

My training by tutorial
from developers.sap.com

<https://developers.sap.com/tutorials/aif-proxy-monitoring-interface-create..html>

Create a Simple Proxy Interface

- Instructions and codes are taken from tutorial "Create a Simple Proxy Interface" (<https://developers.sap.com/tutorials/>).
All screenshots made from my own development performed via this tutorial

Create a package

Display Package

Package Edit Goto Utilities Environment System Help

Repository Browser

Repository Information System

Package

ZST1_AIF_TUT

Object Name

ZST1_AIF_TUT

Package ZST1_AIF_TUT Saved

Properties Use Accesses Package Interfaces Subpackages Package Hierarchy

Basic Data

Short Description	Demo Package for AIF Tutorials
Application Component	
Person Responsible	STUDENT001
Created by	STUDENT001
Created On	06/13/2025
Last Changed By	STUDENT001
Changed on	06/13/2025

Package Properties

Superpackage [Change](#)

☐ Main Package

☐ Adding further objects not possible

Transport Attributes

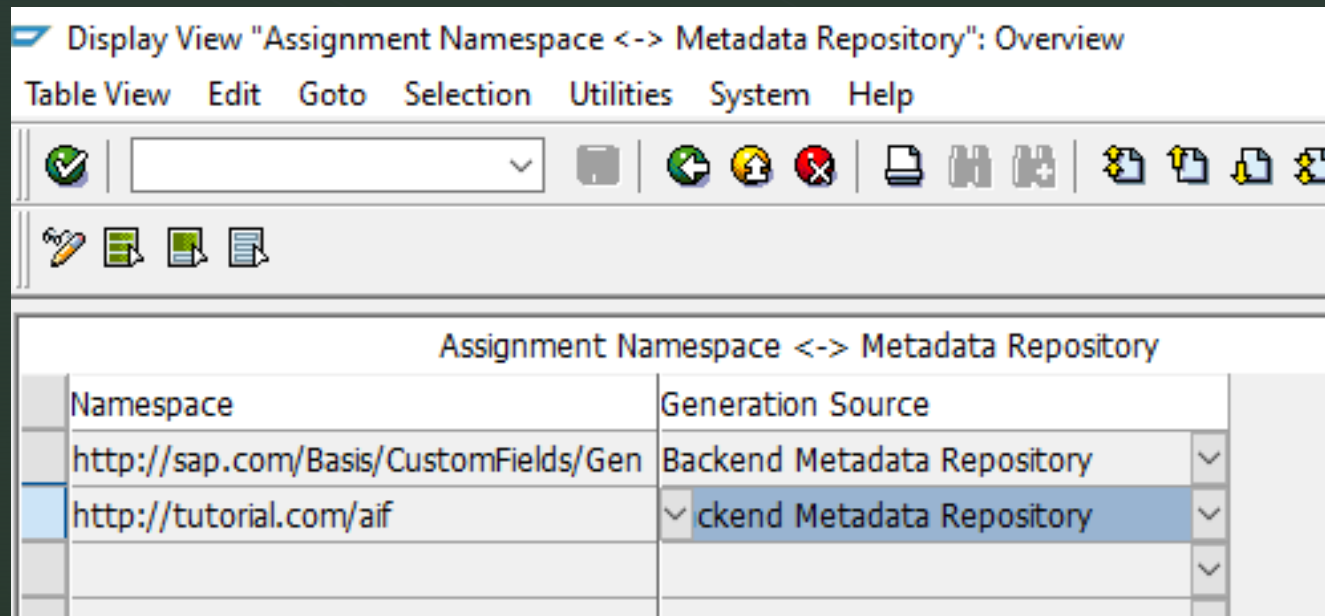
Transport Layer	ZS5M	Transport Layer ZS5M
Software Component	HOME	

☒ Record Object Changes in Transport Requests

Next, you need to assign a new namespace to the Backend Metadata Repository. Run transaction **Assignment Namespace Generating Application** (transaction code **SPXNGENAPPL**).

Switch to **Edit** mode, add a new entry, and enter or select the following details for your new namespace:

Save your changes.



Display View "Assignment Namespace <-> Metadata Repository": Overview

Table View Edit Goto Selection Utilities System Help

Assignment Namespace <-> Metadata Repository

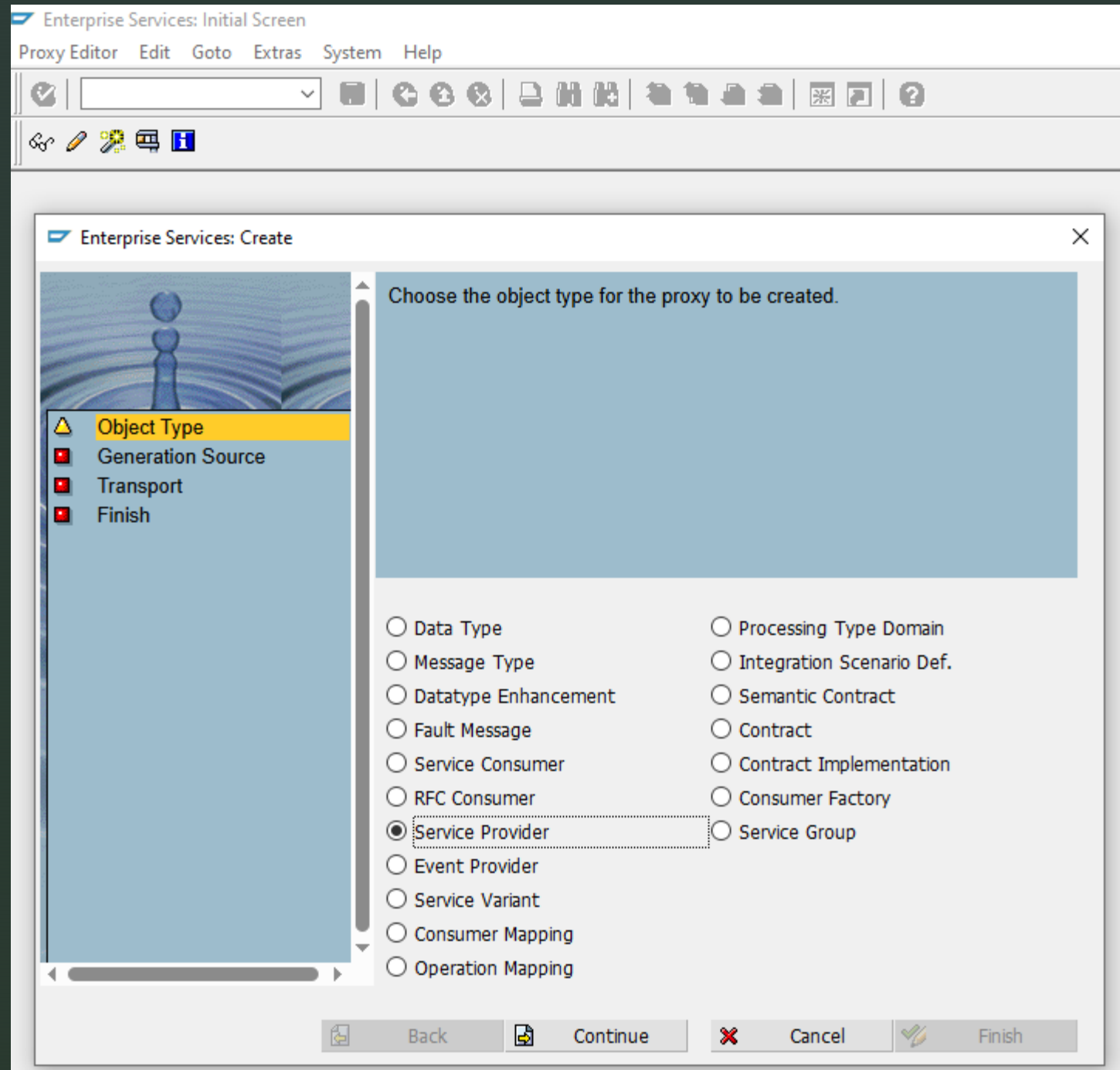
Namespace	Generation Source
http://sap.com/Basis/CustomFields/Gen	Backend Metadata Repository
http://tutorial.com/aif	Backend Metadata Repository

CREATE PROXY

To create a new service interface, run the **proxy editor** (transaction code **SPROXY_START**).

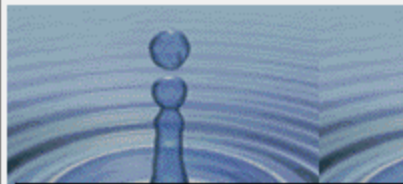
In **Enterprise Services: Initial Screen**, select **Create with Wizard** and carry out the following configuration steps:

1. As **Object Type**, select **Service Provider**. Click **Continue**.
2. As **Kind of Service Provider**, select **Backend**. Click **Continue**.
3. As **Name**, enter and namespace <http://tutorial.com/aif>. Click **Continue**.
4. For the transport options, enter your package **ZST1_AIF_TUT**, select a workbench request, and enter the prefix **ZAIF_**. Select **Continue**.





Enterprise Services: Create



- [Object Type](#)
- Kind of Service Provider**
- Transport
- Finish

Choose the kind of service provider to be generated

- ☒ Backend
- ☐ Enterprise Service Repository
- ☐ Existing ABAP Object (Inside Out)
- ☐ External WSDL/Schema



Back



Continue



Cancel



Finish

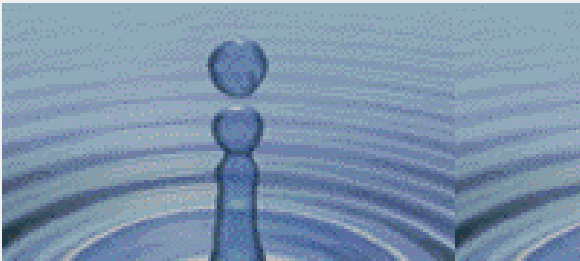
Enterprise Services: Create

Enter external name and namespace (QName) for the **Service Interface** to be created.

☒ [Object Type](#)
☒ [Kind of Service Provider](#)
☒ **QName**
☐ Transport
☐ Finish

Name

Namespace



- [Object Type](#)
- [Kind of Service Provider](#)
- [QName](#)
- ▲ **Transport**
- Finish

Enter the package for the new object or check Local Object to use \$TMP.

If already known you may enter the request to be used.
For proxy objects you have the option to specify a prefix.

☐ Local Object

Package	ZST1_AIF_TUT
Request/Task	S5MK903488
Prefix	ZAIF_



Service Provider

FlightBooking_01_In

New (Revised)

Properties

External View

Internal View

Objects

Configuration

WSDL

Classifications

Related Interfaces

Service Provider

Name	FlightBooking_01_In		
Namespace	http://tutorial.com/aif		
ABAP Object	INTF Interface		
ABAP Name	ZAIF_II_FLIGHT_BOOKING_01_IN		
Prefix	ZAIF_		
Source	Backend Metadata Repository		
Description			
Implementing Class	ZAIF_CL_FLIGHT_BOOKING_01_IN		
WebService Definition	ZFlightBooking_01_In		

General Data

Package	ZST1_AIF_TUT		Semant Version	
Original Language	EN English		Leading BO	
Release Status	Not Released			
Created by		on		00:00:00
Changed by		on		00:00:00

Change Service Provider FlightBooking_01_In

Proxy Edit Goto Utilities System Help



Service Provider

FlightBooking_01_In

Inactive

Properties External View Internal View Objects Configuration WSDL Classifications Related Interfaces



ABAP Name

ZAIF_II_FLIGHT_BOOKING_01_IN

Service Provider

Name	FlightBooking_01_In
Namespace	http://tutorial.com/aif
ABAP Object	INTF Interface
ABAP Name	ZAIF_II_FLIGHT_BOOKING_01_IN
Prefix	ZAIF_
Description	
Pattern	Stateless
Security Level	No

Change Service Provider FlightBooking_01_In

Proxy Edit Goto Utilities System Help



Service Provider

FlightBooking_01_In

Inactive

Properties

External View

Internal View

Objects

Configuration

WSDL

Classifications

Related Interfaces



External Name

FlightBooking_01_In

Service Provider

Define operation name

Name

PostBookings_01



Description

Pattern

Stateless

Security Level

No

DEFINE PROXY STRUCTURES

Next, you need to add an operation and a fault message type in the proxy editor.

1. Switch to the **Internal View** tab. Right-click your service provider and select **Add Operation**. Enter the operation name **PostBookings_01**. With the new operation selected, switch the **Pattern** of the operation to **Not Reliable** to simplify testing.

1. Right-click the operation and select **Set Request > Select Existing Message Type** from the context menu.
2. In the upcoming **Restrict Value Range** dialog, remove all filters.
3. Enter the message type **SXIDAL_FBO_REQUEST_MT** in the **ABAP Name** search filter,
4. and the namespace **<http://sap.com/xi/XI/Demo/Airline>** in the **Namespace** search filter,
5. then press **Enter**. In the search result, select the found entry, and select **Copy**.
2. Right-click the operation and select **Add Fault > Select Existing Fault Message Type**.
3. Similar to the message type search, remove all filters. Then search for the fault message
4. type **CX_SXIDAL_TECHNICAL_PROBLEMS**.
3. Save and activate the proxy.

Proxy Edit Goto Utilities System Help



Service Provider FlightBooking_01_In Inactive

Properties External View Internal View Objects Configuration WSDL Classifications Related Interfaces



ABAP Name

ZAIF_II_FLIGHT_BOOKING_01_IN

POST_BOOKINGS_01

INPUT

AGENCY_DATA

FLIGHT_ID

CLASS_CODE

PASSENGER_NAME

PASSENGER_BIRTHDATE

PASSENGER_FORM_OF_ADDRESS

Operation

Name PostBookings_01

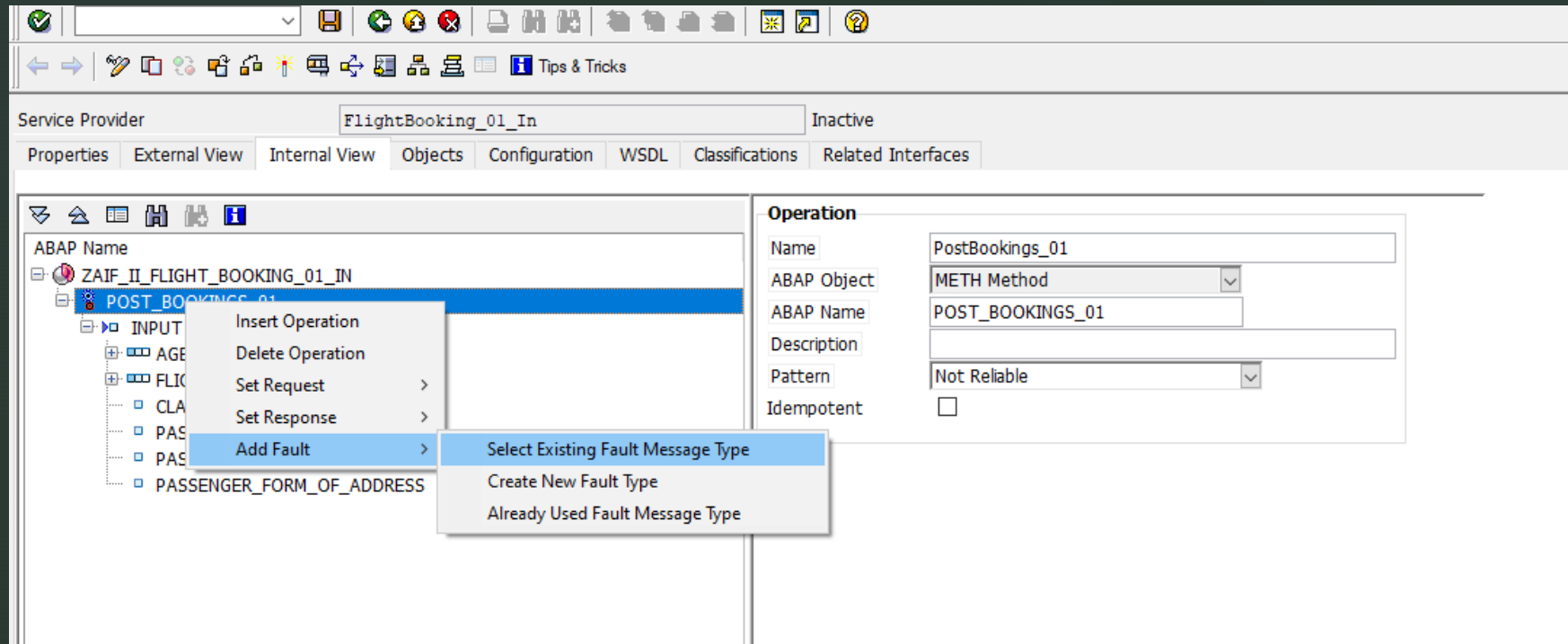
ABAP Object METH Method

ABAP Name POST_BOOKINGS_01

Description

Pattern Not Reliable

Idempotent ☐



Change Service Provider FlightBooking_01_In

Proxy Edit Goto Utilities System Help

Service Provider FlightBooking_01_In Inactive (Revised)

Properties External View Internal View Objects Configuration WSDL Warnings Classifications Related Interfaces

ABAP Name

- ZAIF_II_FLIGHT_BOOKING_01_IN
 - POST_BOOKINGS_01
 - INPUT
 - AGENCY_DATA
 - FLIGHT_ID
 - CLASS_CODE
 - PASSENGER_NAME
 - PASSENGER_BIRTHDATE
 - PASSENGER_FORM_OF_ADDRESS
 - CX_SXIDAL_TECHNICAL_PROBLEMS
 - STANDARD
 - FAULT_TEXT
 - FAULT_URL
 - FAULT_DETAIL

Operation

Name	PostBookings_01
ABAP Object	METH Method
ABAP Name	POST_BOOKINGS_01
Description	
Pattern	Not Reliable
Idempotent	<input type="checkbox"/>

Implement proxy class method

Finally, to book the flights in your test scenario, the proxy class method needs to be implemented. Switch to the **Properties** tab. Double-click the implementing class **ZAIF_CL_FLIGHT_BOOKING_01_IN** and then double-click the method **ZAIF_II_FLIGHT_BOOKING_01_IN~POST_BOOKINGS_01**. Maintain the implementation of the method by copying and pasting the following:

Class Builder: Display Class ZAIF_CL_FLIGHT_BOOKING_01_IN

Class Edit Goto Utilities Environment System Help

Icons: [Checkmark] [Dropdown] [Folder] [Back] [Home] [Close] [Print] [Find] [Find All] [Find in Files] [Find in Project] [Find in Package] [Find in Source] [Find in Test] [Find in Trace] [Find in View] [Find in Window] [Find in Document] [Find in Project] [Find in Package] [Find in Source] [Find in Test] [Find in Trace] [Find in View] [Find in Window] [Find in Document]

Class/Interface ZAIF_CL_FLIGHT_BOOKING_01_IN Implemented / Inactive

Properties Interfaces Friends Attributes Methods Events Types Aliases

Parameters Exceptions Sourcecode [Icons] [Filter]

Method	Level	Visibility	M...	Description
ZAIF_II_FLIGHT_BOOKING_01_IN~POST_B	Instance Method	Public		

method ZAIIF_II_FLIGHT_BOOKING_01_IN~POST_BOOKINGS_01.

DATA: lv_bookkey TYPE bapisbokey,
lv_bookdata TYPE bapisbonew,
lt_bapiret TYPE TABLE OF bapiret2.

** Convert input data*

```
lv_bookdata = VALUE #( airlineid = input-flight_booking_order_request-flight_id-  
airline_id  
                        connectid = input-flight_booking_order_request-flight_id-connection_id  
                        flightdate = input-flight_booking_order_request-flight_id-flight_date  
                        customerid = '1'  
                        class      = input-flight_booking_order_request-class_code  
                        agencynum  = input-flight_booking_order_request-agency_data-  
agency_id  
                        passname   = input-flight_booking_order_request-passenger_name  
                        passform   = input-flight_booking_order_request-  
passenger_form_of_address  
                        passbirth  = input-flight_booking_order_request-passenger_birthdate ).
```

** Call internal flight booking function*

```
CALL FUNCTION 'BAPI_FLBOOKING_CREATEFROMDATA'
```

```
EXPORTING
```

```
  reserve_only = ''
```

```
  booking_data = lv_bookdata
```

```
IMPORTING
```

```
  airlineid    = lv_bookkey-airlineid
```

```
  bookingnumber = lv_bookkey-bookingid
```

```
TABLES
```

```
  return       = lt_bapiret.
```

** error case*

```
IF line_exists( lt_bapiret[ type = 'E' id = 'BAPI' number = '001' ] ).
```

```
  CALL METHOD cl_proxy_fault=>raise
```

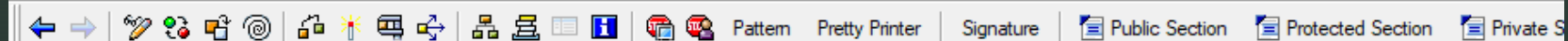
```
  EXPORTING
```

```
    exception_class_name = 'CX_SXIDAL_TECHNICAL_PROBLEMS'
```

```
    bapireturn_tab       = lt_bapiret.
```

```
ENDIF.
```

```
ENDMETHOD.
```

Ty.	Parameter	Typing	Descripti...
INPUT		TYPE SXIDAL_FBO_REQUEST_MT	
	CX_SXIDAL_TECHNICAL_PROBLEMS		

Method ZAIF_II_FLIGHT_BOOKING_01_IN~POST_BOOKINGS_01 active

```

1  method ZAIF_II_FLIGHT_BOOKING_01_IN~POST_BOOKINGS_01.
2      *** ***** INSERT IMPLEMENTATION HERE ***** ***
3      DATA: lv_bookkey TYPE bapisbokey,
4              lv_bookdata TYPE bapisbonew,
5              lt_bapiret TYPE TABLE OF bapiret2.
6
7      * Convert input data
8      lv_bookdata = VALUE #( airlineid = input-flight_booking_order_request-flight_id-airline_id
9                             connectid = input-flight_booking_order_request-flight_id-connection_id
10                            flightdate = input-flight_booking_order_request-flight_id-flight_date
11                            customerid = '1'
12                            class      = input-flight_booking_order_request-class_code
13                            agencynum = input-flight_booking_order_request-agency_data-agency_id
14                            passname  = input-flight_booking_order_request-passenger_name
15                            passform  = input-flight_booking_order_request-passenger_form_of_address
16                            passbirth = input-flight_booking_order_request-passenger_birthdate ).
17
18      * Call internal flight booking function
19      CALL FUNCTION 'BAPI_FLBOOKING_CREATEFROMDATA'
20      EXPORTING
21          reserve_only = ' '
22          booking_data = lv_bookdata
23      IMPORTING
24          airlineid      = lv_bookkey-airlineid
25          booking_number = lv_bookkey-booking_id

```

Scope: \METHOD ZAIF_II_FLIGHT_BOOKING_01_IN~POST_BOOKINGS_01\IF

ABAP

Ln 37 Col 5

Object(s) activated

CREATE NAMESPACE

As interfaces in SAP Application Interface Framework are grouped using namespaces, you must create a namespace.

Go to **Customizing** for SAP Application Interface Framework (transaction code [/n/AIF/CUST](#)) and navigate to **Interface Development > Define Namespace**.

Select **New Entries** and enter the following name and description for your new namespace:

Namespace	Namespace Description
DEMO_2	NS for AIF Proxy tutorials

Display IMG

Implementation Activities Edit Goto Additional Information Utilities System

Navigation bar with icons for back, forward, search, and other functions. Below the icons are three tabs: Existing BC Sets, BC Sets for Activity, and Activated BC Sets for Activity.




Structure

- SAP Application Interface Framework
 - Interface Development
 - Define Namespace
 - Define Interfaces
 - Additional Interface Properties
 - Event Trigger Settings
 - Define Structure Mappings
 - Define Value Mappings
 - Define Fix Values
 - Define Checks
 - Define Actions
 - Interface Variants
 - Error Handling
 - System Configuration

New Entries: Overview of Added Entries

Table View Edit Goto Selection Utilities System Help

Toolbar with icons for: Checkmark, Dropdown menu, Save, Undo, Redo, Delete, Print, Copy, Paste, and Find.

Define Namespace		
NS	Namespace Description	  
DEMO_2	NS for AIF Proxy tutorials	

CREATE INTERFACE

While still in **Customizing** (transaction code **/n/AIF/CUST**), navigate to **Interface Development > Define Interfaces**. In the upcoming dialog, enter your previously created namespace **DEMO_2** and press **Enter**.

Select **New Entries** and enter the following parameters based on your proxy class and implementation.

You can double-check this information in transaction code **SPROXY**

Be aware that entering the **Proxy Class Inbound** automatically fills in **Raw Data Structure**, **Record Type in Raw Structure**, and **Proxy Method**.

Field name	Description	VALUE
Interface Name	Name of the interface to be created, for example (an abbreviation of) the basic type	FLBOOK
Interface version	Version number of the interface	1
Description	Description of the interface	Demo interface 1 for Proxy tutorial
SAP Data structure	Input substructure of the proxy class	SXIDAL_FBO_REQUEST
Raw Data structure	Input structure of the proxy class	SXIDAL_FBO_REQUEST_MT
Record Type in Raw Structure	Main component of the raw data structure	FLIGHT_BOOKING_ORDER_REQUEST
Proxy Class Inbound	Name of the proxy class	ZAIF_CL_FLIGHT_BOOKING_01_IN
Proxy Method	Method name of the generated proxy class	POST_BOOKINGS_01
Interface Direction	Indicates the direction of the interface	Inbound

Save your changes.



Namespace

DEMO_2

Documentation

No display authorizatio

Interface Name

FLBOOK

Interface Version

1

Define Interfaces

Description

Demo interface 1 for Proxy tutorial

SAP Data Structure

SXIDAL_FBO_REQUEST

Raw Data Structure

SXIDAL_FBO_REQUEST_MT

Record Type in Raw Structure

FLIGHT_BOOKING_ORDER_REQUEST

☐ Move Corresponding Structures

Check Function Module

Init Function Before Mapping

Init Function Before Processing

Separate Commit

No Separate Commit

Lifetime of Application Log

☐ Test Mode

Proxy Class Inbound

ZAIF_CL_FLIGHT_BOOKING_01_IN

Proxy Class Outbound

Proxy Method

POST_BOOKINGS_01

Field for the Sending System

Status Handling

☐ Pre-Processing☐ Proxy XML Transformation

Interface Direction

Inbound

Display Service Provider FlightBooking_01_In

Proxy Edit Goto Utilities System Help



Repository Browser

Repository Information System

Enterprise Services Browser



Name	Additional Info
⊕ http://sap.com/srt/scenrio/t	
⊕ http://sap.com/xi/BASIS/Cu:	
⊕ http://sap.com/xi/SAPSCORI	
⊕ http://sap.com/xi/SD-SLS	
⊕ http://soapRuntime/Test	
⊕ http://srt.sap.com/test	
⊕ http://test	
⊖ http://tutorial.com/aif	
⊕ SWCs	
⊕ Packages	
⊖ Object Types	
⊖ Service Providers	
⊕ SWCs	
⊕ Packages	
⊖ Objects	
FlightBookin	http://tutorial.com/aif
⊕ Service Def. (auto.ger	
⊕ Objects	
⊕ ...	
⊕ Object Types	
⊕ Objects	
⊕ SWCs	
⊕ Packages	
⊕ Namespaces	

Service Provider FlightBooking_01_In Active

Properties External View Internal View Objects Configuration WSDL Classifications

Service Provider

Name	FlightBooking_01_In
Namespace	http://tutorial.com/aif
ABAP Object	INTF Interface
ABAP Name	ZAIF_II_FLIGHT_BOOKING_01_IN
Prefix	ZAIF_
Source	Backend Metadata Repository
Description	
Implementing Class	ZAIF_CL_FLIGHT_BOOKING_01_IN
WebService Definition	ZFlightBooking_01_In

General Data

Package	ZST1_AIF_TUT
Original Language	EN English
Release Status	Not Released
Created by	STUDENT001 on 06/13/2025 16:29:26
Changed by	STUDENT001 on 06/14/2025 08:11:57

SPECIFY INTEERFACE ENGINES

Next, you have to select the engines that should be used to handle the messages that are processed.

If you create a new interface, by default, SAP Application Interface Framework handles the messages as proxy messages, so you can keep the default settings.

To double-check the settings, go to **Customizing** for SAP Application Interface Framework (transaction code **/AIF/CUST**)

and navigate to **Interface Development > Additional Interface Properties > Specify Interface Engines**.

In the upcoming dialog, enter your beforehand created namespace **DEMO_2**, and press **Enter**.

Check that the following engines are preselected: