

**WEB APP “18APP”
LINEE GUIDA PER ESERCENTI**

ottobre 2016

La Legge n. 208 del 28 dicembre 2015 assegna un bonus di 500 euro ai cittadini residenti in Italia (in possesso, ove previsto, di permesso di soggiorno in corso di validità) che compiono diciotto anni nel 2016.

La gestione del bonus può avvenire tramite qualunque tipo device (smartphone, tablet, computer) connesso ad Internet, sia da parte dei diciottenni che da parte degli esercenti (soggetti pubblici o privati) presso i quali acquistare i prodotti previsti riconducibili alle seguenti categorie: cinema, concerti, eventi culturali, libri, musei, monumenti, parchi naturali ed aree archeologiche, teatro e danza.

In particolare, nella web app gli esercenti avranno a disposizione le seguenti funzionalità:

1. registrazione al servizio;
2. vendita attraverso esercizio fisico oppure on line
3. controllo e riscossione dei buoni di acquisto
4. fatturazione

1. REGISTRAZIONE AL SERVIZIO

“18app” può essere utilizzato solo dagli esercenti con credenziali di accesso ai Servizi Telematici dell’Agenzia delle Entrate e che hanno incaricato dei soggetti ad operare per proprio conto. Gli esercenti si devono, quindi, obbligatoriamente registrare alla web app per poter vendere i propri prodotti.

Le fasi della registrazione:

- **entro il 30 giugno 2017** registrazione alla web app con indicazione dei dati generali dell’esercente (dati anagrafici, di residenza, codice ATECO, recapiti telefonici, mail ed eventuale sito internet, accettazione della normativa vigente).
- **entro il 30 giugno 2017** indicazione di categorie di prodotto e prodotti vendibili sia nel proprio esercizio fisico che in quello online.

Al termine della registrazione, il sistema assegnerà automaticamente a ciascun esercente un “codice esercente” di 5 caratteri alfanumerici da utilizzare in fase di riscossione dei buoni. Gli esercenti possono modificare le informazioni registrate ogni qual volta vorranno (ad esempio: variazione dati, inserimento/cancellazione dei punti vendita fisici).

2. VENDITA

La web app mette a disposizione degli esercenti 2 tipologie di vendita: nell'esercizio fisico oppure on-line (in fase di registrazione è possibile indicare l'utilizzo anche di entrambe le tipologie). In caso di vendita nell'esercizio fisico, l'esercente può indicare tutti i punti vendita dove i diciottenni potranno esibire i "buoni" di spesa con l'indicazione delle categorie di prodotto e dei prodotti disponibili. I punti vendita potranno essere geolocalizzati nell'applicazione dei 18enni.

In caso di vendita online, l'esercente dovrà prima dotarsi di un certificato da applicare al web-service (**vd. allegato 1**) richiamabile dai propri sistemi per la verifica e validazione dei buoni di spesa.

In particolare dovrà:

- richiedere il certificato in fase di registrazione;
- scaricare il certificato nei propri sistemi;
- verificare l'installazione del certificato tramite specifica chiamata del web-service, come da indicazioni in allegato1.

Si precisa che sarà possibile anche per gli esercenti che scelgono la vendita presso l'esercizio fisico scegliere l'utilizzo delle API services che prevedono il download ed installazione di un certificato nei propri sistemi, secondo il procedimento descritto per la vendita online.

Si precisa, inoltre, che per gli esercenti che utilizzeranno il web-services per la validazione dei buoni sono previsti i seguenti stati di lavorazione legati alle attività di configurazione dei propri sistemi in cui dovrà essere installato il certificato:

- da attivare → certificato da scaricare dall' applicazione 18app
- attivabile → certificato scaricato, ma non installato
- attivo → certificato installato (vd. allegato 1).

3. CONTROLLO E RISCOSSIONE DEI BUONI D'ACQUISTO

A partire da settembre 2016 e fino al 31 dicembre 2017 gli esercenti possono verificare e riscuotere i buoni generati dai 18enni nelle funzionalità dell'applicazione loro dedicate ed identificati da uno specifico codice.

La verifica e validazione è differente in caso di vendita nell'esercizio fisico oppure online.

Verifica in caso di vendita nell'esercizio fisico

Nell'area non autenticata della web app gli esercenti possono;

- indicare il codice del buono esibito dal beneficiario insieme al "codice esercente";
- leggere il QR code o il codice a barre del buono.

In caso di riconoscimento del codice del buono, la web app restituisce le informazioni generali dell'acquisto (categoria di prodotto, bene, importo e cognome/nome del beneficiario); eventualmente l'esercente può verificare il beneficiario anche tramite l'esibizione di un documento d'identità. A questo punto, l'esercente può validare il buono esibito consentendo di "scalare" l'importo dal bonus totale di 500 euro del diciottenne.

Verifica in caso di vendita online

Nel proprio sito internet, l'esercente consente al diciottenne di indicare il codice del buono da verificare e riscuotere tramite chiamata al web-service le cui specifiche sono reperibili nell'allegato 1. In questo caso è prevista la possibilità di "congelare" il buono in attesa di verificare la disponibilità dei prodotti nel proprio magazzino o per altre specifiche situazioni; a valle di questa verifica, l'esercente può convalidare tutto o parte dell'importo di acquisto consentendo di "scalare" dal bonus totale del diciottenne solo l'importo validato.

Gli esercenti possono visualizzare la lista dei buoni riscossi utilizzando eventuali filtri di ricerca per data di generazione del buono e/o per tipologia di vendita (esercizio fisico oppure online), per categoria di prodotto e per buoni fatturati o data fatturare.

Nelle liste verranno visualizzate le seguenti informazioni:

- categoria di prodotto
- dettagli del prodotto acquistato (tipologia ed esercente/punto vendita)
- identificativo del buono
- identificativo della fattura elettronica in cui è stato indicato il bene acquistato

- informazione sull'eventuale download della lista esportabile in formato .csv.

Le liste visualizzate potranno essere scaricate in formato .csv per facilitare la preparazione della fattura elettronica.

Nella "lista dei buoni", resa disponibile nella web app 18app, gli esercenti possono, inoltre, verificare se il numero di buoni convalidati coincide con quello delle fatture.

4. FATTURAZIONE

Agli esercenti viene riconosciuto un credito per ogni buono riscosso da dichiarare in sede di fatturazione elettronica.

L'esercente dovrà preparare la fattura in formato elettronico secondo il tracciato stabilito per la fatturazione elettronica verso la pubblica amministrazione ("Schema del file xml FatturaPA - versione 1.1" reperibile nel sito www.fatturapa.gov.it, sezione Norme e regole, Documentazione FatturaPA).

La valorizzazione degli elementi del tracciato xml dovrà rispettare i requisiti formali e di obbligatorio previsti e, con particolare riferimento al blocco 2.2.1 <DettaglioLinee>, dovrà tener conto delle indicazioni **in corso di definizione** riportate nel documento "**Linee guida fatturazione esercente**" disponibili nell'applicazione 18app.

ALLEGATO 1

Allegato 1 – Specifica WS online

18app

Servizio Web per verifica voucher degli esercenti

Gli esercenti registrati al servizio “18app” per la vendita online potranno validare nei propri sistemi i voucher (abbiamo deciso di chiamarli buoni?) di acquisto di beni da parte dei 18enni utilizzando uno specifico servizio web di cui si forniscono di seguito le specifiche di utilizzo.

In merito agli endpoint, si precisa che al momento l'indirizzo referenziato nel wsdl (<http://bonus.mibact.it/VerificaVoucher>) è provvisorio.

Il servizio web di **verifica del voucher (buono?)** da parte degli esercenti rende disponibili 2 principali operazioni (“**Check**” e “**Confirm**”). La seguente descrizione delle operazioni è necessaria per valorizzare adeguatamente i campi di input. Pertanto le operazioni esposte dall'interfaccia wsdl possono essere descritte come segue:

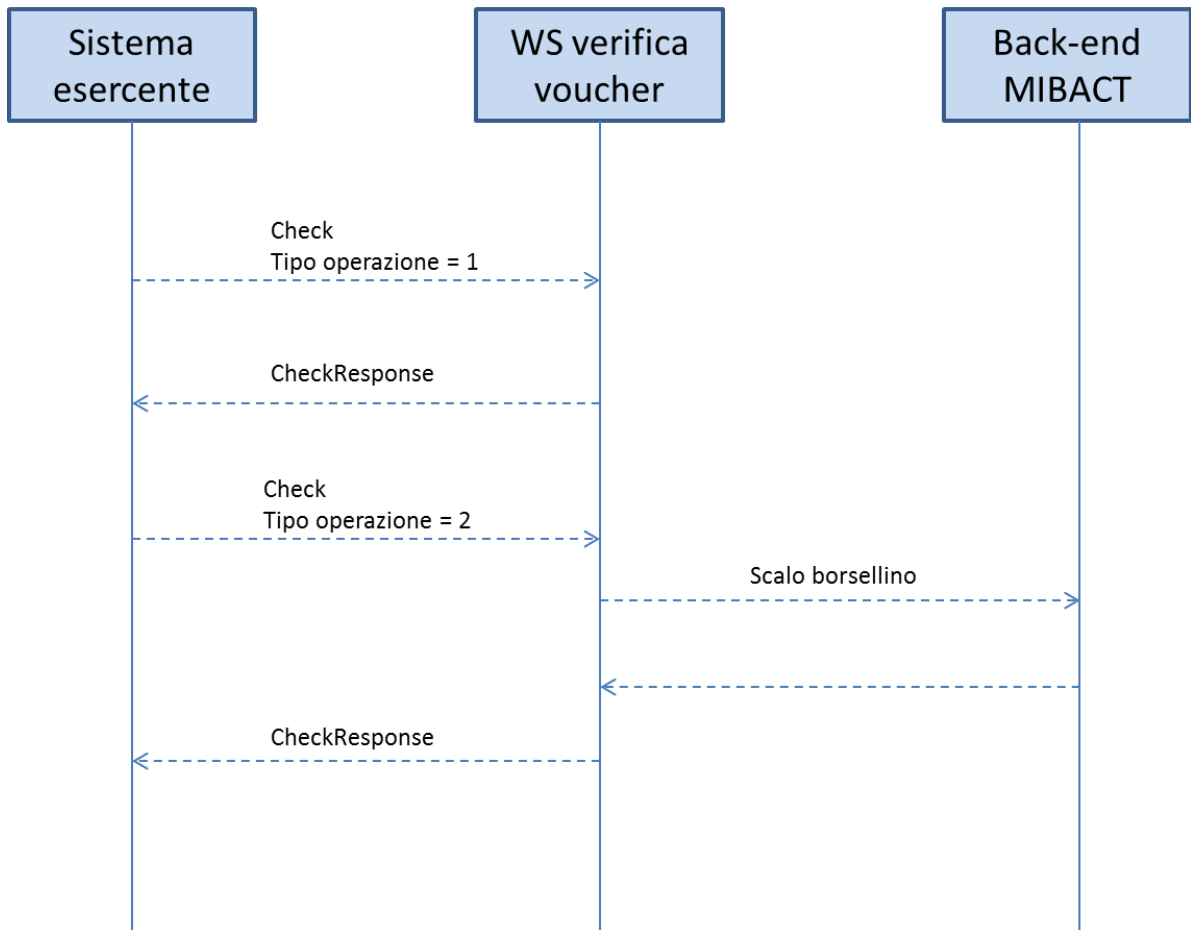
1. Check

INPUT:	tipo operazione	“1”, “2”, “3”
	codice voucher	
OUTPUT:	nominativo beneficiario	CF o Nome e Cognome
	partita IVA esercente	
	ambito	cinema, teatro, libreria...
	bene	libri, spettacoli...
	importo	importo totale del voucher

Se **tipo operazione** verrà valorizzato con “1”, il check del voucher restituirà all'esercente i campi previsti in output senza consumare il voucher e quindi senza scalare l'importo dal borsellino del beneficiario. Questa modalità di utilizzo dell'operazione non è obbligatoria, ma lascia all'esercente la possibilità di eseguire un controllo tra il nominativo del beneficiario e quello del suo cliente in sessione.

Se **tipo operazione** verrà valorizzato con “2”, il check del voucher consumerà direttamente l'importo, scalandolo dal borsellino del beneficiario, e restituirà comunque le informazioni previste in output. L'esercente potrà scegliere di usare direttamente questa modalità oppure effettuare due chiamate successive: la prima per il controllo del beneficiario e la seconda per l'effettivo utilizzo del voucher.

Il sequence diagram seguente descrive ad alto livello l'interazione tra i vari sistemi coinvolti, nei casi fin qui descritti:



Valorizzando il campo **tipo operazione** con “3”, l’esercente potrà impegnare l’importo del voucher congelandolo per il tempo necessario ad eseguire un controllo di disponibilità in magazzino o per altre situazioni specifiche.

In questo caso l’importo del voucher non sarà scalato da borsellino del beneficiario, ma risulterà comunque impegnato e non utilizzabile per altri acquisti.

Per poter sbloccare il voucher ed utilizzarne tutto l’importo o solo parte di esso, l’esercente dovrà invocare l’operazione di **Confirm** di seguito descritta.

2. Confirm

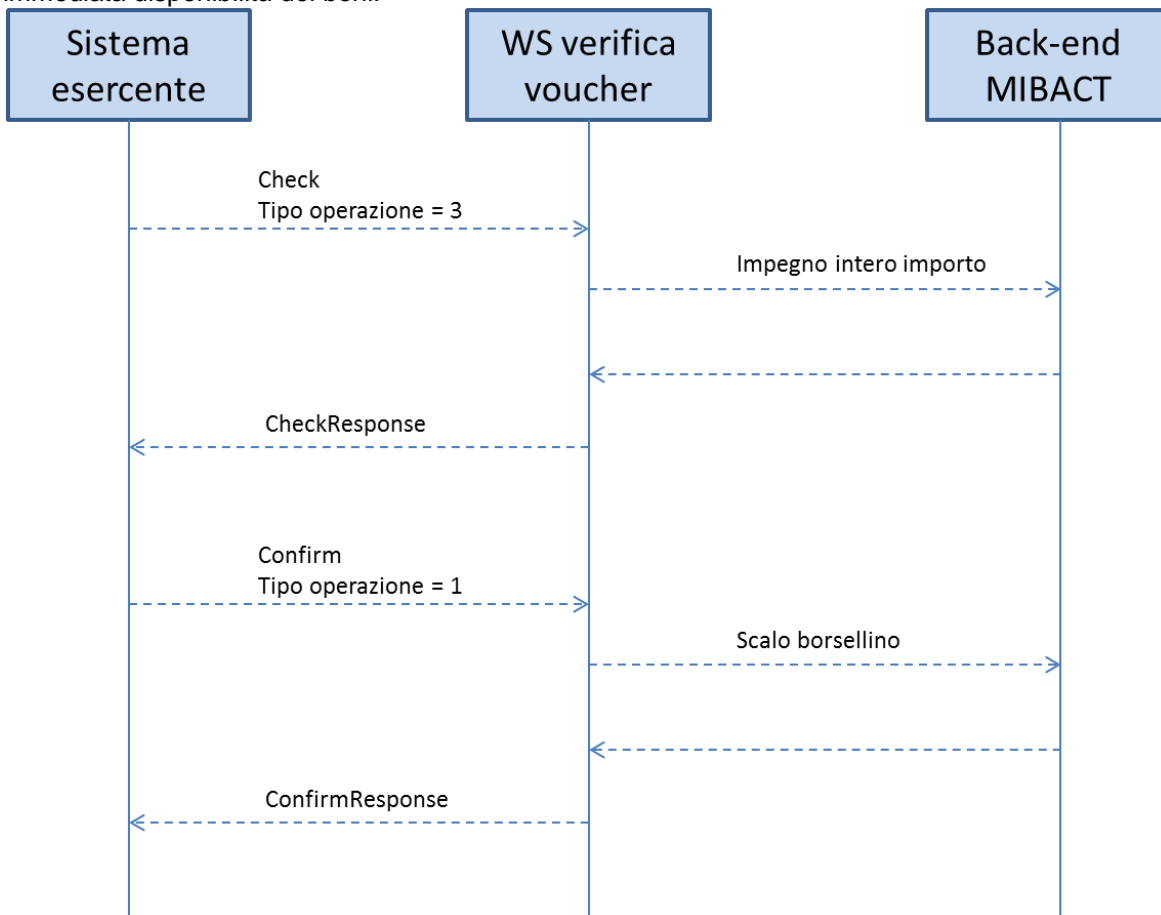
INPUT:	tipo operazione	“1”
	codice voucher	
	importo	<i>importo confermato dall’esercente</i>
OUTPUT:	esito	

In questa versione del servizio il **tipo operazione** verrà valorizzato sempre con “1” e l’esercente potrà comunicare la quota utilizzata rispetto all’importo totale del voucher, momentaneamente impegnato. Il sistema scalerà l’importo dal borsellino del beneficiario, riaccreditando la parte non

utilizzata, calcolata come differenza tra il valore totale del voucher e l'importo comunicato dall'esercente.

L'**esito** dell'operazione ("OK" / "KO") sarà restituito all'esercente che potrà eventualmente fornire un feedback al beneficiario.

Il seguente sequence diagram descrive ad alto livello l'interazione tra i sistemi nel caso di non immediata disponibilità dei beni:



Modalità di autenticazione

Per consumare il web service di verifica del voucher, ogni esercente dovrà essere dotato di un **certificato di autenticazione** da installare nel proprio client del servizio e da utilizzare nella chiamata SOAP per effettuare l'autenticazione in modalità SSL con certificato client.

Tale certificato X509 sarà generabile e scaricabile in formato .cer direttamente tramite l'applicazione web dedicata agli esercenti, in area autenticata. In particolare il processo di generazione del certificato prevede due step:

1. Il primo step di richiesta del certificato; a seguito di questa operazione il sistema prende in carico la richiesta.
2. Il secondo step di verifica esito della richiesta; questa operazione controlla se è pronto il certificato emesso da CA dedicata ed eventualmente lo rende disponibile per il download.

Durante il primo step sarà necessario caricare un file .der rappresentante la richiesta di certificato alla CA dedicata al progetto. Tale csr deve presentare le seguenti caratteristiche:

- Algoritmo generazione chiavi: RSA
- Lunghezza chiavi: 2048 bit

Una volta scaricato il certificato X509 va installato, insieme alla corrispondente chiave privata, nel client utilizzato per il servizio di verifica voucher. Pertanto l'evento di download del certificato non può rappresentare la definitiva attivazione dell'esercente. E' stato previsto uno step di attivazione, di tipo "Check" con i seguenti valori di input:

- tipo operazione = 1
- codice voucher = 11aa22bb

Questa operazione equivale ad una transazione di attivazione, il cui unico effetto è quello di portare l'esercente nello stato attivo. Da questo momento in poi i beneficiari potranno generare voucher reali per tale esercente.

Endpoint del servizio

Il servizio risponde ai seguenti endpoint

<https://wstest.18app.italia.it/VerificaVoucherWEB/VerificaVoucher> (ambiente di prova)
<https://ws.18app.italia.it/VerificaVoucherWEB/VerificaVoucher> (ambiente reale)

Codici di errore

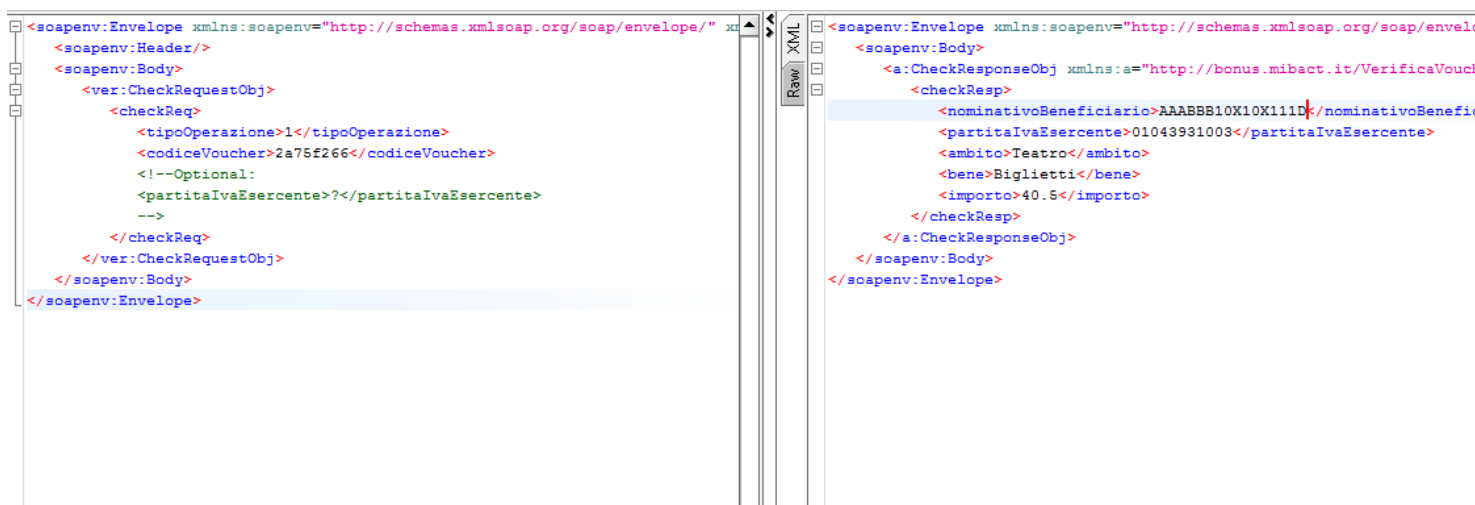
La seguente tabella rappresenta i possibili errori gestiti dal sistema:

Codice/Code	Descrizione/Description
01	Errore nel formato dei parametri in input, verificarli e riprovare
	Error in the input parameters, check and try again
02	Il buono richiesto non è disponibile sul sistema o è già stato riscosso o annullato
	The requested voucher is not available on the system. It could be already collected or canceled
03	Impossibile attivare l'esercente. Verificare che i dati siano corretti e che l'esercente non sia già stato attivato
	Impossible to activate the user. Please verify input parameters and that the user has not been already activated.
04	L'importo richiesto è superiore all'importo del buono selezionato
	The amount claimed is greater than the amount of the selected voucher
05	Non si può verificare o consumare il buono poichè l'esercente risulta non attivo
	User inactive, voucher impossible to verify.
06	Ambito e bene del buono non coincidono con ambiti e beni trattati dall'esercente
	Category and type of this voucher are not aligned with category and type managed by the user.

Esempi di request/response

Di seguito si riportano due esempi di request e relativa response, sia per l'operation "Check" che per l'operation "Confirm".

"Check"



The image shows two XML editors side-by-side. The left editor displays a SOAP request XML, and the right editor displays the corresponding SOAP response XML. Both are structured as SOAP envelopes with headers and bodies. The request body contains a 'CheckRequestObj' with fields like 'tipoOperazione', 'codiceVoucher', and 'partitaIvaEsercente'. The response body contains a 'CheckResponseObj' with fields like 'nominativoBeneficiario', 'partitaIvaEsercente', 'ambito', 'bene', and 'importo'.

Check request:

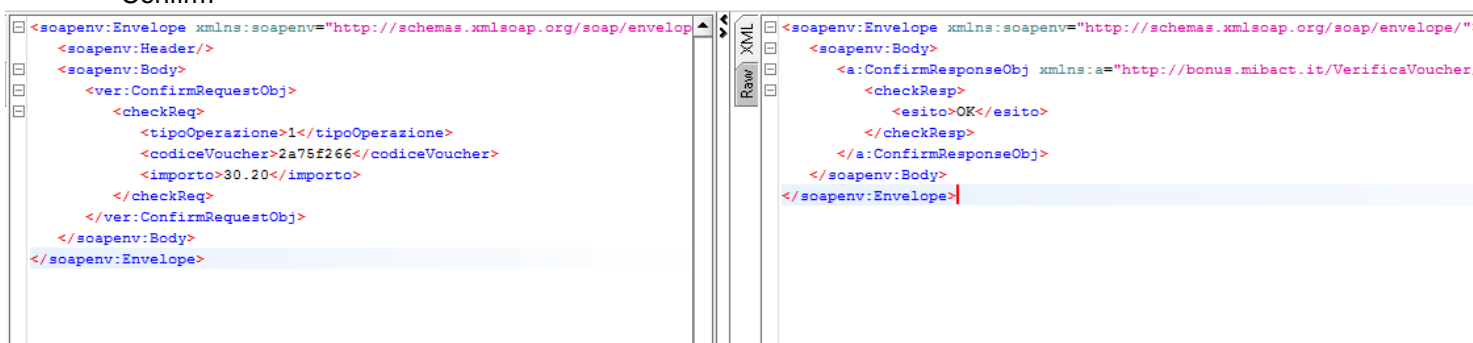
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ver="http://bonus.mibact.it/VerificaVoucher/">
  <soapenv:Header/>
  <soapenv:Body>
    <ver:CheckRequestObj>
      <checkReq>
        <tipoOperazione>1</tipoOperazione>
        <codiceVoucher>2a75f266</codiceVoucher>
        <!--Optional:
        <partitaIvaEsercente>?</partitaIvaEsercente>
        -->
      </checkReq>
    </ver:CheckRequestObj>
  </soapenv:Body>
</soapenv:Envelope>
```

Check response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <a:CheckResponseObj xmlns:a="http://bonus.mibact.it/VerificaVoucher/">
      <checkResp>
        <nominativoBeneficiario>AAABBB10X10X111D</nominativoBeneficiario>
        <partitaIvaEsercente>01043931003</partitaIvaEsercente>
      </checkResp>
    </a:CheckResponseObj>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<ambito>Teatro</ambito>  
<bene>Biglietti</bene>  
<importo>40.5</importo>  
</checkResp>  
</a:CheckResponseObj>  
</soapenv:Body>  
</soapenv:Envelope>
```

“Confirm”



Confirm request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:ver="http://bonus.mibact.it/VerificaVoucher/">  
  <soapenv:Header/>  
  <soapenv:Body>  
    <ver:ConfirmRequestObj>  
      <checkReq>  
        <tipoOperazione>1</tipoOperazione>  
        <codiceVoucher>2a75f266</codiceVoucher>  
        <importo>30.20</importo>  
      </checkReq>  
    </ver:ConfirmRequestObj>  
  </soapenv:Body>  
</soapenv:Envelope>
```

Confirm response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">  
  <soapenv:Body>  
    <a:ConfirmResponseObj xmlns:a="http://bonus.mibact.it/VerificaVoucher/">  
      <checkResp>  
        <esito>OK</esito>  
      </checkResp>  
    </a:ConfirmResponseObj>  
  </soapenv:Body>  
</soapenv:Envelope>
```

WSDL VerificaVoucher.wsdl

targetnamespace: <http://bonus.mibact.it/VerificaVoucher/>

services	bindings	porttypes	messages	types
VerificaVoucher	VerificaVoucherSOAP	VerificaVoucher	CheckRequest	Check
			CheckResponse	CheckRequestObj
			ConfirmRequest	CheckResponse
			ConfirmResponse	CheckResponseObj
				Confirm
				ConfirmRequestObj
				ConfirmResponse
				ConfirmResponseObj

attributeFormDefault:

elementFormDefault:

targetNamespace: <http://bonus.mibact.it/VerificaVoucher/>

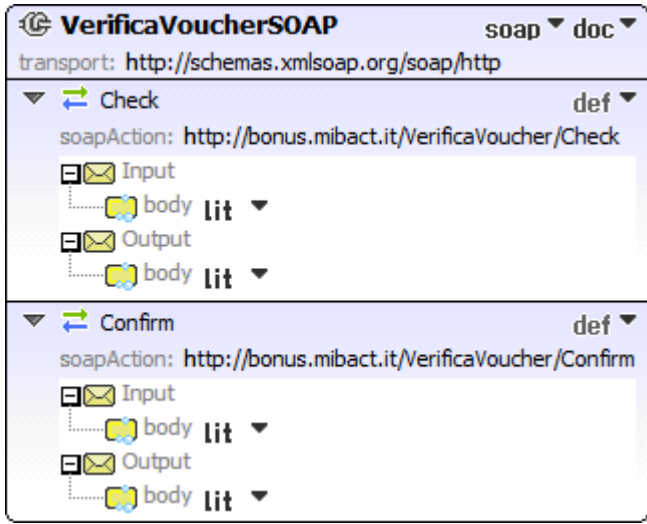
Elements	Complex types
CheckRequestObj	Check
CheckResponseObj	CheckResponse
ConfirmRequestObj	Confirm
ConfirmResponseObj	ConfirmResponse

service VerificaVoucher

diagram	
ports	VerificaVoucherSOAP binding tns:VerificaVoucherSOAP extensibility <code><soap:address location="https://bonus.mibact.it/VerificaVoucherWEB/VerificaVoucher"/></code>
source	<pre><wsdl:service name="VerificaVoucher"> <wsdl:port name="VerificaVoucherSOAP" binding="tns:VerificaVoucherSOAP"></pre>

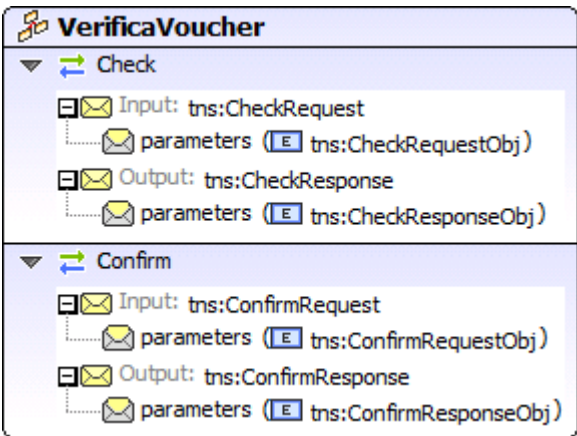
	<pre><soap:address location="https://bonus.mibact.it/VerificaVoucherWEB/VerificaVoucher"/> </wsdl:port> </wsdl:service></pre>
--	---

binding **VerificaVoucherSOAP**

diagram	
type	tns:VerificaVoucher
extensibility	<code><soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/></code>
operations	<p>Check</p> <p>extensibility <code><soap:operation soapAction="http://bonus.mibact.it/VerificaVoucher/Check"/></code></p> <p>input <code><soap:body use="literal"/></code></p> <p>output <code><soap:body use="literal"/></code></p> <p>Confirm</p> <p>extensibility <code><soap:operation soapAction="http://bonus.mibact.it/VerificaVoucher/Confirm"/></code></p> <p>input <code><soap:body use="literal"/></code></p> <p>output <code><soap:body use="literal"/></code></p>
used by	Port VerificaVoucherSOAP in Service VerificaVoucher
source	<pre><wsdl:binding name="VerificaVoucherSOAP" type="tns:VerificaVoucher"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="Check"> <soap:operation soapAction="http://bonus.mibact.it/VerificaVoucher/Check"/> </wsdl:operation> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:binding></pre>

	<pre> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> <wsdl:operation name="Confirm"> <soap:operation soapAction="http://bonus.mibact.it/VerificaVoucher/Confirm"/> <wsdl:input> <soap:body use="literal"/> </wsdl:input> <wsdl:output> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> </pre>
--	--

porttype **VerificaVoucher**

diagram	 <p>The diagram shows a port type named VerificaVoucher. It contains two operations: Check and Confirm. Check operation: - Input: tns:CheckRequest - Parameters: tns:CheckRequestObj - Output: tns:CheckResponse - Parameters: tns:CheckResponseObj Confirm operation: - Input: tns:ConfirmRequest - Parameters: tns:ConfirmRequestObj - Output: tns:ConfirmResponse - Parameters: tns:ConfirmResponseObj</p>
operations	<p>Check input tns:CheckRequest output tns:CheckResponse</p> <p>Confirm input tns:ConfirmRequest output tns:ConfirmResponse</p>
used by	binding VerificaVoucherSOAP
source	<pre> <wsdl:portType name="VerificaVoucher"> <wsdl:operation name="Check"> <wsdl:input message="tns:CheckRequest"/> <wsdl:output message="tns:CheckResponse"/> </wsdl:operation> <wsdl:operation name="Confirm"> </pre>

	<pre><wsdl:input message="tns:ConfirmRequest"/> <wsdl:output message="tns:ConfirmResponse"/> </wsdl:operation> </wsdl:portType></pre>
--	---

message **CheckRequest**

parts	parameters element tns:CheckRequestObj
used by	Operation Check in PortType VerificaVoucher
source	<pre><wsdl:message name="CheckRequest"> <wsdl:part name="parameters" element="tns:CheckRequestObj"/> </wsdl:message></pre>

message **CheckResponse**

parts	parameters element tns:CheckResponseObj
used by	Operation Check in PortType VerificaVoucher
source	<pre><wsdl:message name="CheckResponse"> <wsdl:part name="parameters" element="tns:CheckResponseObj"/> </wsdl:message></pre>


message **ConfirmRequest**

parts	parameters element tns:ConfirmRequestObj
used by	Operation Confirm in PortType VerificaVoucher
source	<pre><wsdl:message name="ConfirmRequest"> <wsdl:part name="parameters" element="tns:ConfirmRequestObj"/> </wsdl:message></pre>

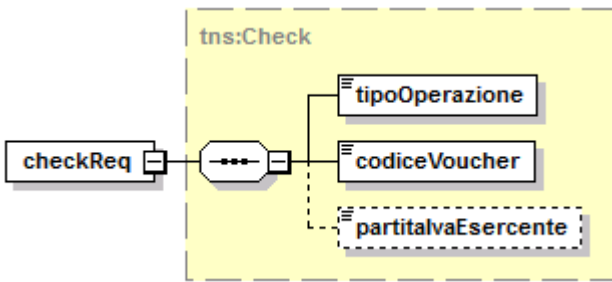
message **ConfirmResponse**

parts	parameters element tns:ConfirmResponseObj
used by	Operation Confirm in PortType VerificaVoucher
source	<pre><wsdl:message name="ConfirmResponse"> <wsdl:part name="parameters" element="tns:ConfirmResponseObj"/> </wsdl:message></pre>

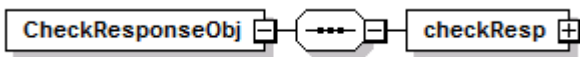
element CheckRequestObj

diagram	
namespace	http://bonus.mibact.it/VerificaVoucher/
properties	content complex
children	checkReq
source	<pre><xsd:element name="CheckRequestObj"> <xsd:complexType> <xsd:sequence> <xsd:element name="checkReq" type="tns:Check"/> </xsd:sequence> </xsd:complexType> </xsd:element></pre>

element CheckRequestObj/checkReq

diagram	
type	tns:Check
properties	content complex
children	tipoOperazione codiceVoucher partitalvaEsercente
source	<pre><xsd:element name="checkReq" type="tns:Check"/></pre>

element CheckResponseObj

diagram	
namespace	http://bonus.mibact.it/VerificaVoucher/
properties	content complex
children	checkResp
source	<pre><xsd:element name="CheckResponseObj"></pre>

	<pre><xsd:complexType> <xsd:sequence> <xsd:element name="checkResp" type="tns:CheckResponse"/> </xsd:sequence> </xsd:complexType> </xsd:element></pre>
--	--

element **CheckResponseObj/checkResp**

diagram	
type	tns:CheckResponse
properties	content complex
children	nominativoBeneficiario partitalvaEsercente ambito bene importo
source	<pre><xsd:element name="checkResp" type="tns:CheckResponse"/></pre>

element **ConfirmRequestObj**

diagram	
namespace	http://bonus.mibact.it/VerificaVoucher/
properties	content complex
children	checkReq
source	<pre><xsd:element name="ConfirmRequestObj"> <xsd:complexType> <xsd:sequence> <xsd:element name="checkReq" type="tns:Confirm"/> </xsd:sequence> </xsd:complexType> </xsd:element></pre>

element **ConfirmRequestObj/checkReq**

diagram	<p>The diagram shows a box labeled 'checkReq' connected to a dashed box labeled 'tns:Confirm'. Inside the dashed box, there are three child elements: 'tipoOperazione', 'codiceVoucher', and 'importo'.</p>
type	tns:Confirm
properties	content complex
children	tipoOperazione codiceVoucher importo
source	<code><xsd:element name="checkReq" type="tns:Confirm"/></code>

element **ConfirmResponseObj**

diagram	<p>The diagram shows a box labeled 'ConfirmResponseObj' connected to a dashed box containing a box labeled 'checkResp'.</p>
namespace	<code>http://bonus.mibact.it/VerificaVoucher/</code>
properties	content complex
children	checkResp
source	<pre> <xsd:element name="ConfirmResponseObj"> <xsd:complexType> <xsd:sequence> <xsd:element name="checkResp" type="tns:ConfirmResponse"/> </xsd:sequence> </xsd:complexType> </xsd:element> </pre>

element **ConfirmResponseObj/checkResp**

diagram	<p>The diagram shows a box labeled 'checkResp' connected to a dashed box labeled 'tns:ConfirmResponse'. Inside the dashed box, there is one child element: 'esito'.</p>
type	tns:ConfirmResponse

properties	content complex
children	esito
source	<code><xsd:element name="checkResp" type="tns:ConfirmResponse"/></code>

complexType Check

diagram	
namespace	http://bonus.mibact.it/VerificaVoucher/
children	tipoOperazione codiceVoucher partitalvaEsercente
used by	element CheckRequestObj/checkReq
source	<pre> <xsd:complexType name="Check"> <xsd:sequence> <xsd:element name="tipoOperazione" type="xsd:string" minOccurs="1" maxOccurs="1"/> <xsd:element name="codiceVoucher" type="xsd:string" minOccurs="1" maxOccurs="1"/> <xsd:element name="partitalvaEsercente" type="xsd:string" minOccurs="0" maxOccurs="1"/> </xsd:sequence> </xsd:complexType> </pre>

element Check/tipoOperazione

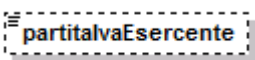
diagram	
type	xsd:string
properties	content simple
source	<pre> <xsd:element name="tipoOperazione" type="xsd:string" minOccurs="1" maxOccurs="1"/> </pre>

element Check/codiceVoucher

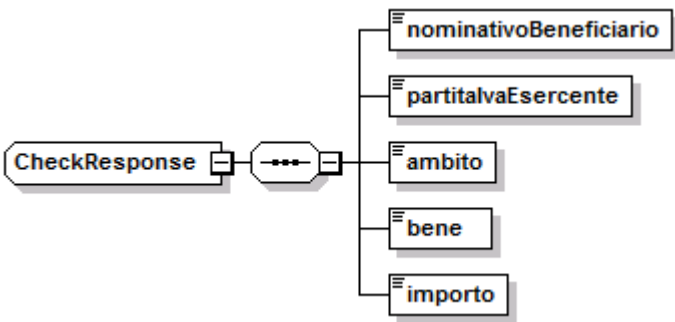
diagram	
---------	--

type	xsd:string
properties	content simple
source	<code><xsd:element name="codiceVoucher" type="xsd:string" minOccurs="1" maxOccurs="1"/></code>

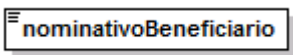
element **Check/partitalvaEsercente**

diagram	
type	xsd:string
properties	minOcc 0 maxOcc 1 content simple
source	<code><xsd:element name="partitalvaEsercente" type="xsd:string" minOccurs="0" maxOccurs="1"/></code>

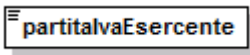
complexType **CheckResponse**

diagram	
namespace	http://bonus.mibact.it/VerificaVoucher/
children	nominativoBeneficiario partitalvaEsercente ambito bene importo
used by	element CheckResponseObj/checkResp
source	<pre><xsd:complexType name="CheckResponse"> <xsd:sequence> <xsd:element name="nominativoBeneficiario" type="xsd:string" minOccurs="1" maxOccurs="1"/> <xsd:element name="partitalvaEsercente" type="xsd:string" minOccurs="1" maxOccurs="1"/> <xsd:element name="ambito" type="xsd:string" minOccurs="1" maxOccurs="1"/> <xsd:element name="bene" type="xsd:string" minOccurs="1" maxOccurs="1"/> <xsd:element name="importo" type="xsd:double" minOccurs="1" maxOccurs="1"/> </xsd:sequence> </xsd:complexType></pre>

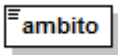
element **CheckResponse/nominativoBeneficiario**

diagram	
type	xsd:string
properties	content simple
source	<code><xsd:element name="nominativoBeneficiario" type="xsd:string" minOccurs="1" maxOccurs="1"/></code>

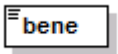
element **CheckResponse/partitalvaEsercente**

diagram	
type	xsd:string
properties	content simple
source	<code><xsd:element name="partitalvaEsercente" type="xsd:string" minOccurs="1" maxOccurs="1"/></code>


element **CheckResponse/ambito**

diagram	
type	xsd:string
properties	content simple
source	<code><xsd:element name="ambito" type="xsd:string" minOccurs="1" maxOccurs="1"/></code>

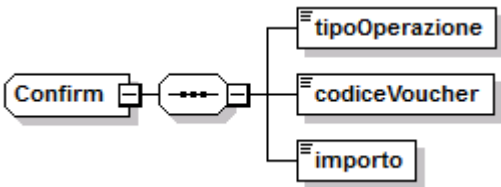
element **CheckResponse/bene**

diagram	
type	xsd:string
properties	content simple
source	<code><xsd:element name="bene" type="xsd:string" minOccurs="1" maxOccurs="1"/></code>

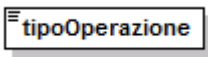
element **CheckResponse/importo**

diagram	
type	xsd:double
properties	content simple
source	<code><xsd:element name="importo" type="xsd:double" minOccurs="1" maxOccurs="1"/></code>

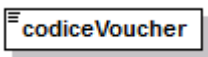
complexType **Confirm**

diagram	
namespace	http://bonus.mibact.it/VerificaVoucher/
children	tipoOperazione codiceVoucher importo
used by	element ConfirmRequestObj/checkReg
source	<pre><xsd:complexType name="Confirm"> <xsd:sequence> <xsd:element name="tipoOperazione" type="xsd:string" minOccurs="1" maxOccurs="1"/> <xsd:element name="codiceVoucher" type="xsd:string" minOccurs="1" maxOccurs="1"/> <xsd:element name="importo" type="xsd:double" minOccurs="1" maxOccurs="1"/> </xsd:sequence> </xsd:complexType></pre>


element **Confirm/tipoOperazione**

diagram	
type	xsd:string
properties	content simple
source	<code><xsd:element name="tipoOperazione" type="xsd:string" minOccurs="1" maxOccurs="1"/></code>


element **Confirm/codiceVoucher**

diagram	
type	xsd:string
properties	content simple
source	<code><xsd:element name="codiceVoucher" type="xsd:string" minOccurs="1" maxOccurs="1"/></code>


element **Confirm/importo**

diagram	
type	xsd:double
properties	content simple
source	<code><xsd:element name="importo" type="xsd:double" minOccurs="1" maxOccurs="1"/></code>

complexType **ConfirmResponse**

diagram	
namespace	http://bonus.mibact.it/VerificaVoucher/
children	esito
used by	element ConfirmResponseObj/checkResp
source	<code><xsd:complexType name="ConfirmResponse"> <xsd:sequence> <xsd:element name="esito" type="xsd:string" minOccurs="1" maxOccurs="1"/> </xsd:sequence> </xsd:complexType></code>

element **ConfirmResponse/esito**

diagram	
type	xsd:string
properties	content simple
source	<code><xsd:element name="esito" type="xsd:string" minOccurs="1" maxOccurs="1"/></code>