

**This is a Practice Based Working Document. Everyone can change by their own requirements.**

### **Scenario:**

The scenario is when a Purchase Order Document creates a workflow will send an approval request email to the approver. That email will be received in Gmail Inbox or any other email server.

The approver will open that email in any device by using Gmail or mail server app or website. He will have two options Approve or Reject the purchase order.

He will just click on approve/reject button in email and the effect will happen in SAP Inbox.

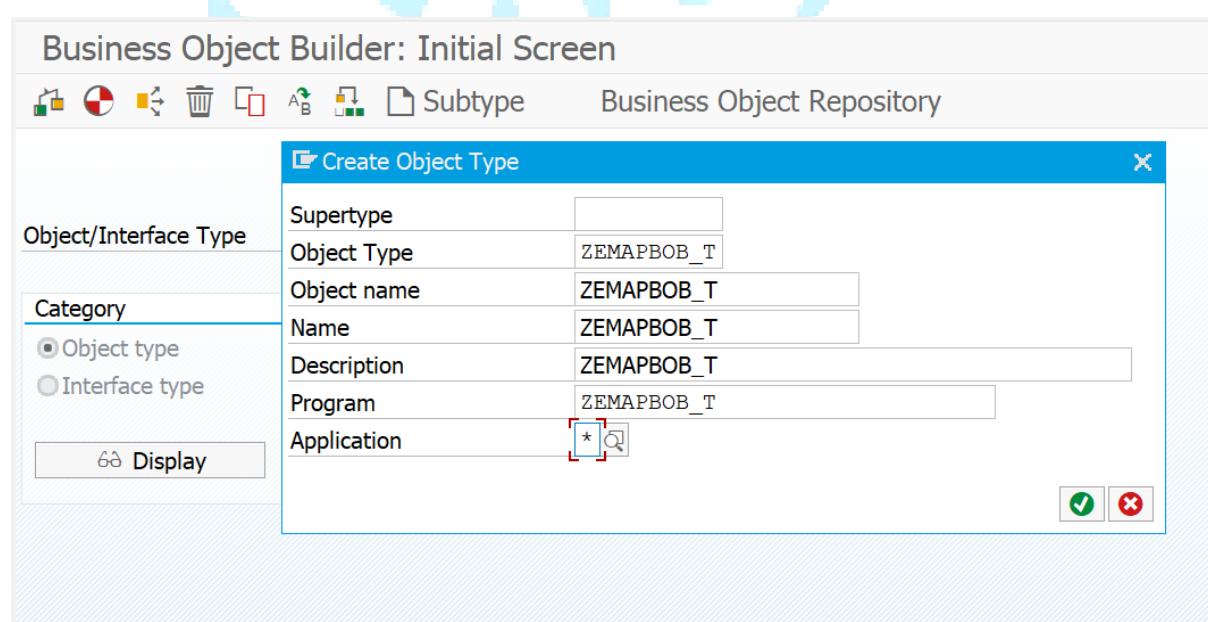
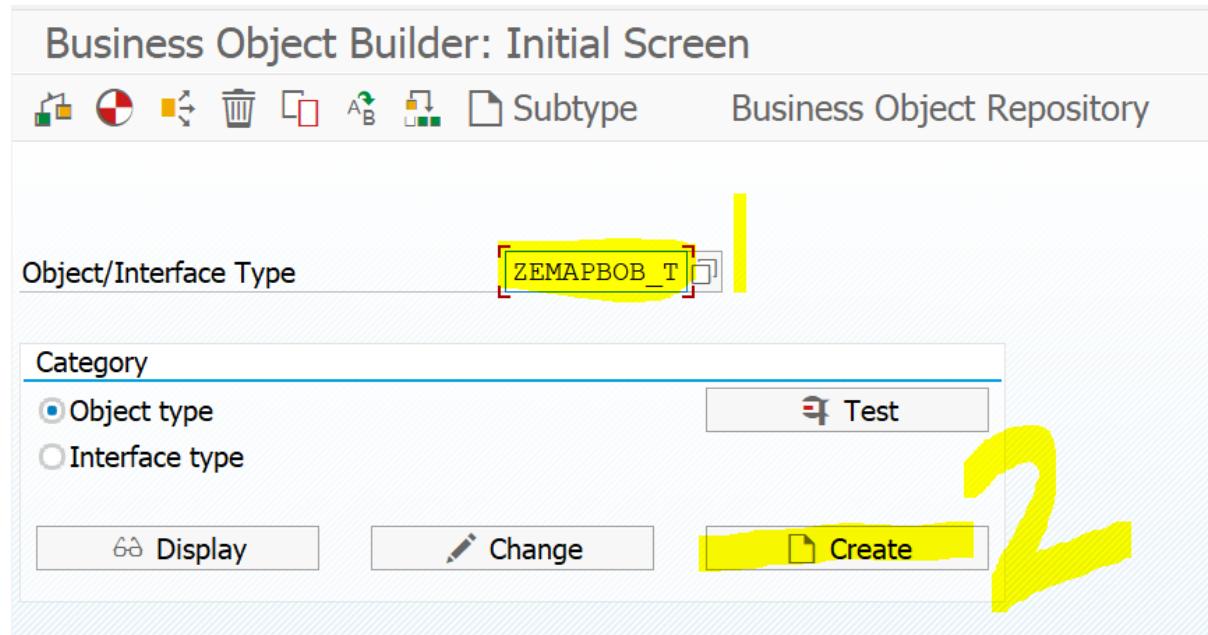
The Purchase Order will be approved or rejected directly in SAP.

### **Transaction Codes Used:**

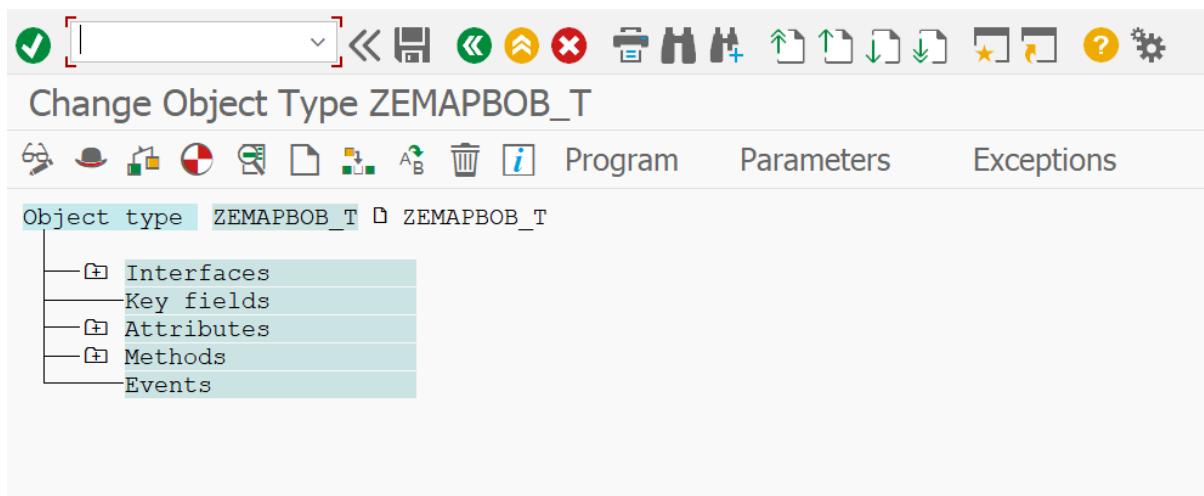
1. SWO1 – To create business object.
2. SE37 – To create Function Module for Email Body
3. PFTC – To Create Workflow Task and Template
4. SEGW – Create ODATA
5. SOST – To Check Emails in Inbox

**Go to SW01:**

Create a business object by follow these the steps.

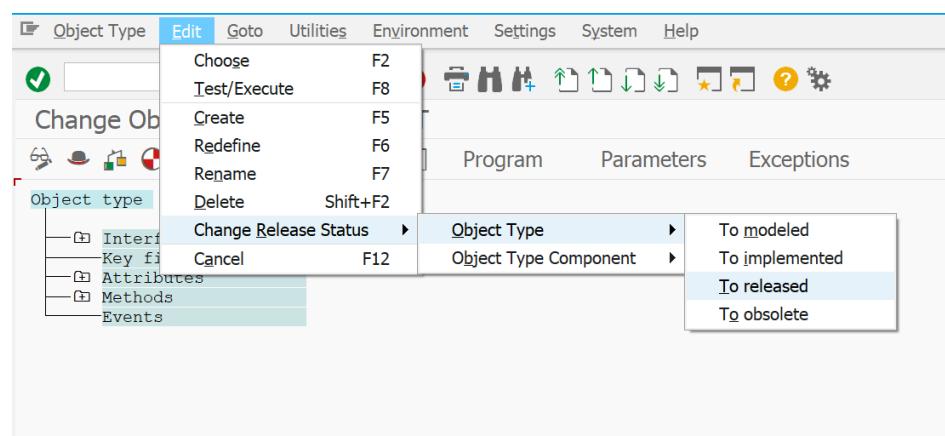
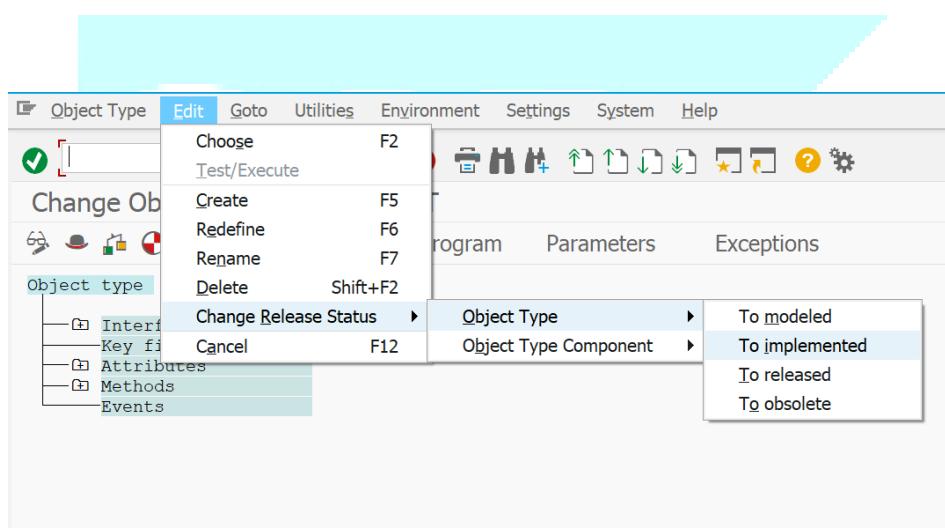
**Step 1:**

The Business Object is created. Now we have to arrange it according to the need.



## Step 2:

Must Implement and Release the Business object to make it available for use.



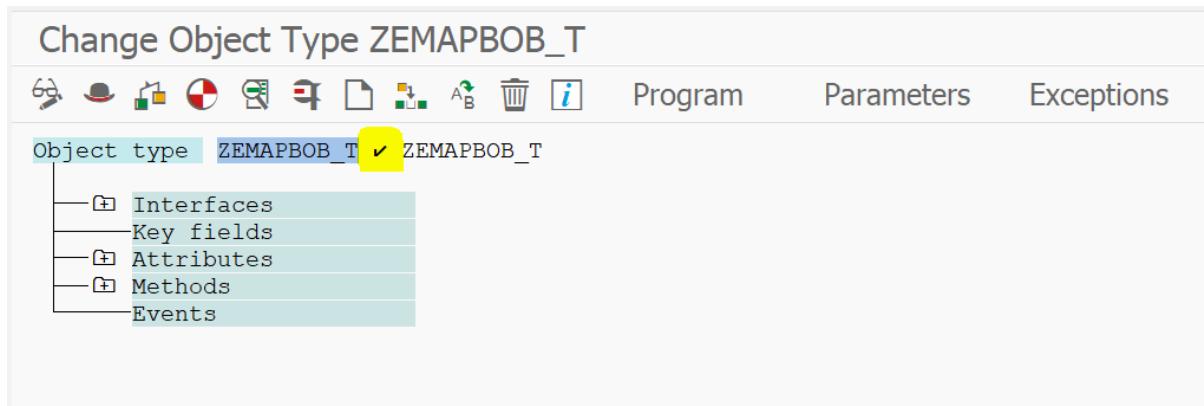
Now, it is available to use for Workflow.

Change Object Type ZEMAPBOB\_T

Object type ZEMAPBOB\_T ✓ ZEMAPBOB\_T

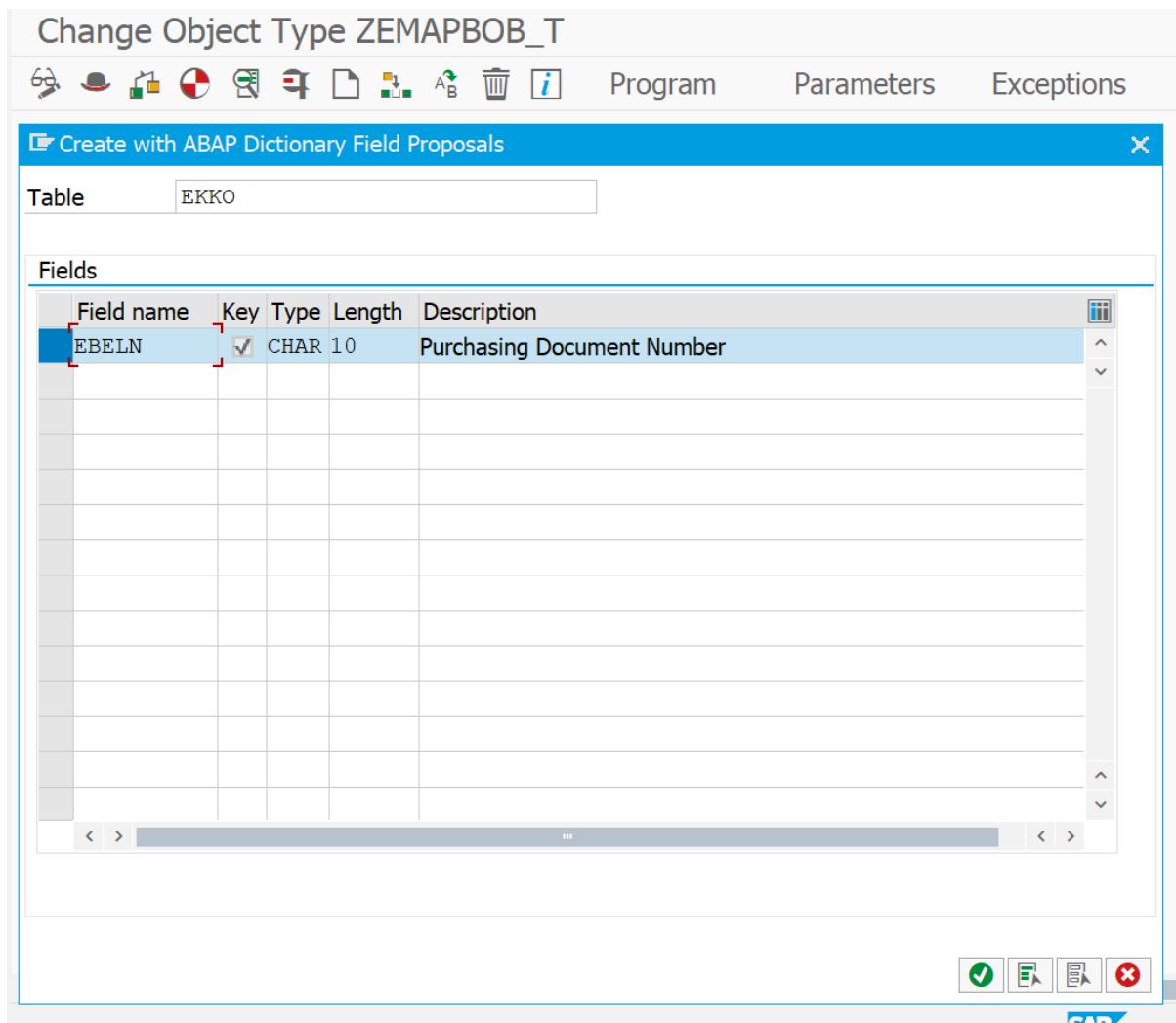
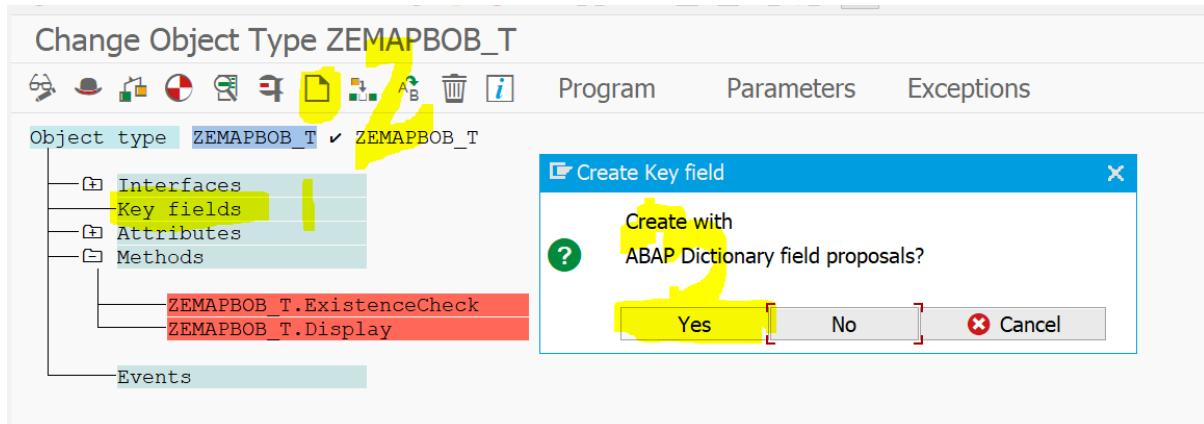
Program Parameters Exceptions

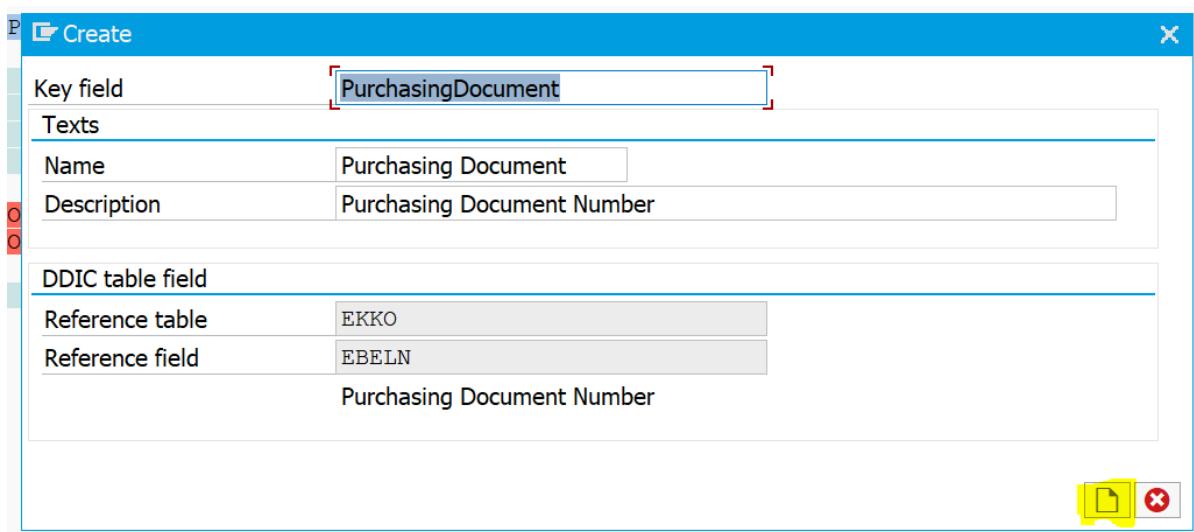
Interfaces Key fields Attributes Methods Events



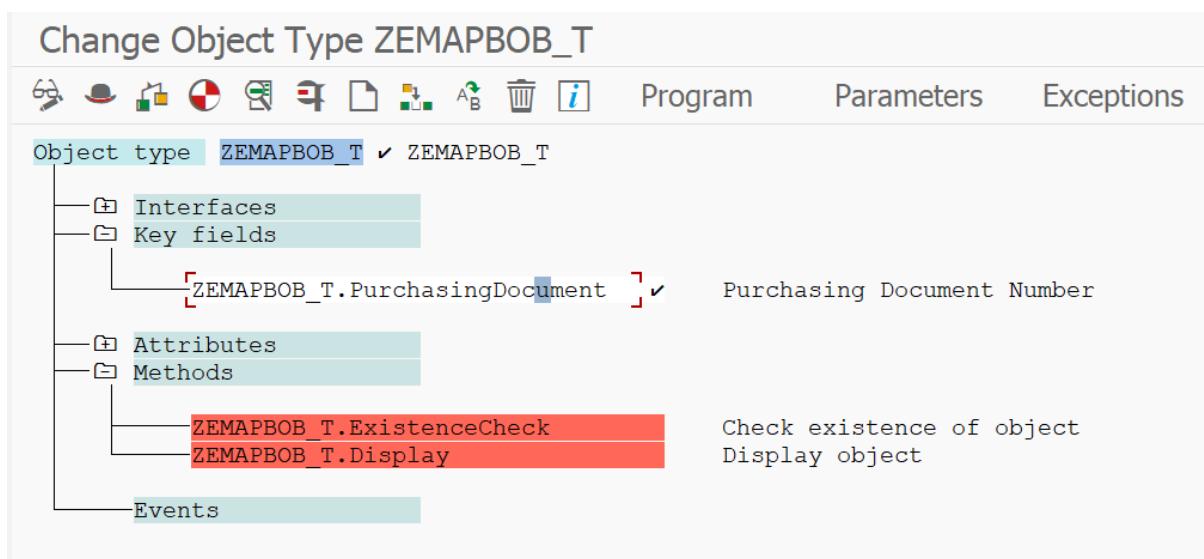
### Step 3:

We must create Key Field that is Purchase Order Number in Our case.



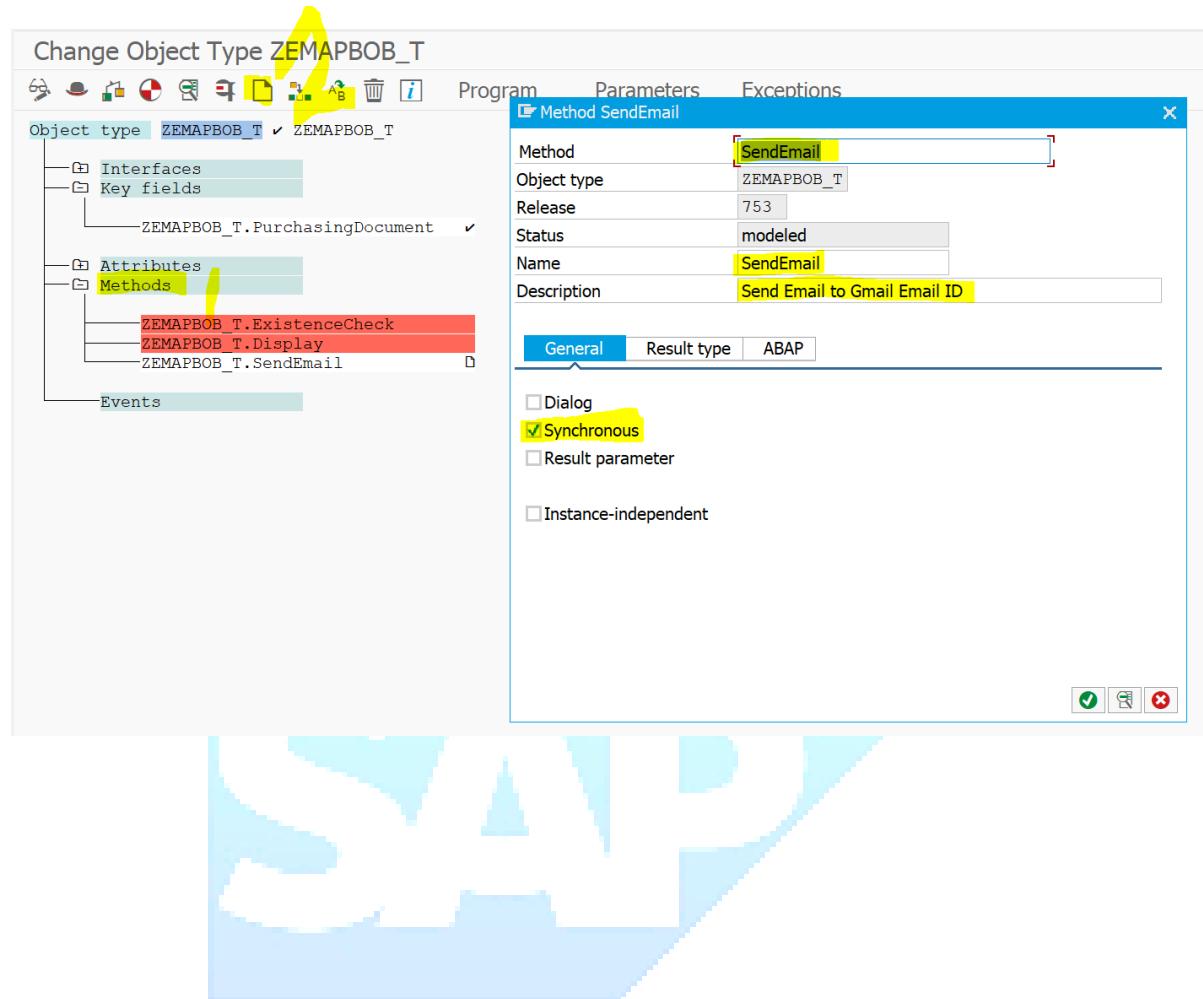


The Key Field is created successfully.



## Step 4:

We have to create a method in this Business Object to send Email to the recipient.



Method SendEmail

Method	SendEmail
Object type	ZEMAPBOB_T
Release	753
Status	modeled
Name	SendEmail
Description	Send Email to Gmail Email ID

General    Result type    ABAP

ABAP Dictionary

Reference table  
Reference field  
Search Help

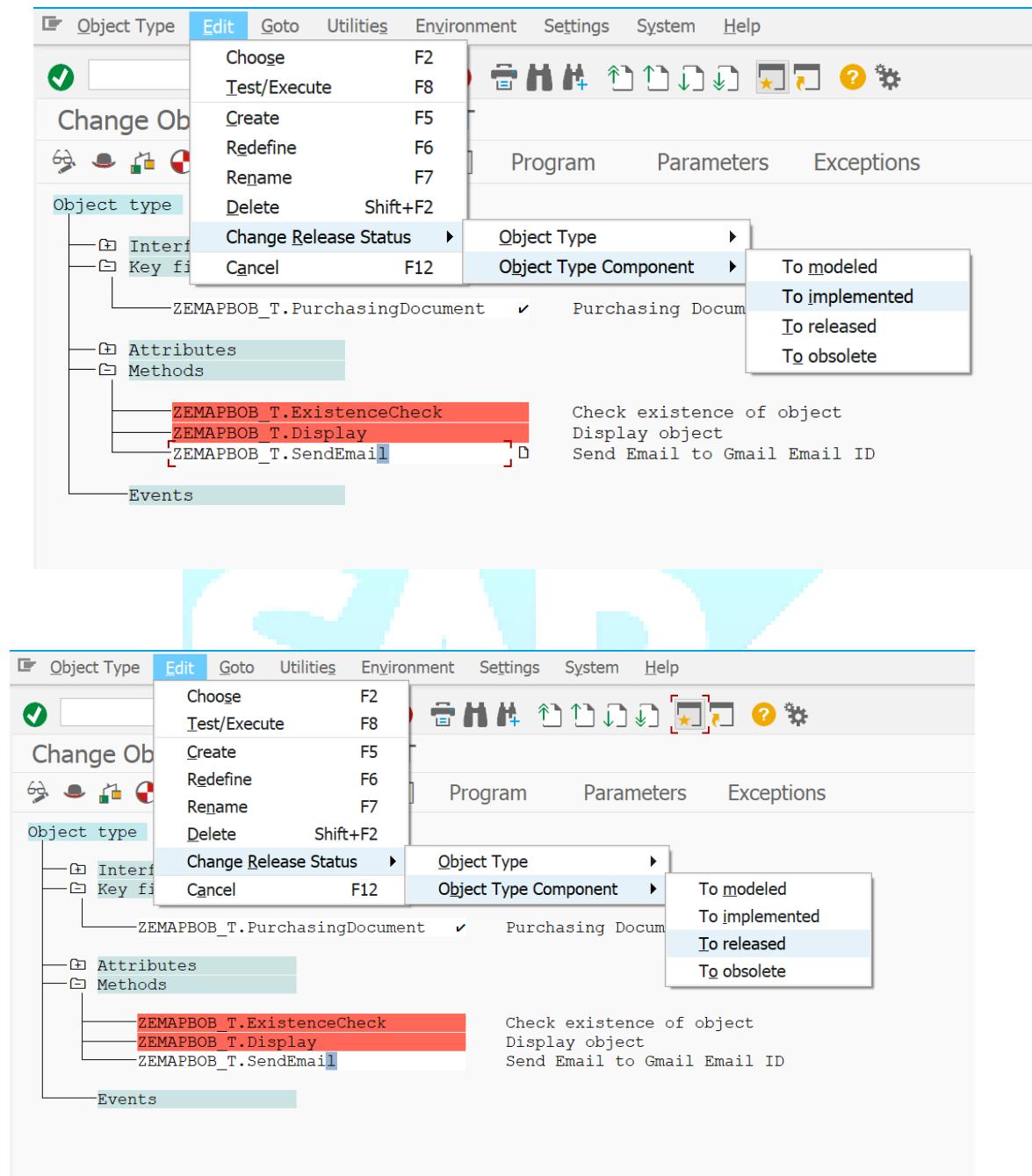
Object type    ZEMAPBOB\_T

Multiline

Note: Must step after creating any object in BO.

We must need to Implement and Release the Method to make it available for use in Workflow Tasks.



Method is now available to use.

Change Object Type ZEMAPBOB\_T

Object type ZEMAPBOB\_T ✓ ZEMAPBOB\_T

Interfaces

Key fields

ZEMAPBOB\_T.PurchasingDocument ✓ Purchasing Document Number

Attributes

Methods

ZEMAPBOB\_T.ExistenceCheck Check existence of object  
ZEMAPBOB\_T.Display Display object  
ZEMAPBOB\_T.SendEmail Send Email to Gmail Email ID

Events



## Step 5:

Create the parameters for the method that should be passed to that method during workflow running.

Object Type ZEMAPBOB\_T: Edit Parameters for Method SENDEMAIL

Other View      Program      Exceptions

Create with ABAP Dictionary Field Proposals

Table EKKO

Fields

Field name	Key	Type	Length	Description
MANDT	<input checked="" type="checkbox"/>	CLNT	3	Client
EBELN	<input checked="" type="checkbox"/>	CHAR	10	Purchasing Document Number
BUKRS	<input type="checkbox"/>	CHAR	4	Company Code
BSTYP	<input type="checkbox"/>	CHAR	1	Purchasing Document Category
BSART	<input type="checkbox"/>	CHAR	4	Purchasing Document Type
BSAKZ	<input type="checkbox"/>	CHAR	1	Control indicator for purchasing document type
LOEKZ	<input type="checkbox"/>	CHAR	1	Deletion indicator in purchasing document
STATU	<input type="checkbox"/>	CHAR	1	Status of Purchasing Document
AEDAT	<input type="checkbox"/>	DATS	8	Date on which the record was created
ERNAM	<input type="checkbox"/>	CHAR	12	Name of Person who Created the Object
LASTCHANG...	<input type="checkbox"/>	DEC	21	Change Time Stamp
PINCR	<input type="checkbox"/>	NUMC	5	Item Number Interval
LPONR	<input type="checkbox"/>	NUMC	5	Last Item Number
LIFNR	<input type="checkbox"/>	CHAR	10	Vendor's account number

Buttons: ✓, ⌂, ⌂, ✖

Object Type ZEMAPBOB\_T: Edit Parameters for Method SENDEMAIL

Other View      Program      Exceptions

Create

Parameter PurchasingDocument

Texts

Name	Purchasing Document
Description	Purchasing Document Number

Parameter attributes

<input checked="" type="checkbox"/> Import	<input type="checkbox"/> Mandatory
<input checked="" type="checkbox"/> Export	<input type="checkbox"/> Supplied from Key
<input type="checkbox"/> Multiline	

DDIC table field

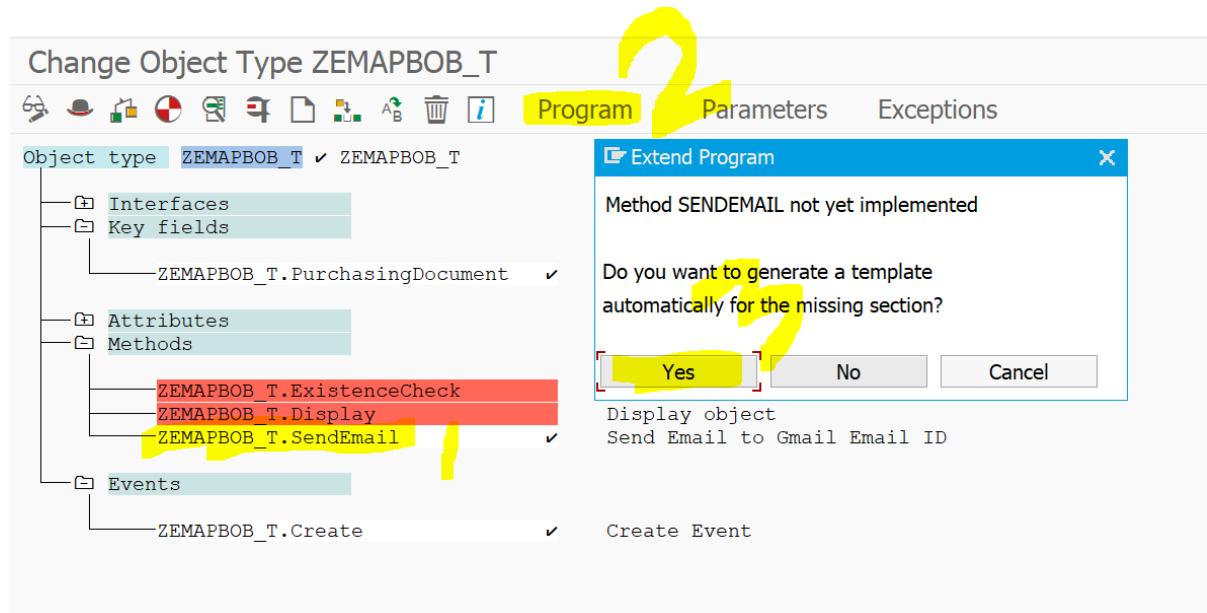
Reference table	EKKO
Reference field	EBELN
Purchasing Document Number	

Buttons: ⌂, ✖

## Step 6:

Now we have to write code in the created method to send Email.

**Note:** The code is written for practice only. You can write by your own way with the new ABAP syntax as well.



### Code for the Email Send:

```
BEGIN_METHOD SENDEMAIL CHANGING CONTAINER.  
DATA: PURCHASINGDOCUMENT TYPE EKKO-EBELN.  
SWC_GET_ELEMENT CONTAINER 'PurchasingDocument' PURCHASINGDOCUMENT.
```

CONSTANTS:

```
*-- Constants used in the body of the Email (HTML)  
C_HTM      TYPE CHAR3  VALUE 'HTM',  
C_STYLE_START TYPE CHAR255 VALUE '<FONT face=Arial size=2>',  
C_NEW_LINE  TYPE CHAR255 VALUE '<br>',  
C_LINK_START TYPE CHAR128 VALUE '<A href="www.w3schools.com">',  
C_LINK_TEXT  TYPE CHAR32  VALUE 'Link for learning HTML',  
C_LINK_END   TYPE CHAR4  VALUE '</A>',  
C_SPACE(6)   TYPE C    VALUE '&nbsp';
```

```
DATA: GT_CONT TYPE STANDARD TABLE OF BAPIASCONT255,
```

GW\_CONT LIKE LINE OF GT\_CONT.

```
DATA: SEND_REQUEST TYPE REF TO CL_BCS,
      MAILSUBJECT TYPE SO_OBJ_DES,
      MAILTEXT   TYPE BCSY_TEXT,
      DOCUMENT   TYPE REF TO CL_DOCUMENT_BCS,
      SENDER     TYPE REF TO CL_CAM_ADDRESS_BCS,
      RECIPIENT_TO TYPE REF TO CL_CAM_ADDRESS_BCS,
      RECIPIENT_CC TYPE REF TO CL_CAM_ADDRESS_BCS,
      RECIPIENT_BCC TYPE REF TO CL_CAM_ADDRESS_BCS,
      BCS_EXCEPTION TYPE REF TO CX_BCS.
```

TRY.

```
SEND_REQUEST = CL_BCS=>CREATE_PERSISTENT().
```

```
MAILSUBJECT = 'Release PO Notification'.
```

```
CALL FUNCTION 'ZSENDEMAIL_THROUGH_WORKFLOW'
```

```
EXPORTING
```

```
EBELN = PURCHASINGDOCUMENT
```

```
TABLES
```

```
GT_CONT = GT_CONT.
```

```
DOCUMENT = CL_DOCUMENT_BCS=>CREATE_DOCUMENT(
```

```
I_TYPE = C_HTM
```

```
I_TEXT = GT_CONT
```

```
I_LANGUAGE = SY-LANGU
```

```
I SUBJECT = MAILSUBJECT ).
```

```
SEND_REQUEST->SET_DOCUMENT( DOCUMENT ).
```

```
SENDER = CL_CAM_ADDRESS_BCS=>CREATE_INTERNET_ADDRESS( 'practice@gmail.com' ).
```

```
SEND_REQUEST->SET_SENDER( SENDER ).
```

```
RECIPIENT_TO = CL_CAM_ADDRESS_BCS=>CREATE_INTERNET_ADDRESS( 'practice@gmail.com' ).
```

```
SEND_REQUEST->ADD_RECIPIENT( I_RECIPIENT = RECIPIENT_TO ).
```

```
RECIPIENT_CC = CL_CAM_ADDRESS_BCS=>CREATE_INTERNET_ADDRESS( 'practice@gmail.com' ).
```

```

SEND_REQUEST->ADD_RECIPIENT( I_RECIPIENT = RECIPIENT_CC
I_COPY = 'X' ).

RECIPIENT_BCC = CL_CAM_ADDRESS_BCS=>CREATE_INTERNET_ADDRESS( 'practice@gmail.com' ).

SEND_REQUEST->ADD_RECIPIENT( I_RECIPIENT = RECIPIENT_BCC
I_BLIND_COPY = 'X' ).

DATA(LV_SENT_TO_ALL) = SEND_REQUEST->SEND( ).

IF LV_SENT_TO_ALL = 'X'.

*   WRITE 'Email sent to all recipients'.

ELSE.

*   WRITE 'Email could not be sent to all recipients!'.

ENDIF.

COMMIT WORK.

CATCH CX_BCS INTO BCS_EXCEPTION.

*   WRITE: 'Error occurred while sending email: Error Type', BCS_EXCEPTION->ERROR_TYPE.

ENDTRY.

SWC_SET_ELEMENT CONTAINER 'PurchasingDocument' PURCHASINGDOCUMENT.

END_METHOD.

```

## Function Module Code: ZSENDEMAIL\_THROUGH\_WORKFLOW

```

DATA: G_URL_FIOR1 TYPE TEXT200,
      G_URL_FIOR2 TYPE TEXT200,
      LV_WIID  TYPE C LENGTH 12,
      LV_PO    TYPE C LENGTH 25.

" READ SELECTED WORK ITEM ID
SELECT WI_ID FROM SWWIHEAD INTO TABLE @DATA(IT_WI_ID) WHERE TOP_TASK EQ 'WS90000135' AND WI_STAT EQ 'STARTED' AND WI_RH_TASK EQ 'TS90000200'.
SORT IT_WI_ID DESCENDING BY WI_ID.
READ TABLE IT_WI_ID INTO DATA(WA_WI_ID) INDEX 1.
WA_WI_ID-WI_ID = WA_WI_ID-WI_ID + 1.
WA_WI_ID-WI_ID = |{ WA_WI_ID-WI_ID ALPHA = IN }|.
MOVE WA_WI_ID-WI_ID TO LV_WIID.
CLEAR: LV_PO,WA_WI_ID,IT_WI_ID.

" CREATE LINK TO APPROVE WORK ITEM
MOVE EBELN TO LV_PO.
CONCATENATE LV_PO 'A' LV_WIID INTO LV_PO SEPARATED BY SPACE.
CONCATENATE 'https://-----/sap/opu/odata/SAP/ZAPPROVEPO_SRV/poSet(' "" LV_PO "" ')' INTO G_URL_FIOR1.

" CREATE LINK TO REJECT WORK ITEM
CLEAR: LV_PO.
MOVE EBELN TO LV_PO.

```

CONCATENATE LV\_PO 'R' LV\_WIID INTO LV\_PO SEPARATED BY SPACE.  
 CONCATENATE 'https://-----/sap/opu/odata/SAP/ZAPPROVEPO\_SRV/poSet('''' LV\_PO '''')' INTO G\_URL\_FIOR2.

CONSTANTS:

\*-- Constants used in the body of the Email (HTML)  
 C\_HTM TYPE CHAR3 VALUE 'HTM',  
 C\_STYLE\_START TYPE CHAR255 VALUE '<FONT face=Arial size=2>',  
 C\_NEW\_LINE TYPE CHAR255 VALUE '<br>',  
 C\_LINK\_START TYPE CHAR128 VALUE '<A href="www.w3schools.com">',  
 C\_LINK\_TEXT TYPE CHAR32 VALUE 'Link for learning HTML',  
 C\_LINK\_END TYPE CHAR4 VALUE '</A>',  
 C\_SPACE(6) TYPE C VALUE '&nbsp;'.

DATA: GW\_CONT LIKE LINE OF GT\_CONT.

\*start-> === Filling Email header ======  
 GW\_CONT-LINE = '<html>'."  
 APPEND GW\_CONT TO GT\_CONT.

CLEAR GW\_CONT.  
 MOVE '<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">' TO GW\_CONT-LINE.  
 APPEND GW\_CONT TO GT\_CONT.

CLEAR GW\_CONT.  
 MOVE '<head>' TO GW\_CONT-LINE.  
 APPEND GW\_CONT TO GT\_CONT.

CLEAR GW\_CONT.  
 GW\_CONT-LINE = '<em><font ' .  
 APPEND GW\_CONT TO GT\_CONT.

CLEAR GW\_CONT.  
 GW\_CONT-LINE = 'color="#0000FF" size="+7" face="Arial,'.  
 APPEND GW\_CONT TO GT\_CONT.

CLEAR GW\_CONT.  
 GW\_CONT-LINE = 'Helvetica, sans-serif"> </font></em>'."  
 APPEND GW\_CONT TO GT\_CONT.

CLEAR GW\_CONT.  
 MOVE '</head>' TO GW\_CONT-LINE.  
 APPEND GW\_CONT TO GT\_CONT.

CLEAR GW\_CONT.  
 GW\_CONT-LINE = '<br>'."  
 APPEND GW\_CONT TO GT\_CONT.

CLEAR GW\_CONT.

CONCATENATE 'Dear' 'Sir,' INTO GW\_CONT-LINE SEPARATED BY SPACE.

APPEND GW\_CONT TO GT\_CONT.

CLEAR GW\_CONT.  
 GW\_CONT-LINE = '<br>'."  
 APPEND GW\_CONT TO GT\_CONT.

CONCATENATE 'Please Release PO' " INTO GW\_CONT-LINE SEPARATED BY SPACE.

```
GW_CONT-LINE = '<body>'.
```

```
APPEND GW_CONT TO GT_CONT.
```

```
CONCATENATE '<p>Purchase number' EBELN 'is waiting for your decision,'  
'kindly login the following applications to approve the document:</p>' INTO GW_CONT-  
LINE SEPARATED BY SPACE.
```

```
APPEND GW_CONT TO GT_CONT.
```

*"Insert URLs in bullets*

```
GW_CONT-LINE = '<ul>'.
```

```
APPEND GW_CONT TO GT_CONT.
```

*"Option 1*

```
CONCATENATE '<li>'
```

```
'<a href=""  
G_URL_FIOR1  
"">'
```

```
'Approve</a>  
'</li>'
```

```
INTO GW_CONT-LINE.
```

```
APPEND GW_CONT TO GT_CONT.
```

*"Option 2*

```
CONCATENATE '<li>'
```

```
'<a href=""  
G_URL_FIOR2  
"">'
```

```
'Reject</a>  
'</li>'
```

```
INTO GW_CONT-LINE.
```

```
APPEND GW_CONT TO GT_CONT.
```

```
GW_CONT-LINE = '</ul>'.
```

```
APPEND GW_CONT TO GT_CONT.
```

```
GW_CONT-LINE = '</body>'.
```

```
APPEND GW_CONT TO GT_CONT.
```

```
GW_CONT-LINE = '<br><p>Regards</P>'.
```

```
APPEND GW_CONT TO GT_CONT.
```

```
GW_CONT-LINE = '<p><i>This e-mail was generated automatically from'.
```

```
APPEND GW_CONT TO GT_CONT.
```

```
GW_CONT-LINE = 'SAP; please do not reply.</i></P>'.
```

```
APPEND GW_CONT TO GT_CONT.
```

```
GW_CONT-LINE = '</body></html>'.
```

```
APPEND GW_CONT TO GT_CONT.
```

```
CLEAR GW_CONT.
```

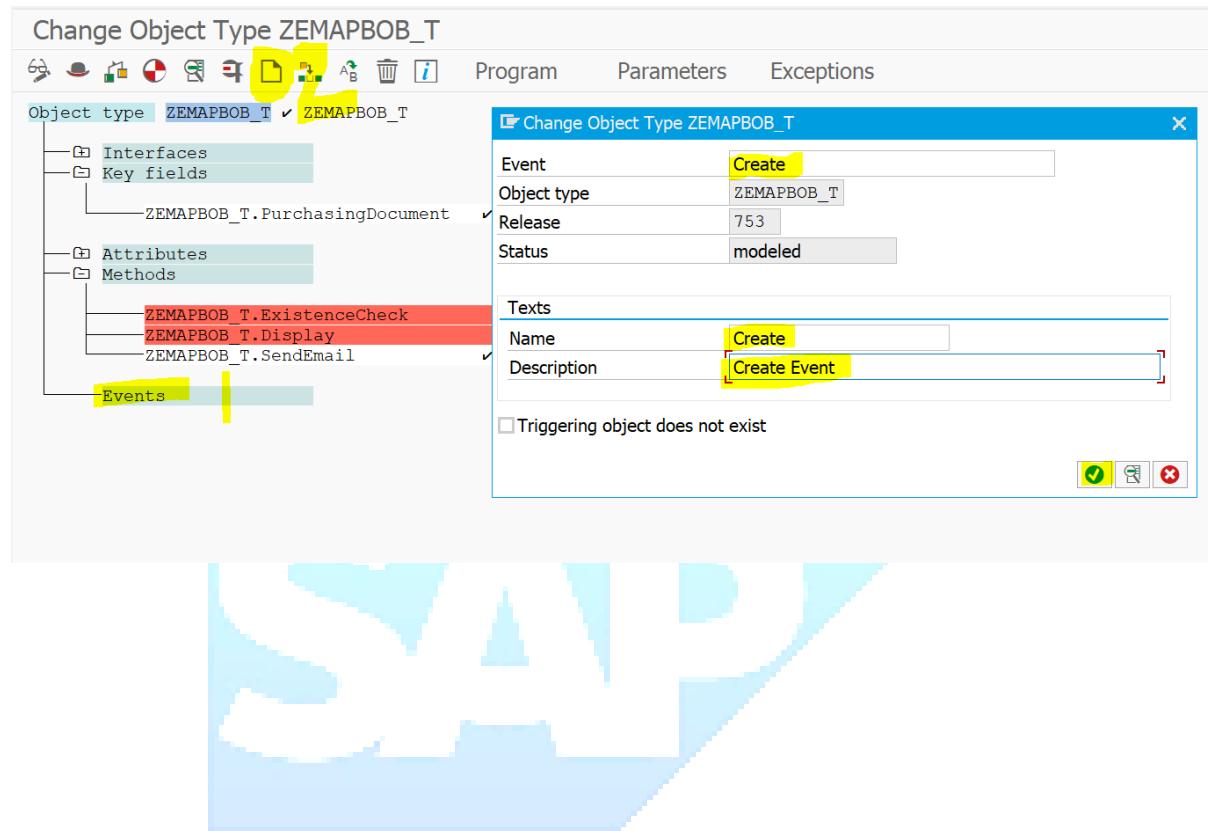
```
GW_CONT-LINE = '<br>'.
```

```
APPEND GW_CONT TO GT_CONT.
```

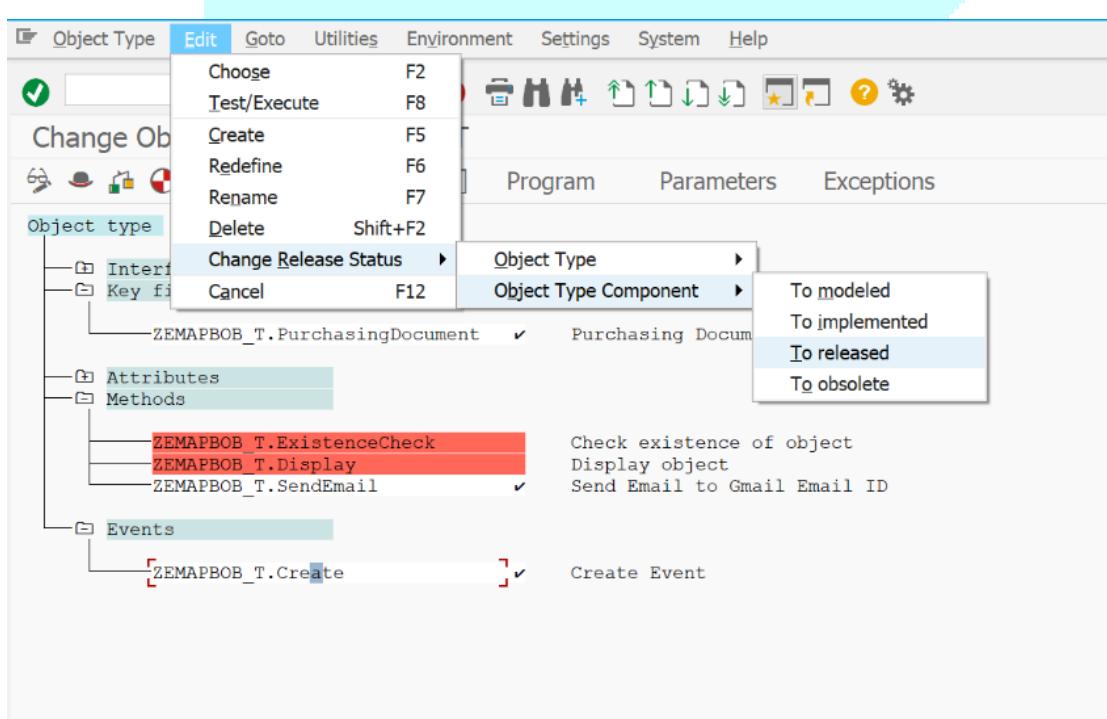
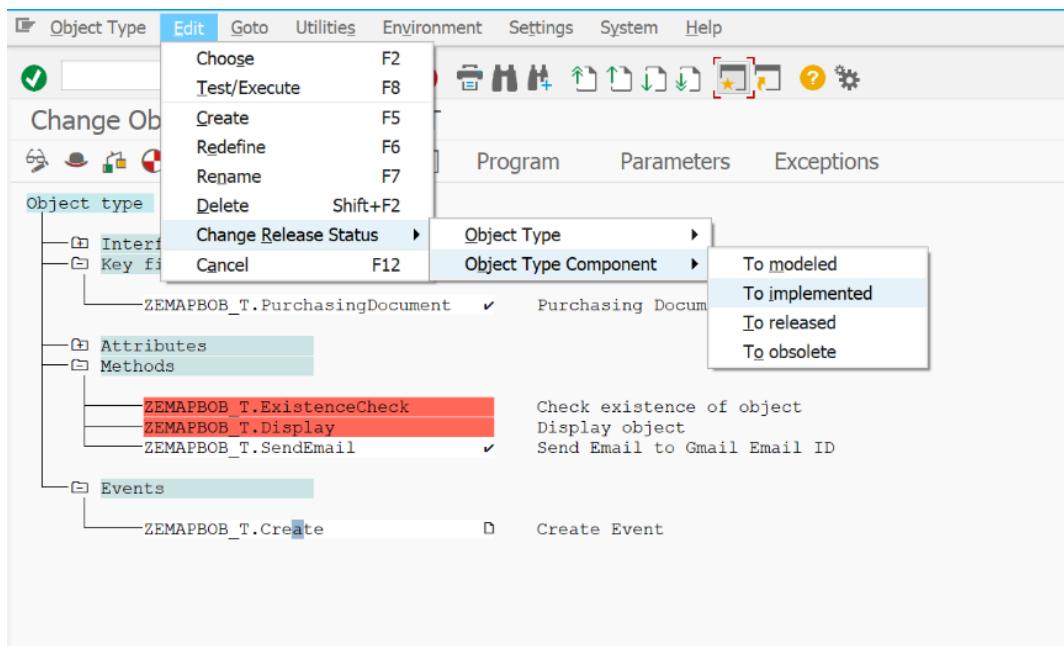
**Note:** After writing code in the BO Method must need to click on generate button to make it available.

## Step 7:

Now we have to create event to use in Workflow.



Same as before, must need to implement and release it to make it available for use.



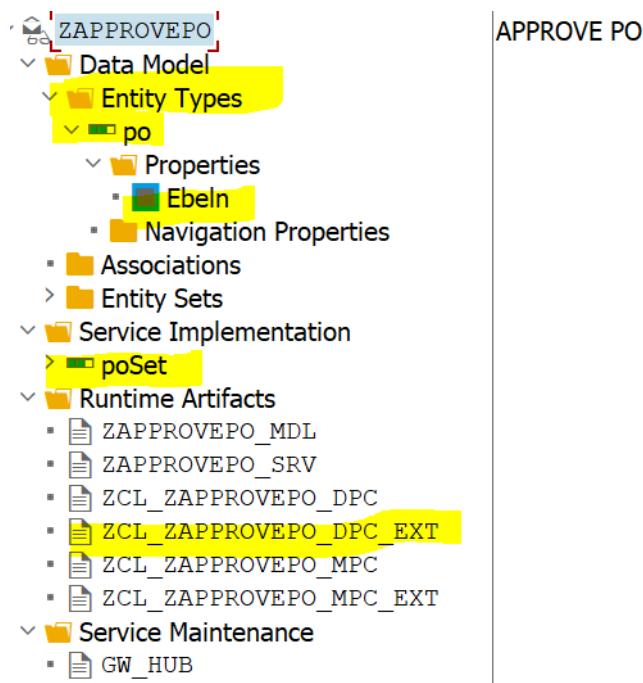
## Step 8:

We need an API that will send a request of approve or reject case selected by approver from email.

In Our case, We Create ODATA and Implement Get Entity Method:

**Go To Transaction Code SEGW and create a project with the following objects:**

**Note:** I am not telling ODATA Creation in Detail here. Also change length of Ebeln to 25.



After Creating and Generating the ODATA Project we have to **redefine** the Get Entity Method.

Why we use get entity method?

As we have to send 2 parameters from Email to Back SAP.

1. Purchase Order Number
2. Approve or Reject Selection

**Code for Get Entity Method:**

```
DATA: WA_KEY_TAB LIKE LINE OF IT_KEY_TAB,  
      LV_EBELN  TYPE EBELN,  
      LV_VAR1   TYPE C LENGTH 10,  
      LV_VAR2   TYPE C LENGTH 1,  
      LV_VAR3   TYPE C LENGTH 12.  
  
DATA: WORKITEM_ID      TYPE SWR_STRUCT-WORKITEMID.  
  
READ TABLE IT_KEY_TAB INTO WA_KEY_TAB WITH KEY NAME = 'Ebeln'.  
  
SPLIT WA_KEY_TAB-VALUE AT SPACE INTO LV_VAR1 LV_VAR2 LV_VAR3.  
  
IF SY-SUBRC EQ 0.  
  
  IF LV_VAR2 EQ 'A'.  
  
    MOVE LV_VAR3 TO WORKITEM_ID.  
    CALL FUNCTION 'SAP_WAPI_FORWARD_WORKITEM'  
      EXPORTING  
        WORKITEM_ID = WORKITEM_ID.  
  
    COMMIT WORK AND WAIT.  
  
  ELSEIF LV_VAR2 EQ 'R'.  
  
    MOVE LV_VAR3 TO WORKITEM_ID.  
    CALL FUNCTION 'SAP_WAPI_WORKITEM_DELETE'  
      EXPORTING  
        WORKITEM_ID = WORKITEM_ID.  
  
    COMMIT WORK AND WAIT.  
  
ENDIF.  
  
ENDIF.  
MOVE LV_VAR1 TO LV_EBELN.  
SELECT SINGLE * FROM EKKO INTO ER_ENTITY WHERE EBELN EQ LV_EBELN.
```

**Step 9:**

Create Workflow and Workflow Task:

Go to PFTC:

**Task: Maintain**

**Workflow Template**

**Name**

**Workflow Pattern**

**Basic data**

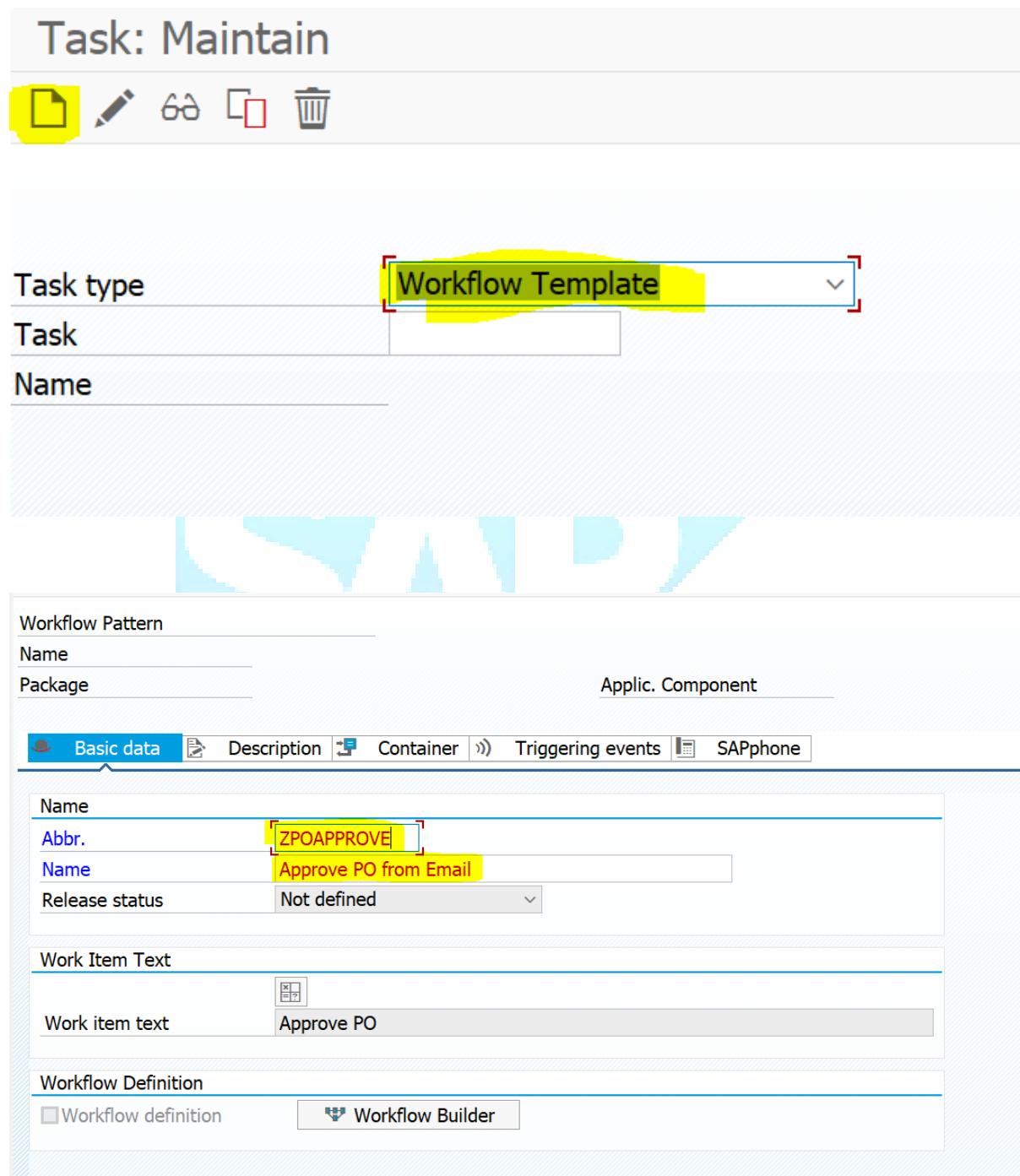
**Name**  
**Abbr.** ZPOAPPROVE  
**Name** Approve PO from Email  
**Release status** Not defined

**Work Item Text**

**Work item text** Approve PO

**Workflow Definition**

Workflow definition [Workflow Builder](#)



Link the event here that was created in Business Object.

Workflow Pattern: Display

Workflow Pattern 90000135 ZPOAPPROVE  
Name Approve PO from Email  
Package Z\_ASAD      Applic. Component

Basic data Description Container Triggering events SAPphone

Standard events

Binding...	Object Category	Object Type	Event	Name
<input checked="" type="checkbox"/>	BOR Object Type	ZEMAPBOB_T	CREATE	MAPBOB_T Create
<input type="checkbox"/>				



## Step 10:

Create workflow.

**Workflow Pattern: Display**

The screenshot shows the SAP Workflow Pattern Display screen. The workflow pattern is identified by ID 90000135 and name ZPOAPPROVE. It is part of the package Z\_ASAD. The 'Basic data' tab is selected. The 'Name' field contains 'ZPOAPPROVE'. The 'Abbr.' field contains 'ZPOAPPROVE'. The 'Name' field under 'Work Item Text' contains 'Approve PO from Email'. The 'Release status' dropdown is set to 'Not defined'. The 'Workflow definition' section shows a checked checkbox labeled 'Workflow definition' next to a button labeled 'Workflow Builder'. A large yellow arrow points downwards towards the Workflow Builder interface.

We have to create 2 activities only for simplicity and proper understanding.

**Workflow Builder - Display 'ZPOAPPROVE'**

The screenshot shows the SAP Workflow Builder interface. The 'Information Area' displays the workflow ID WS90000135, version 0000 (0000) Definition, and status Active, Saved. The 'Navigation Area' shows steps 000004 sendemail and 000008 Release of purchase order. The 'Workflow Container' lists various elements. The 'Graphical Model' pane shows a process flow starting with a 'Create Event' step, followed by a 'sendemail' activity, then a 'Release of purchase order' activity. A decision diamond labeled 'Purchase order' leads to two parallel regions. The left region leads to a 'Not used' activity, and the right region leads to an 'Purchase Order Changed Significantly' activity, which then leads to a final 'Workflow' activity. A large yellow arrow points from the 'Workflow Builder' button in the previous screenshot to this interface.

## Activity 1:

We have to create this Task in PFTC T-Code.

Workflow Builder - Display 'ZPOAPPROVE'

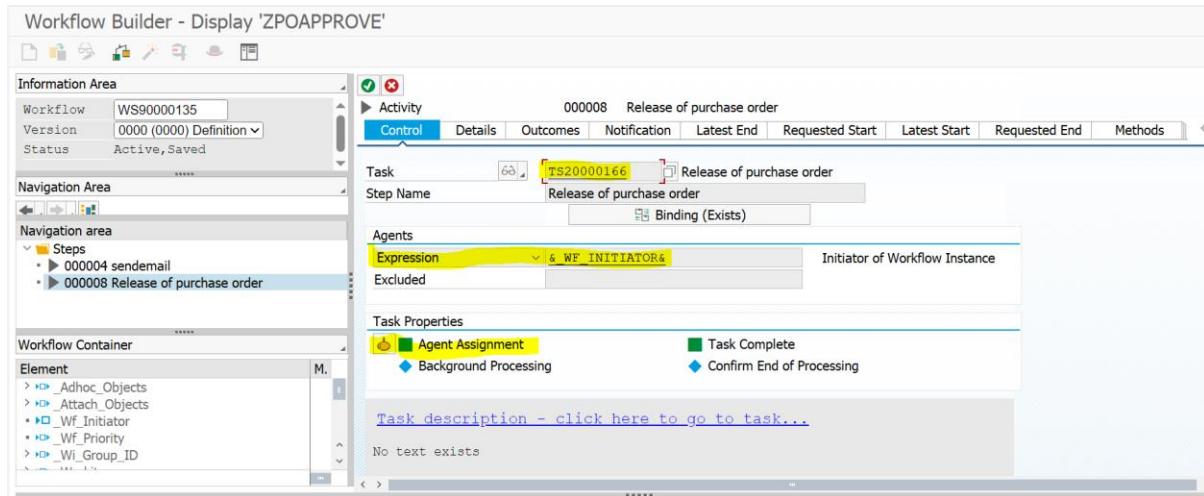
The screenshot shows the SAP Workflow Builder interface for a workflow named 'ZPOAPPROVE'. In the 'Information Area', the workflow ID is WS90000135, version 0000 (0000) Definition, and the status is Active, Saved. In the 'Task' section, a new task is being created with the step name 'sendemail' and task ID TS90000200. The 'Task Properties' section includes options for Agent Assignment, Background Processing, Task Complete, and Confirm End of Processing. A task description placeholder 'Task description - click here to go to task...' is present. The 'Navigation Area' and 'Workflow Container' panes are also visible.

Standard Task: Display

The screenshot shows the SAP Standard Task configuration for a task named 'sendemail' with ID 90000200. The 'Basic data' tab is selected, showing details like Name (sendemail), Package (Z\_ASAD), and Applic. Component. The 'Object method' tab is active, displaying the following settings: Object Category (BOR Object Type), Object Type (ZEMAPBOB\_T), Method (SENDEMAIL), and SendEmail. Under Execution, the 'Background processing' checkbox is checked. Other tabs include Description, Container, Triggering events, Terminating events, Default rules, and SAPphone.

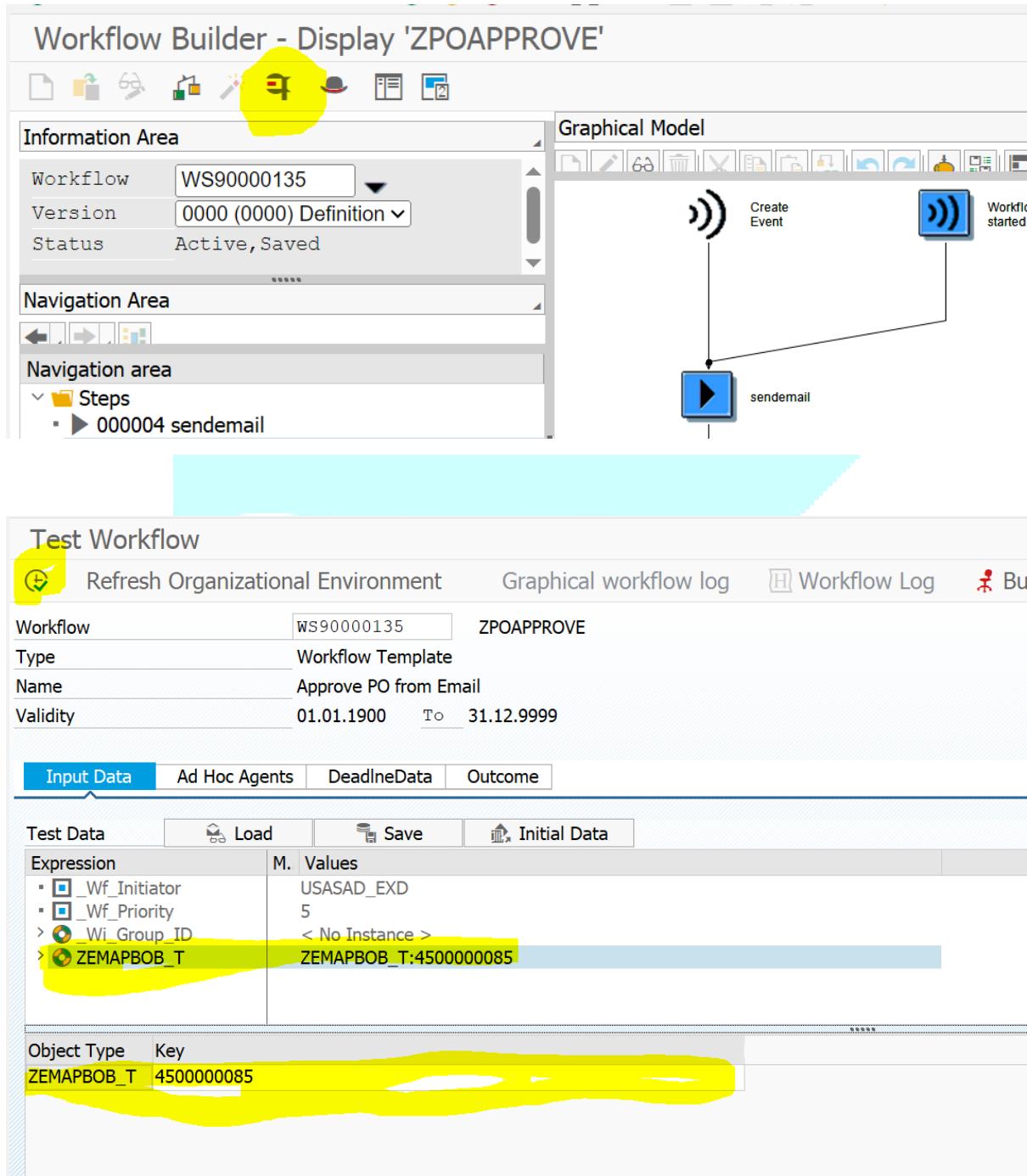
## Activity 2:

We have to use standard task here for sending notification in Inbox.



## Step 11:

Let's Run the Workflow and Test it.



Test Workflow

Refresh Organizational Environment Graphical workflow log Workflow Log Business Workplace

Workflow WS90000135 ZPOAPPROVE

Type Workflow Template

Name Approve PO from Email

Validity 01.01.1900 To 31.12.9999

Input Data Ad Hoc Agents DeadlineData Outcome

Test Data Load Save Initial Data

Expression	M. Values
• <input type="checkbox"/> _Wf_Initiator	USASAD_EXD
• <input type="checkbox"/> _Wf_Priority	5
> <input checked="" type="checkbox"/> _Wi_Group_ID	< No Instance >
> <input checked="" type="checkbox"/> ZEMAPBOB_T	ZEMAPBOB_T:4500000085

Object Type	Key
ZEMAPBOB_T	4500000085

Task started under work item ID 260275 (current state: In Process)



## We can see our inbox the notification arrived to approve Purchase Order.

The screenshot shows the SAP Business Workplace interface for user ASAD\_EXD. The left sidebar displays the 'Workplace: ASAD\_EXD' tree, with 'Inbox' selected. The main area is the 'Inbox' view, which lists messages with columns for 'Class', 'Title', 'Date received', and 'Author'. One message is highlighted: 'Please release purchase order' by 'TESTING' on 05.10.2024.

Class	Title	Date received	Author
	Please release purchase order	16.10.2024	
	TESTING	05.10.2024	
	TESTING	05.10.2024	
	Please approve Budget Request 0000090083	05.10.2024	

Also, see in Transaction **SOST** the Email arrived to send.

The screenshot shows the SAP Transaction SOST (Send Requests) screen. It displays a table of send requests with columns for 'Status', 'Send Method', 'Doc. Title', 'Sender', and 'Recipient'. Two entries are visible: 'By E-Mail' for 'Release PO Notification' from 'saptechall@gmail.com' to 'saptechall@gmail.com' and 'm.uzairkoreja@gmail.com'.

Status	Send Method	Doc. Title	Sender	Recipient
▲	By E-Mail	Release PO Notification	saptechall@gmail.com	saptechall@gmail.com
▲	By E-Mail	Release PO Notification	saptechall@gmail.com	m.uzairkoreja@gmail.com

Select Email line and Click on Display to see Email Body.

We can see Approve and Reject buttons.

The screenshot shows the email body for the 'Release PO Notification'. The subject is 'Display Document: Release PO Notification'. The email content includes a greeting, a note about a purchase number waiting for approval, and two buttons: 'Approve' and 'Reject'. The bottom of the email states: 'This e-mail was generated automatically from SAP; please do not reply.'

Release PO Notification

**Created** SAP Workflow

Dear Sir,

Purchase number 4500000085 is waiting for your decision, kindly login the following applications to approve the document:

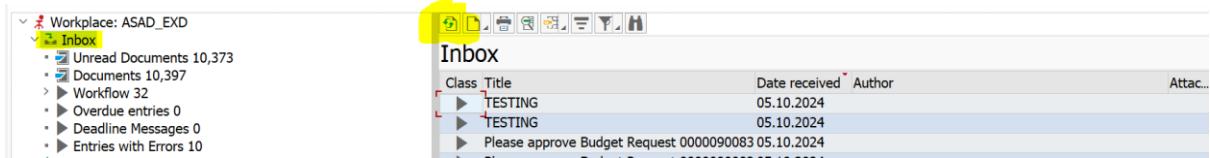
- [Approve](#)
- [Reject](#)

Regards

*This e-mail was generated automatically from SAP; please do not reply.*

**When we click on Approve or Reject Button.**

**The notification will be removed from inbox.**



A screenshot of the SAP Fiori inbox interface. On the left, there's a sidebar with 'Workplace: ASAD\_EXD' and a 'Inbox' section containing statistics: Unread Documents (10,373), Documents (10,397), Workflow (32), Overdue entries (0), Deadline Messages (0), and Entries with Errors (10). The main area is titled 'Inbox' and shows three notifications:

Class	Title	Date received	Author	Attac...
►	TESTING	05.10.2024		
►	TESTING	05.10.2024		
►	Please approve Budget Request 0000090083	05.10.2024		



