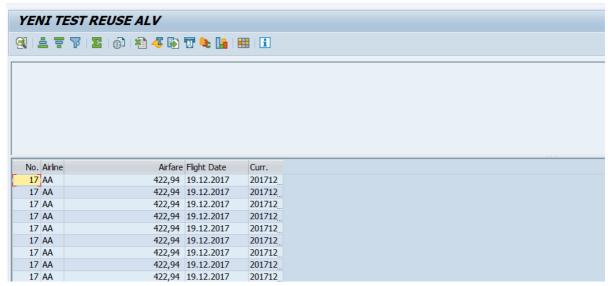
ReuseALV | Callback, Events, Reuse Alv Events Get

ReuseALV Callback, ALV tablosunda bir satırın değiştirildiği, silindiği veya seçildiği gibi olaylar gerçekleştikçe otomatik olarak tetiklenen özel bir işlevdir. ReuseALV Callback işlevi, kullanıcının tanımlayabileceği bir ABAP fonksiyon modülüdür. Bu modül, ALV tablosu olaylarını dinler ve olaylar gerçekleştiğinde çağrılır. Events, ReuseALV Callback fonksiyonunu çağıran ve ALV tablosundaki belirli bir olay gerçekleştiğinde tetiklenen özel bir işlevdir. Örneğin, bir satır seçildiğinde veya bir sütun sıralandığında bir olay tetiklenebilir. Bu olaylar, ALV tablosu işlevlerini genişletmek ve özelleştirmek için kullanılabilir.

ReuseALV Events Get, ALV tablosundaki olayların kontrolünü sağlamak için kullanılan bir işlevdir. Bu işlev, hangi olayların tetikleneceğini ve bu olaylar gerçekleştiğinde hangi işlevlerin çağrılacağını belirlemek için kullanılır. ReuseALV Events Get, kullanıcının ALV tablosu olaylarını özelleştirmesine izin verir ve böylece ALV tablosu işlevselliğini daha da geliştirir.

Ve görüldüğü gibi top alanında bir sayfa açmış, yani boşluk bırakmış oldu.



Burada önemli olan kısım I_CALLBACK_TOP_OF_PAGE kısmına karşılık gelen alan ve o alanda yazan isimle aynı isme sahip bir form oluşturulmasıdır. Eğer isimler aynı olursa problem yok fakat isimler farklı olursa bile yine de bu son görselde oluşan page çıkacaktır fakat veriler işin içine dahil olduğunda veya düzenlemeler dahil olduğunda bu sefer içerisine müdahale edemiyor oluyoruz çünkü I_CALLBACK_TOP_OF_PAGE kısmına ne atarsak atayalım bu alanı zaten bize açmış oluyor fakat sadece görünüşte kalmış olur.

Bu nedenle verilen isim ve form adı aynı olmalı!

EVENTS ALANI:

```
*" REFERENCE (I_SAVE) DEFAULT SPACE

*" REFERENCE (IS_VARIANT) LIKE DISVARIANT STRUCTURE DISVARIAN

*" OPTIONAL

*" REFERENCE (IT_EVENTS) TYPE SLIS_T_EVENT_OPTIONAL

*" REFERENCE (IT_EVENT_EXIT) TYPE SLIS_T_EVENT_EXIT OPTIONAL

*" REFERENCE (IS_PRINT) TYPE SLIS_PRINT_ALV OPTIONAL

*" REFERENCE (IS_REPREP_ID) TYPE SLIS_REPREP_ID OPTIONAL

data: gt_events type SLIS_T_EVENT,
    gs_event type

- end of slis_alv_event.

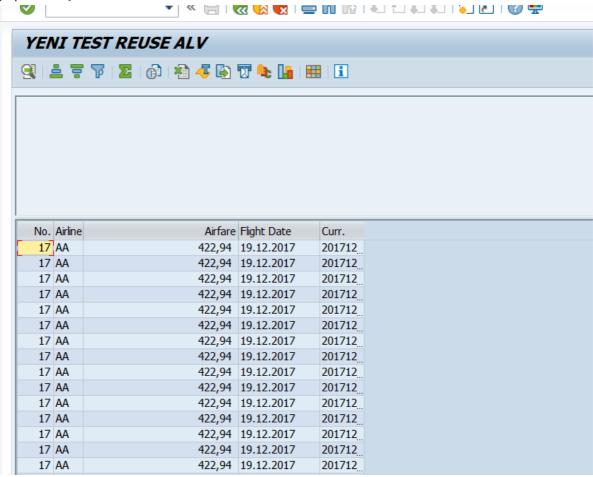
types: slis_t_event type slis_alv_event occurs 0.

*--- Structure for key information

types: begin of slis_keyinfo_alv,
    header01 type slis_fieldname.
```

Aşağıda görülen ekran görüntüsünde belirtilen alanlar developer edition'da yok, el ile yazmak gerekiyor. CTRL + space ile gelmiyor.

Sonuç olarak az önce callback tarafında yapmış olduğum şeyleri events yardımıyla da yapabilmiş olundu;



Bu method adı konusunda callback ile aynı duruma sahiptir, isimleri aynı olmalıdır. Yoksa events yapısı zaten bir form olacakmış diyerekten top of page açıyor ama formumuzla alakası olmuyor, isimler farklı olduğu sürece. İsimler aynıysa zaten herhangi bir problem yok.

Ve her iki yapıda da I_CALLBACK_PROGRAM aktif olmalı;

Bir diğer kullanım şekli;

```
☐ FORM display_alv .

gs_event-name = slis_ev_top_of_page.
gs_event-form = 'TOP_OF_PAGE'.

APPEND gs_event to gt_events.

gs_event-name = slis_ev_end_of_list.
gs_event-form = 'END_OF_LIST'.

APPEND gs event to gt events.

CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'

EXPORTING

** I_INTERFACE_CHECK = ''

** I_BYPASSING_BUFFER = ''

-* I_BUFFER_ACTIVE = ''

I_CALLBACK_PROGRAM = sy-repid

** I_CALLBACK_PF_STATUS_SET = ''

** I_CALLBACK_USER_COMMAND = ''

** I_CALLBACK_TOP_OF_PAGE = 'TOP_OF_PAGE'

** I_CALLBACK_HTML_TOP_OF_PAGE = 'TOP_OF_PAGE'

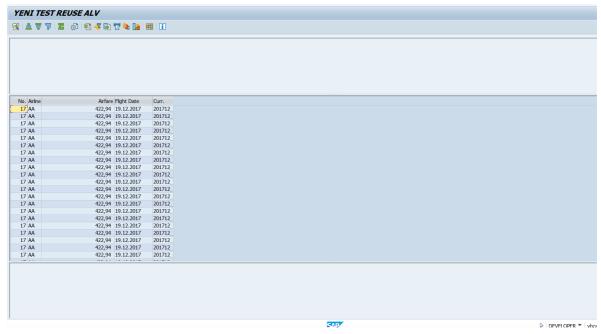
** I_CALLBACK_HTML_TOP_OF_PAGE = ''

I_CALLBACK_HTML_TOP_OF_PAGE = ''

** I_CALLBACK_HTML_TOP_OF_PAGE = ''

** I_CALLBACK_HTML_TOP_OF_PAGE = ''
```

Bu da alt tarafta alan yaratır;



Reuse ALV Events Get

