

# **GestPayS2S**

## **Specifiche tecniche**

### **WEBSERVICE**

### **XML**

## Sommario

Informazioni documento .....	3
Informazioni versione .....	4
Istruzioni per l'utilizzo del modulo GestPayS2S .....	5
Descrizione oggetto GestPayS2S.....	5
Metodi di GestPayS2S. ....	7
Operazioni effettuabili con GestPayS2S.....	9
Richiesta di autorizzazione .....	9
Movimentazione .....	12
Cancellazione.....	13
Storno.....	14
Annulla.....	15
Interrogazione stato transazione .....	15
Transazioni Verified by Visa .....	19
Transazioni con carte Visa non 3D .....	19
Transazioni con carte Visa 3D .....	19
Gestione campo CVV .....	23
Ordini di pagamento in ambiente di Test.....	24
Links .....	24

## Informazioni documento

Nome progetto	GestPayS2S WEB SERVICE XML
Titolo documento	GestPayS2S - Specifiche tecniche
Data creazione	2/12/2008 12.00
Lingua	Italiano
Società	Easy Nolo S.p.A.

## Informazioni versione

Versione	Descrizione	Data	Autore
1.0.0	Versione iniziale	2/12/2008	EasyNolo

## Istruzioni per l'utilizzo del modulo GestPayS2S con interfaccia WEB SERVICE

Il presente documento contiene le istruzioni necessarie per l'utilizzo del webservice **GestPayS2S**, che consente l'utilizzo di GestPay in modalità Server – Server.

L'utilizzo di GestPay e quindi anche del webservice **GestPayS2S** è riservato alle aziende che abbiano stipulato gli appositi contratti con EasyNolo e Banca Sella, per la gestione delle transazioni con carta di credito online e siano quindi diventate "Esercenti convenzionati".

Un esercente convenzionato GestPay è riconosciuto da un codice identificativo detto "Shop\_Login" ed ha diritto ad utilizzare l'ambiente di Back Office tramite uno o più operatori registrati al portale Sella.it (almeno un operatore, con la qualifica di Amministratore, è necessario per l'attivazione del servizio).

Il modulo **GestPayS2S web service**, in estrema sintesi, può essere considerato come una "scatola nera" che può essere chiamata per l'esecuzione di una transazione con carta di credito. La "scatola nera" provvederà a:

- formattare in modo opportuno tali informazioni
- trasmetterle ai server di GestPay
- ricevere da GestPay l'esito dell'operazione
- rendere facilmente reperibili le informazioni restituite in formato **XML**.

Poiché GestPay consente ad ogni esercente di decidere quali informazioni trasmettere al sistema (al di là di un insieme minimo obbligatorio), l'esercente dovrà configurare opportunamente il sistema mediante la sezione "Campi & Parametri" del Back Office.

GestPay, inoltre, accetterà le chiamate solo se provenienti da un indirizzo IP associato allo Shop\_Login dell'esercente chiamante. E' quindi necessario, sempre dal Back Office (sezione indirizzi IP), inserire l'indirizzo IP del server che eseguirà le chiamate al web service **GestPayS2S**.

### Descrizione WebService GestPayS2S

La comunicazione server to server tra GestPay e l'esercente verrà automaticamente gestita dal webservice **GestPayS2S**. Tale componente dovrà essere richiamato dalle pagine web predisposte dall'esercente per gestire l'invio dei dati della transazione e la lettura dell'esito comunicato da GestPay in formato XML.

Indirizzo del servizio al seguente URL:

**<https://ecomms2s.sella.it/gestpay/gestpayws/WSs2s.asmx>**

per codici di test

**<https://testecomm.sella.it/gestpay/gestpayws/WSs2s.asmx>**

## Generazione Classe Proxy per utilizzare le funzionalità del WebService da diversi linguaggi

La classe proxy nel linguaggio preferito può essere creata automaticamente tramite l'utilizzo in questo caso dal programma wsdl.exe (in questo caso fornito da Microsoft ) semplicemente specificando il file di contratto relativo al WebService in questo caso: Indirizzo descrizione del servizio al seguente URL

Per i codici di produzione

<https://ecomms2s.sella.it/gestpay/gestpayws/Ws2s.asmx?WSDL>

per codici di test

<https://testecomm.sella.it/gestpay/gestpayws/Ws2s.asmx?WSDL>

ad esempio

wsdl /language:VB /out: wss2sProxyClass.vb

<https://testecomm.sella.it/gestpay/gestpayws/Ws2s.asmx?WSDL>

verrà generato il file .vb con la gestione della classe proxy relativa al WebService che andrà semplicemente importato nel progetto ed utilizzato

Verificare nell'esempio successivo come la classe proxy **WSs2s** generata ed importata nel progetto può essere richiamata

## Elenco chiamate disponibili del webservice GestPayS2S.

Di seguito è riportato l'elenco completo dei metodi dell'oggetto GestPayS2S, suddivisi nelle 3 sezioni.

Metodi WBSERVICE	
Nome metodo	Descrizione
callDeleteS2S	Cancella una transazione autorizzata
callPagamS2S	Effettua la richiesta di autorizzazione
callReadTrxS2S	Interroga lo stato ed i dettagli di una transazione
callRefundS2S	Storna una transazione movimentata
callRenounceS2S	Storna una transazione se movimentata o la cancella se autorizzata
callSettleS2S	Movimenta una transazione autorizzata
callVerifyCardS2S	Verifica la validità di una carta di credito

Le varie chiamate ai metodi sono gestite come chiamate a funzione dal WebService quindi non con il passaggio di una stringa XML.

I valori della varie chiamate devono essere passati come parametri

Descrizione elenco possibili parametri di Input

Parametri Input	
Nome metodo	Descrizione
amount (val)	Assegna l'importo
bankTransactionID (val)	Assegna il codice attribuito dalla banca alla transazione
buyerEmail (val)	Assegna l'indirizzo e-mail del compratore
buyerName (val)	Assegna il nome del compratore
cardNumber (val)	Assegna il numero di carta
uicCode(val)	Assegna il codice divisa
customInfo (val)	Assegna una stringa contenente eventuali parametri personalizzati
Cvv (val)	Assegna il codice di sicurezza stampato sulla carta
Encryption	Attiva la crittografia locale
expiryMonth (val)	Assegna il mese di scadenza della carta
expiryYear (val)	Assegna l'anno di scadenza della carta
languageId (val)	Assegna la lingua per le mail al compratore
SetMIN (val)	Non utilizzato
PARes (val)	Assegna stringa cifrata restituita da Issuer (solo operazioni VbV)
shopLogin (val)	Assegna l'identificativo del negozio
shopTransactionID (val)	Assegna il codice attribuito dall'esercente alla transazione
transKey (val)	Assegna l'identificativo transazione per operazioni VbV

Il file XML e' descritto e puo' essere validato tramite il relativo file GestPayS2S.xsd che qui andiamo a descrivere nel dettaglio

Valori XML restituiti	
Nome metodo	Descrizione
TransactionType	Tipo di richiesta eseguita puo' assumere i seguenti valori:  PAGAM SETTLE DELETE REFUND VERCARD RENOUNCE QUERYTX
TransactionResult	Restituisce l'esito della transazione con valori OK e KO
TransactionState	Restituisce lo stato (Autorizzata, Stornata, ecc.) della transazione
ShopTransactionID	Restituisce il codice attribuito dall'esercente alla transazione
BankTransactionID	Restituisce il codice attribuito dalla banca alla transazione
AuthorizationCode	Restituisce il codice di autorizzazione
Currency	Restituisce il codice della divisa
Country	Restituisce nazionalità istituto emittente la carta di credito
Company	Restituisce la compagnia della carta Visa MasterCard etc ..
Events	Restituisce l'elenco degli eventi di una transazione suddivisi come segue
	EventType tipo evento con i seguenti valori:  AUT MOV CAN REQ STO
	EventAmount Valore relative all'evento
	EventDate Data dell'evento
CustomInfo	Restituisce gli eventuali parametri aggiuntivi
BuyerType	Restituisce l'indirizzo nome e e-mail del compratore divisi come segue
	BuyerEmail Restituisce l'indirizzo e-mail del compratore
	BuyerName Restituisce il nome del compratore
Get3DLevel	Restituisce il livello di autenticazione per transazioni Visa VbV / Mastercard Securcode (Maestro SecureCode
ErrorCode	Restituisce un codice relativo all'esito della transazione
ErrorDescription	Restituisce la descrizione associata al valore di ErrorCode
AlertCode	Restituisce il codice di violazione dei criteri di risk management
AlertDescription	Restituisce la descrizione associata al valore di AlertCode
Amount	Restituisce l'importo della transazione
TransKey	Restituisce l'identificativo transazione per operazioni VbV
VBVType	Restituisce i valori per il VbV come segue
	VbV Restituisce il flag Verified by Visa (OK oppure KO)
	VbVBuyer Restituisce il flag transazione Full Verified by Visa
	VbVRisp Restituisce la stringa cifrata ottenuta dalla Visa directory



## Operazioni effettuabili con il WEB SERVICE GestPayS2S

Di seguito sono elencate le operazioni che possono essere effettuate tramite il web service **GestPayS2S** e le modalità con cui ciascuna operazione deve essere eseguita.

### **Richiesta di autorizzazione**

E' l'operazione di pagamento vera e propria. Consiste nell'invio al server della banca di tutte le informazioni necessarie per l'effettuazione di una transazione di pagamento e nel reperimento dell'esito successivamente all'elaborazione da parte della banca.

La funzione di **GestPayS2S** da utilizzare per questo scopo è **CallPagamS2S**.

Prima di richiamare tale metodo, però, è necessario valorizzare tutte le informazioni necessarie, per il passaggio dei parametri :

- **shopLogin** (Codice Esercente)
- **uicCode** (Codice Divisa)
- **Amount** (Importo)
- **shopTransactionID** (Codice identificativo transazione)
- **cardNumber** (Numero carta di credito)
- **expiryMonth** (Mese di scadenza della carta)
- **expiryYear** (Anno di scadenza della carta)

Facoltativamente, potranno essere passati anche i seguenti parametri, previa opportuna configurazione della sezione "Configurazione cliente | Campi & Parametri":

- **buyerName** (Nome del compratore)
- **buyerEmail** (Indirizzo e-mail del compratore)
- **languageId** (Codice della lingua)
- **cvv** (Stringa contenente il codice di sicurezza stampato sulla carta di credito, come specificato nell'apposito capitolo "[Gestione campo CVV](#)")
- **min** (non utilizzato passare vuoto)
- **transKey** (verificare nell'apposito paragrafo Transazioni Verified by Visa)
- **PARes** (verificare nell'apposito paragrafo Transazioni Verified by Visa)
- **customInfo** (Stringa contenente gli eventuali campi personalizzati, dichiarati come "Parametro" nell'apposita sezione del Back Office)
- **IDEA** (non utilizzato passare vuoto)

Il metodo **CallPagamS2S** trasmetterà a GestPay tutti i dati precedentemente assegnati; GestPay controllerà la corrispondenza tra i dati ricevuti e la configurazione impostata tramite Back Office e, in caso positivo effettuerà l'operazione.

Se la configurazione M.O.T.O. prevede la separazione tra la fase di autorizzazione e quella di movimentazione, l'attività di GestPay sarà limitata alla richiesta di autorizzazione.

Se, invece, la configurazione M.O.T.O. prevede la contestualità tra la fase di autorizzazione e quella di movimentazione, allora GestPay effettuerà la richiesta di autorizzazione e, in caso di risposta positiva, provvederà di seguito ad effettuarne la movimentazione.

Successivamente all'esecuzione del metodo **CallPagamS2S**, è possibile conoscere l'esito dell'operazione tramite il file XML restituito.

## Esempio di Chiamata CallPagamS2S in VB.NET

```
Dim MyService As New WSs2s

Try
    XmlReturn.LoadXml(MyService.callPagamS2S(txtShopLogin.Text, txtUicCode.Text,
    txtAmount.Text, txtShopID.Text, txtCardNumber.Text, txtExpMM.Text, txtExpYY.Text,
    txtBuyName.Text, txtBuyEmail.Text, txtLanguage.Text, txtCVV.Text, txtMin.Text,
    txtTransKey.Text, txtPares.Text, txtCustom.Text, txtIdea.Text).OuterXml)

Catch ex As Exception
    Dim message As String = ex.Message
    MsgBox("A run-time error occurred:" & ex.Message)
    Exit Sub
End Try
```

Verificare nelle note precedenti come generare una classe proxy WSs2s da utilizzare

**Esempio di file XML di risposta alla chiamata CallPagamS2S**

```
<?xml version="1.0" encoding="utf-8"?>
<GestPayS2S>
  <TransactionType>PAGAM</TransactionType>
  <TransactionResult>OK</TransactionResult>
  <ShopTransactionID>34az85ord19</ShopTransactionID>
  <BankTransactionID>18</BankTransactionID>
  <AuthorizationCode>940499</AuthorizationCode>
  <Currency>242</Currency>
  <Country>ITALIA</Country>
  <Buyer>
    <BuyerName>Mario.bianchi</BuyerName>
    <BuyerEmail>Mario.bianchi@isp.it</BuyerEmail>
  </Buyer>
  <ErrorCode>0</ErrorCode>
  <ErrorDescription>Transazione correttamente effettuata</ErrorDescription>
  <AlertCode></AlertCode>
  <AlertDescription></AlertDescription>
  <TransactionKey></TransactionKey>
  <VbV>
    <VbVFlag>OK</VbVFlag>
    <VbVBuyer>
      </VbVBuyer>
    <VbVRisp>
      </VbVRisp>
    </VbV>
  </GestPayS2S>
```

- Per prima cosa, verificare il **TransactionResult** che restituirà la stringa “OK” se la transazione è stata autorizzata o la stringa “KO” se la transazione non è stata autorizzata.
- Se **TransactionResult** restituisce “KO”, allora è possibile sapere se il fallimento è dovuto a motivi tecnici o ad una risposta negativa dei circuiti autorizzativi, utilizzando il valore **ResultCode**:
  - Se **ResultCode** restituisce un valore **<> 0**, allora la transazione ha avuto un esito negativo; il valore restituito sarà diverso a seconda del motivo specifico che ha causato il fallimento. Il valore **ResultDescription** restituirà una descrizione (nella lingua impostata per il Back Office) del motivo del fallimento.
- Se **TransactionResult** restituisce “OK”, allora la transazione è stata autorizzata (e, in caso di M.O.T.O. contestuale, anche movimentata). Sarà allora possibile ottenere ulteriori informazioni in merito alla transazione, con i seguenti valori:
  - **BankTransactionID** per ottenere il codice univoco assegnato alla transazione da GestPay
  - **AuthorizationCode** per ottenere il codice di autorizzazione assegnato alla transazione dai circuiti internazionali
  - **AlertCode** (ed il corrispondente **AlertDescription**) nel caso in cui siano stati impostati dei criteri di Risk Management tramite Back Office e la transazione ne abbia violato uno; se nessun criterio di Risk Management è stato violato, **AlertCode** sarà vuoto.
  - **Get3DLevel** per conoscere il livello di autenticazione utilizzato dal titolare di carta in transazioni Visa VBV o Mastercard Securecode o Maestro SecureCode
- Tutti gli altri valori restituiti potranno essere utilizzati
- Il file XML restituito potrà essere verificato nella sua validità tramite il relativo file descrittivo **GESTPAYS2S.XSD** scaricabile dal sito di EasyNolo

## Movimentazione

E' l'operazione di conferma finanziaria che può seguire una autorizzazione positiva. Può esistere solo per esercenti con una configurazione M.O.T.O. che prevede la separazione tra la fase di autorizzazione e la fase di movimentazione.

La funzione di **GestPayS2S WebService** da utilizzare per questo scopo è **CallSettleS2S**.

Prima di utilizzare tale metodo, però, è necessario passare a **GestPayS2S** tutte le informazioni necessarie per identificare in modo certo la transazione da movimentare.

Una transazione può essere identificata in modo univoco dalla coppia

**< Codice Esercente > + < Codice univoco assegnato da GestPay alla transazione >**

E' però possibile utilizzare anche il codice assegnato alla transazione dall'esercente, avendo cura di accertarsi che lo stesso identifichi univocamente una transazione; in caso contrario **GestPayS2S** restituirà il codice di errore relativo a "Transazione non univocamente identificata".

Poiché una transazione può essere movimentata anche per un importo inferiore a quello per cui è stata autorizzata, è necessario assegnare a **GestPayS2S** anche l'importo da movimentare ed il codice della divisa in cui è espresso (ovviamente la divisa deve essere la stessa dell'autorizzazione).

Per fare tutto ciò è sufficiente passare i seguenti parametri :

- **shopLogin** (Codice Esercente)
- **bankTransID** (Codice assegnato alla transazione da GestPay) o **shopTransID** (Codice assegnato alla transazione dall'esercente)
- **uicCode** (Codice divisa: 242 = EUR)
- **amount** (Importo)

A differenza dell'operazione di **richiesta autorizzazione**, l'operazione di **movimentazione** non può avere esito negativo, se non per cause tecniche. Per conoscere l'esito dell'operazione è quindi sufficiente consultarne il risultato tecnico, mediante la lettura dei valori seguenti dal file XML di risposta:

- **ResultCode** (e **ResultDescription**)

## Esempio di utilizzo del metodo CallSettleS2S in linguaggio VB.NET

```
Dim MyService As New WSs2s

Dim XmlReturn As New XmlDocument
Dim ReturnNode As New XmlDocument
Dim ms As System.IO.MemoryStream

Try
    XmlReturn.LoadXml(MyService.callSettleS2S(txtShopLoginS.Text,
    txtUICCodeS.Text, txtAmountS.Text, txtShopIdS.Text, txtBankIdS.Text).OuterXml)
Catch ex As Exception
    Dim message As String = ex.Message
    MsgBox("A run-time error occurred:" & ex.Message)
    Exit Sub
End Try
```

Verificare nelle note precedenti come generare una classe proxy WSs2s da utilizzare

### **Cancelazione**

E' l'operazione di abbandono di una transazione autorizzata. Può essere effettuata solo in caso di impostazione M.O.T.O. con fasi separate.

La chiamata al WebService **GestPayS2S** da utilizzare per questo scopo è **CallDeleteS2S**.

Prima di utilizzare tale chiamata, però, è necessario passare a **GestPayS2S** tutte le informazioni necessarie per identificare in modo certo la transazione da cancellare.

Una transazione può essere identificata in modo univoco dalla coppia

**< Codice Esercente > + < Codice univoco assegnato da GestPay alla transazione >**

E' però possibile utilizzare anche il codice assegnato alla transazione dall'esercente, avendo cura di accertarsi che lo stesso identifichi univocamente una transazione; in caso contrario **GestPayS2S** restituirà il codice di errore relativo a "Transazione non univocamente identificata".

Per fare ciò è sufficiente passare i seguenti parametri :

- **shopLogin** (Codice Esercente)
- **bankTransactionID** (Codice assegnato alla transazione da GestPay) o **shopTransactionID** (Codice assegnato alla transazione dall'esercente)

A differenza dell'operazione di **richiesta autorizzazione**, l'operazione di **cancellazione** non può avere esito negativo, se non per cause tecniche. Per conoscere l'esito dell'operazione è quindi sufficiente consultarne il risultato tecnico, mediante la lettura dei valori seguenti dal file XML di risposta:

- **ResultCode** (e **ResultDescription**)

### **Esempio di utilizzo del metodo CallDeleteS2S in linguaggio VB.NET**

```
Try

    XmlReturn.LoadXml(MyService.callDeleteS2S(txtShopLoginD.Text, txtShopIdD.Text,
txtBankIdD.Text).OuterXml)

Catch ex As Exception
    Dim message As String = ex.Message
    MsgBox("A run-time error occurred:" & ex.Message)
    Exit Sub
End Try
```

Verificare nelle note precedenti come generare una classe proxy Ws2s da utilizzare

## Storno

E' l'operazione di annullo (totale o parziale) di una movimentazione. Contabilmente consiste nell'addebito dell'esercente e conseguente riaccredito della somma sulla carta del compratore.

Ovviamente, può essere effettuata solo su transazioni già movimentate.

Una transazione movimentata può essere stornata anche più di una volta, per importi parziali, fino ad un totale pari alla somma per cui è stata movimentata.

La funzione di **GestPayS2S** Web Service da utilizzare per questo scopo è **CallRefundS2S**.

Prima di utilizzare tale chiamata, però, è necessario passare a **GestPayS2S** tutte le informazioni necessarie per identificare in modo certo la transazione da stornare.

Una transazione può essere identificata in modo univoco dalla coppia

**< Codice Esercente > + < Codice univoco assegnato da GestPay alla transazione >**

E' però possibile utilizzare anche il codice assegnato alla transazione dall'esercente, avendo cura di accertarsi che lo stesso identifichi univocamente una transazione; in caso contrario **GestPayS2S** restituirà il codice di errore relativo a "Transazione non univocamente identificata".

Poiché una transazione può essere stornata anche per un importo inferiore a quello per cui è stata movimentata, è necessario assegnare a **GestPayS2S** anche l'importo da stornare ed il codice della divisa in cui è espresso (ovviamente la divisa deve essere la stessa della movimentazione).

Per fare tutto ciò è sufficiente passare i seguenti parametri:

- **shopLogin** (Codice Esercente)
- **bankTransactionID** (Codice assegnato alla transazione da GestPay) o **shopTransactionID** (Codice assegnato alla transazione dall'esercente)
- **uicCode** (Codice divisa: 18 = ITL, 242 = EUR)
- **amount** (Importo)

A differenza dell'operazione di **richiesta autorizzazione**, l'operazione di **storno** non può avere esito negativo, se non per cause tecniche. Per conoscere l'esito dell'operazione è quindi sufficiente consultarne il risultato tecnico, mediante la lettura dei valori seguenti dal file XML di risposta:

- **ResultCode** (e **ResultDescription**)

## Esempio di utilizzo del metodo CallRefundS2S in linguaggio VB.NET

```
Dim MyService As New WSs2s
MyService.Proxy = myProxy

Try
    XmlReturn.LoadXml(MyService.callRefundS2S(txtShopLoginRf.Text,
    txtUICCodeRf.Text, txtAmountRf.Text, txtShopIDRf.Text, txtBankIdRf.Text).OuterXml)
Catch ex As Exception
    Dim message As String = ex.Message
    MsgBox("A run-time error occurred:" & ex.Message)
    Exit Sub
End Try
```

Verificare nelle note precedenti come generare una classe proxy WSs2s da utilizzare

## Annullo

In aggiunta ai metodi **CallDeleteS2S** e **CallRefundS2S** usati per annullare rispettivamente una transazione Autorizzata o Movimentata, **GestPayS2S** WebService prevede anche la possibilità di annullare una transazione di cui si ignori lo stato (solo autorizzata o anche movimentata).

In tal modo sarà possibile ordinare a GestPay di cancellare la transazione specificata qualora la stessa sia in stato "Autorizzato" oppure di stornarla qualora la stessa sia in stato "Movimentato"; ovviamente la transazione dovrà essere in uno dei due stati, altrimenti nessuna operazione sarà effettuata.

Trattandosi inoltre di una operazione di annullo, nel caso in cui questa dovesse risolversi in uno storno, sarà considerato uno storno totale: non sarà possibile specificare un importo.

La funzione di **GestPayS2S** da utilizzare per questo scopo è **CallRenounceS2S**.

Prima di utilizzare tale metodo, però, è necessario passare a **GestPayS2S** tutte le informazioni necessarie per identificare in modo certo la transazione da cancellare o stornare. Una transazione può essere identificata in modo univoco dalla coppia

**< Codice Esercente > + < Codice univoco assegnato da GestPay alla transazione >**

E' però possibile utilizzare anche il codice assegnato alla transazione dall'esercente, avendo cura di accertarsi che lo stesso identifichi univocamente una transazione; in caso contrario **GestPayS2S** restituirà il codice di errore relativo a "Transazione non univocamente identificata".

Per fare ciò è sufficiente passare i seguenti parametri:

- **shopLogin** (Codice Esercente)
- **bankTransactionID** (Codice assegnato alla transazione da GestPay) o **shopTransactionID** (Codice assegnato alla transazione dall'esercente)

A differenza dell'operazione di **richiesta autorizzazione**, l'operazione di **annullo** non può avere esito negativo, se non per cause tecniche. Per conoscere l'esito dell'operazione è quindi sufficiente consultarne il risultato tecnico, mediante la lettura dei valori seguenti dal file XML di risposta:

- **ResultCode** (e **ResultDescription**)

## Esempio di utilizzo del metodo CallRenounceS2S in linguaggio VB.NET

```
Dim MyService As New WSs2s
MyService.Proxy = myProxy

Dim XmlReturn As New XmlDocument
Dim ReturnNode As New XmlDocument

Try
    XmlReturn.LoadXml(MyService.callRenounceS2S(txtShopLoginRn.Text,
    txtShopIDRn.Text, txtBankIdRn.Text).OuterXml)
Catch ex As Exception
    Dim message As String = ex.Message
    MsgBox("A run-time error occurred:" & ex.Message)
    Exit Sub
End Try
```

Verificare nelle note precedenti come generare una classe proxy WSs2s da utilizzare

## Interroga

E' l'operazione che consente di conoscere lo stato ed i dettagli di una specifica transazione. La funzione di **GestPayS2S** Web Service da utilizzare per questo scopo è **CallReadTrxS2S**.

Prima di utilizzare tale metodo, però, è necessario assegnare a **GestPayS2S** tutte le informazioni necessarie per identificare in modo certo la transazione di cui interrogare lo stato.

Una transazione può essere identificata in modo univoco dalla coppia

**< Codice Esercente > + < Codice univoco assegnato da GestPay alla transazione >**

E' però possibile utilizzare anche il codice assegnato alla transazione dall'esercente, avendo cura di accertarsi che lo stesso identifichi univocamente una transazione; in caso contrario **GestPayS2S** restituirà il codice di errore relativo a "Transazione non univocamente identificata".

Per fare ciò è sufficiente valorizzare i seguenti parametri:

- **shopLogin** (Codice Esercente)
- **bankTransactionID** (Codice assegnato alla transazione da GestPay) o **shopTransactionID** (Codice assegnato alla transazione dall'esercente)
- In caso di transazione correttamente identificata, è poi possibile ottenere tutte le informazioni in merito ad essa mediante la lettura dei valori nel XML di risposta:
  - **TransactionResult** per sapere se la transazione è stata autorizzata ("OK" = Autorizzata; "KO" = Non Autorizzata)
  - **BankTransactionID** per ottenere il codice univoco assegnato alla transazione da GestPay
  - **AuthorizationCode** per ottenere il codice di autorizzazione assegnato alla transazione dai circuiti internazionali (in caso di transazione autorizzata)
  - **ErrorCode** (ed il corrispondente **ErrorDescription**) per conoscere la causa della mancata autorizzazione (**ErrorCode = 0** in caso di transazione autorizzata)
  - **AlertCode** (ed il corrispondente **AlertDescription**) nel caso in cui siano stati impostati dei criteri di Risk Management tramite Back Office e la transazione ne abbia violato uno; se nessun criterio di Risk Management è stato violato, **AlertCode** restituirà 0.
  - **Currency** per conoscere la divisa in cui è espressa la transazione
  - **BuyerName** per conoscere il nome digitato dal compratore nella pagina di pagamento
  - **BuyerEmail** per conoscere l'indirizzo e-mail digitato dal compratore nella pagina di pagamento
  - **AuthorizationAmount** per conoscere l'importo per cui è stata richiesta l'autorizzazione
  - **TransactionState** per conoscere lo stato della transazione. I valori possibili ed i loro significati sono i seguenti:
    - **AUT** E' avvenuta la richiesta di autorizzazione. Non è avvenuta alcun'altra operazione
    - **MOV** La transazione è stata autorizzata positivamente ed in seguito è stata movimentata. Potrebbe essere stata stornata, ma solo parzialmente
    - **CAN** La transazione è stata autorizzata ed in seguito è stata cancellata
    - **STO** La transazione è stata autorizzata, movimentata e poi stornata completamente
  - **Eseguire un ciclo su valori Events per avere le diverse operazione eseguite sulla transazione**
  - **EventType (AUT,MOV,CAN,REQ,STO)SettledAmount** per conoscere l'importo per cui la transazione è stata movimentata (0 se la movimentazione non è avvenuta)
  - **EventAmount** per conoscere l'importo dell'operazione relativa
  - **EventDateTime** per conoscere la data-ora dell'operazione relativa
  - **Get3DLevel** per conoscere il livello di autenticazione utilizzato dal titolare di carta in transazioni Visa VBV o Mastercard Securcode o Maestro SecureCode



### Esempio di utilizzo del metodo CallReadTrxS2S in linguaggio VB.NET

```
Dim MyService As New WSs2s
MyService.Proxy = myProxy

Dim XmlReturn As New XmlDocument
Dim ReturnNode As New XmlDocument
Dim ms As System.IO.MemoryStream

Try

    XmlReturn.LoadXml(MyService.callReadTrxS2S(txtShoploginR.Text,
txtShopIDR.Text, txtBankIDR.Text).OuterXml)

Catch ex As Exception
    Dim message As String = ex.Message
    MsgBox("A run-time error occurred:" & ex.Message)
    Exit Sub
End Try
```

**Verificare nelle note precedenti come generare una classe proxy WSs2s da utilizzare**

### Verifica carta

E' l'operazione di controllo della validità di una carta. Consiste nell'invio al server della banca dei dati della carta (numero e scadenza) e nel reperimento dell'esito successivamente all'elaborazione da parte della banca.

La funzione di **GestPayS2S** Web Service da utilizzare per questo scopo è **CallVerifyCardS2S**.

Prima di richiamare tale metodo, però, è necessario assegnare a **GestPayS2S** tutte le informazioni necessarie, tramite la valorizzazione dei seguenti parametri:

- **shopLogin** (Codice Esercente)
- **shopTransactionID** (Codice identificativo transazione)
- **cardNumber** (Numero carta di credito)
- **expMonth** (Mese di scadenza della carta)
- **expYear** (Anno di scadenza della carta)
- **CVV2** (Stringa contenente il codice di sicurezza stampato sulla carta di credito, come specificato nell'apposito capitolo "[Gestione campo CVV](#)")

Il metodo **CallVerifyCardS2S** trasmetterà a GestPay tutti i dati precedentemente assegnati; GestPay li utilizzerà per effettuare una richiesta di autorizzazione senza effetti contabili e restituirà a **GestPayS2S** l'esito dell'operazione.

Successivamente all'esecuzione del chiamata **CallVerifyCardS2S**, è possibile conoscere l'esito dell'operazione utilizzando gli valori nella relativa risposta XML:

- Per prima cosa, è possibile utilizzare il metodo **TransactionResult** che restituirà la stringa "**OK**" se la transazione è stata autorizzata o la stringa "**KO**" se la transazione non è stata autorizzata.
- Se **TransactionResult** restituisce "**KO**", allora è possibile sapere se il fallimento è dovuto a motivi tecnici o ad una risposta negativa dei circuiti autorizzativi, utilizzando il metodo **ResultCode**:
  - Se **GetResultCode** restituisce un valore  $\neq 0$ , allora la transazione ha avuto un esito negativo; il valore restituito sarà diverso a seconda del motivo specifico che ha causato il fallimento. Il metodo **ResultDescription** restituirà una descrizione (nella lingua impostata per il Back Office) del motivo del fallimento.
  - Se **ResultCode** restituisce **0**, allora la transazione non è fallita per motivi tecnici. Tramite il metodo **ErrorCode** sarà possibile ottenere il codice della causa del fallimento; la descrizione in lingua sarà ottenuta tramite il metodo **ErrorDescription**.
- Se **TransactionResult** restituisce "**OK**", allora la transazione è stata autorizzata e la carta è da considerarsi valida.

### Esempio di utilizzo del metodo CallVerifyCardS2S in linguaggio VB.NET

```
Dim MyService As New WSs2s
MyService.Proxy = myProxy

Dim XmlReturn As New XmlDocument
Dim ReturnNode As New XmlDocument
Dim ms As System.IO.MemoryStream

Try

    XmlReturn.LoadXml(MyService.callVerifycardS2S(txtShopLoginVC.Text,
    txtShopIdVC.Text, txtCardNumberVC.Text, txtExpMMVC.Text, txtExpYYVC.Text).OuterXml)

Catch ex As Exception
    Dim message As String = ex.Message
    MsgBox("A run-time error occurred:" & ex.Message)
Exit Sub
End Try
```

Verificare nelle note precedenti come generare una classe proxy WSs2s da utilizzare

## Transazioni Verified by Visa

Le transazioni effettuate con carte di credito Visa, se l'esercente è abilitato al servizio Verified by Visa, garantiscono la non ripudiabilità della transazione da parte del titolare di carta. Si possono distinguere due casi:

- Transazioni con carte Visa non 3D effettuate su esercenti abilitati al servizio Verified by Visa
- Transazioni con carte Visa 3D effettuate su esercenti abilitati al servizio Verified by Visa

### **Transazioni con carte Visa non 3D**

Dal punto di vista funzionale la transazione viene processata come una normale richiesta di autorizzazione. Le transazioni Verified by Visa saranno caratterizzate da un flag (valorizzato **OK** oppure **KO**) leggibile utilizzando il valore **VbV nella risposta XML**

### **Transazioni con carte Visa 3D**

Le transazioni con carte di credito 3D richiedono l'autenticazione del titolare. Il compratore dovrà essere rediretto sul sito dell'istituto che ha emesso la carta di credito e digitare una password che gli è stata rilasciata dall'istituto stesso. Se il processo di autenticazione si conclude con esito positivo la transazione può continuare e generare un esito positivo o negativo in funzione della risposta ricevuta dai circuiti.

Dal punto di vista funzionale il processo si svolge in 3 fasi:

#### Fase I: richiesta di autorizzazione

Si procede con una normale richiesta di autorizzazione. Se la carta è riconosciuta come una carta di credito 3D l'esito della richiesta è un codice d'errore specifico (**8006**) leggibile utilizzando il valore **ErrorCode**. La descrizione dell'errore (**Verify By Visa**) sarà leggibile utilizzando il valore **ErrorDescription**.

In questa fase vengono inoltre restituite delle informazioni aggiuntive, necessarie durante il processo di pagamento, specifiche delle transazioni Verified by Visa. In particolare è necessario acquisire l'identificativo della transazione leggibile utilizzando il valore **TransKey** e una stringa cifrata da utilizzare nella fase successiva leggibile utilizzando il valore **VbVRisp** sempre nella risposta XML.

#### Fase II: autenticazione titolare di carta

In questa fase è necessario permettere al compratore di autenticarsi presso l'istituto che ha emesso la carta di credito. Si deve indirizzare il browser del compratore su una pagina predisposta da Banca Sella che si occuperà di interfacciare il servizio di autenticazione e di indirizzare il compratore all'istituto emittente la carta con tutte le informazioni necessarie all'autenticazione.

La pagina da richiamare risponde alla seguente url:

`https://ecomm.sella.it/gestpay/pagamvisa3d.asp`

per codici di test

`https://testecomm.sella.it/gestpay/pagamvisa3d.asp`

La pagina deve essere richiamata passando 3 parametri:

- a** shop\_login
- b** stringa cifrata acquisita nella fase precedente tramite il valore **VbVRisp**
- c** url sul sito esercente al quale il compratore sarà rediretto dopo l'autenticazione

Eventuali parametri aggiuntivi non saranno restituiti nella risposta alla seconda chiamata, si consiglia, pertanto, di conservarli in apposite variabili durante la fase I.

Un'esempio di form html per la redirectione potrebbe essere:

```
<html>
<body>
  <form action="https://ecommm.sella.it/gestpay/pagamvisa3d.asp" method="post">
    <input type="hidden" name="a" value="9000001">
    <input type="hidden" name="b" value="dffbihuihiuhe656erh12t5">
    <input type="hidden" name="c" value="http://www.sitoesercente.it/rientro.asp">
    <input type="submit" value="Autenticazione">
  </form>
</body>
</html>
```

Alla fine del processo di autenticazione il compratore sarà rediretto sul sito dell'esercente all'url specificata al parametro c nella redirectione.

La pagina predisposta dall'esercente per il rientro del compratore sarà richiamata con un parametro **PARES** (stringa cifrata contenente l'esito dell'autenticazione) che dovrà essere acquisito dall'esercente ed inoltrato a Banca Sella nella fase successiva.

#### Fase III: conclusione transazione

A questo punto l'esercente dispone di tutte le informazioni necessarie a concludere la transazione. Dovrà effettuare una nuova richiesta di autorizzazione ( **CallPagamS2S**).

Prima di richiamare tale chiamata al Web Service , però, è necessario assegnare a **GestPayS2S** tutte le informazioni necessarie, tramite la valorizzazione delle variabili:

- **shopLogin** (Codice Esercente)
- **uicCode** (Codice Divisa)
- **amount** (Importo)
- **shopTransactionID** (Codice identificativo transazione)
- **transKey** (identificativo transazione acquisito in fase I)
- **PARes** (stringa cifrata contenente esito autenticazione acquisita in fase II)

L'esito della transazione verrà restituito da Gestpay e sarà interpretabile come descritto nel paragrafo Richiesta di autorizzazione.

## Esempio di Transazioni con carte Verified by Visa in linguaggio VB.NET

### FASE I

```

myshoplogin= "9000001"
mycurrency= "242"
myamount= "56.28"
myshoptransactionID= "34az85ord19"
mycardnumber = "4444444444444448"
myexpmonth = "04"
myexpyear="08"
mybuyername="Mario Bianchi"
mybuyeremail="Mario.bianchi@isp.it"
myLanguage=""
myCVV=""
myMin=""
MyTransKey=""
MyPares=""
MyCuston=""
MyIdea=""

Dim MyService As New WSs2s
Dim XmlReturn As New XmlDocument

Try
    XmlReturn.LoadXml(MyService.callPagamS2S(myshoplogin, mycurrency, myamount,
myshoptransactionID, mycardnumber, myexpmonth, myexpyear, mybuyername, mybuyeremail,
myLanguage, myCVV, myMin, myTransKey, myPares, myCustom, myIdea).OuterXml)

Catch ex As Exception
    Dim message As String = ex.Message
    MsgBox("A run-time error occurred:" & ex.Message)
    Exit Sub
End Try

Dim root As XmlElement = doc.DocumentElement

If root.SelectSingleNode("TransactionType").InnerText = "PAGAM" Then
    If root.SelectSingleNode("ErrorCode").InnerText = "8006" Then
        MyTransactionKey=root.SelectSingleNode("transactionKey").InnerText
        MyVbvRisp=root.SelectSingleNode("VbvRisp").InnerText
    Else
        If root.SelectSingleNode("ErrorCode").InnerText = "0" Then
            MsgBox "Chiamata OK"
        Else
            MsgBox root.SelectSingleNode("ErrorCode").InnerText
            MsgBox root.SelectSingleNode("ErrorDescription").InnerText
        End if
    End If
End If

'procedo alla FASE 2

```

### FASE II

```

<%
'redirigo il cliente verso l'issuer per l'autenticazione
%>
<html>
<body>
    <form action="https://ecommerce.sella.it/gestpay/pagamvisa3d.asp" method="post">
        <input type="hidden" name="a" value="9000001">
        <input type="hidden" name="b" value="<%=myVbvRisp%>">
        <input type="hidden" name="c" value="http://www.sitoesercente.it/rientro.asp">
        <input type="submit" value="Autenticazione">
    </form>
</body>
</html>

```

### FASE III ( <http://www.sitoesercente.it/rientro.asp> )

```
'ricevo dal cliente il parametro PARES
Pares = request("PARES")
'istanzio il parametro transKey acquisito in risposta da GestPay nella FASE I
transKey = mytransKey
myshoplogin= "9000001"

mycurrency= "242"
myamount= "56.28"
myshoptransactionID= "34az85ord19"
mycardnumber = ""
myexpmonth = ""
myexpyear=""
mybuyername=""
mybuyeremail=""
myLanguage=""
myCVV=""
myMin=""
MyTransKey=transkey
MyPares=Pares
MyCustom=""
MyIdea=""

Dim MyService As New WSs2s
Dim XmlReturn As New XmlDocument

Try
    XmlReturn.LoadXml(MyService.callPagamS2S(myshoplogin, mycurrency, myamount,
myshoptransactionID, mycardnumber, myexpmonth, myexpyear, mybuyername, mybuyeremail,
myLanguage, myCVV, myMin, myTransKey, myPares, myCustom, myIdea).OuterXml)

Catch ex As Exception
    Dim message As String = ex.Message
    MsgBox("A run-time error occurred:" & ex.Message)
    Exit Sub
End Try

Dim root As XmlElement = doc.DocumentElement

If root.SelectSingleNode("TransactionType").InnerText = "PAGAM" Then
    If root.SelectSingleNode("ErrorCode").InnerText = "0" Then
        "Chiamata OK procedere per esito positivo"
    Else
        "Chiamata KO procedere per esito negativo"
        MsgBox root.SelectSingleNode("ErrorCode").InnerText
        MsgBox root.SelectSingleNode("ErrorDescription").InnerText
    End if
End If
```

## Gestione campo CVV

Nel presente documento chiameremo CVV il dato formato da 3 o 4 cifre stampate sulla carta di credito, chiamato con nomi diversi a seconda della compagnia emittente la carta (CVV2 per Visa, 4DBC per American Express, CVC2 per Mastercard).

In tutti i casi si tratta di un dato utilizzato per aumentare la sicurezza dei pagamenti, in quanto diminuisce le probabilità che la carta di credito sia utilizzata da chi non ne è in possesso.

Per effettuare transazioni inviando anche il campo CVV, è sufficiente effettuare le seguenti operazioni:

- 1) Richiedere alla Banca l'abilitazione al servizio CVV per le compagnie desiderate (o per tutte).  
Si tratta di un'operazione da effettuare una volta sola (ripetibile per altre compagnie se si richiede di abilitarne solo alcune).
- 2) Configurare il campo CVV come "PARAMETRO" nella sezione "Configurazione Cliente" - "Campi & Parametri" dell'ambiente di Back Office, relativamente alla pagina pubblica. Una volta visualizzato l'elenco dei campi che compongono la pagina, è sufficiente clickare sul nome del campo ("cvv") per accedere alla finestra di dettaglio del campo stesso. Da lì basta attivare la casella di controllo "Parametro" e confermare la modifica.

Per pagine create prima dell'anno 2005, il campo cvv potrebbe non essere presente. In tal caso occorre contattare il servizio di assistenza per ottenere la creazione di una nuova pagina, identica alla precedente, con il campo cvv in più.  
A questo punto sarà possibile procedere alla configurazione del campo come "PARAMETRO".

**Attenzione: occorre sempre pubblicare la pagina per rendere effettive le modifiche.**

- 3) A questo punto, per le carte appartenenti alle compagnie per cui il servizio è stato abilitato, è possibile effettuare transazioni contenenti il CVV, allo stesso modo in cui si effettuano le transazioni senza CVV ad eccezione del fatto che occorre comunicare al WebService GestPayS2S il valore del dato mediante la valorizzazione della variabile **CVV**

## Ordini di pagamento in ambiente di Test

Vi ricordiamo che per simulare l'autorizzazione di un ordine di pagamento in ambiente di test è necessario utilizzare una carta di credito in corso di validità.

Gli importi relativi agli ordini di pagamento autorizzati verranno prenotati nel plafond della carta utilizzata e non verranno mai addebitati, consigliamo pertanto di effettuare ordini di pagamento di importi esigui in modo da non decrementare completamente il plafond della carta utilizzata per i test.

## Links

Codici di Test (<http://service.easynolo.it/download.asp> )

Supporto Tecnico ([http://www.easynolo.it/ecommerce/assistenza/richiedi\\_assistenza.jsp?p=com\\_42](http://www.easynolo.it/ecommerce/assistenza/richiedi_assistenza.jsp?p=com_42))

F.A.Q. ([http://www.easynolo.it/ecommerce/assistenza/faq\\_ecommerce.jsp?p=com\\_55](http://www.easynolo.it/ecommerce/assistenza/faq_ecommerce.jsp?p=com_55))

Forum (<http://service.easynolo.it/forum.asp> )

E-Commerce su Sella.it (<https://www.sella.it/gbs/shop/ecommerce/gestpay/index.jsp> )

Ambiente di **Back Office** per esercenti **effettivi**

(<https://ecomm.sella.it/gestpay/backoffice/LoginGestPay.asp>)

Ambiente di **Back Office** per esercenti di **test**

(<https://testecomm.sella.it/gestpay/backoffice/logingestpay.asp>)

F I N E   D O C U M E N T O