

```
1 CLASS zcl_jt_auth_view DEFINITION
2 PUBLIC
3 FINAL
4 CREATE PUBLIC .
5
6 PUBLIC SECTION.
7     TYPES: ty_r_xubname TYPE RANGE OF xubname,
8            ty_r_agrname TYPE RANGE OF agr_name,
9            ty_r_value   TYPE RANGE OF xuvalue,
10           ty_r_auth    TYPE RANGE OF xuprofile.
11
12     TYPES: BEGIN OF ty_mytype,
13            uname       TYPE uname,
14            name_text   TYPE name_text,
15            prnam       TYPE agr_name,
16            title       TYPE agr_title,
17            END OF ty_mytype.
18
19     TYPES: BEGIN OF ty_auth,
20            uname       TYPE uname,
21            tcode      TYPE xupname,
22            object     TYPE xuobject,
23            field      TYPE xufield,
24            low        TYPE xuval,
25            END OF ty_auth.
26
27     TYPES: BEGIN OF ty_auth_users,
28            uname       TYPE uname,
29            object     TYPE xuobject,
30            ptext      TYPE xutext,
31            END OF ty_auth_users.
32
33     METHODS:
34         constructor
35         IMPORTING
36             i_bname TYPE ty_r_xubname
37             i_prnam TYPE ty_r_agrname
38             i_tcode TYPE seocpdname
39             i_mname TYPE seocpdname,
40         check_error
41         RETURNING
42             VALUE(r_error) TYPE bool.
43
44
45 PRIVATE SECTION.
46     METHODS:
47         user_roles,
48         user_profiles,
49         user_tcode_check,
50         tcode_check,
51         role_user_list,
52         profile_user_list,
53         display_alv
54         CHANGING
55             c_tab TYPE STANDARD TABLE,
56         set_layout,
57         set_variant,
58         rename_fields,
59         create_fcat
60         IMPORTING
61             i_tab TYPE STANDARD TABLE.
62
63     DATA: alv_grid TYPE REF TO cl_gui_alv_grid,
64           r_bname  TYPE ty_r_xubname,
```

```

65         r_pnam      TYPE ty_r_agrname,
66         v_tcode     TYPE seocpdname,
67         v_name      TYPE seocpdname,
68         v_error     TYPE flag,
69         s_layout    TYPE lvc_s_layo,
70         s_variant   TYPE disvariant,
71         t_fieldcat  TYPE lvc_t_fcat,
72         t_objects   TYPE STANDARD TABLE OF ty_auth,
73         t_users     TYPE STANDARD TABLE OF ty_auth_users,
74         t_out       TYPE STANDARD TABLE OF ty_mytype.
75
76 ENDCLASS.
77
78 CLASS zcl_jt_auth_view IMPLEMENTATION.
79
80 METHOD constructor.
81     me->r_bname = i_bname. " uživatelské jméno
82     me->r_pnam  = i_pnam.  " název role/profilu
83     me->v_tcode = i_tcode. " kód transakce
84     me->v_name  = i_mname. " typ metody
85
86     CALL METHOD me->(me->v_name). " volání metody
87 ENDMETHOD.
88
89 METHOD check_error.
90
91     r_error = xsdbool( me->v_error IS NOT INITIAL ).
92
93 ENDMETHOD.
94
95 METHOD display_alv.
96     me->set_layout( ). " nastavení layoutu
97     me->set_variant( ). " nastavení varianty
98     me->create_fcat( c_tab ).
99     IF c_tab IS NOT INITIAL.
100        " vytvoření gridu ALV
101        me->alv_grid = NEW #( cl_gui_container=>default_screen ).
102        " nastavení ALV gridu k zobrazení
103        me->alv_grid->set_table_for_first_display(
104            EXPORTING
105                is_variant      = me->s_variant
106                is_layout      = me->s_layout
107                i_save          = 'A'
108            CHANGING
109                it_outtab      = c_tab
110                it_fieldcatalog = me->t_fieldcat ).
111        ELSE.
112            me->v_error = 'X'.
113        ENDIF.
114 ENDMETHOD.
115
116 METHOD user_roles.
117
118     SELECT DISTINCT uname, name_text, usr~agr_name, text FROM agr_users AS usr
119         INNER JOIN agr_texts AS txt ON usr~agr_name = txt~agr_name
120         LEFT JOIN usr2l ON usr2l~bname = usr~uname
121         LEFT JOIN adrp ON usr2l~persnumber = adrp~persnumber
122         INTO TABLE @me->t_out
123         WHERE usr~uname IN @me->r_bname
124         AND txt~spras = @sy-langu
125         ORDER BY uname, usr~agr_name.
126
127     DELETE ADJACENT DUPLICATES FROM me->t_out COMPARING name_text prnam.
128     display_alv( CHANGING c_tab = me->t_out ).

```

```

129
130 ENDMETHOD.
131
132 METHOD user_profiles.
133
134     SELECT DISTINCT ust04~bname, name_text, profile, ptext FROM ust04
135         INNER JOIN usrl1 ON usrl1~profn = ust04~profile AND langu = @sy-langu
136         LEFT JOIN usr21 ON usr21~bname = ust04~bname
137         LEFT JOIN adrp ON usr21~persnumber = adrp~persnumber
138     INTO TABLE @me->t_out
139     WHERE ust04~bname IN @me->r_bname
140     ORDER BY ust04~bname, profile.
141
142     DELETE ADJACENT DUPLICATES FROM me->t_out COMPARING name_text prnam.
143     display_alv( CHANGING c_tab = me->t_out ).
144
145 ENDMETHOD.
146
147 METHOD user_tcode_check.
148     DATA lt_usobt TYPE STANDARD TABLE OF ty_auth.
149
150     LOOP AT me->r_bname INTO DATA(wa_bname).
151         SELECT name AS tcode, object, field, low FROM usobt
152             INTO CORRESPONDING FIELDS OF TABLE @lt_usobt
153             WHERE name = @me->v_tcode
154             AND low <> ''
155             AND type = 'TR'.
156         SELECT ust04~bname, profile FROM ust04
157             INTO TABLE @DATA(lt_profiles)
158             WHERE ust04~bname = @wa_bname-low.
159         IF lt_profiles IS NOT INITIAL.
160             LOOP AT lt_profiles ASSIGNING FIELD-SYMBOL(<row>).
161                 <row>-profile = |{ <row>-profile }00|.
162             ENDLOOP.
163             LOOP AT lt_usobt INTO DATA(wa_usobt).
164                 SELECT COUNT( * ) FROM ust12
165                     FOR ALL ENTRIES IN @lt_profiles
166                     WHERE objct = @wa_usobt-object
167                     AND field = @wa_usobt-field
168                     AND ( von = @wa_usobt-low OR von = '*' )
169                     AND auth = @lt_profiles-profile.
170                 IF sy-subrc = 0.
171                     DELETE lt_usobt.
172                 ENDIF.
173             ENDLOOP.
174         ENDIF.
175
176         LOOP AT lt_usobt ASSIGNING FIELD-SYMBOL(<name>).
177             <name>-uname = wa_bname-low.
178             APPEND <name> TO me->t_objects.
179         ENDLOOP.
180     ENDLOOP.
181     IF me->t_objects IS NOT INITIAL.
182         display_alv( CHANGING c_tab = me->t_objects ).
183     ELSEIF lt_profiles IS NOT INITIAL.
184         MESSAGE i002(zhp_auth) DISPLAY LIKE 'S'.
185         me->v_error = 'X'.
186     ENDIF.
187 ENDMETHOD.
188
189 METHOD tcode_check.
190     DATA: r_values TYPE ty_r_value,
191           wa_values LIKE LINE OF r_values,
192           lt_auth TYPE STANDARD TABLE OF auth,

```

```

193         r_auth      TYPE ty_r_auth,
194         wa_auth     LIKE LINE OF r_auth,
195         len         TYPE i.
196
197     SELECT object, field, low FROM usobt
198     INTO TABLE @DATA(lt_usobt)
199     WHERE name = @me->v_tcode
200     AND low <> ''.
201
202     LOOP AT lt_usobt INTO DATA(wa_usobt).
203     CLEAR: wa_values, r_values.
204     wa_values-sign = 'I'.
205     wa_values-option = 'EQ'.
206     wa_values-low = wa_usobt-low.
207     APPEND wa_values TO r_values.
208
209     SELECT auth FROM ust12
210     INTO TABLE @DATA(lt_auth_temp)
211     WHERE objct = @wa_usobt-object
212     AND field = @wa_usobt-field
213     AND ( von IN @r_values OR von = '*' ).
214
215     SORT lt_auth_temp BY auth.
216
217     IF lt_auth IS INITIAL.
218     APPEND LINES OF lt_auth_temp TO lt_auth.
219     ELSE.
220     LOOP AT lt_auth INTO DATA(temp).
221     READ TABLE lt_auth_temp INTO DATA(temp2) WITH KEY auth = temp BINARY
SEARCH.
222     IF sy-subrc <> 0.
223     DELETE lt_auth.
224     ENDIF.
225     ENDLOOP.
226     ENDIF.
227
228     ENDLOOP.
229
230     SORT lt_auth.
231     DELETE ADJACENT DUPLICATES FROM lt_auth.
232
233     LOOP AT lt_auth ASSIGNING FIELD-SYMBOL(<row>).
234     wa_auth-sign = 'I'.
235     wa_auth-option = 'EQ'.
236     IF <row>(3) = 'T-H'.
237     len = strlen( <row> ).
238     len = len - 2.
239     wa_auth-low = <row>(len).
240     ELSEIF <row>(2) = '&_'.
241     wa_auth-low = <row>+2.
242     ELSE.
243     wa_auth-low = <row>.
244     ENDIF.
245     APPEND wa_auth TO r_auth.
246     ENDLOOP.
247
248     SELECT bname AS uname, profile AS object, ptext FROM ust04
249     LEFT JOIN usr11 ON usr11~profn = ust04~profile AND langu = @sy-langu
250     INTO CORRESPONDING FIELDS OF TABLE @me->t_users
251     WHERE profile IN @r_auth.
252
253     SORT me->t_users BY uname.
254     IF me->t_users IS NOT INITIAL.
255     display_alv( CHANGING c_tab = me->t_users ).

```

```

256     ELSE.
257         me->v_error = 'X'.
258     ENDIF.
259
260     ENDMETHOD.
261
262     METHOD role_user_list.
263
264     SELECT DISTINCT uname, name_text, usr~agr_name, text FROM agr_users AS usr
265         INNER JOIN agr_texts AS txt ON usr~agr_name = txt~agr_name
266         LEFT JOIN usr21 ON usr21~bname = usr~uname
267         LEFT JOIN adrp ON usr21~persnumber = adrp~persnumber
268         INTO TABLE @me->t_out
269         WHERE usr~agr_name IN @me->r_pnam
270         AND txt~spras = @sy-langu
271         ORDER BY uname, usr~agr_name.
272
273     DELETE ADJACENT DUPLICATES FROM me->t_out COMPARING name_text pnam.
274     display_alv( CHANGING c_tab = me->t_out ).
275
276     ENDMETHOD.
277
278     METHOD profile_user_list.
279
280     SELECT DISTINCT ust04~bname, adrp~name_text, profile, ptext FROM ust04
281         INNER JOIN usr11 ON usr11~profn = ust04~profile AND langu = @sy-langu
282         LEFT JOIN usr21 ON usr21~bname = ust04~bname
283         LEFT JOIN adrp ON usr21~persnumber = adrp~persnumber
284         INTO TABLE @me->t_out
285         WHERE profile IN @me->r_pnam
286         ORDER BY ust04~bname, profile.
287
288     DELETE ADJACENT DUPLICATES FROM me->t_out COMPARING name_text pnam.
289     display_alv( CHANGING c_tab = me->t_out ).
290
291     ENDMETHOD.
292
293     METHOD set_layout.
294     me->s_layout-cwidth_opt = 'X'.
295     me->s_layout-zebra = 'X'.
296     ENDMETHOD.
297
298
299     METHOD set_variant.
300     me->s_variant-report = sy-repid.
301     me->s_variant-variant = |/{ me->v_name }|.
302     ENDMETHOD.
303
304     METHOD create_fcat.
305     DATA: lr_tbdscr TYPE REF TO cl_abap_structdescr,
306         lr_data TYPE REF TO data,
307         lt_dfies TYPE ddfields,
308         ls_dfies TYPE dfies,
309         ls_fcat TYPE lvc_s_fcat.
310
311     CREATE DATA lr_data LIKE LINE OF i_tab.
312
313     lr_tbdscr ?= cl_abap_structdescr=>describe_by_data_ref( lr_data ).
314     lt_dfies = cl_salv_data_descr=>read_structdescr( lr_tbdscr ).
315
316     LOOP AT lt_dfies INTO ls_dfies.
317         CLEAR ls_fcat.
318         MOVE-CORRESPONDING ls_dfies TO ls_fcat.
319         APPEND ls_fcat TO me->t_fieldcat.

```

```
320     ENDMETHOD.  
321  
322     DELETE me->t_fieldcat WHERE fieldname = 'MANDT'.  
323     rename_fields( ).  
324 ENDMETHOD.  
325  
326 METHOD rename_fields.  
327  
328     LOOP AT me->t_fieldcat ASSIGNING FIELD-SYMBOL(<line>).  
329         IF <line>-fieldname = 'PRNAM'.  
330             <line>-reptext = 'Název'.  
331         ENDIF.  
332         IF <line>-fieldname = 'TITLE'.  
333             <line>-reptext = 'Označení'.  
334         ENDIF.  
335     ENDMETHOD.  
336  
337 ENDMETHOD.  
338  
339 ENDMETHOD.
```