

## Obiettivo

La terza parte richiede, tramite REST **GET** sincrono, un servizio fornito dal backend di ECC mediante funzione RFC. Il servizio è esposto per il tramite di SAP PI.

La Figura 1 e la Figura 2 descrivono l'interfaccia esposta dalla RFC **Z\_BOL\_KEEP\_ALIVE**.

Display Message Type Request

Search  Go

Name	Category	Type	Description
▼ Z_BOL_KEEP_ALIVE	Element		
ZBANCA	Element	xsd:string	Bank key
ZCONTO	Element	xsd:string	Proxy Data Element (Generated)
ZDIVISA	Element	xsd:string	Currency Key
ZIDPAG	Element	xsd:string	R/2 table
ZIMPORTO	Element	xsd:string	BSI: Tax class amount
ZTIMESTAMP	Element	xsd:string	Timestamp

Figura 1

Display Message Type Response

Search  Go

Name	Category	Type	Occurrence	Details	Description
▼ Z_BOL_KEEP_ALIVE.Response	Element				
CODERRORE	Element	xsd:string	0..1	maxLength="2"	2 byte integer (signed)
MESSAGE	Element	xsd:string	0..1	maxLength="50"	Comment
ZZBANCA	Element	xsd:string	0..1	maxLength="15"	Bank key
ZZCONTO	Element	xsd:string	0..1	maxLength="35"	Proxy Data Element (Generated)
ZZDIVISA	Element	xsd:string	0..1	maxLength="5"	Currency Key
ZZIDPAG	Element	xsd:string	0..1	maxLength="5"	R/2 table
ZZIMPORTO	Element	xsd:string	0..1	maxLength="9"; pattern="ld+"	BSI: Tax class amount
ZZTIMESTAMP	Element	xsd:string	0..1	maxLength="19"	2 byte integer (signed)

Figura 2

Il consumer invia il seguente payload in XML tramite la query variable **DATI**

```
<?xml version="1.0" encoding="UTF-8" ?><Input><ServiceID>1</ServiceID><TransactionID>1</TransactionID><Importo>1</Importo><Divisa>eur</Divisa><Timestamp>1</Timestamp></Input> <?xml version="1.0" encoding="UTF-8" ?><Input><ServiceID>1</ServiceID><TransactionID>1</TransactionID><Importo>1</Importo><Divisa>eur</Divisa><Timestamp>1</Timestamp></Input>
```

# Enterprise Services Builder

## Repository oggetti Sender e Receiver

Software Component Version	SC_VODAFONE of nick4name.eu
Namespace	http://vodafone.com/xi/bridge/ISP/KeepAlive

### Architettura

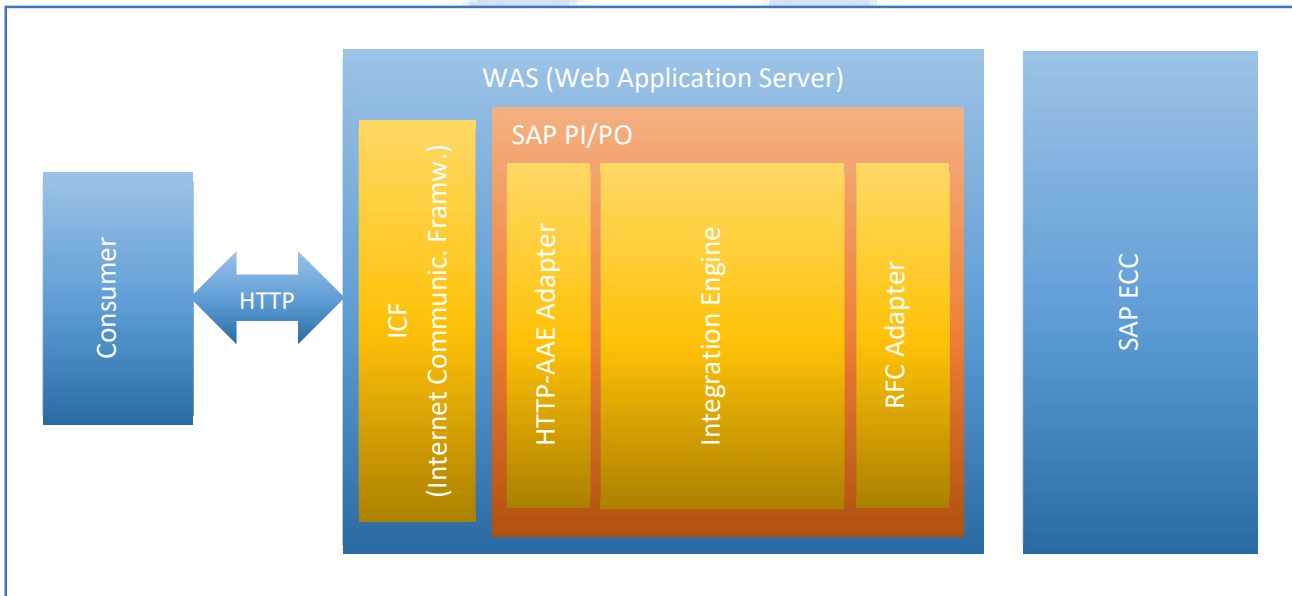


Figura 3

### Mapping request

Il passaggio dei parametri dal canale http al mapping avviene tramite la ASMA (Adapter Specific Message Attribute). A questo scopo, nel mapping in Figura 4, è stato introdotto un nodo fittizio, una *variabile*, DATI che implementa la UDF (User Defined Function) getDynConfHttpURLParamOne().

The screenshot shows the 'Display Message Mapping' interface. The top section displays metadata for the mapping: Name (MM\_KeepAlive\_request), Namespace (http://vodafone.com/xi/bridge/ISP/KeepAlive), and Software Component Version (SC\_VODAFONE 1.0 of nick4name.eu). Below this, the 'Definition' tab is active, showing the mapping between the 'Message Type: MT\_KeepAlive\_request' and the 'RFC Message: Z\_BOL\_KEEP\_ALIVE'. The 'Structure' table shows the mapping of fields, including a custom variable '\*\_DATI' which is implemented via a UDF 'getDynCo...'. The 'Structure' table for the RFC message lists fields like ZBANCA, ZCONTO, ZDIVISA, ZIDPAG, ZIMPORTO, and ZTIMESTAMP.

Structure	Occurrences	Type	Description
MT_KeepAlive_request	1..1	ns0:DT_KeepAlive_request	
ServiceID	1..1	xsd:integer	
TransactionID	1..1	xsd:integer	
Importo	1..1	xsd:decimal	
Divisa	1..1	xsd:string	
Timestamp	1..1	xsd:string	

Structure	Occurrences	Type	Description
Z_BOL_KEEP_ALIVE	1..1		
*_DATI	0..1		
ZBANCA	0..1	xsd:string	Bank key
ZCONTO	1..1	xsd:string	Proxy Data Element (Generated)
ZDIVISA	1..1	xsd:string	Currency Key
ZIDPAG	1..1	xsd:string	R/2 table
ZIMPORTO	1..1	xsd:string	BSI: Tax class amount
ZTIMESTAMP	1..1	xsd:string	Timestamp

Figura 4



La `getDynConfHttpURLParamOne()` legge dalla Dynamic Configuration, utilizzata dalla ASMA, da uno dei sei `URLParameters` previsti dall'HTTP-AAE sender adapter, `URLParameterOne`, il payload ricevuto tramite la query variable `DATI`. Con il payload restituito dalla UDF viene valorizzata la variabile `_DATI`

```
public String getDynConfHttpURLParamOne(String paramName, Container container) throws
StreamTransformationException{
// paramName = URLParamOne|URLParamTwo | URLParamThree | URLParamFour | URLParamFive | URLParamSix

    DynamicConfiguration dc = (DynamicConfiguration) container
        .getTransformationParameters().get(
            StreamTransformationConstants.DYNAMIC_CONFIGURATION);

    DynamicConfigurationKey k1 = DynamicConfigurationKey.create("http://sap.com/xi/XI/System",
        paramName);

    return dc.get(k1);
}
```

Sorgente 1

Tutti i TAG sono mappati secondo il template in Figura 5 dove la UDF `retTagValue()` restituisce il valore del TAG al secondo parametro a fronte dell'XML nella variabile `_DATI`.

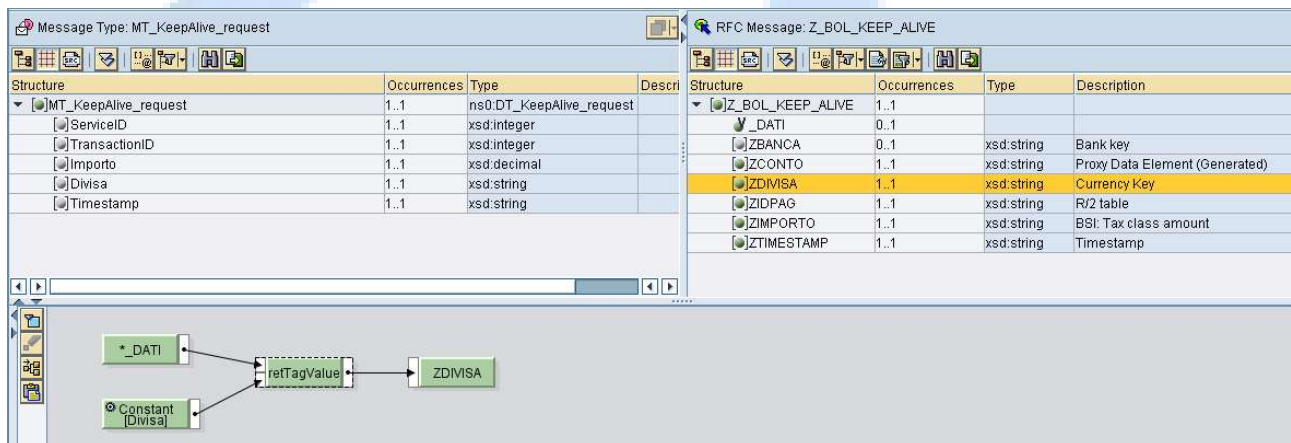


Figura 5

Con riferimento all'architettura in Figura 3, dal momento che a ricevere il messaggio REST è un componente esterno a PI, l'ICF, e quindi non un adapter, la modalità con la quale questo passa la chiamata a PI non consente a quest'ultimo di agganciare l'ICO, *Integrated Configuration*. Per questa ragione, è necessaria una prima conversione di adattamento del messaggio tramite un mapping **XSLT**.

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:msxsl="urn:schemas-microsoft-com:xslt" exclude-result-prefixes="msxsl">
  <xsl:output method="xml" indent="yes"/>

  <xsl:template match="/">
    <ns0:MT_KeepAlive_request xmlns:ns0="http://vodafone.com/xi/bridge/ISP/KeepAlive">
      <xsl:apply-templates select="/Input"/>
    </ns0:MT_KeepAlive_request>
  </xsl:template>

  <xsl:template match="Input">
    <xsl:copy-of select="*/>
  </xsl:template>
</xsl:stylesheet>
```

Sorgente 2 - ReRootKeepAlive\_request.xsl

L'XSLT request copia l'intero payload ricevuto sotto il tag

```
ns0:MT_KeepAlive_request xmlns:ns0=http://vodafone.com/xi/bridge/ISP/KeepAlive
```

così da adeguarlo alla Sender Interface *SI\_KeepAlive\_out*

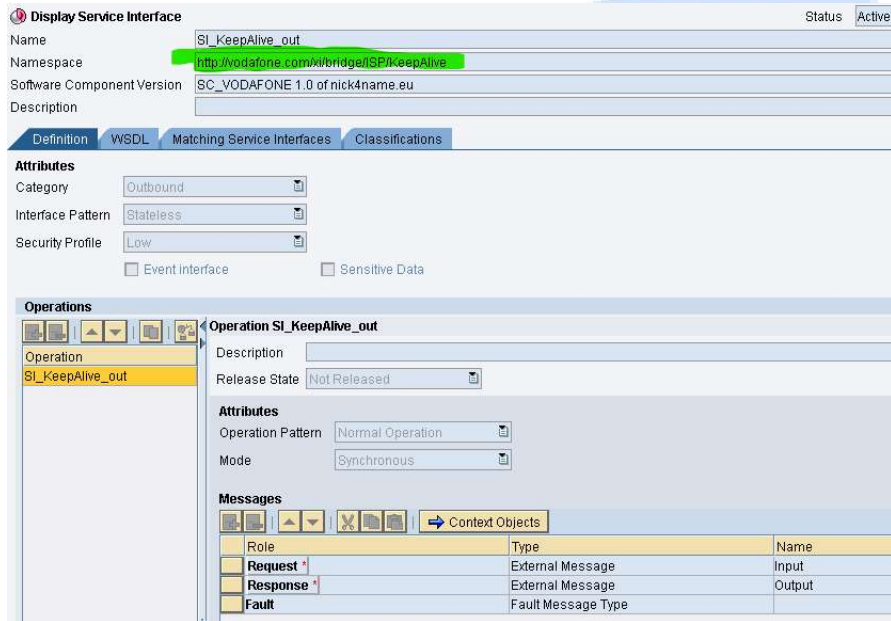


Figura 6

Il mapping XSLT deve precedere quello grafico come mostrato nell'Operation Mapping in Figura 7 nella sezione *Request*.

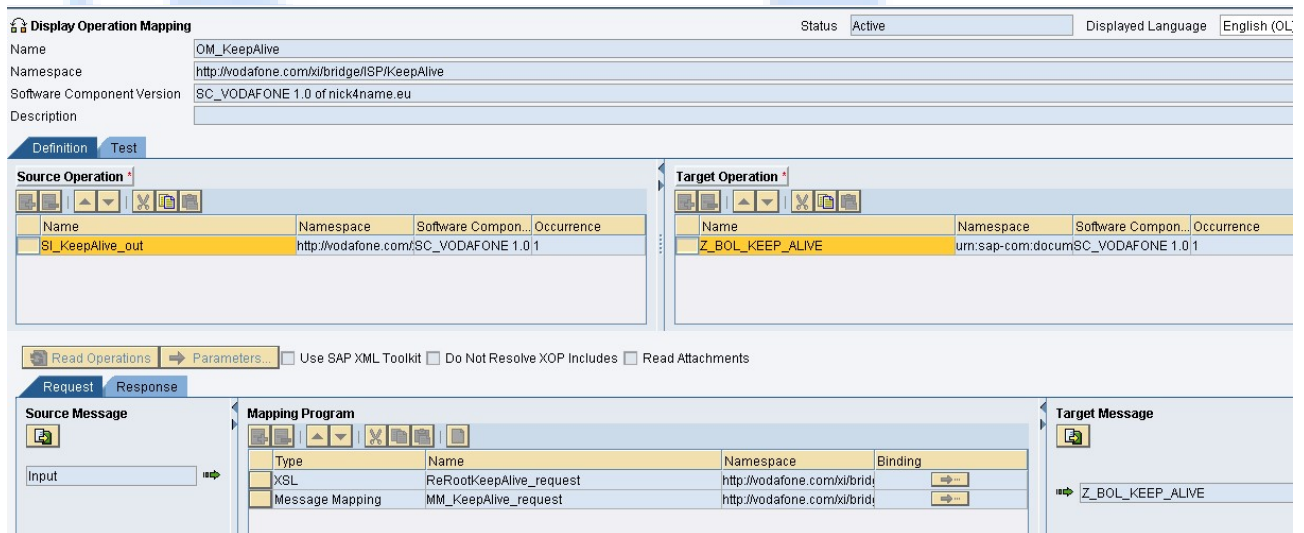


Figura 7

## Mapping response

Quando la RFC Z\_BOL\_KEEP\_ALIVE produce la risposta, restituisce il seguente payload

```
<?xml version="1.0" encoding="UTF-8"?><rfc:Z_BOL_KEEP_ALIVE.Response
xmlns:rfc="urn:sap-com:document:sap:rfc:functions"><CODERRORE>00</CODERRORE><MESSAGE>No
error.</MESSAGE><ZZBANCA></ZZBANCA><ZZCONTO>1</ZZCONTO><ZZDIVISA>eur</ZZD
IVISA><ZZIDPAG>1</ZZIDPAG><ZZIMPORTO>000000001</ZZIMPORTO><ZZTIMESTAMP>1<
/ZZTIMESTAMP></rfc:Z_BOL_KEEP_ALIVE.Response>
```

La risposta attesa dal consumer è la seguente e non prevede il namespace che sarebbe invece prodotto dal mapping grafico

```
<Output>
```

```
  <ServiceID/>
  <TransactionID/>
  <Importo/>
  <Divisa/>
  <Timestamp/>
  <CodErrore/>
  <Message/>
```

```
</Output>
```

Per conformare la risposta è necessario l'impiego di un nuovo mapping XSLT di response da eseguirsi a valle del mapping grafico. Lo script è il seguente

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:msxsl="urn:schemas-microsoft-com:xslt"
  xmlns:ns1="http://vodafone.com/xi/bridge/ISP/KeepAlive"
  exclude-result-prefixes="msxsl ns1">

  <xsl:output method="xml" indent="yes"/>

  <xsl:template match="/">
    <Output>
      <xsl:apply-templates select="ns1:MT_KeepAlive_response"/>
    </Output>
  </xsl:template>

  <xsl:template match="ns1:MT_KeepAlive_response"
    xmlns:ns1="http://vodafone.com/xi/bridge/ISP/KeepAlive">
    <ServiceID>
      <xsl:value-of select="ServiceID"/>
    </ServiceID>
    <TransactionID>
      <xsl:value-of select="TransactionID"/>
    </TransactionID>
    <Importo>
      <xsl:value-of select="Importo"/>
    </Importo>
    <Divisa>
      <xsl:value-of select="Divisa"/>
    </Divisa>
    <Timestamp>
      <xsl:value-of select="Timestamp"/>
    </Timestamp>
    <CodErrore>
      <xsl:value-of select="CodErrore"/>
    </CodErrore>
  </xsl:template>

```

```

</CodErrore>
<Message>
  <xsl:value-of select="Message"/>
</Message>
</xsl:template>
</xsl:stylesheet>

```

Sorgente 3 - *ReRootKeepAlive\_response.xsl*

L'Operation Mapping nella sezione *response* si configura come in Figura 8.

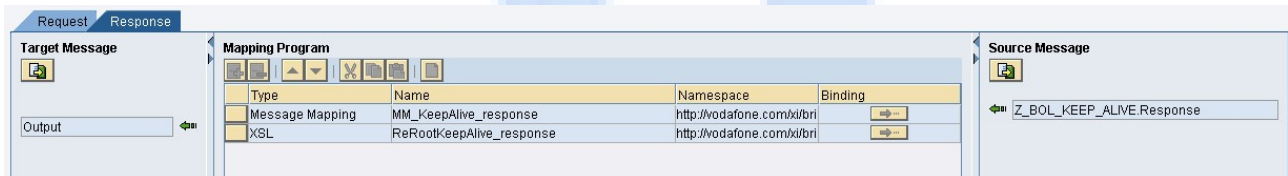


Figura 8

L'output generato dallo script *ReRootKeepAlive\_response.xsl* è il seguente, conformemente al requisito

```

<?xml version="1.0" encoding="UTF-8"?>
<Output>
  <ServiceID>1</ServiceID>
  <TransactionID>1</TransactionID>
  <Importo>000000001</Importo>
  <Divisa>eur</Divisa>
  <Timestamp>1</Timestamp>
  <CodErrore>00</CodErrore>
  <Message>No error.</Message>
</Output>

```

## Deploy degli XSLT

L'estensione degli script deve essere **XSL**.

Il deploy deve avvenire all'interno di un file **.zip** il quale deve essere importato nel nodo *Imported Archives*

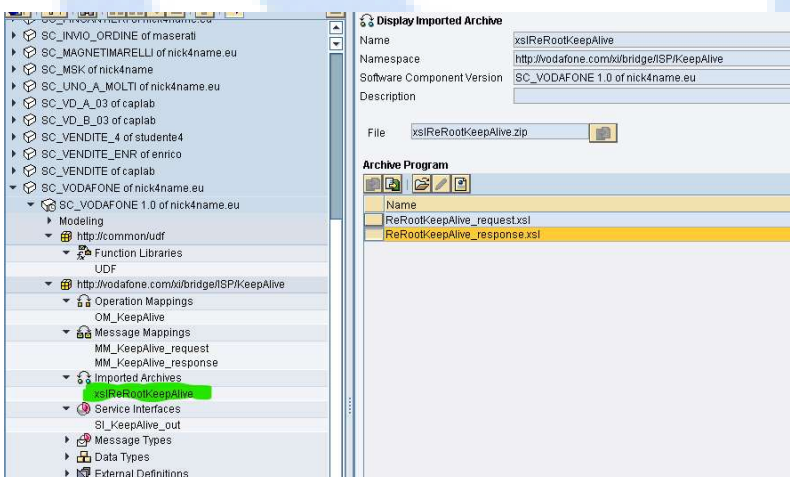


Figura 9



## Import delle strutture consumer

Le strutture REST di request e response sono importate tramite xsd nella sezione *External Definition*

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" attributeFormDefault="unqualified"
elementFormDefault="qualified">
  <xs:element name="Input">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ServiceID" type="xs:integer" />
        <xs:element name="TransactionID" type="xs:integer" />
        <xs:element name="Importo" type="xs:decimal" />
        <xs:element name="Divisa" type="xs:string" />
        <xs:element name="Timestamp" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

*EX\_KeepAlive\_request.xsd*

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" attributeFormDefault="unqualified"
elementFormDefault="qualified">
  <xs:element name="Output">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ServiceID" type="xs:integer" />
        <xs:element name="TransactionID" type="xs:integer" />
        <xs:element name="Importo" type="xs:decimal" />
        <xs:element name="Divisa" type="xs:string" />
        <xs:element name="Timestamp" type="xs:string" />
        <xs:element name="CodErrore" type="xs:string" />
        <xs:element name="Message" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

*EX\_KeepAlive\_response.xsd*

## Import delle strutture consumer

L'URL di chiamata del servizio va composto dal team PI sulla base di un template standard e comunicato al *consumer*. Il template è il seguente

`http://<pi_server>:<port>/HttpAdapter/HttpMessageServlet?interfaceNamespace=namespace_sender&interface=serviceinterface_sender&senderService=businesscomponent_sender&senderParty=&qos=BE`

e quindi l'URL è il seguente

`http://n4n-nw73.n4n.eu:50000/HttpAdapter/HttpMessageServlet?interfaceNamespace=http://vodafone.com/xi/bridge/ISP/KeepAlive&interface=SI_KeepAlive_out&senderService=BC_ISP&qos=BE&DATI=<?xml version="1.0" encoding="UTF-8"?><Input><ServiceID>1</ServiceID><TransactionID>1</TransactionID><Importo>1</Importo><Divisa>eur</Divisa><Timestamp>1</Timestamp></Input>`



## Integration Builder

Lo scenario è **CS\_VODA\_KEEPLIVE**.

**Display Configuration Scenario**

Configuration Scenario: CS\_VODA\_KEEPLIVE  
Description:

ES Repository Model | **Objects** | Configuration Overview

Search:  Go

Type	Object
Business System	BS_V17_800
Business Component	BC_ISP
Communication Channel	BC_ISP   CC_HTTPAAE_ISP_SND
Communication Channel	BS_V17_800   CC_RFC_LOOKUP_00_RCV
Integrated Configuration	BC_ISP   SI_KeepAlive_out

Figura 10

## Communication channel sender

**Display Communication Channel** Status: Active

Communication Channel: CC\_HTTPAAE\_ISP\_SND  
Party:  
Communication Component: BC\_ISP  
Description:

Parameters | Identifiers | Module

Adapter Type: HTTP\_AAE | http://sap.com/xi/XI/System | SAP BASIS 7.31  
 Sender  Receiver  
 Transport Protocol: HTTP 1.1  
 Message Protocol: POST  
 Adapter Engine: Central Adapter Engine

General | Advanced

**Request Details**

Set Form  
 Main Payload Parameter Name: DATI  
 Set Multipart

Figura 11



General **Advanced**

**Adapter-Specific Message Properties**

Set Adapter-Specific Message Properties

HTTP Header Fields

URL Parameters

Parameter 1 (URLParamOne)

Parameter 2 (URLParamTwo)

Parameter 3 (URLParamThree)

Parameter 4 (URLParamFour)

Parameter 5 (URLParamFive)

Parameter 6 (URLParamSix)

HTTP Header Fields (Synchronous Response)

**Adapter Status**

Status

**Additional Parameters**

Set Additional Parameters

Figura 12

Quello evidenziato in Figura 12 è il parametro che va citato nell'UDF di Figura 4 e che accoglie la *query variable* DATI con il payload di input.

## Communication channel receiver

**Display Communication Channel** Status

Communication Channel

Party

Communication Component

Description

**Parameters Identifiers Module**

Adapter Type

Sender  Receiver

Transport Protocol

Message Protocol

Adapter Engine

**Target Advanced**

**RFC Client Parameter**

RFC Server Type

Load Balancing

Application Server

System Number

Authentication Mode

Logon User

Logon Password

Logon Language

Logon Client

Maximum Connections

Advanced Mode

**RFC Metadata Repository Parameter**

Use Alternative RFC Metadata Repository

## Integrated configuration

**Display Integrated Configuration** Status: Active

**Sender**

Communication Party: \_\_\_\_\_  
 Communication Component: BC\_ISP  
 Interface: SI\_KeepAlive\_out  
 Namespace: http://vodafone.com/ki/bridge/ISP/KeepAlive

**Receiver**

Communication Party: \_\_\_\_\_  
 Communication Component: \_\_\_\_\_  
 Description: \_\_\_\_\_

**Configuration for Interface SI\_KeepAlive\_out**

Communication Channel: CC\_HTTPAAE\_ISP\_SND  
 Adapter Type: HTTP\_AAE | http://sap.com/ki/XU/System | SAP BASIS 7.31  
 Adapter Engine: Central Adapter Engine  
 Software Component Version of Sender Interface: SC\_VODAFONE 1.0 of nick4name.eu  
 Virus Scan: Use Global Configuration  
 Schema Validation:  No Validation  Validation by Adapter

Figura 13

**Receiver**

Type of Receiver Determination:  Standard  Extended

**Configured Receivers**

Condition	Communication Party	Communication Component
		BS_V17_800

Figura 14

**Receiver Interfaces**

Condition	Operation Mapping	Name	Namespace	Software Component Version	Multiplicity	Parameters
	OM_KeepAlive	Z_BOL_KEEP_ALIVE	urn:sap-com:document:sap:rfc:functio	SC_VODAFONE 1.0 of nick4name.eu	1	<input type="checkbox"/>

Figura 15

**Configuration for Interface Z\_BOL\_KEEP\_ALIVE | urn:sap-com:document:sap:rfc:functions | SC\_VODAFONE 1.0 of nick4name.eu**

Communication Channel: CC\_RFC\_LOOKUP\_00\_RCV  
 Adapter Type: RFC | http://sap.com/ki/XU/System | SAP BASIS 7.31  
 Adapter Engine: Central Adapter Engine  
 Software Component Version of Receiver Interface: SC\_VODAFONE 1.0 of nick4name.eu  
 Virus Scan: Use Global Configuration  
 Schema Validation:  No Validation  Validation by Adapter

**Header Mapping**

Sender Communication Party  
 Sender Communication Component  
 Receiver Communication Party  
 Receiver Communication Component

**Principal Propagation Properties**

Propagate Principal

Figura 16



## Test dello scenario

Tramite soapUI.

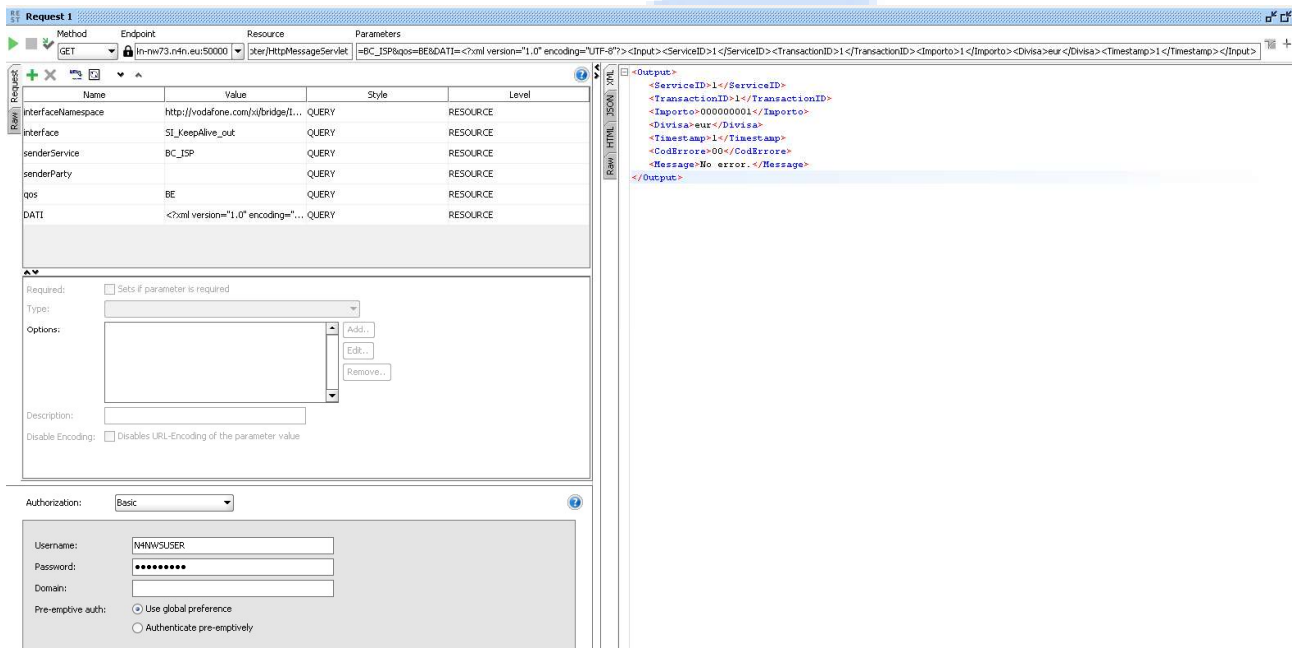


Figura 17

## Limitazioni dell'adapter HTTP-AAE rispetto all'adapter REST

Di seguito, alcune limitazioni dell'adapter HTTP-AAE della 7.31 rispetto all'adapter REST disponibile da SAP PI 7.4.

Adapter HTTP-AAE	Adapter REST
Supporta solo GET e POST	Supporta tutti i verbi CRUD
Struttura vincolante dell'URL di chiamata	Nessun vincolo nella definizione dell'URL
I payload di request e, eventualmente, di response devono essere XML. Non è supportato JSON e neppure valori non strutturati	Supporta XML, JSON e valori non strutturati
Il numero massimo di valori che è possibile passare dipende dagli <i>URLParam*</i> nel channel sender ed è pari a 6.	Nessun limite

Tabella 1