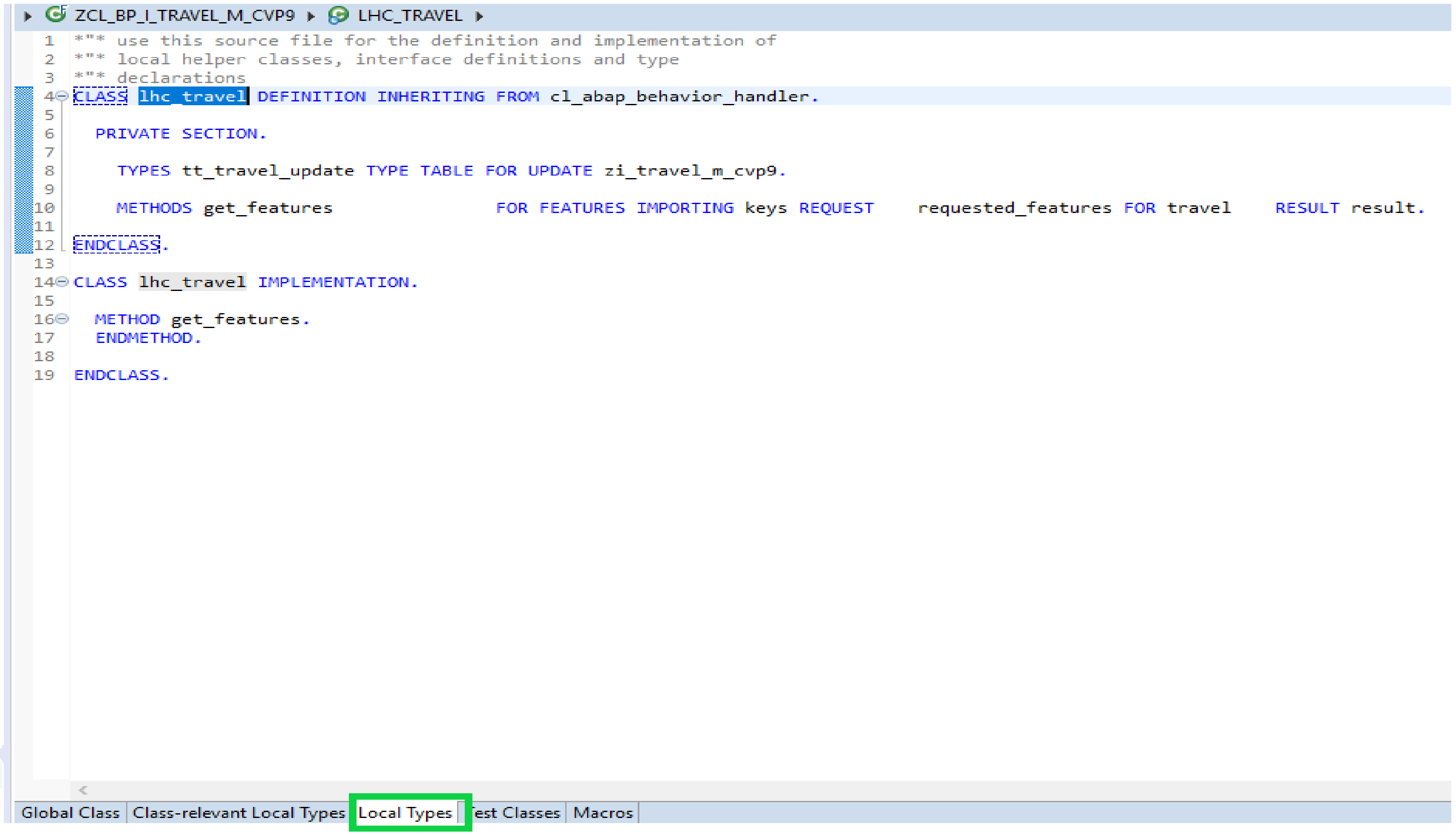
Target group

- Developers
- People interested in learning about development in ABAP env in SAP Cloud Platform with focus on RAP model.

Goal

The goal of this exercise is to enhance behavior implementation with actions and validations.

Open the behavior implementation `ZCL_BP_I_TRAVEL_M_XXX` and switch to local types to replace your code


ZCL_BP_I_TRAVEL_M_CVP9 ▸ LHC_TRAVEL ▸
Global Class | Class-relevant Local Types | Local Types | Test Classes | Macros

```
*"* use this source file for the definition and implementation of
*"* local helper classes, interface definitions and type
*"* declarations
CLASS lhc_travel DEFINITION INHERITING FROM cl_abap_behavior_handler.

  PRIVATE SECTION.

    TYPES tt_travel_update TYPE TABLE FOR UPDATE zi_travel_m_cvp9.

    METHODS get_features              FOR FEATURES IMPORTING keys REQUEST    requested_features FOR travel    RESULT result.

ENDCLASS.

CLASS lhc_travel IMPLEMENTATION.

  METHOD get_features.
  ENDMETHOD.

ENDCLASS.
```

In your **local types** replace your code with following:

```
*"* use this source file for the definition and implementation of

*"* local helper classes, interface definitions and type

*"* declarations

CLASS lhc_travel DEFINITION INHERITING FROM cl_abap_behavior_handler.


PRIVATE SECTION.


  TYPES tt_travel_update TYPE TABLE FOR UPDATE zi_travel_m_xxx.


  METHODS validate_customer          FOR VALIDATION travel~validateCustomer IMPORTING keys FOR travel.

  METHODS validate_dates             FOR VALIDATION travel~validateDates    IMPORTING keys FOR travel.

  METHODS validate_travel_status     FOR VALIDATION travel~validateStatus   IMPORTING keys FOR travel.


  METHODS set_status_completed       FOR MODIFY IMPORTING   keys FOR ACTION travel~acceptTravel
RESULT result.

  METHODS get_features               FOR FEATURES IMPORTING keys REQUEST    requested_features FOR travel
RESULT result.
```

```
ENDCLASS.


CLASS lhc_travel IMPLEMENTATION.


**********************************************************************
*
* Validate customer data when saving travel data
*
**********************************************************************
METHOD validate_customer.


READ ENTITY zi_travel_m_xxx\\travel FROM VALUE #(
    FOR <root_key> IN keys ( %key     = <root_key>
                             %control = VALUE #( customer_id = if_abap_behv=>mk-on ) ) )
    RESULT DATA(lt_travel).
```

```
  DATA lt_customer TYPE SORTED TABLE OF /dmo/customer WITH UNIQUE KEY customer_id.


  " Optimization of DB select: extract distinct non-initial customer IDs

  lt_customer = CORRESPONDING #( lt_travel DISCARDING DUPLICATES MAPPING customer_id = customer_id
EXCEPT * ).

  DELETE lt_customer WHERE customer_id IS INITIAL.

  CHECK lt_customer IS NOT INITIAL.


  " Check if customer ID exist

  SELECT FROM /dmo/customer FIELDS customer_id

    FOR ALL ENTRIES IN @lt_customer

    WHERE customer_id = @lt_customer-customer_id

    INTO TABLE @DATA(lt_customer_db).


  " Raise msg for non existing customer id

  LOOP AT lt_travel INTO DATA(ls_travel).

    IF ls_travel-customer_id IS NOT INITIAL AND NOT line_exists( lt_customer_db[ customer_id = ls_travel-customer_id ]
).
```

```
      APPEND VALUE #(  travel_id = ls_travel-travel_id ) TO failed.

      APPEND VALUE #(  travel_id = ls_travel-travel_id

                       %msg      = new_message( id       = /dmo/cx_flight_legacy=>customer_unkown-msgid

                                                number   = /dmo/cx_flight_legacy=>customer_unkown-msgno

                                                v1       = ls_travel-customer_id

                                                severity = if_abap_behv_message=>severity-error )

                       %element-customer_id = if_abap_behv=>mk-on ) TO reported.

    ENDIF.


  ENDLOOP.


ENDMETHOD.



**********************************************************************

*

* Check validity of date
```

```
*
**********************************************************************
METHOD validate_dates.

  READ ENTITY zi_travel_m_xxx\\travel FROM VALUE #(
      FOR <root_key> IN keys ( %key     = <root_key>
                               %control = VALUE #( begin_date = if_abap_behv=>mk-on
                                                   end_date   = if_abap_behv=>mk-on ) ) )
      RESULT DATA(lt_travel_result).

  LOOP AT lt_travel_result INTO DATA(ls_travel_result).

    IF ls_travel_result-end_date < ls_travel_result-begin_date.  "end_date before begin_date

      APPEND VALUE #( %key        = ls_travel_result-%key
                      travel_id   = ls_travel_result-travel_id ) TO failed.
  APPEND VALUE #( %key     = ls_travel_result-%key
                %msg     = new_message( id       = /dmo/cx_flight_legacy=>end_date_before_begin_date-msgid
                                        number   = /dmo/cx_flight_legacy=>end_date_before_begin_date-msgno
                                        v1       = ls_travel_result-begin_date
                                        v2       = ls_travel_result-end_date
                                        v3       = ls_travel_result-travel_id
                                        severity = if_abap_behv_message=>severity-error )
                %element-begin_date = if_abap_behv=>mk-on
                %element-end_date   = if_abap_behv=>mk-on ) TO reported.

ELSEIF ls_travel_result-begin_date < cl_abap_context_info=>get_system_date( ).  "begin_date must be in the future

  APPEND VALUE #( %key        = ls_travel_result-%key
                travel_id   = ls_travel_result-travel_id ) TO failed.

  APPEND VALUE #( %key = ls_travel_result-%key
                %msg = new_message( id       = /dmo/cx_flight_legacy=>begin_date_before_system_date-msgid
                                    number   = /dmo/cx_flight_legacy=>begin_date_before_system_date-msgno
                                    severity = if_abap_behv_message=>severity-error )
                    %element-begin_date = if_abap_behv=>mk-on
                    %element-end_date   = if_abap_behv=>mk-on ) TO reported.
   ENDIF.

  ENDLOOP.

ENDMETHOD.

**********************************************************************
*
* Validate travel status when saving travel data
*
**********************************************************************
METHOD validate_travel_status.
```

```
READ ENTITY zi_travel_m_xxx\\travel FROM VALUE #(
   FOR <root_key> IN keys ( %key     = <root_key>
                            %control = VALUE #( overall_status = if_abap_behv=>mk-on ) ) )
   RESULT DATA(lt_travel_result).


LOOP AT lt_travel_result INTO DATA(ls_travel_result).
   CASE ls_travel_result-overall_status.
     WHEN 'O'.  " Open
     WHEN 'X'.  " Cancelled or rejected
     WHEN 'A'.  " Accepted


     WHEN OTHERS.
       APPEND VALUE #( %key = ls_travel_result-%key ) TO failed.


       APPEND VALUE #( %key = ls_travel_result-%key
                       %msg = new_message( id       = /dmo/cx_flight_legacy=>status_is_not_valid-msgid
                                           number   = /dmo/cx_flight_legacy=>status_is_not_valid-msgno
```

```
                                            v1       = ls_travel_result-overall_status
                                            severity = if_abap_behv_message=>severity-error )
                        %element-overall_status = if_abap_behv=>mk-on ) TO reported.
    ENDCASE.

  ENDLOOP.

ENDMETHOD.



**********************************************************************
*
* Implements travel action (in our case: for setting travel overall_status to completed)
*
**********************************************************************
METHOD set_status_completed.
```

```
  " Modify in local mode: BO-related updates that are not relevant for authorization checks
  MODIFY ENTITIES OF zi_travel_m_xxx IN LOCAL MODE
         ENTITY travel
            UPDATE FROM VALUE #( for key in keys ( travel_id = key-travel_id
                                                   overall_status = 'A' " Accepted
                                                   %control-overall_status = if_abap_behv=>mk-on ) )
         FAILED   failed
         REPORTED reported.

ENDMETHOD.

*********************************************************************************************
*
* Implements the dynamic feature handling for travel instances
*
*********************************************************************************************
METHOD get_features.
```

```
  READ ENTITY zi_travel_m_xxx FROM VALUE #( FOR keyval IN keys
                                            (  %key                   = keyval-%key
                                               %control-travel_id      = if_abap_behv=>mk-on
                                               %control-overall_status = if_abap_behv=>mk-on ) )
                            RESULT DATA(lt_travel_result).



  result = VALUE #( FOR ls_travel IN lt_travel_result
                   ( %key                         = ls_travel-%key
                     %features-%action-acceptTravel = COND #( WHEN ls_travel-overall_status = 'A'
                                                  THEN if_abap_behv=>fc-o-disabled ELSE if_abap_behv=>fc-o-enabled   )
                  ) ).


ENDMETHOD.
```

```
ENDCLASS.
```

Click Save (Ctrl-S) and Activate (Ctrl-F3).

Replace XXX with the name which you had choosed earlier.

An example :

```
*"* use this source file for the definition and implementation of
*"* local helper classes, interface definitions and type
*"* declarations
CLASS lhc_travel DEFINITION INHERITING FROM cl_abap_behavior_handler.

PRIVATE SECTION.

  TYPES tt_travel_update TYPE TABLE FOR UPDATE zi_travel_m_cvp9.

  METHODS validate_customer          FOR VALIDATION travel~validateCustomer IMPORTING keys FOR travel.
  METHODS validate_dates             FOR VALIDATION travel~validateDates    IMPORTING keys FOR travel.
  METHODS validate_travel_status     FOR VALIDATION travel~validateStatus   IMPORTING keys FOR travel.

  METHODS set_status_completed       FOR MODIFY IMPORTING   keys FOR ACTION travel~acceptTravel              RESULT
result.
  METHODS get_features               FOR FEATURES IMPORTING keys REQUEST    requested_features FOR travel    RESULT
result.

ENDCLASS.

CLASS lhc_travel IMPLEMENTATION.

**********************************************************************
*
* Validate customer data when saving travel data
*
**********************************************************************
METHOD validate_customer.

READ ENTITY zi_travel_m_cvp9\\travel FROM VALUE #(
      FOR <root_key> IN keys ( %key     = <root_key>
                               %control = VALUE #( customer_id = if_abap_behv=>mk-on ) ) )
      RESULT DATA(lt_travel).

  DATA lt_customer TYPE SORTED TABLE OF /dmo/customer WITH UNIQUE KEY customer_id.
  " Optimization of DB select: extract distinct non-initial customer IDs
  lt_customer = CORRESPONDING #( lt_travel DISCARDING DUPLICATES MAPPING customer_id = customer_id EXCEPT * ).
  DELETE lt_customer WHERE customer_id IS INITIAL.
  CHECK lt_customer IS NOT INITIAL.

  " Check if customer ID exist
  SELECT FROM /dmo/customer FIELDS customer_id
    FOR ALL ENTRIES IN @lt_customer
    WHERE customer_id = @lt_customer-customer_id
    INTO TABLE @DATA(lt_customer_db).

  " Raise msg for non existing customer id
  LOOP AT lt_travel INTO DATA(ls_travel).
    IF ls_travel-customer_id IS NOT INITIAL AND NOT line_exists( lt_customer_db[ customer_id = ls_travel-customer_id
] ).
      APPEND VALUE #(  travel_id = ls_travel-travel_id ) TO failed.
      APPEND VALUE #(  travel_id = ls_travel-travel_id
                       %msg      = new_message( id       = /dmo/cx_flight_legacy=>customer_unkown-msgid
                                                number   = /dmo/cx_flight_legacy=>customer_unkown-msgno
                                                v1       = ls_travel-customer_id
                                                severity = if_abap_behv_message=>severity-error )
                       %element-customer_id = if_abap_behv=>mk-on ) TO reported.
    ENDIF.

  ENDLOOP.

ENDMETHOD.


**********************************************************************
*
* Check validity of date
*
**********************************************************************
METHOD validate_dates.

  READ ENTITY zi_travel_m_cvp9\\travel FROM VALUE #(
      FOR <root_key> IN keys ( %key     = <root_key>
                               %control = VALUE #( begin_date = if_abap_behv=>mk-on
                                                   end_date   = if_abap_behv=>mk-on ) ) )
      RESULT DATA(lt_travel_result).

  LOOP AT lt_travel_result INTO DATA(ls_travel_result).

    IF ls_travel_result-end_date < ls_travel_result-begin_date.  "end_date before begin_date

      APPEND VALUE #( %key        = ls_travel_result-%key
                      travel_id   = ls_travel_result-travel_id ) TO failed.

      APPEND VALUE #( %key     = ls_travel_result-%key
                      %msg     = new_message( id       = /dmo/cx_flight_legacy=>end_date_before_begin_date-msgid
                                              number   = /dmo/cx_flight_legacy=>end_date_before_begin_date-msgno
                                              v1       = ls_travel_result-begin_date
                                              v2       = ls_travel_result-end_date
                                              v3       = ls_travel_result-travel_id
                                              severity = if_abap_behv_message=>severity-error )
                      %element-begin_date = if_abap_behv=>mk-on
                      %element-end_date   = if_abap_behv=>mk-on ) TO reported.

    ELSEIF ls_travel_result-begin_date < cl_abap_context_info=>get_system_date( ).  "begin_date must be in the future

      APPEND VALUE #( %key        = ls_travel_result-%key
                      travel_id   = ls_travel_result-travel_id ) TO failed.

      APPEND VALUE #( %key = ls_travel_result-%key
                      %msg = new_message( id       = /dmo/cx_flight_legacy=>begin_date_before_system_date-msgid
                                          number   = /dmo/cx_flight_legacy=>begin_date_before_system_date-msgno
                                          severity = if_abap_behv_message=>severity-error )
                      %element-begin_date = if_abap_behv=>mk-on
                      %element-end_date   = if_abap_behv=>mk-on ) TO reported.
    ENDIF.

  ENDLOOP.

ENDMETHOD.

**********************************************************************
```

```
*
* Validate travel status when saving travel data
*
**********************************************************************
METHOD validate_travel_status.

 READ ENTITY zi_travel_m_cvp9\\travel FROM VALUE #(
    FOR <root_key> IN keys ( %key     = <root_key>
                             %control = VALUE #( overall_status = if_abap_behv=>mk-on ) ) )
    RESULT DATA(lt_travel_result).

 LOOP AT lt_travel_result INTO DATA(ls_travel_result).
    CASE ls_travel_result-overall_status.
      WHEN 'O'.  " Open
      WHEN 'X'.  " Cancelled or rejected
      WHEN 'A'.  " Accepted

      WHEN OTHERS.
        APPEND VALUE #( %key = ls_travel_result-%key ) TO failed.

        APPEND VALUE #( %key = ls_travel_result-%key
                        %msg = new_message( id       = /dmo/cx_flight_legacy=>status_is_not_valid-msgid
                                            number   = /dmo/cx_flight_legacy=>status_is_not_valid-msgno
                                            v1       = ls_travel_result-overall_status
                                            severity = if_abap_behv_message=>severity-error )
                        %element-overall_status = if_abap_behv=>mk-on ) TO reported.
    ENDCASE.

  ENDLOOP.

ENDMETHOD.


********************************************************************************
*
* Implements travel action (in our case: for setting travel overall_status to completed)
*
********************************************************************************
METHOD set_status_completed.
```

```
  " Modify in local mode: BO-related updates that are not relevant for authorization checks
  MODIFY ENTITIES OF zi_travel_m_cvp9 IN LOCAL MODE
         ENTITY travel
            UPDATE FROM VALUE #( for key in keys ( travel_id = key-travel_id
                                                    overall_status = 'A' " Accepted
                                                    %control-overall_status = if_abap_behv=>mk-on ) )
         FAILED   failed
         REPORTED reported.

ENDMETHOD.

********************************************************************************
*
* Implements the dynamic feature handling for travel instances
*
********************************************************************************
METHOD get_features.

  READ ENTITY zi_travel_m_cvp9 FROM VALUE #( FOR keyval IN keys
                                                     (  %key                    = keyval-%key
                                                        %control-travel_id      = if_abap_behv=>mk-on
                                                        %control-overall_status = if_abap_behv=>mk-on ) )
                              RESULT DATA(lt_travel_result).


  result = VALUE #( FOR ls_travel IN lt_travel_result
                      ( %key                           = ls_travel-%key
                        %features-%action-acceptTravel = COND #( WHEN ls_travel-overall_status = 'A'
                                                                    THEN if_abap_behv=>fc-o-disabled ELSE
if_abap_behv=>fc-o-enabled   )
                     ) ).

ENDMETHOD.


ENDCLASS.
```