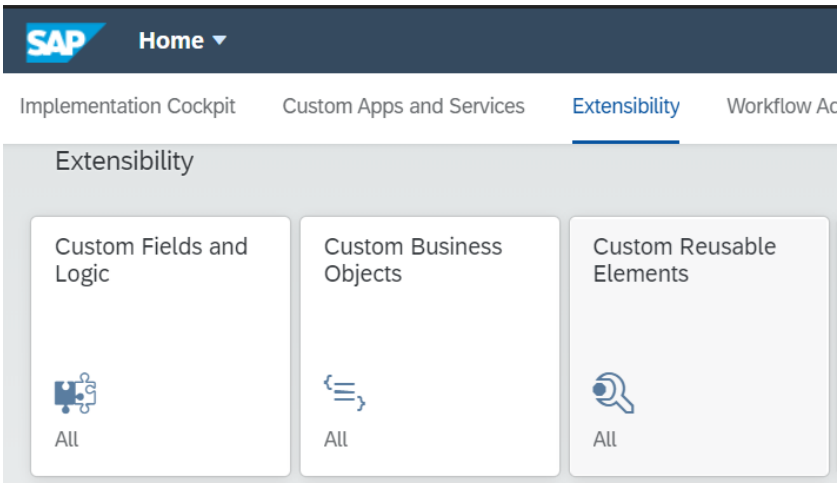
Création de custom reusable elements

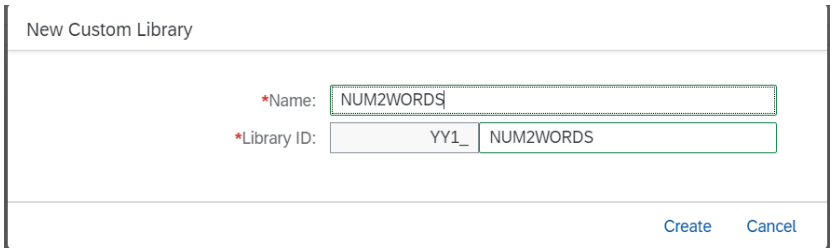
La deuxième étape consiste à créer la logique de conversion des nombres en mots.

Pour ce faire, sélectionnez la tuile custom reusable elements dans le groupe Extensibility.



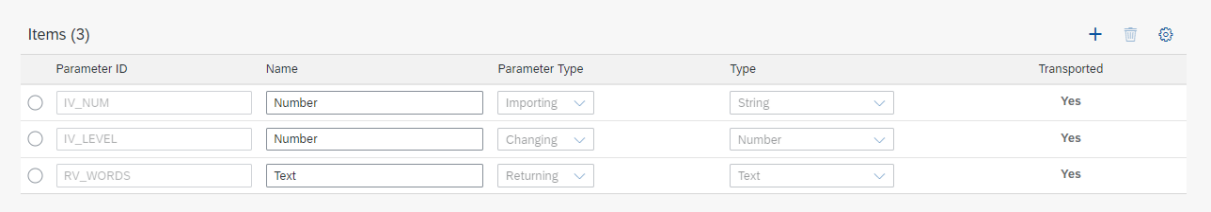
Créez une nouvelle custom library en cliquant sur le bouton + sous l'onglet custom library.

Gérez les éléments suivants et cliquez sur create



Ajouter une nouvelle méthode dans la bibliothèque custom library créée.

Ajouter une nouvelle méthode dans la custom library nouvellement créée



Enregistrez et publiez la custom library. Vous ne pourrez ajouter de la logique à la méthode qu'après la publication.

Une fois la bibliothèque publiée, cliquez sur method id.

L'implémentation de la méthode s'ouvre alors.

Cliquez sur Create Draft et saisissez le code suivant.

```
*  Méthode alternative de SPELL-AMOUNT BY YOUSSEF SOUDOU

   DATA lv_int TYPE i.
   DATA lv_p1 TYPE p decimals 0.
   DATA lv_int2 TYPE i.
   DATA lv_num TYPE P LENGTH 16 DECIMALS 14.
   DATA lv_div TYPE string.
   DATA lv_ret TYPE string.
   DATA lv_numSt TYPE string.

   TYPES: BEGIN OF ty_ns,
            num   TYPE I ,
            word  TYPE string,
          END OF ty_ns.

   DATA: lt_unite TYPE TABLE OF ty_ns,
         lt_disaine TYPE TABLE OF ty_ns,
         ls_dd TYPE  ty_ns .

CLEAR ls_dd.
```

```
CLEAR lt_unite.
CLEAR lt_disaine.

 ls_dd-num = 0 .
 ls_dd-word  = 'Zero'.
   APPEND ls_dd TO lt_unite .
  ls_dd-num = 1 .
  ls_dd-word  = 'UN'.
   APPEND ls_dd TO lt_unite .
ls_dd-num =  2.
ls_dd-word  = 'DEUX'.
 APPEND ls_dd TO lt_unite .
ls_dd-num = 3.
ls_dd-word  = 'TROIS'.
 APPEND ls_dd TO lt_unite .
ls_dd-num = 4.
ls_dd-word  = 'QUATRE'.
 APPEND ls_dd TO lt_unite .
ls_dd-num = 5.
ls_dd-word  = 'CINQ'.
 APPEND ls_dd TO lt_unite .
ls_dd-num = 6.
ls_dd-word  = 'SIX'.
 APPEND ls_dd TO lt_unite .
 ls_dd-num = 7.
ls_dd-word  = 'SEPT' .
 APPEND ls_dd TO lt_unite .
ls_dd-num = 8.
ls_dd-word  = 'HUIT'.
 APPEND ls_dd TO lt_unite .
ls_dd-num = 9 .
ls_dd-word  = 'NEUF' .
 APPEND ls_dd TO lt_unite .
ls_dd-num = 11.
ls_dd-word  = 'ONZE'.
 APPEND ls_dd TO lt_unite .
ls_dd-num = 12.
ls_dd-word  = 'DOUZE'.
 APPEND ls_dd TO lt_unite .
ls_dd-num = 13.
ls_dd-word  = 'TREIZE'.
 APPEND ls_dd TO lt_unite .
ls_dd-num = '14'.
ls_dd-word  = 'QUATORZE'.
 APPEND ls_dd TO lt_unite .
ls_dd-num = '15'.
ls_dd-word  = 'QUINZE'.
 APPEND ls_dd TO lt_unite .
ls_dd-num = 16.
ls_dd-word  = 'SEIZE'.
 APPEND ls_dd TO lt_unite .
ls_dd-num = 17.
```

```
 ls_dd-word  = 'DIX-SEPT'.
  APPEND ls_dd TO lt_unite .
 ls_dd-num = 18.
 ls_dd-word  = 'DIX-HUIT'.
  APPEND ls_dd TO lt_unite .
 ls_dd-num =  19 .
 ls_dd-word  = 'DIX-NEUF' .
   APPEND ls_dd TO lt_unite .



 ls_dd-num = 0.
 ls_dd-word  = 'Zero'.
 APPEND ls_dd TO lt_disaine.
 ls_dd-num = 10.
 ls_dd-word  = 'DIX'.
  APPEND ls_dd TO lt_disaine .
ls_dd-num = 20.
 ls_dd-word  = 'VINGT' .
  APPEND ls_dd TO lt_disaine .
 ls_dd-num = 30.
 ls_dd-word  = 'TRENTE'.
  APPEND ls_dd TO lt_disaine .
 ls_dd-num = 40.
 ls_dd-word  = 'QUARANTE'.
  APPEND ls_dd TO lt_disaine .
 ls_dd-num = 50.
 ls_dd-word  = 'CINQUANTE'.
  APPEND ls_dd TO lt_disaine .
 ls_dd-num = 60.
 ls_dd-word  = 'SOIXANTE'.
  APPEND ls_dd TO lt_disaine .
 ls_dd-num =  70.
 ls_dd-word  = 'SOIXANTE-DIX'.
  APPEND ls_dd TO lt_disaine .
 ls_dd-num = 80.
 ls_dd-word  = 'QUATRE VINGT'.
  APPEND ls_dd TO lt_disaine .
 ls_dd-num = 90.
 ls_dd-word  = 'QUATRE-VINGT-DIX'.
  APPEND ls_dd TO lt_disaine .


 CLEAR ls_dd .

   SPLIT IV_NUM AT '.' INTO DATA(lv_int5) DATA(lv_int6) .

   lv_int = lv_int5.

   IF lv_int EQ 0.
      RV_WORDS  = 'ZERO'.
```

```
    return.
  ENDIF.

  IF  lv_int  GE 1000000000.
    lv_div = floor( lv_int / 1000000000 ).
    lv_numSt = lv_div.

    DATA(output) = YY1_NUM2WORD S( EXPORTING iv_num  = lv_numSt
                          CHANGING iv_level = iv_level ).
    RV_WORDS  = |{ rv_words  } { output } MILLIARD|.
    IF floor( lv_int / 1000000000 ) GT 1.
      rv_words  = |{ rv_words  }S|.
    ELSE.
      rv_words  = |{ rv_words  }|.
    ENDIF.
      lv_int = lv_int MOD 1000000000.
  ENDIF.


  IF  lv_int GE 1000000.
    lv_div = floor ( lv_int / 1000000 ).
    lv_numSt = lv_div.
    DATA(output2) = YY1_NUM2WORD S( EXPORTING iv_num  = lv_numSt
                          CHANGING iv_level = iv_level ).
    RV_WORDS  = |{ rv_words  } { output2 } MILLION|.
    IF floor ( lv_int / 1000000 ) GT 1.
      rv_words  = |{ rv_words  }S|.
    ELSE.
      rv_words  = |{ rv_words  }|.
    ENDIF.
      lv_int = lv_int MOD 1000000.
  ENDIF.


  IF lv_int  GE 1000.
    lv_div = floor( lv_int / 1000 ).
    lv_numSt = lv_div.
    DATA(output3) = YY1_NUM2WORD S( EXPORTING iv_num  = lv_numSt
                          CHANGING iv_level = iv_level ).
    RV_WORDS  = |{ rv_words  } { output3 } MILLE|.

    IF lv_div EQ 1.
      REPLACE SUBSTRING 'UN' IN RV_WORDS  WITH ''.
    ENDIF.

    IF lv_div GT 1.
      rv_words  = |{ rv_words  }S|.
    ELSE.
      rv_words  = |{ rv_words  }|.
    ENDIF.
      lv_int = lv_int MOD 1000.
  ENDIF.
```

```
IF  lv_int GE 100 .
   lv_div = floor( lv_int / 100 ).
   lv_numSt = lv_div.
   DATA(output4) = YY1_NUM2WORD S( EXPORTING iv_num  = lv_numSt
                        CHANGING iv_level = iv_level ).
   RV_WORDS  = |{ rv_words  } { output4 } CENT|.

   IF lv_div EQ 1.
      REPLACE SUBSTRING 'UN' IN RV_WORDS  WITH ''.
   ENDIF.

   IF lv_div GT 1.
      rv_words  = |{ rv_words  }S|.
   ELSE.
      rv_words  = |{ rv_words  }|.
   ENDIF.
      lv_int = lv_int MOD 100.
ENDIF.

IF lv_int GT 0.

   IF lv_int lt 20.

      IF lv_int EQ 10.
         READ TABLE lt_disaine REFERENCE INTO DATA(ls_disaine) WITH KEY num = lv_int.
         rv_words  = |{ rv_words  } { ls_disaine->word  }|.
      ELSE.
         READ TABLE lt_unite REFERENCE INTO DATA(ls_unite) WITH KEY num = lv_int.
         rv_words  = |{ rv_words  } { ls_unite->word  }|.
      ENDIF.
   ELSE.

      IF ( ( lv_int GT 70 ) AND ( lv_int lt 80 ) ) OR ( ( lv_int GT 90 ) AND ( lv_int lt 100  ) ) .
         READ TABLE lt_unite REFERENCE INTO DATA(ls_unite2) INDEX ( ( lv_int MOD 10 ) + 10 ).
         lv_div = floor( lv_int / 10 ).
         lv_p1 = lv_div.
         READ TABLE lt_disaine REFERENCE INTO ls_disaine  INDEX  lv_p1.
         IF lv_int MOD 10 EQ 1.

            rv_words  = |{ rv_words  } { ls_disaine->word  } ET { ls_unite2->word  }|.

         ELSE.

            rv_words  = |{ rv_words } { ls_disaine->word  }-{ ls_unite2->word  }|.

         ENDIF.
      ELSE.
         lv_div = floor( lv_int / 10 ).
         lv_p1 = lv_div.
         READ TABLE lt_disaine REFERENCE INTO DATA(ls_disaine2)  INDEX  lv_p1 + 1.
```

```
          rv_words  = |{ rv_words } { ls_disaine2->word  }|.

          IF lv_int MOD 10 GT 0.
            READ TABLE lt_unite REFERENCE INTO DATA(ls_unite3) INDEX ( ( lv_int MOD 10 ) + 1 ).
            IF lv_int MOD 10 EQ 1.

              rv_words  = |{ rv_words } ET { ls_unite3->word  }|.

            ELSE.

              rv_words  = |{ rv_words }-{ ls_unite3->word  }|.

            ENDIF.

          ENDIF.

        ENDIF.

      ENDIF.


  ENDIF.

  IF lv_int6 IS NOT INITIAL.

    DATA(output6) = YY1_NUM2WORD S( EXPORTING iv_num  = lv_int6
                          CHANGING iv_level = iv_level ).
    IF output6 ne 'ZERO' AND output6 ne 'UN'.
      rv_words  = |{ rv_words } / { output6 } CENTIMES|.
    ELSEIF output6 EQ 'UN'.
      rv_words  = |{ rv_words } / { output6 } CENTIME|.
    ENDIF.

  ENDIF.
```