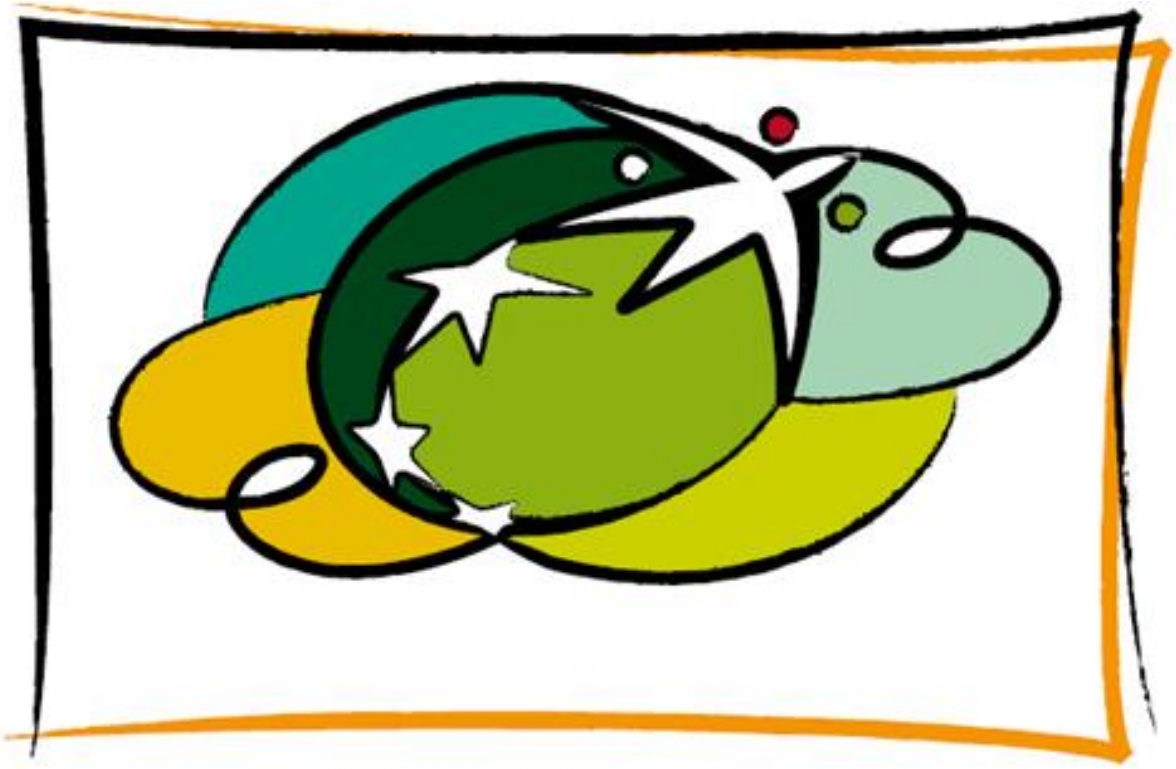


MANUALE DI INTEGRAZIONE
API & WEB SERVICES

BNL POSITIVITY
GRUPPO BNP PARIBAS

BNL POSITIVITY
GRUPPO BNP PARIBAS

Indice

1. INFORMAZIONI SULL' ASSISTENZA

La guida per l'integrazione di **e–POSitivity** che state leggendo, si rivelerà essere il riferimento per l'integrazione della vostra applicazione con i nostri sistemi autorizzativi.

Se dopo aver letto la documentazione non avete trovato una risposta alle vostre domande, contattate il Servizio Esercenti BNL POSitivity scrivendo all'indirizzo e-mail **ecommercebnlp@bnlpositivity.it** oppure chiamando il Numero Verde **800 955 954** e selezionando l'opzione **4**.

2. INTRODUZIONE

Nel presente documento vengono illustrate le specifiche tecniche dei **web services** e delle **API**, necessari per l'integrazione del proprio portale di e-commerce, con il sistema di pagamento elettronico **e-POSitivity**. I portali web, che offrono servizi di e-commerce prevedono un carrello con il riepilogo dei prodotti acquistati e un pulsante per azionare una richiesta di pagamento.

FIGURA 1 FLUSSO E-COMMERCE

Ultimata la fase di selezione dei prodotti, il cliente viene veicolato su una pagina di conferma dell'acquisto. In questa pagina, dopo aver selezionato lo strumento da utilizzare per il pagamento, viene invitato a premere il pulsante relativo alla funzione di "acquisto/compra" o di uguale

significato. Ricevuta la conferma di acquisto, l'esercente invia al gateway **e-POSitivity** la richiesta di pagamento.

3. CIRCUITI DI PAGAMENTO SUPPORTATI

e–POSitivity consente di accettare una vasta gamma di circuiti eseguendo una sola integrazione:

- L'accettazione delle carte Visa, Visa Electron, MasterCard e Maestro è compresa nel contratto di convenzionamento stipulato con **BNL POSitivity**. Per alcune categorie merceologiche è possibile che vi siano delle restrizioni nell'accettazione delle carte MasterCard e Maestro: in tali casi BNL POSitivity fornisce un avviso preventivo in fase di definizione del contratto

- L'iscrizione a Verified By Visa e 3D Secure Code è a carico di **BNL POSitivity** (avviene durante il censimento della pratica)

- Per i circuiti American Express, Diners e JCB è necessario sottoscrivere gli appositi moduli di convenzione con i rispettivi circuiti; la modulistica è disponibile contattando la propria Agenzia BNL, il proprio Agente **BNL POSitivity** o il Servizio Esercenti. **BNL POSitivity** provvederà ad inoltrare il contratto e ad attivare il circuito automaticamente al momento della ricezione del codice di convenzione assegnato.

- Laddove il cliente desiderasse anche l'attivazione del circuito Findomestic al fine di accettare transazioni con carte Aura e Nova, sarà necessario che sottoscriva il contratto; il modulo potrà essere richiesto direttamente a Findomestic o mediante la rete di vendita di **BNL POSitivity**.

- Per l'attivazione di Masterpass rivolgersi alla rete di vendita di **BNL POSitivity** o al servizio di assistenza e-commerce.

- Per l'attivazione di MyBank rivolgersi alla rete di vendita di **BNL POSitivity** o al servizio di assistenza e-commerce.

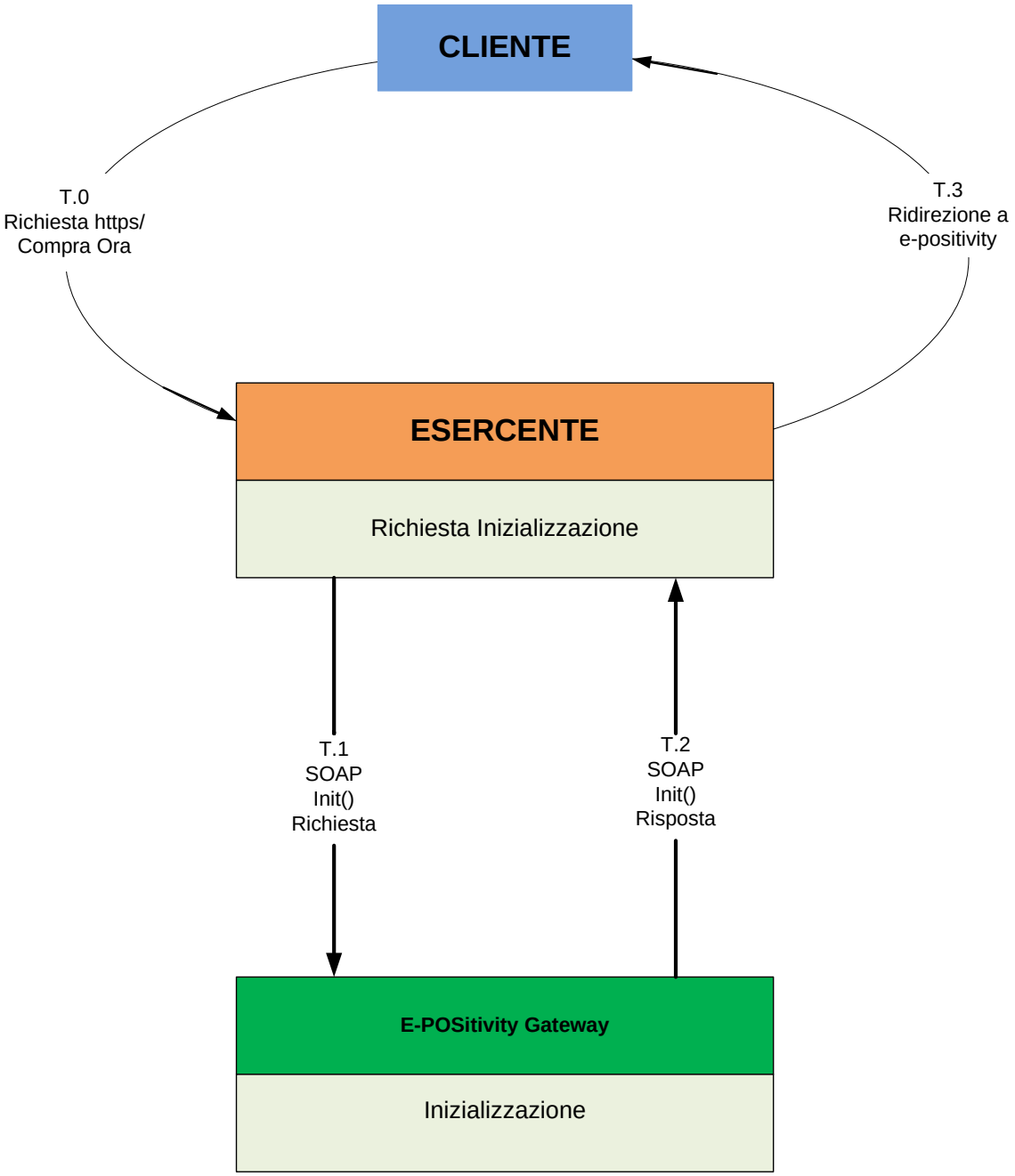
- L'esercente che vuole attivare PayPal sul POS virtuale **e–POSitivity** deve:

  - attivare un conto PayPal Business dal sito dal sito www.paypal.it e, in caso di necessità di assistenza, chiedere ad un operatore di PayPal chiamando il numero verde 800 976 359 del Gruppo Attivazione Commercianti;

  - richiedere le abilitazioni necessarie alle API PayPal per consentire l'esecuzione di transazioni da e–POSitivity;

- l'esercente, ricevuta l'abilitazione da PayPal, segnala a **ecommercebnlp@bnlpositivity.it** l'esito fornendo:
  - ✓ API Username
  - ✓ API Password
  - ✓ Signature
- BNL POSitivity provvede ad abilitare PayPal sul POS virtuale **e-POSitivity**.

FIGURA 2 DESCRIZIONE DEL PROCESSO



4. PRODOTTI

L'esercente può inviare richieste di autorizzazione verso **e-POSitivity** con due modalità diverse:

A. **Pagamenti on line (BUY NOW)**
B. **Pagamenti diretti**

Le tecnologie di integrazione con le richieste di autorizzazione elencate sono:

@ **API**

@ **WEB SERVICES**

A. PAGAMENTI ONLINE (BUY NOW)

Questa è l' opzione più semplice e sicura, consiste nell'utilizzo di una pagina di pagamento fornita da **BNL POSitivity** e residente sui nostri server. In questo caso, il titolare di carta verrà indirizzato sulle pagine di **BNL POSitivity** al momento del pagamento e potrà inserire i dati della carta di credito direttamente sul sito di **BNL POSitivity**, protetto con certificato SSL. Successivamente, il cliente titolare di carta verrà reindirizzato nuovamente al vostro sito Web, a cui trasmetteremo i dettagli relativi all'esito del pagamento (Figura 2).

Le fasi del processo di richiesta di autorizzazione utilizzando **Buy Now** possono essere riepilogate nei seguenti step:

@ Il Cliente (titolare di carta) visita il web store e riempie il carrello. Al termine della scelta degli articoli di acquisto, il web store indirizza il cliente al form **Buy Now**. Nessuna informazione relativa alla carta di credito verrà conservata dall'esercente;

@ Il web server di **Buy Now** mostra il form per il pagamento, raccoglie i dati (nome e cognome del titolare di carta, PAN, scadenza carta, CCV2) immessi attraverso una connessione SSL;

@ **e-POSitivity** provvede all'interrogazione del Directory Server per la verifica dell'iscrizione al servizio di **3D Secure;**

@ Nel caso di risposta positiva, il Merchant Plug In (MPI) passa il controllo all'Access Control Server (ACS) dell'emittente carta (issuer) che permette al cliente (titolare di carta) di inserire la password (in caso di transazione con VbV o 3D Securecode);

@ Verificata l'iscrizione al servizio **3D Secure** si procede all'invio della richiesta di autorizzazione verso i circuiti.

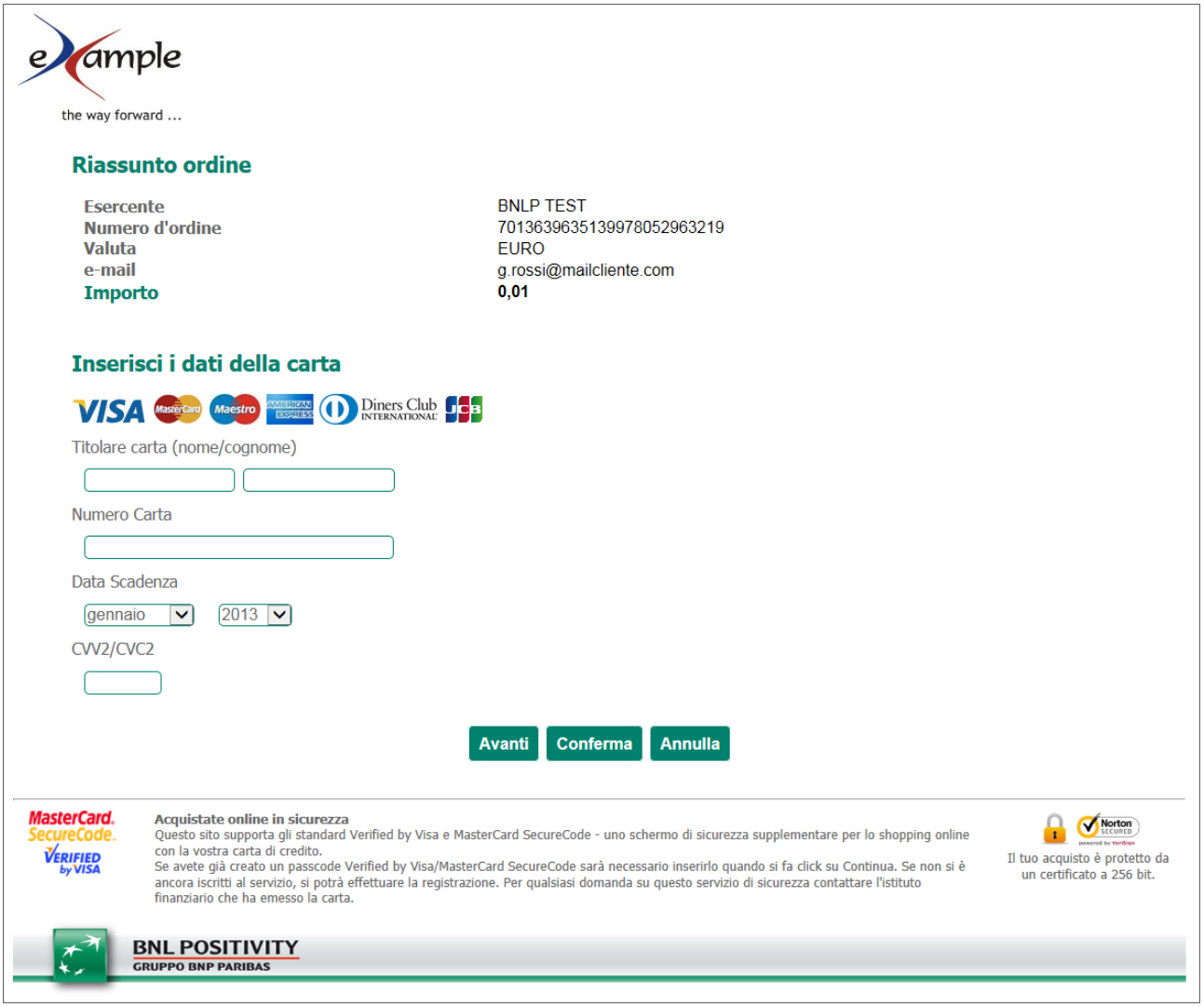
La risposta con l'esito dell'operazione verrà inviata sia all'esercente sia al titolare di carta. Il titolare di carta, oltre a visualizzarla sul suo browser, riceve anche un'e-mail di conferma.

Questa è la pagina di pagamento di **e-POSitivity**; il layout della stessa potrà in parte essere modificato inoltrando una richiesta a **BNL POSitivity**.

E' possibile personalizzare:

- @ l'immagine di **intestazione**
- @ l'immagine a **piè di pagina**
- @ il testo sulla **destra**
- @ il colore dello **sfondo**
- @ il font, la dimensione, lo stile (corsivo, grassetto...) e il colore dei **testi**
- @ il colore dei **pulsanti**
- @ il colore del **link**

FIGURA 3


example
the way forward ...
Riassunto ordine
Esercente
BNLP TEST
Numero d'ordine
70136396351399780529632 19
Valuta
EURO
e-mail
g.rossi@mailcliente.com
Importo
0,01
Inserisci i dati della carta
VISA
MasterCard
Maestro
AMERICAN EXPRESS
Diners Club INTERNATIONAL
JCB
Titolare carta (nome/cognome)
Numero Carta
Data Scadenza
gennaio
2013
CVV2/CVC2
Avanti
Conferma
Annulla
MasterCard SecureCode
VERIFIED by VISA
Acquistate online in sicurezza
Questo sito supporta gli standard Verified by Visa e MasterCard SecureCode - uno schermo di sicurezza supplementare per lo shopping online con la vostra carta di credito.
Se avete già creato un passcode Verified by Visa/MasterCard SecureCode sarà necessario inserirlo quando si fa click su Continua. Se non si è ancora iscritti al servizio, si potrà effettuare la registrazione. Per qualsiasi domanda su questo servizio di sicurezza contattare l'istituto finanziario che ha emesso la carta.
Norton SECURED
Il tuo acquisto è protetto da un certificato a 256 bit.
BNL POSITIVITY
GRUPPO BNP PARIBAS

B. PAGAMENTI DIRETTI

In questa modalità, la pagina di pagamento risiede sul server dell'esercente, quindi, dopo che il titolare di carta (cliente) ha concluso gli acquisti, cliccando su "paga" dal carrello verrà indirizzato in una pagina (modulo), dove inserirà i dati necessari per effettuare una transazione (PAN, data scadenza, CCV2).

Il sistema effettuerà una chiamata verso **e-POSitivity** passando i parametri opportunamente settati. Anche con questa modalità lo scambio delle informazioni sensibili viene effettuato con protocolli sicuri (**TLS 1.0**).

L'esercente in questo caso è responsabile dei dati della carta inseriti dai clienti e deve pertanto attenersi alle regole di PCI-DSS, **BNL POSitivity** è tenuta a controllare l'aderenza dei propri esercenti alle regole di PCI-DSS, pertanto potrà fare degli audit sui propri clienti che utilizzano questa modalità, al fine di garantirne la compliance in materia di sicurezza.

5. COLLOQUIO ESERCENTE → E-POSITIVITY

I messaggi scambiati tra l'applicazione che effettua le richieste (Customer **Buy Now**) e **e-POSitivity** sono conformi alle specifiche **SOAP 1.1**. Il protocollo utilizzato è l' HTTP 1.1 (RFC 2616), l'integrità dei dati e la cifratura delle informazioni vengono garantite attraverso il protocollo **TLS 1.0** (rfc2246.txt).

I servizi online offerti dal prodotto sono descritti dal file **WSDL** fornito.

6. COLLOQUIO E-POSITIVITY → DIRECTORY SERVER

Il protocollo **3D Secure** prevede l'invio delle richieste di Verifica Iscrizione Carta (VEREQ) su connessione cifrata (HTTPS).

In particolare per MasterCard e JCB è prevista una mutua autenticazione, Visa, invece, fornisce delle credenziali da inserire nel messaggio applicativo.

7. DATI NECESSARI

Per iniziare l'integrazione è fondamentale richiedere a **BNL POSitivity** l'accesso ai servizi. Dopo l'abilitazione verranno forniti i seguenti parametri da valorizzare durante le inizializzazioni per le richieste di autorizzazione.

Parametri necessari per l'utilizzo della tecnologia **Web Services:**

- @ Terminal Id
- @ Wsdl url (url del servizio web)

- File wsdl

Parametri necessari per l'utilizzo della tecnologia **API:**

- Terminal Id
- Ksig
- Server url
- Riferimenti da aggiungere al progetto (dll, jar, classi php)

8. INTEGRAZIONE E IMPLEMENTAZIONE

Questa sezione propone un semplice esempio per integrare il vostro sito Web con **e–POSitivity**. Le pagine di esempio sono proposte utilizzando i seguenti linguaggi di programmazione:

- vb.net
- c#
- php
- java

Si presuppone che lo sviluppatore abbia una conoscenza di base del linguaggio scelto.

A. INTEGRAZIONE API

L' integrazione con le API presuppone l'importazione dei riferimenti (dll, jar, classi php) nel proprio progetto e quindi richiamare i metodi messi a disposizione per l'inizializzazione della transazione. Riportiamo tutti i metodi resi disponibili per le API con i parametri attesi.

1. Pagamenti online

@ ClasseIgfsCgInit

Utilizzata per eseguire una inizializzazione della richiesta di pagamento

Sommario Properties Input	
Tipo[Dimensione]	**Property**
URL	**ServerURL** Indirizzo del server di destinazione della richiesta
Integer	**Timeout** Timeout massimo espresso in millisecondi di completamento di una richiesta
String[64]	**KSig** Chiave per firmare il messaggio
String[16]	**Tid** Codice terminale dell'esercente
String[256]	**ShopID** Chiave esterna identificante il pagamento
String[256]	**ShopUserRef** Identificativo chiente
PURCHASE, AUTH, VERIFY	**TrType** Tipologia di una richiesta
Long[12]	**Amount** Importo associato ad una richiesta in virgola virtuale (es. 100 = 1,00 EUR)
EUR	**CurrencyCode** Valuta associata ad una richiesta
IT, EN	**LangID** Lingua relativa alla pagina di inserimento dei dati sensibili associata ad una richiesta
URL[512]	**NotifyURL** URL relativo alla pagina di notifica esito di una richiesta

URL[512]	**ErrorURL** URL relativo alla pagina di errore associata ad una richiesta
String[256]	**AddInfo1** Campo a disposizione dell' esercente
String[256]	**AddInfo2** Campo a disposizione dell' esercente
String[256]	**AddInfo3** Campo a disposizione dell' esercente
String[256]	**AddInfo4** Campo a disposizione dell' esercente
String[256]	**AddInfo5** Campo a disposizione dell' esercente
String[100]	**Description** Causale di pagamento
String[268]	**FreeText** Testo libero

Per i dettagli sulla valorizzazione del campo freeText si rimanda al paragrafo seguente

String [512]	**callBackUrl** Pagina sulla quale ricevere la notifica server-to-server descritta nel paragrafo seguente

Metodi	
Tipo[Dimensione]	**Property**
Boolean	**execute()** Esegue la transazione
	resetFields() Azzeramento parametri di richiesta

Sommario Properties Output	
Tipo[Dimensione]	Property
String[16]	**Rc** Esito della richiesta
String[80]	**ErrorDesc** Descrizione di un errore/anomalia
String[32]	**PaymentID** Codice paymentID associato ad una richiesta
URL[512]	**RedirectURL** Url associato alla pagina di "buynow"

CallBackUrl server to server

In fase di richiesta autorizzativa, se è popolato il campo "CallBackUrl", il server invia una comunicazione sull' esito della transazione alla pagina esposta dall' esercente. I tentativi di comunicazione seguono le seguenti regole:

1. i dati postati sono identici a quelli inviati sulle url; responseSuccessURL, responseFailURL.
2. i tentativi verranno fatti finché il server non risponde.
3. il primo tentativo è fatto al termine della transazione dell' utente.
4. se il primo tentativo non va a buon fine i contatti verso il server dell' esercente verranno fatti successivamente, selezionando le transazioni eseguite nelle ultime 24 ore e fino ad un ora prima dell' esecuzione dell' autorizzazione.

Insegna Dinamica (campo freeText)

I merchant che ne richiedono il servizio, ad ogni richiesta autorizzativa se valorizzano il campo "freetext" con un codice alfanumerico di **6 caratteri**, tale codice sarà postato sull' estratto conto prodotto dalla banca emittente della carta di credito utilizzata dall' utente, comprensivo dell' insegna del punto vendita.

Esempio: se l'insegna del negozio è "PIZZA ITALY", e in fase di richiesta autorizzativa viene postato nel campo "freetext" il codice 34FGNO, sull' estratto conto dell' utente ci sarà; "PIZZA ITALY-34FGNO". Il prodotto si chiama insegna dinamica, perché il codice alfanumerico può cambiare ad ogni operazione di acquisto.

@ ClasseIgfsCgVerfy

Utilizzata per eseguire un'operazione di verifica dati della richiesta di pagamento.

Sommario Properties Input	
Tipo[Dimensione]	**Property**
URL	**ServerURL** Indirizzo del server di destinazione della richiesta
Integer	**Timeout** Timeout massimo espresso in millisecondi di completamento di una richiesta
String[64]	**KSig** Chiave per firmare il messaggio
String[16]	**Tid** Codice terminale dell'esercente
String[256]	**ShopID** Chiave esterna identificante il pagamento
String[256]	**PaymentID** Codice paymentID associato ad una richiesta

Metodi	
Tipo[Dimensione]	**Property**
Boolean	**execute()** Esegue la transazione
	resetFields() Azzeramento parametri di richiesta

Sommario Properties Output	
Tipo [Dim]	**Property**
String [16]	**RC** Esito della richiesta
String [80]	**Error Desc** Descrizione di un errore/anomalia
Long [16]	**TranID** Codice Ordine processato

String [32]	**AuthCode** Codice di autorizzazione restituito dall' issuer
String [1]	**EnrStatus** Stato di iscrizione carta al servizio 3D Secure
String [1]	**AuthStatus** Esito autenticazione carta al servizio 3D Secure
String [8]	**Brand** Brand carta di credito es. (VISA, MASTERCARD, …)
String [2,4]	**PayInstr** Strumento di pagamento utilizzato es. (CC, MYBK, …)

2. Pagamenti diretti con carte di credito

@ ClasseIgfsCgAuth

Viene utilizzata per le inizializzazioni delle richieste di autorizzazione dirette su carta di credito.

Sommario Properties Input	
Tipo[Dimensione]	**Property**
URL	**ServerURL** Indirizzo del server di destinazione della richiesta
Integer	**Timeout** Timeout massimo espresso in millisecondi di completamento di una richiesta
String[64]	**KSig** Chiave per firmare il messaggio
String[16]	**Tid** Codice terminale dell'esercente
String[256]	**ShopID** Chiave esterna identificante il pagamento
String[256]	**ShopUserRef** Identificativo cliente (es:email)
PURCHASE, AUTH, VERIFY	**TrType** Tipologia della richiesta
Long[12]	**Amount** Importo in virgola virtuale (es. 100 = 1,00 EUR)
EUR, USD	**CurrencyCode** Valuta
String[19]	**Pan** Numero di carte
String[4]	**Cvv2** Numero di sicurezza sul retro della carta
Integer[2]	**ExpireMonth** Mese di scadenza
Integer[4]	**ExpireYear** Anno di scadenza
String[256]	**AddInfo1** Campo a disposizione dell'esercente
String[256]	**AddInfo2** Campo a disposizione dell'esercente

String[256]	**AddInfo3** Campo a disposizione dell'esercente
String[256]	**AddInfo4** Campo a disposizione dell'esercente
String[256]	**AddInfo5** Campo a disposizione dell'esercente
String[1]	**EnrStatus** Dato ricevuto in risposta dall'MPI
String[1]	**AuthStatus** Dato ricevuto in risposta dall'MPI
String[28]	**Cavv** Dato ricevuto in risposta dall'MPI
String[28]	**Xid** Dato ricevuto in risposta dall'MPI
String[100]	**Description** Causale di pagamento
Boolean	**Recurrent** Pagamento ricorrente
String[268]	**FreeText** Testo libero

Metodi	
Tipo[Dimensione]	**Property**
Boolean	**execute()** Esegue la transazione
	resetFields() Azzeramento parametri di richiesta

Sommario Properties Output	
Tipo[Dimensione]	**Property**
String[16]	**Rc** Esito della richiesta
String[80]	**ErrorDesc** Descrizione dell'errore/anomalia
Long[16]	**TranID** Codice Ordine processato
String[32]	**AuthCode** Codice di autorizzazione restituito dall' issuer
String[8]	**Brand** Brand carta di credito es. (VISA, MASTERCARD,…)

@ ClasseIgfsCgConfirm

Viene utilizzata per movimentare un'autorizzazione fatta con carta di credito.

Sommario Properties Input	
Tipo[Dimensione]	**Property**
URL	**ServerURL** Indirizzo del server di destinazione della richiesta
Integer	**Timeout** Timeout massimo espresso in millisecondi di completamento di una richiesta
String[64]	**KSig** Chiave per firmare il messaggio
String[16]	**Tid** Codice terminale dell'esercente
String[256]	**ShopID** Chiave esterna identificante il pagamento
Long[12]	**Amount** Importo in virgola virtuale (es. 100 = 1,00 EUR)
Long[16]	**RefTranID** Codice Ordine relativo alla transazione da movimentare
Boolean	**SplitTran** è true se la conferma è parziale

Metodi	
Tipo[Dimensione]	**Property**
Boolean	**execute()** Esegue la transazione
	resetFields() Azzeramento parametri di richiesta

Sommario Properties Output	
Tipo[Dimensione]	**Property**
String[16]	**Rc** Esito della richiesta
String[80]	**ErrorDesc** Descrizione dell'errore/anomalia
Long[16]	**TranID** Codice Ordine processato
String[256]	**AddInfo1** Dati inviati in fase di autorizzazione dall'esercente
String[256]	**AddInfo2** Dati inviati in fase di autorizzazione dall'esercente
String[256]	**AddInfo3** Dati inviati in fase di autorizzazione dall'esercente
String[256]	**AddInfo4** Dati inviati in fase di autorizzazione dall'esercente
String[256]	**AddInfo5** Dati inviati in fase di autorizzazione dall'esercente
Long[16]	**PendingAmount** Eventuale importo non confermato

@ ClasseIgfsCgVoidAuth

Viene utilizzata per stornare una autorizzazione effettuata con carta di credito.

Sommario Properties Input	
Tipo[Dimensione]	**Property**
URL	**ServerURL** Indirizzo del server di destinazione della richiesta
Integer	**Timeout** Timeout massimo espresso in millisecondi di completamento di una richiesta
String[64]	**KSig** Chiave per firmare il messaggio
String[16]	**Tid** Codice terminale dell'esercente
String[256]	**ShopID** Chiave esterna identificante il pagamento
Long[12]	**Amount** Importo in virgola virtuale (es. 100 = 1,00 EUR)
Long[16]	**RefTranID** Codice Ordine relativo alla transazione da annullare

Metodi	
Tipo[Dimensione]	**Property**
Boolean	**execute()** Esegue la transazione
	resetFields() Azzeramento parametri di richiesta

Sommario Properties Output	
Tipo[Dimensione]	**Property**
String[16]	**Rc** Esito della richiesta
String[80]	**ErrorDesc** Descrizione dell'errore/anomalia
Long[16]	**TranID** Codice Ordine processato
String[256]	**AddInfo1** Dati inviati in fase di autorizzazione dall'esercente
String[256]	**AddInfo2** Dati inviati in fase di autorizzazione dall'esercente
String[256]	**AddInfo3** Dati inviati in fase di autorizzazione dall'esercente
String[256]	**AddInfo4** Dati inviati in fase di autorizzazione dall'esercente
String[256]	**AddInfo5** Dati inviati in fase di autorizzazione dall'esercente

@ ClasseIgfsCgCredit

Viene utilizzata per riaccreditare un'autorizzazione effettuata con carta di credito.

Sommario Properties Input	
Tipo[Dimensione]	**Property**
URL	**ServerURL** Indirizzo del server di destinazione della richiesta
Integer	**Timeout** Timeout massimo espresso in millisecondi di completamento di una richiesta
String[64]	**KSig** Chiave per firmare il messaggio
String[16]	**Tid** Codice terminale dell'esercente
String[256]	**ShopID** Chiave esterna identificante il pagamento
Long[12]	**Amount** Importo in virgola virtuale (es. 100 = 1,00 EUR)
Long[16]	**RefTranID** Codice Ordine relativo alla transazione da riaccreditare
Boolean	**SplitTran** è true se la conferma è parziale

Metodi	
Tipo[Dimensione]	**Property**
Boolean	**execute**() Esegue la transazione
	resetFields() Azzeramento parametri di richiesta

Sommario Properties Output	
Tipo[Dimensione]	**Property**
String[16]	**Rc** Esito della richiesta
String[80]	**ErrorDesc** Descrizione dell'errore/anomalia
Long[16]	**TranID** Codice Ordine processato
String[256]	**AddInfo1** Dati inviati in fase di autorizzazione dall'esercente
String[256]	**AddInfo2** Dati inviati in fase di autorizzazione dall'esercente
String[256]	**AddInfo3** Dati inviati in fase di autorizzazione dall'esercente
String[256]	**AddInfo4** Dati inviati in fase di autorizzazione dall'esercente
String[256]	**AddInfo5** Dati inviati in fase di autorizzazione dall'esercente

3. Funzionalità MPI

@ ClasseIgfsCGMPIEnroll

Viene utilizzata per verificare se la carta è inscritta al servizio **3D Secure**.

Sommario Properties Input	
Tipo[Dimensione]	**Property**
URL	**ServerURL** Indirizzo del server di destinazione della richiesta
Integer	**Timeout** Timeout massimo espresso in millisecondi di completamento di una richiesta
String[64]	**KSig** Chiave per firmare il messaggio
String[16]	**Tid** Codice terminale dell'esercente
String[256]	**ShopID** Chiave esterna identificante il pagamento
String[256]	**ShopUserRef** Identificativo cliente (es:email)
Long[12]	**Amount** Importo in virgola virtuale (es. 100 = 1,00 EUR)
EUR, USD	**CurrencyCode** Valuta
String[19]	**Pan** Numero di carte
Integer[2]	**ExpireMonth** Mese di scadenza
Integer[4]	**ExpireYear** Anno di scadenza
URL[512]	**TermURL** URL dove ACS ridirige a fine del processo di Auth
String[100]	**Description** Valore riportato nella form di ACS
String[256]	**AddInfo1** Campo a disposizione dell'esercente
String[256]	**AddInfo2** Campo a disposizione dell'esercente

String[256]	**AddInfo3** Campo a disposizione dell'esercente
String[256]	**AddInfo4** Campo a disposizione dell'esercente
String[256]	**AddInfo5** Campo a disposizione dell'esercente

Sommario Metodi	
Tipo[Dimensione]	**Property**
Boolean	**execute**() Esegue la transazione
	resetFields() Azzeramento parametri di richiesta

Sommario Properties Output	
Tipo[Dimensione]	**Property**
String[16]	**Rc** Esito della richiesta
String[80]	**ErrorDesc** Descrizione dell'errore/anomalia
String[28]	**Xid** Identificativo della transazione
String[1]	**EnrStatus** Y - Authentication Available Yes No N - Cardholder Not Enrolled No Yes U - Unable to Authenticate No Yes E - any error message here
String	**PaReq** Process auth response
String	**Md** Merchant Data
URL[512]	**AcsURL** URL dall'ACS dove il browser deve essere rediretto. E' necessario postare i seguenti parametri all URL ricevuta - **PaReq** - **MD** - **TermURL**
String	**AcsPage** Codice HTML da inviare al Browser per arrivare alla pagine ACS

@ ClasseIgfsCGMPIAuth

Viene utilizzata per verificare il risultato dell'autenticazione **3D Secure** del titolare.

Sommario Properties Input	
Tipo[Dimensione]	**Property**
URL	**ServerURL** Indirizzo del server di destinazione della richiesta
Integer	**Timeout** Timeout massimo espresso in millisecondi di completamento di una richiesta
String[64]	**KSig** Chiave per firmare il messaggio
String[16]	**Tid** Codice terminale dell'esercente
String[256]	**ShopID** Chiave esterna identificante il pagamento
String	**PaRes** Process auth response (da msg enrollment)
String	**Md** Merchant Data

Sommario Metodi	
Tipo[Dimensione]	**Property**
Boolean	**execute**() Esegue la transazione
	resetFields() Azzeramento parametri di richiesta

Sommario Properties Output	
Tipo[Dimensione]	**Property**
String[16]	**Rc** Esito della richiesta
String[80]	**ErrorDesc** Descrizione dell'errore/anomalia
String[28]	**Xid** Identificativo della transazione
String[1]	**AuthStatus** Y - Authenticated N - Cardholder Not Authenticated U - Unable to Authenticate E - any error
String[28]	**Cavv** CardHolder Auth. Verification Value
String[8]	**Eci** Electronic Commerce Indicator

B. INTEGRAZIONE WEB SERVICES

L'integrazione con i servizi web di **e-POSitivity** presuppone l'importazione di file **wsdl** nel proprio progetto. Riportiamo tutte le proprietà di richiesta e risposta.

1. Pagamenti online

@ Metodo init()

Utilizzato per eseguire una inizializzazione della richiesta di pagamento

PaymentInitRequest	
Tipo[Dimensione]	Property
String	**Signature** Firma del messaggio composta dalla concatenazione dei campi: - Tid - ShopID - ShopUserRef - TrType - Amount - CurrencyCode - LangID - NotifyURL - ErrorURL - AddInfo1 - AddInfo2 - AddInfo3 - AddInfo4 - AddInfo5 Per il calcolo della firma si veda l'*APPENDICE A.*
String[16]	**Tid** Codice terminale dell'esercente
String[256]	**ShopID** Chiave esterna identificante il pagamento
String[256]	**ShopUserRef** Identificativo cliente (es:email)

PURCHASE, AUTH, VERIFY	**TrType** Tipologia della richiesta
Long[12]	**Amount** Importo in virgola virtuale (es. 100 = 1,00 EUR)
EUR	**CurrencyCode** Valuta
IT, EN	**LangID** Codice iso 639-2 relativo alla pagina di inserimento dei dati di pagamento
URL[512]	**NotifyURL** URL relativo alla pagina di notifica esito
URL[512]	**ErrorURL** URL relativo alla pagina di errore
String[256]	**AddInfo1** Campo a disposizione dell'esercente
String[256]	**AddInfo2** Campo a disposizione dell'esercente
String[256]	**AddInfo3** Campo a disposizione dell'esercente
String[256]	**AddInfo4** Campo a disposizione dell'esercente
String[256]	**AddInfo5** Campo a disposizione dell'esercente
String[100]	**Description** Causale di pagamento
String[268]	**FreeText** Testo libero

PaymentInitResponse	
Tipo[Dimensione]	**Property**
Boolean	**Error** Restituisce true in presenza di un errore/anomalia
String[16]	**Rc** Esito della richiesta
String[80]	**ErrorDesc** Descrizione dell'errore/anomalia
String[32]	**PaymentID** Codice paymentID associato alla richiesta
URL[512]	**RedirectURL** Url associato alla pagina di "buynow"

Insegna Dinamica (campo freeText)

I merchant che ne richiedono il servizio, ad ogni richiesta autorizzativa se valorizzano il campo "freetext" con un codice alfanumerico di 6 caratteri, tale codice sarà postato sull' estratto conto prodotto dalla banca emittente della carta di credito utilizzata dall' utente, comprensivo dell' insegna del punto vendita.

Esempio: se l'insegna del negozio è "PIZZA ITALY", e in fase di richiesta autorizzativa viene postato nel campo "freetext" il codice 34FGNO, sull' estratto conto dell' utente ci sarà; "PIZZA ITALY-34FGNO". Il prodotto si chiama insegna dinamica, perché il codice alfanumerico può cambiare ad ogni operazione di acquisto.

@ Metodo verify()

Utilizzato per eseguire un'operazione di verifica dati della richiesta di pagamento

PaymentVerifyRequest	
Tipo[Dimensione]	**Property**
String	**Signature** Firma del messaggio composta dalla concatenazione dei campi: - Tid - ShopID - PaymentID Per il calcolo della firma si veda l'*APPENDICE A.*
String[16]	**Tid** Codice terminale dell'esercente
String[256]	**ShopID** Chiave esterna identificante il pagamento
String[32]	**PaymentID** Codice paymentID associato alla richiesta

PaymentVerifyResponse	
Tipo[Dimensione]	Property
Boolean	**Error** Restituisce true in presenza di un errore/anomalia
String[16]	**Rc** Esito della richiesta
String[80]	**ErrorDesc** Descrizione di un errore/anomalia
Long[16]	**TranID** Codice Ordine processato
String[32]	**AuthCode** Codice di autorizzazione restituito dall' issuer
String[1]	**EnrStatus** Stato di iscrizione carta al servizio 3D Secure
String[1]	**AuthStatus** Esito autenticazione carta al servizio 3D Secure
String[32]	**Brand** Brand carta di credito es. (VISA, MASTERCARD,…)

2. Pagamenti diretti

@ Metodo Auth()

Utilizzato per eseguire una inizializzazione della richiesta di pagamento diretta con carta di credito.

PaymentAuthRequest	
Tipo[Dimensione]	**Property**
String	**Signature** Firma del messaggio composta dalla concatenazione dei campi: - Tid - ShopID - ShopUserRef - TrType - Amount - CurrencyCode - Pan - Cvv2 - ExpireMonth - ExpireYear - AddInfo1 - AddInfo2 - AddInfo3 - AddInfo4 - AddInfo5 Per il calcolo della firma si veda l'*APPENDICE A.*
String[16]	**Tid** Codice terminale dell'esercente
String[256]	**ShopID** Chiave esterna identificante il pagamento
String[256]	**ShopUserRef** Identificativo cliente (es:email)

PURCHASE, AUTH, VERIFY	**TrType** Tipologia della richiesta
Long[12]	**Amount** Importo in virgola virtuale (es. 100 = 1,00 EUR)
EUR, USD	**CurrencyCode** Valuta
String[19]	**Pan** Numero di carte
String[4]	**Cvv2** Numero di sicurezza sul retro della carta
Integer[2]	**ExpireMonth** Mese di scadenza
Integer[4]	**ExpireYear** Anno di scadenza
String[256]	**AddInfo1** Campo a disposizione dell'esercente
String[256]	**AddInfo2** Campo a disposizione dell'esercente
String[256]	**AddInfo3** Campo a disposizione dell'esercente
String[256]	**AddInfo4** Campo a disposizione dell'esercente
String[256]	**AddInfo5** Campo a disposizione dell'esercente
String[1]	**EnrStatus** Dato ricevuto in risposta dall'MPI
String[1]	**AuthStatus** Dato ricevuto in risposta dall'MPI
String[28]	**Cavv** Dato ricevuto in risposta dall'MPI
String[28]	**Xid** Dato ricevuto in risposta dall'MPI
String[100]	**Description** Causale di pagamento
Boolean	**Recurrent** Pagamento ricorrente
String[268]	**FreeText** Testo libero

PaymentAuthResponse	
Tipo[Dimensione]	Property
Boolean	**Error** Restituisce true in presenza di un errore/anomalia
String[16]	**Rc** Esito della richiesta
String[80]	**ErrorDesc** Descrizione dell'errore/anomalia
Long[16]	**TranID** Codice Ordine processato
String[32]	**AuthCode** Codice di autorizzazione restituito dall' issuer
String[32]	**Brand** Brand carta di credito es. (VISA, MASTERCARD,…)

@ Metodo confirm()

Utilizzato per movimentare una autorizzazione effettuata con carta di credito

PaymentConfirmRequest	
Tipo[Dimensione]	Property
String	**Signature** Firma del messaggio composta dalla concatenazione dei campi: - Tid - ShopID - Amount - RefTranID Per il calcolo della firma si veda l'*APPENDICE A.*
String[16]	**Tid** Codice terminale dell'esercente
String[256]	**ShopID** Chiave esterna identificante il pagamento
Long[12]	**Amount** Importo in virgola virtuale (es. 100 = 1,00 EUR)
Long[16]	**RefTranID** Codice Ordine relativo alla transazione da movimentare
Boolean	**SplitTran** è true se la conferma è parziale

PaymentConfirmResponse	
Tipo[Dimensione]	Property
Boolean	**Error** Restituisce true in presenza di un errore/anomalia
String[16]	**Rc** Esito della richiesta
String[80]	**ErrorDesc** Descrizione dell'errore/anomalia
Long[16]	**TranID** Codice Ordine processato
String[256]	**AddInfo1** Dati inviati in fase di autorizzazione dall'esercente
String[256]	**AddInfo2** Dati inviati in fase di autorizzazione dall'esercente
String[256]	**AddInfo3** Dati inviati in fase di autorizzazione dall'esercente
String[256]	**AddInfo4** Dati inviati in fase di autorizzazione dall'esercente
String[256]	**AddInfo5** Dati inviati in fase di autorizzazione dall'esercente
Long[12]	**PendingAmount** Eventuale importo non confermato

@ Metodo voidAuth()

Utilizzata per stornare un'autorizzazione eseguita con carta di credito

PaymentVoidAuthRequest	
Tipo[Dimensione]	**Property**
String	**Signature** Firma del messaggio composta dalla concatenazione dei campi: - Tid - ShopID - Amount - RefTranID Per il calcolo della firma si veda l'*APPENDICE A.*
String[16]	**Tid** Codice terminale dell'esercente
String[256]	**ShopID** Chiave esterna identificante il pagamento
Long[12]	**Amount** Importo in virgola virtuale (es. 100 = 1,00 EUR)
Long[16]	**RefTranID** Codice Ordine relativo alla transazione da annullare

PaymentVoidAuthResponse	
Tipo[Dimensione]	Property
Boolean	**Error** Restituisce true in presenza di un errore/anomalia
String[16]	**Rc** Esito della richiesta
String[80]	**ErrorDesc** Descrizione dell'errore/anomalia
Long[16]	**TranID** Codice Ordine processato
String[256]	**AddInfo1** Dati inviati in fase di autorizzazione dall'esercente
String[256]	**AddInfo2** Dati inviati in fase di autorizzazione dall'esercente
String[256]	**AddInfo3** Dati inviati in fase di autorizzazione dall'esercente
String[256]	**AddInfo4** Dati inviati in fase di autorizzazione dall'esercente
String[256]	**AddInfo5** Dati inviati in fase di autorizzazione dall'esercente

@ Metodo credit()

Utilizzato per riaccreditare un'autorizzazione effettuata con carta di credito

PaymentCreditRequest	
Tipo[Dimensione]	**Property**
String	**Signature** Firma del messaggio composta dalla concatenazione dei campi: - Tid - ShopID - Amount - RefTranID Per il calcolo della firma si veda l'*APPENDICE A.*
String[16]	**Tid** Codice terminale dell'esercente
String[256]	**ShopID** **Chiave esterna identificante il pagamento**
Long[12]	**Amount** Importo in virgola virtuale (es. 100 = 1,00 EUR)
Long[16]	**RefTranID** Codice Ordine relativo alla transazione da riaccreditare
Boolean	**SplitTran** è true se la conferma è parziale

PaymentCreditResponse	
Tipo[Dimensione]	Property
Boolean	**Error** Restituisce true in presenza di un errore/anomalia
String[16]	**Rc** Esito della richiesta
String[80]	**ErrorDesc** Descrizione dell'errore/anomalia
Long[16]	**TranID** Codice Ordine processato
String[256]	**AddInfo1** Dati inviati in fase di autorizzazione dall'esercente
String[256]	**AddInfo2** Dati inviati in fase di autorizzazione dall'esercente
String[256]	**AddInfo3** Dati inviati in fase di autorizzazione dall'esercente
String[256]	**AddInfo4** Dati inviati in fase di autorizzazione dall'esercente
String[256]	**AddInfo5** Dati inviati in fase di autorizzazione dall'esercente

3. Funzioni MPI

@ Metodo enroll()

Utilizzata per verificare se la carta è inscritta a **3D Secure**.

MPIEnrollRequest	
Tipo[Dimensione]	**Property**
String	**Signature** Firma del messaggio composta dalla concatenazione dei campi: - Tid - ShopID - ShopUserRef - Amount - CurrencyCode - Pan - ExpireMonth - ExpireYear - TermUrl - Description - AddInfo1 - AddInfo2 - AddInfo3 - AddInfo4 - AddInfo5 Per il calcolo della firma si veda l'*APPENDICE A.*
String[16]	**Tid** Codice terminale dell'esercente
String[256]	**ShopID** Chiave esterna identificante il pagamento
String[256]	**ShopUserRef** Identificativo cliente (es:email)
Long[12]	**Amount** Importo in virgola virtuale (es. 100 = 1,00 EUR)
EUR, USD	**CurrencyCode** Valuta

String[19]	**Pan** Numero di carte
Integer[2]	**ExpireMonth** Mese di scadenza
Integer[4]	**ExpireYear** Anno di scadenza
URL[512]	**TermURL** URL dove ACS ridirige a fine del processo di Auth
String[100]	**Description** Valore riportato nella form di ACS
String[256]	**AddInfo1** Campo a disposizione dell'esercente
String[256]	**AddInfo2** Campo a disposizione dell'esercente
String[256]	**AddInfo3** Campo a disposizione dell'esercente
String[256]	**AddInfo4** Campo a disposizione dell'esercente
String[256]	**AddInfo5** Campo a disposizione dell'esercente

MPIEnrollResponse	
Tipo[Dimensione]	Property
Boolean	**Error** Restituisce true in presenza di un errore/anomalia
String[16]	**Rc** Esito della richiesta
String[80]	**ErrorDesc** Descrizione dell'errore/anomalia
String[28]	**Xid** Identificativo della transazione
String[1]	**EnrStatus** Y - Authentication Available Yes No N - Cardholder Not Enrolled No Yes U - Unable to Authenticate No Yes E - any error message here
String	**PaReq** Process auth response
String	**Md** Merchant Data
URL[512]	**AcsURL** URL dall'ACS dove il browser deve essere rediretto E' necessario aggiungere i seguenti parametri all URL ricevuta - **PaReq** - **MD** - **TermURL**
String	**AcsPage** Codice HTML da inviare al Browser per arrivare alla pagine ACS

@ Metodo auth()

Utilizzata per verificare il risultato dell'autorizzazione **3D Secure** del titolare.

MPIAuthRequest	
Tipo[Dimensione]	**Property**
String	**Signature** Firma del messaggio composta dalla concatenazione dei campi: - Tid - ShopID - PaRes - Md Per il calcolo della firma si veda l'*APPENDICE A.*
String[16]	**Tid** Codice terminale dell'esercente
String[256]	**ShopID** Chiave esterna identificante il pagamento
String	**PaRes** Process auth response (da msg enrollment)
String	**Md** Merchant Data

MPIAuthResponse	
Tipo[Dimensione]	**Property**
Boolean	**Error** Restituisce true in presenza di un errore/anomalia
String[16]	**Rc** Esito della richiesta
String[80]	**ErrorDesc** Descrizione dell'errore/anomalia
String[28]	**Xid** Identificativo della transazione
String[1]	**AuthStatus** Y - Authenticated N - Cardholder Not Authenticated U - Unable to Authenticate E - any error
String[28]	**Cavv** CardHolder Auth. Verification Value
String[8]	**Eci** Electronic Commerce Indicator

C. Procedura delle chiamate

Riportiamo di seguito la corretta procedura di chiamata delle classi nei servizi di pagamento che richiedono più step per la corretta esecuzione della transazione.

@ Pagamenti BuyNow autorizzazione o purchase:

La classe richiede una prima inizializzazione che indirizza il cliente sulla pagina di pagamento di **BNL POSitivity** utilizzando la classe "**IgfsCgInit o init()"**. Dopo la transazione il gateway di **BNL POSitivity** indirizza il browser dell'utente sulla pagina di notifica, dove deve essere istanziata la classe "**IgfsCgVerfy o verify".** Se non viene correttamente eseguita questa procedura la transazione sarà annullata dal sistema.

@ Pagamenti diretti verifica del 3DSecure:

@ Prima di procedere con i pagamenti diretti è necessario verificare se la carta è iscritta al servizio 3DSecure, quindi chiamare la directory dell'issuer per verificare se la carta richiede il token/password, utilizzando la classe "igfsCGMPIEnroll", che con risposta positiva indirizza l'utente sulla pagina dell'emittente carta per l'inserimento della password. Successivamente la banca indirizza l'utente sulla pagina dell'esercente, quindi deve essere richiamata la classe "IgfsCGMPIAuth" per la verifica dell'esito.

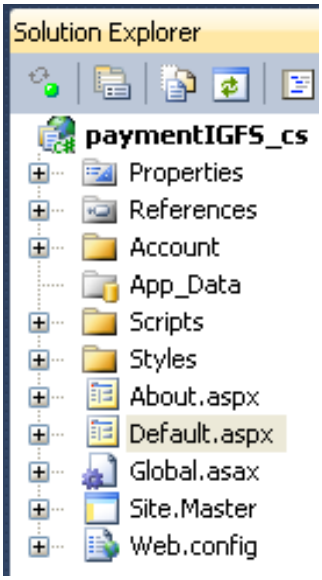
D. IMPLEMENTAZIONE API

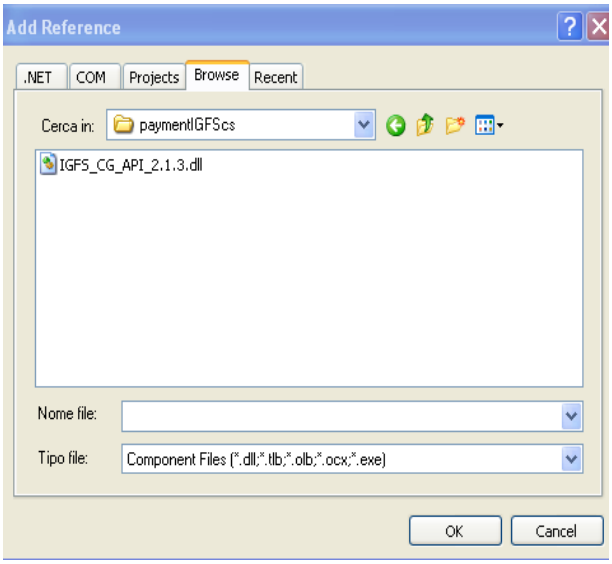
Al fine di facilitare la comprensione dei metodi concernenti, l'utilizzo delle API sopra descritte riportiamo script di esempio in diversi linguaggi.

@ DOT.NET

Riportiamo un'integrazione fatta con Visual Studio 2010:

a. Dalla solution del progetto aggiungere il riferimento alla dll **IGFS_CG_API_2.1.3.dll**





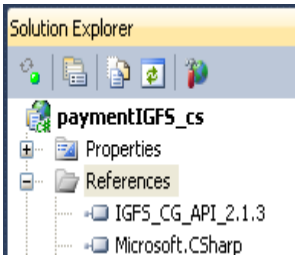


TABELLA 1 VB:NET PAGAMENTO ONLINE

```
Protected Sub btnPagamentiOnlineAuth_Click(ByVal sender As Object, ByVal e As EventArgs) Handles btnPagamentiOnlineAuth.Click

    Dim result As String = ""
    Dim tranID As String = ""
    Dim brand As String = ""

    'DICHIARARE L'URL DELLA PAGINA BUY NOW DI BNLPositivity
    Dim serverURL As String = "https://test.bnlpositivity.it/BNL_CG_SERVICES/services"
    'DICHIARARE UNA PAGINA DI REDIRECT IN CASO DI ERRORE
    Dim errorURL As String = "http://localhost:2637/errorPayment.aspx"
    'DICHIARARE UNA PAGINA DI REDIRECT DOPO LA TRANSAZIONE
    Dim notifyURL As String = "http://localhost:2637/notifyPayment.aspx"


    'DICHIARARE I PARAMETRI PER L'INIZIALIZZAZIONE DELLA RICHIESTA DI PAGAMENTO
    'TUTTI I PARAMETRI RIPORTATI SONO SOLO A TITOLO DI ESEMPIO
    Dim objInit As New IgfsCgInit
    With objInit
       .ServerURL = New Uri(serverURL)
       .Timeout = 1500000
       .Tid = "BNL_SELECT"
       .KSig = " xxxxiSb08fs8Bxxx9Yhq3Ub99E8="
       .ShopID = "00000001"
       .ShopUserRef = "email@clienteemail.it"
       .TrType = it.netsw.apps.igfs.cg.coms.api.init.TrType.AUTH
       .CurrencyCode = it.netsw.apps.igfs.cg.coms.api.CurrencyCode.EUR
       .LangID = it.netsw.apps.igfs.cg.coms.api.LangID.IT
       .Amount = Me.txtIMPORTO.Text
       .ErrorURL = New Uri(errorURL)
       .NotifyURL = New Uri(notifyURL)
    End With


    'ESECUZIONE RICHIESTA DI TRANSAZIONE
    If Not objInit.execute() Then
```

```
        'RINDIRIZZAMENTO IN CASO DI ERRORE
        Response.Redirect(errorURL & "?rc=" & objInit.Rc & "&errorDesc=" & objInit.ErrorDesc)
    End If

    'RIPRENDO IL PAYMENTID PER EVENTUALI SALVATAGGI ES. SUL DB
    Dim paymentId As String = objInit.PaymentID

    'REDIRECT VERSO LA PAGINA BUY NOW DI BNLPositivity
    Dim redirectURL As Uri = objInit.RedirectURL
    Response.Redirect(redirectURL.ToString())

End Sub
```

TABELLA 2 VB.NET PAGAMENTO DIRETTO

```
Protected Sub btnPagamentiDirettiAuth_Click(ByVal sender As Object, ByVal e As EventArgs) Handles btnPagamentiDirettiAuth.Click
    Dim result As String = ""
    Dim tranID As String = ""
    Dim brand As String = ""

    'DICHIARARE L'URL DELLA PAGINA BUY NOW DI BNLPositivity
    Dim serverURL As String = "https://test.bnlpositivity.it/BNL_CG_SERVICES/services"

    'DICHIARARE UNA PAGINA DI REDIRECT IN CASO DI ERRORE
    Dim errorURL As String = "http://localhost:2637/errorPayment.aspx"

    'DICHIARARE I PARAMETRI PER L'INIZIALIZZAZIONE DELLA RICHIESTA DI PAGAMENTO
    'TUTTI I PARAMETRI RIPORTATI SONO SOLO A TITOLO DI ESEMPIO
    Dim objAuth As New IgfsCgAuth
    With objAuth
        .ServerURL = New Uri(serverURL)
        .Timeout = 15000
        .Tid = "BNL_SELECT"
        .KSig = " xxxxiSb08fs8Bxxx9Yhq3Ub99E8="
        .ShopID = "00000003"
        .ShopUserRef = "email@clienteemail.it"
        .TrType = it.netsw.apps.igfs.cg.coms.api.init.TrType.AUTH
        .CurrencyCode = it.netsw.apps.igfs.cg.coms.api.CurrencyCode.EUR
        .Amount = Convert.ToInt64(Me.txtIMPORTO.Text)
        .Pan = Me.txtPAN.Text
        .ExpireMonth = Me.txtSCADENZA_MESE.Text
        .ExpireYear = Me.txtSCADENZA_ANNO.Text
        .Cvv2 = Me.txtCVV.Text
    End With


    'ESECUZIONE RICHIESTA DI TRANSAZIONE
    If Not objAuth.execute() Then
        'REDIRECT IN CASO DI ERRORE
        Response.Redirect(errorURL & "?rc=" & objAuth.Rc & "&errorDesc=" & objAuth.ErrorDesc)
    End If

    'RECUPERARE IL RISULTATO DELLA TRANSAZIONE
    If Not IsNothing(objAuth.Rc) Then result = objAuth.Rc
    If Not IsNothing(objAuth.TranID) Then tranID = objAuth.TranID
    If Not IsNothing(objAuth.Brand) Then brand = objAuth.Brand

End Sub
```

TABELLA 3 C# PAGAMENTO ONLINE

```
protected void btnPagamentiOnlineAuth_Click(object sender, EventArgs e)
  {

    string result = "";
    string tranID = "";
    string brand = "";

    //DICHIARARE L'URL DELLA PAGINA BUY NOW DI BNLPositivity
    string serverURL = "https://test.bnlpositivity.it/BNL_CG_SERVICES/services";
    //DICHIARARE UNA PAGINA DI REDIRECT IN CASO DI ERRORE
    string errorURL = "http://localhost:2637/errorPayment.aspx";
    //DICHIARARE UNA PAGINA DI REDIRECT DOPO LA TRANSAZIONE
    string notifyURL = "http://localhost:2637/notifyPayment.aspx";

    //DICHIARARE I PARAMETRI PER L'INIZIALIZZAZIONE DELLA RICHIESTA DI PAGAMENTO
    //TUTTI I PARAMETRI RIPORTATI SONO SOLO A TITOLO DI ESEMPIO
    IgfsCgInit objInit = new IgfsCgInit();
    var _with1 = objInit;
    objInit.ServerURL = new Uri(serverURL);
    objInit.Timeout = 1500000;
    objInit.Tid = "BNL_SELECT";
    objInit.KSig = " xxxxiSb08fs8Bxxx9Yhq3Ub99E8=";
    objInit.ShopID = "00000001";
    objInit.ShopUserRef = "email@clienteemail.it";
    objInit.TrType = it.netsw.apps.igfs.cg.coms.api.init.TrType.AUTH;
    objInit.CurrencyCode = it.netsw.apps.igfs.cg.coms.api.CurrencyCode.EUR;
    objInit.LangID = it.netsw.apps.igfs.cg.coms.api.LangID.IT;
    objInit.Amount = Convert.ToInt64(this.txtIMPORTO.Text) * 100;
    objInit.ErrorURL = new Uri(errorURL);
    objInit.NotifyURL = new Uri(notifyURL);



    //ESECUZIONE RICHIESTA DI TRANSAZIONE
    if (!objInit.execute())
    {
      //REDIRECT IN CASO DI ERRORE
      Response.Redirect(errorURL + "?rc=" + objInit.Rc + "&errorDesc=" + objInit.ErrorDesc);
    }

    //RIPRENDO IL PAYMENTID PER EVENTUALI SALVATAGGI ES. SUL DB
    string paymentId = objInit.PaymentID;

    //REDIRECT VERSO LA PAGINA BUY NOW DI BNLPositivity
    Uri redirectURL = objInit.RedirectURL;
    Response.Redirect(redirectURL.ToString());
  }
```

TABELLA 4 PAGAMENTO DIRETTO

```
protected void btnPagamentiDirettiAuth_Click(object sender, EventArgs e)
  {

    string result = "";
    string tranID = "";
    string brand = "";

    //DICHIARARE L'URL DELLA PAGINA BUY NOW DI BNLPositivity
    string serverURL = "https://test.bnlpositivity.it/BNL_CG_SERVICES/services";
```

```
        //DICHIARARE UNA PAGINA DI REDIRECT IN CASO DI ERRORE
        string errorURL = "http://localhost:2637/errorPayment.aspx";

        //DICHIARARE I PARAMETRI PER L'INIZIALIZZAZIONE DELLA RICHIESTA DI PAGAMENTO
        //TUTTI I PARAMETRI RIPORTATI SONO SOLO A TITOLO DI ESEMPIO
        IgfsCgAuth objAuth = new IgfsCgAuth();
        objAuth.ServerURL = new Uri(serverURL);
        objAuth.Timeout = 15000;
        objAuth.Tid = "BNLP_SELE";
        objAuth.KSig = " xxxxiSb08fs8Bxxx9Yhq3Ub99E8=";
        objAuth.ShopID = "00000003";
        objAuth.ShopUserRef = "email@clienteemail.it";
        objAuth.TrType = it.netsw.apps.igfs.cg.coms.api.tran.TrType.AUTH;
        objAuth.CurrencyCode = it.netsw.apps.igfs.cg.coms.api.CurrencyCode.EUR;
        objAuth.Amount = Convert.ToInt64(this.txtIMPORTO.Text) * 100;
        objAuth.Pan = this.txtPAN.Text;
        objAuth.ExpireMonth = Convert.ToInt16(this.txtSCADENZA_MESE.Text);
        objAuth.ExpireYear = Convert.ToInt16(this.txtSCADENZA_ANNO.Text);
        objAuth.Cvv2 = this.txtCVV.Text;


        //ESECUZIONE RICHIESTA DI TRANSAZIONE
        if (!objAuth.execute())
        {
          //REDIRECT IN CASO DI ERRORE
          Response.Redirect(errorURL + "?rc=" + objAuth.Rc + "&errorDesc=" + objAuth.ErrorDesc);
        }

        //RECUPERARE IL RISULTATO DELLA TRANSAZIONE
        if ((objAuth.Rc != null))
          result = objAuth.Rc.ToString();
        if ((objAuth.TranID != null))
          tranID = objAuth.TranID.ToString();
        if ((objAuth.Brand != null))
          brand = objAuth.Brand.ToString();

      }
```

@ PHP

TABELLA 5 SCRIPT PHP PAGAMENTO ONLINE

```
        require('IGFS_CG_API/init/IgfsCgInit.php');

        require('IGFS_CG_API/init/IgfsCgAuth.php');



          //Importo del pagamento

      //"amount" è un parametro POST rappresentante l'importo del pagamento, in centesimi
```

```
$amount = $_POST['amount'];

$amount = 100*$amount;


//Inizializzo i parametri

$init = new IgfsCgInit();

$init->serverURL = "https://test.bnlpositivity.it/BNL_CG_SERVICES/services";

$init->errorURL = "http://localhost/testigfs/error.php";

$init->notifyURL = "http://localhost/testigfs/notify.php";

$init->tid="BNL_SELECT";

$init->kSig="xxxxiSb08fs8Bxxx9Yhq3Ub99E8=";

$init->shopID = "00000001";

$init->shopUserRef="email@clienteemail.it";

$init->trType="AUTH";

$init->currencyCode = "EUR";

$init->landID = "IT";

$init->timeout=150000;

$init->amount=$amount;


//Eseguo la richiesta

if(!$init->execute()){

        //In caso di errore, redirect su pagina di errore

        header("location:        error.php?rc=".urlencode($init->rc)."&errorDesc=".urlencode($init->errorDesc));

        return;

}


//Salvo in paymentId il valore dell'id del pagamento (es: da riportare nel DB)

$paymentId = $init->paymentID;


//Senza errori, redirect

header("location:        ".$init->redirectURL);

return;
```

@ JSP

Integrazione del jar

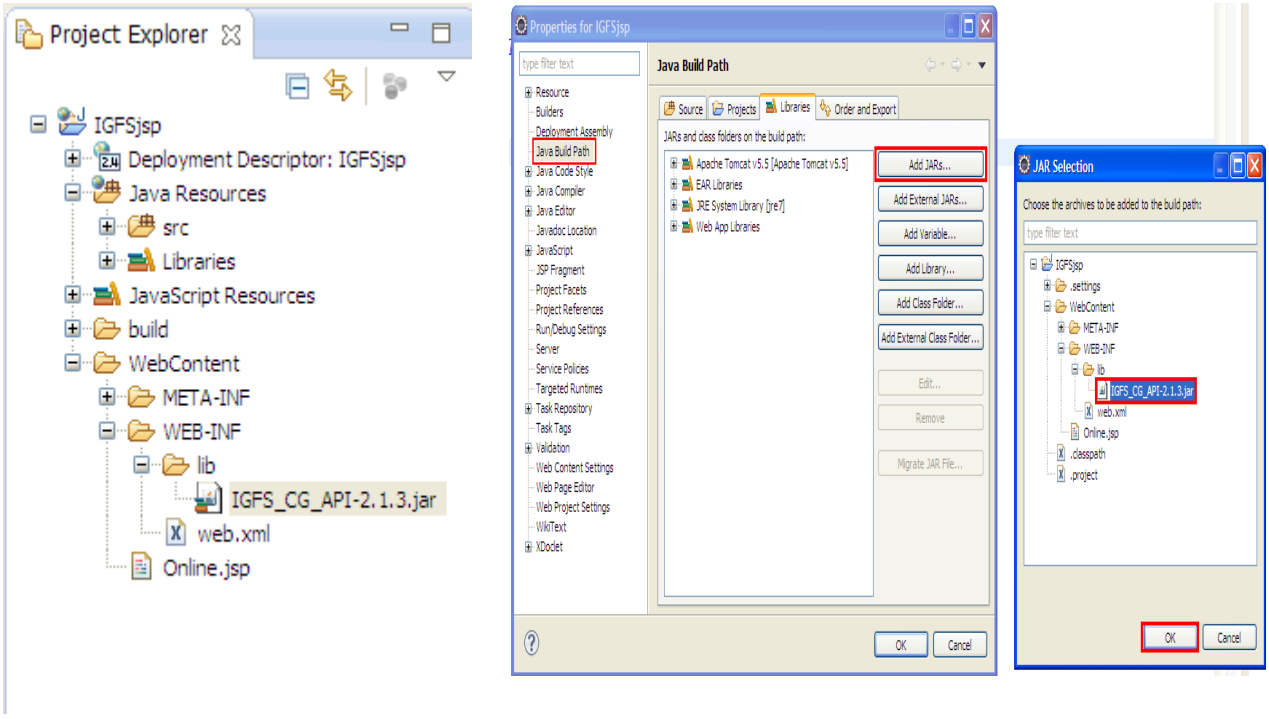


TABELLA 6 PAGAMENTO ONLINE

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

   pageEncoding="ISO-8859-1"%>

<%@page import="it.netsw.apps.igfs.cg.coms.api.BaseIgfsCg.LangID"%>

<%@page import="it.netsw.apps.igfs.cg.coms.api.BaseIgfsCg.CurrencyCode"%>

<%@ page import="it.netsw.apps.igfs.cg.coms.api.init.IgfsCgInit"%>

<%@ page import="it.netsw.apps.igfs.cg.coms.api.init.IgfsCgInit.*"%>

<%@page import="java.net.URL"%>

   <%

      //Definisco i valori dei parametri

      String serverURL = "https://test.bnlpositivity.it/BNL_CG_SERVICES/services";

      String errorURL = "http://localhost:8080/IGFSjsp/errorPayment.jsp";

      String notifyURL = "http://localhost:8080/IGFSjsp/notifyPayment.jsp";

      String tId = "BNL_SELECT";
```

```
String KSig = "xxxxiSb08fs8Bxxx9Yhq3Ub99E8=";

String shopId = "00000001";

String shopRef = "email@clienteemail.it";

TrType trType = TrType.AUTH;

CurrencyCode curr = CurrencyCode.EUR;

LangID langId = LangID.IT;

//Importo del pagamento

long amount = 100;


//Imposto i parametri sul'oggetto IgfsCgInit

IgfsCgInit init = new IgfsCgInit();

init.setServerURL(new URL(serverURL));

init.setErrorURL(new URL(errorURL));

init.setNotifyURL(new URL(notifyURL));

init.setTimeout(150000);

init.setTid(tId);

init.setKSig(KSig);

init.setShopID(shopId);

init.setShopUserRef(shopRef);

init.setTrType(trType);

init.setCurrencyCode(curr);

init.setLangID(langId);

init.setAmount(amount);



//Eseguo la richiesta /*

if (!init.execute())

{

URL redirectURL = new URL(errorURL + "?rc=" + init.getRc() + "&errorDesc=" + init.getErrorDesc());

   response.sendRedirect(redirectURL.toString());

   return;

}
```

```
    //Salvo l'id del pagamento, da riportare ad esempio su DB

    String paymentId = init.getPaymentID();


    //Redirezione

    URL redirectUrl = init.getRedirectURL();

    response.sendRedirect(redirectUrl.toString());



  %>
```

TABELLA 7 PAGAMENTO DIRETTO

```
<%@page import="it.netsw.apps.igfs.cg.coms.api.tran.IgfsCgAuth.*"%>

<%@page import="it.netsw.apps.igfs.cg.coms.api.tran.IgfsCgAuth"%>

<%@page import="it.netsw.apps.igfs.cg.coms.api.BaseIgfsCg.CurrencyCode"%>

<%@page import="java.net.URL"%>

        <%

                //Parametri pagamento

                String serverURL = "https://test.bnlpositivity.it/BNL_CG_SERVICES/services";

                String errorURL = "http://localhost:8080/IGFSjsp/errorPayment.jsp";

                String tId = "BNL_SELECT";

                String KSig = "xxxxiSb08fs8Bxxx9Yhq3Ub99E8=";

                String shopId = "00000003";

                String shopRef = "email@clienteemail.it";

                TrType trType = TrType.AUTH;

                CurrencyCode curr = CurrencyCode.EUR;


                //Importo del pagamento

                long amount = Long.parseLong(request.getParameter("amount"));


                //Dati carta
```

```
//Numero carta
String pan = request.getParameter("pan");
//Mese scadenza
int expireMonth = Integer.parseInt(request
                        .getParameter("expireMonth"));
//Anno scadenza
int expireYear = Integer.parseInt(request
                        .getParameter("expireYear"));
//CVV
String cvv = request.getParameter("cvv");

//Imposto i parametri sul'oggetto IgfsCgAuth
IgfsCgAuth auth = new IgfsCgAuth();
auth.setServerURL(new URL(serverURL));
auth.setTimeout(150000);
auth.setTid(tId);
auth.setKSig(KSig);
auth.setShopID(shopId);
auth.setShopUserRef(shopRef);
auth.setTrType(trType);
auth.setCurrencyCode(curr);
auth.setAmount(amount);
auth.setPan(pan);
auth.setExpireMonth(expireMonth);
auth.setExpireYear(expireYear);
auth.setCvv2(cvv);

//Eseguo la richiesta
if (!auth.execute()) {
        URL redirectURL = new URL(errorURL + "?rc=" + auth.getRc()
                        + "&errorDesc=" + auth.getErrorDesc());
        response.sendRedirect(redirectURL.toString());
```

```
                return;
        }

        //Risultato
        String res = "";
        //Id transazione
        String tranID = "";
        //Brand
        String brand = "";

        if (auth.getRc() != null) {
                res = auth.getRc();
        }
        if (auth.getTranID() != null) {
                tranID = auth.getTranID().toString();
        }
        if (auth.getBrand() != null) {
                brand = auth.getBrand();
        }
    //Restituisco risposta
        response.getWriter().write("res: " + res + ", trandID: " + tranID
                        + ", brand: "+ brand);
        response.getWriter().flush();
%>
```

E. IMPLEMENTAZIONE WEB SERVICES

Al fine di facilitare la comprensione dei metodi relativi all' utilizzo dei web services riportiamo script di esempio.

@ JAVA

TABELLA 8 WS INIT()

```
<%
// ==================================================================
// =          importazione classi di riferimento                    =
// ==================================================================
%>
<%@page import="import java.io.File" %>
<%@page import="import java.net.MalformedURLException" %>
<%@page import="import java.net.URL" %>
<%@page import="import javax.xml.namespace.QName" %>
<%

// ==================================================================
// = impostazione parametri per l'inizializzazione richiesta di     =
// = pagamento.                                                     =
// = NB: I parametri riportati sono solo a titolo di esempio        =
// ==================================================================
URL wsdlURL = new
URL("https://IPGATEWAY/IGFS_CG_SERVICES/services/PaymentInitGatewayPort?wsd
l");
QName SERVICE_NAME = new
QName("http://services.api.web.cg.igfs.apps.netsw.it/",
"PaymentInitGateway");

PaymentInitGateway_Service ss = new PaymentInitGateway_Service(wsdlURL,
SERVICE_NAME);
PaymentInitGateway port = ss.getPaymentInitGatewayPort();

String tid           = "123456";
String kSig          = "ondkmctaf9/MI3I5AZ4LskbmRiw=";
String shopID        = "5687010820272485455";
String email         = "user@customer.it";
String trType        = "AUTH";
long amount          = 100;
String curCode       = "EUR";
String langID        = "IT";
String errorURL      = "https://merchant/error.jsp";
String notifyURL     = "https://merchant/notify.jsp";

String signature = getSignature(kSig, // KSIGN
```

```
  tid, // TID
  shopID, // SHOPID
  shopUserRef, // SHOPUSERREF
  trType,// TRTYPE
  amount, // AMOUNT
  currencyCode, // CURRENCYCODE
  langID, // LANGID
  notifyURL, // NOTIFYURL
  errorURL); // ERRORURL

Init _init_parameters = new Init();
PaymentInitRequest _init_parametersRequest = new  PaymentInitRequest();
_init_parametersRequest.setTid(tid);
_init_parametersRequest.setSignature(signature);
_init_parametersRequest.setShopID(shopID);
_init_parametersRequest.setShopUserRef(shopUserRef);
_init_parametersRequest.setTrType(trType);
_init_parametersRequest.setAmount(amount);
_init_parametersRequest.setCurrencyCode(currencyCode);
_init_parametersRequest.setLangID(langID);
_init_parametersRequest.setNotifyURL(notifyURL);
_init_parametersRequest.setErrorURL(errorURL);
_init_parameters.setRequest(_init_parametersRequest);

// ==================================================================
// =              esecuzione richiesta di inizializzazione          =
// ==================================================================
InitResponse _init_return = port.init(_init_parameters);

if (_init_return.getResponse().isError()) {
  // ================================================================
  // = redirect del client su pagina di errore definita dall'esercente  =
  // ================================================================
  response.sendRedirect(errorURL + "?rc=" +
_init_return.getResponse().getRc() + "&errorDesc=" +
_init_return.getResponse().getErrorDesc());
  return;
}

String paymentID = _init_return.getResponse().getPaymentID();
// NOTA: Salvo il paymentID relativo alla richiesta (es. sul DB)...

// ==================================================================
// =              redirect del client verso URL IGFS BuyNow         =
// ==================================================================
String redirectURL = _init_return.getResponse().getRedirectURL();
response.sendRedirect(redirectURL.toString());
%>
```

TABELLA 9 WS VERIFY()

```
<%
// ==================================================================
// =          importazione classi di riferimento                    =
// ==================================================================
%>
<%@page import="import java.io.File" %>
<%@page import="import java.net.MalformedURLException" %>
<%@page import="import java.net.URL" %>
<%@page import="import javax.xml.namespace.QName" %>
<%

// ==================================================================
// = impostazione parametri per l'inizializzazione richiesta di      =
// = pagamento.                                                      =
// = NB: I parametri riportati sono solo a titolo di esempio         =
// ==================================================================
URL wsdlURL = new URL("https://IPGATEWAY/IGFS_CG_SERVICES
/services/PaymentInitGatewayPort?wsdl");
QName SERVICE_NAME = new
QName("http://services.api.web.cg.igfs.apps.netsw.it/",
"PaymentInitGateway");

PaymentInitGateway_Service ss = new PaymentInitGateway_Service(wsdlURL,
SERVICE_NAME);
PaymentInitGateway port = ss.getPaymentInitGatewayPort();

String tid           = "123456";
String kSig          = "ondkmctaf9/MI3I5AZ4LskbmRiw=";
String shopID        = "5687010820272485455";
String paymentID     = // NOTA: Leggo il paymentID rilasciato in fase di
init (es. dal DB)...
String errorURL      = "https://merchant/error.jsp";
String esitoURL     = "https://merchant/esito.jsp";

String signature = getSignature(kSig, // KSIGN
  tid, // TID
  shopID, // SHOPID
  paymentID); // PAYMENTID

Verify _verify_parameters = new Verify();
PaymentVerifyRequest _verify_parametersRequest = new
PaymentVerifyRequest();
_verify_parametersRequest.setTid(tid);
_verify_parametersRequest.setSignature(signature);
_verify_parametersRequest.setShopID(shopID);
_verify_parametersRequest.setPaymentID(paymentID);
_verify_parameters.setRequest(_verify_parametersRequest);

// ==================================================================
// =              esecuzione richiesta di verifica                   =
// ==================================================================
```


```
VerifyResponse _verify_return = port.verify(_verify_parameters);

if (_verify_return.getResponse().isError()) {
  // ==================================================================
  // = redirect del client su pagina di errore definita dall'esercente  =
  // ==================================================================
  response.sendRedirect(errorURL + "?rc=" +
_verify_return.getResponse().getRc() + "&errorDesc=" +
_verify_return.getResponse().getErrorDesc());
  return;
}

// ==================================================================
// =          redirect del client verso URL Esito Pagamento Merchant  =
// ==================================================================
StringBuffer resultUrl = new StringBuffer();
resultUrl.append(esitoURL);
resultUrl.append("?rc=" + verify.getRc());
resultUrl.append("&tranID=" + verify.getTranID());
resultUrl.append("&enrStatus=" + verify.getEnrStatus());
resultUrl.append("&authStatus=" + verify.getAuthStatus());
response.sendRedirect(resultUrl.toString());
%>
```

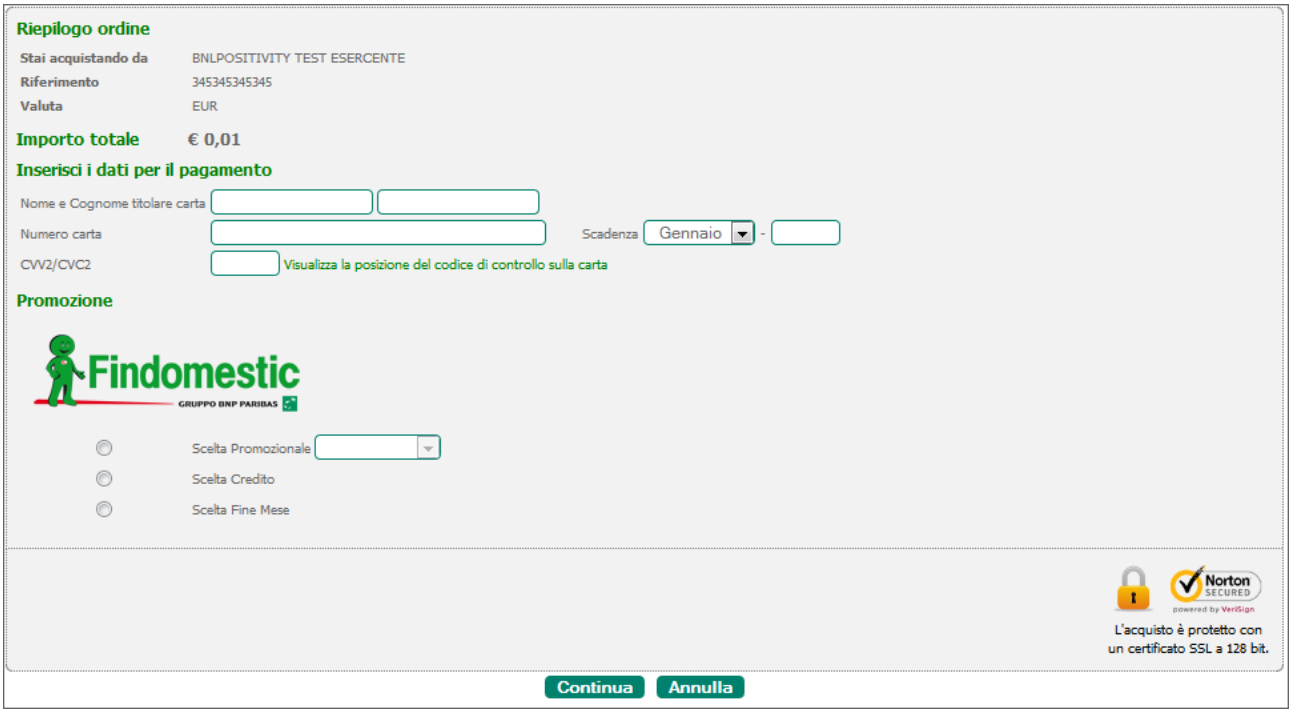
F. Integrazione con Findomestic

L'integrazione del circuito Findomestic prevede la valorizzazione dei parametri riportati nei paragrafi:

@ **8.A.1** ClasseIgfsCgInit, pagamenti BuyNow (autorizzazione o movimentazione)
@ **8.A.2** ClasseIgfsCgAuth, pagamenti diretti (autorizzazione o movimentazione)
  @ **8.A.2** ClasseIgfsCgConfirm, pagamenti diretti (conferma movimento autorizzato)
  @ **8.A.2** ClasseIgfsCgVoidAuth, pagamenti diretti (storno di un autorizzazione)
  @ **8.A.2** ClasseIgfsCgCredit, pagamenti diretti (credito di una movimentazione)
@ **8.B.1** Metodo init(), pagamenti BuyNow (autorizzazione o movimentazione)
@ **8.B.2** Metodo Auth(), pagamenti diretti (autorizzazione o movimentazione)
  @ **8.B.2** Metodo confirm(), pagamenti diretti (conferma movimento autorizzato)
  @ **8.B.2** Metodo voidAuth(), pagamenti diretti (storno di un autorizzazione)
  @ **8.B.2** Metodo credit(), pagamenti diretti (credito di una movimentazione)

Durante la richiesta di autorizzazione il cliente verrà indirizzato sulla pagina di pagamento di **BNLPOSitivity,** per i pagamenti descritti nei paragrafi: 8.A.1, 8.B.1, mentre per i pagamenti diretti, paragrafi 8.A.2 e 8.B.2 l'esercente deve predisporre sul proprio web store i campi necessari per i dati della carta e la scelta della promozione.

L'immagine sotto riporta il layout della pagina, da copiare nelle integrazioni con pagamenti diretti.



Il circuito Findomestic oltre al risultato della transazioni prevede eventuali spese aggiuntive, riportate in risposta nel campo "**additional_fee**".

G. Integrazione con MasterPass, gestione indirizzi

Masterpass è il nuovo servizio di pagamento offerto da **Mastercard**, consente il pagamento su piattaforme e-commerce utilizzando un wallet. L'integrazione dall' sito e-commerce può essere fatta tramite API e web services con soluzione on-line, quindi in aggiunta ai parametri della classe **igfscgInit** e **igfscgVerify** a discrezione dell' esercente possono essere gestiti i parametri riportati nella tabella sotto, per postare o recuperare gli indirizzi di fatturazione o spedizione, in particolare:

- **dalla igfscgInit**, in fase di richiesta autorizzativa, se l'esercente valorizza i parametri della classe **level3Info**, posta sul sito di Masterpass gli indirizzi di spedizione e fatturazione registrati sulla base dati dell' e-commerce.
- **dalla igfscgVerify**, in fase di verifica della transazione, inizializzata ed eseguita nello step precedente, richiamando i parametri della classe **level3Info,** è possibile recuperare l'indirizzo di spedizione e fatturazione che il titolare del wallet ha selezionato in fase di conferma del pagamento, direttamente sul sito di masterpass

Tipo	Descrizione	Formattazione
invoiceNumber	Numero fattura	String[32]
senderPostalCode	Codice postale mittente	String[8]
senderCountryCode	Codice nazionale mittente nel formato ISO 3122 A3	String[3]
destinationName	Rif. spedizione: Destinatario	String[32]
destinationStreet	Rif. spedizione: Dati Indirizzo	String[100]
destinationStreet2	Rif. spedizione: Dati Indirizzo	String[100]
destinationCity	Rif. spedizione: Città	String[40]
destinationState	Rif. spedizione: Stato (se presente)	String[32]
destinationPostalCode	Rif. spedizione: Codice Postale	String[8]
destinationCountryCode	Rif. spedizione: Codice nazionale nel formato ISO 3122 A3	String[3]
billingName	Rif. fatturazione: Destinatario	String[32]
billingStreet	Rif. fatturazione : Dati Indirizzo	String[100]
billingStreet2	Rif. fatturazione : Dati Indirizzo	String[100]
billingCity	Rif. fatturazione: Città	String[40]
billingState	Rif. fatturazione: stato (se presente)	String[32]
billingPostalCode	Rif. fatturazione: codice postale	String[8]
billingCountryCode	Rif. fatturazione: Codice nazionale nel formato ISO 3122 A3	String[3]
freightAmount	Importo totale senza spedizione	String[12]
taxAmount	Tasse	String[12]
vat	Codice fiscale / Partita I.V.A.	String[64]
note	Note	String[255]
product	Lista articoli presenti nell'ordine	Livel3InfoProduct

H. Integrazione con Masterpass, gestione del carrello

L'esercente ha la possibilità di valorizzare sulla pagina di Masterpass il carrello dei prodotti che l'utente ha deciso di acquistare.
In fase di inizializzazione del pagamento, quindi richiamando la classe **igfscginit**, l' esercente deve valorizzare i parametri della classe **level3info.product,** quindi aggiungendo oggetti valorizzati con i parametri riportati sotto.

Tipo	Descrizione	Formattazione
productCode	Codice prodotto	String[32]
productDescription	Descrizione prodotto	String[32]
items	Numero articoli	String[8]
amount	Importo in virgola mobile (es. 100=1,00 EUR)	String[12]
imgURL	Indirizzo dell'immagine del prodotto	String[256]

I. Integrazione con MyBank

BNLPOSitivity, fornisce la possibilità di accettare pagamenti con MyBank, il nuovo servizio di pagamento elettronico tramite bonifico, effettuato direttamente dalla propria home banking. il servizio è disponibile solo per integrazioni e-commerce on-line e non prevede parametri aggiuntivi al normale flusso di pagamento e-commerce con carta di credito.

NOTA BENE: "L'unica accortezza che bisogna prendere in considerazione in sede di implementazione con il circuito MyBank è che il campo "ShopID" deve essere alfanumerico di 35"

nei parametri di ritorno avrete come tutti gli altri circuiti alcuni campi in output.

Il campo Brand a differenza delle carte di credito non sarà valorizzato. Il circuito di pagamento sarà reperibile nel campo PayInstr:

- PayInstr = MYBK

Sul Back Office che BNLPOSitivity offre agli esercenti, viene visualizzato l'esito della transazione e la banca di riferimento.

Per l'attivazione di MyBank sul vostro e-commerce, contattate il nostro servizio di assistenza o il commerciale di riferimento.

J. Integrazione con PayPal

la gestione del pagamento eseguito con PayPal segue le logiche di Masterpass, quindi seguire il paragrafo "G" e "H" sia per la gestione degli indirizzi che dei prodotti del carrello.

APPENDICE A - CALCOLO SIGNATURE

L' HMACSHA256 è un tipo di algoritmo con chiave costruito dalla funzione di hash SHA-256 e utilizzato come codice HMAC.

A titolo di esempio, riportiamo il calcolo dell' algoritmo HMACCSHA256 sul campo "signature" attraverso la tecnologia java.

```
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;

….

// ================================================================
//   calcolo della signature attraverso l'algoritmo SHA256
//
//   key    = chiave segreta
//   fields = parametri del messaggio
// ================================================================
public String getSHA256Signature(String key, Object... fields) throws
Exception {
      StringBuilder sb = new StringBuilder();
      for (Object field : fields) {
            if (field != null) {
                  sb.append(field.toString());
            }
      }

      byte data[] = sb.toString().getBytes();

      String alg = "HmacSHA256";
      SecretKeySpec sk1 = new SecretKeySpec(key.getBytes(), alg);
      Mac mac = Mac.getInstance(alg);
      mac.init(sk1);
      byte sig[] = mac.doFinal(data);
      return new String(Base64.encode(sig));
}
```

e attraverso la tecnologia .Net

```
 using System.Security.Cryptography;
...

        static String getSignature(String key, params Object[] fields)
        {
            StringBuilder sb = new StringBuilder();
            for (int i = 0; i < fields.Length; i++)
            {
                Object field = fields[i];
                if (field != null)
                {
                    sb.Append(field.ToString());
                }
            }
            byte[] keyByte = Encoding.UTF8.GetBytes(key);
            HMACSHA256 hmacsha256 = new HMACSHA256(keyByte);
            byte[] data = Encoding.UTF8.GetBytes(sb.ToString());
            byte[] mac = hmacsha256.ComputeHash(data);
            return System.Convert.ToBase64String(mac);
        }
```

APPENDICE B - CODICI DI RITORNO

CODICE	DESCRIZIONE
IGFS_000	TRANSAZIONE OK
IGFS_001	DESTINATARIO SCONOSCIUTO
IGFS_00155	ID BATCH NON VALIDO
IGFS_00156	ID BATCH NON UNIVOCO
IGFS_00157	STRUMENTO PAGAMENTO NON VALIDO
IGFS_00158	NUMERO CARTA NON NUMERICO
IGFS_00159	NUMERO CARTA NON PRESENTE
IGFS_002	CARTA SCADUTA
IGFS_00260	L'IMPORTO DEL CREDITO SUPERA L'IMPORTO DEL MOVIMENTO
IGFS_00261	L'IMPORTO DEL MOVIMENTO SUPERA L'IMPORTO DELL' AUTORIZZAZIONE
IGFS_003	CARTA ERRATA
IGFS_004	CARTA IN BLACK LIST
IGFS_00452	CODICE TERMINALE NON PRESENTE
IGFS_00456	CODICE TERMINALE ERRATO
IGFS_005	ERRORE DI FORMATO
IGFS_006	ERRORE FILE SYSTEM
IGFS_007	ERRORE DI COMUNICAZIONE
IGFS_00701	BATCH ID NON PROCESSATO
IGFS_00704	BATCH ID NON NUMERICO
IGFS_00705	BATCH ID NON PRESENTE
IGFS_008	AUTORIZZAZIONE NEGATA
IGFS_009	RITIRARE CARTA
IGFS_00950	DIRECTORY BATCH UPLOAD NON PRESENTE
IGFS_00951	DIRECTORY BATCH DOWNLOAD NON PRESENTE
IGFS_00952	NOME DIRECTORY ARCHIVIAZIONE BATCH NON PRESENTE
IGFS_010	MERCHANT NON ABILITATO
IGFS_01000	TRANSAZIONE NEGATA DAL SISTEMA ANTIFRODE
IGFS_011	CONTATTARE ACQUIRER
IGFS_014	MERCHANT NON CONVENZIONATO
IGFS_015	CARTA NON GESTITA
IGFS_016	CARTA IN RANGE NEGATIVO O STRANIERA
IGFS_018	CARTA INESISTENTE
IGFS_020	CARTA INVALIDA
IGFS_021	CODICE MERCHANT ERRATO
IGFS_029	DATA SCADENZA ERRATA
IGFS_030	FONDI INSUFFICIENTI
IGFS_032	IMPORTO NON VALIDO

CODICE	DESCRIZIONE
IGFS_033	TRANSAZIONE ORIGINALE NON TROVATA
IGFS_083	ERRORE CIFRATURA TRANSAZIONE
IGFS_085	CODICE DIVISA ERRATO
IGFS_086	MALFUNZIONAMENTO SISTEMA
IGFS_087	ACQUIRER NON RAGGIUNGIBILE
IGFS_088	MANCATA RISPOSTA DA ACQUIRER
IGFS_091	MALFUNZIONAMENTO SISTEMA ACQUIRER
IGFS_092	TRANSAZIONE SCONOSCIUTA
IGFS_093	CONFERMA GIA' PRESENTE
IGFS_095	STORNO PER NOTIFICA INESISTENTE
IGFS_096	STORNO PER AUTORIZZAZIONE INESISTENTE
IGFS_097	CONFERMA PER AUTORIZZAZIONE INESISTENTE
IGFS_098	IMPORTO SUPERIORE AD IMPORTO AUTORIZZATO
IGFS_10000	CARATTERI NON VALIDI
IGFS_101	MAC ERRATO
IGFS_102	SOSPETTA FRODE
IGFS_104	CARTA SOGGETTA A RESTRIZIONI
IGFS_107	CONTATTARE ISSUER
IGFS_108	CONTATTARE ISSUER: CASO SPECIALE
IGFS_112	INSERIRE PIN
IGFS_115	FUNZIONE NON SUPPORTATA SU CARTA
IGFS_117	PIN ERRATO
IGFS_118	CONTO NON TROVATO O NON ABILITATO
IGFS_119	OPERAZIONE NON PERMESSA AL TITOLARE
IGFS_121	SUPERATO LIMITE IMPORTO
IGFS_122	ERRORE SICUREZZA
IGFS_123	SUPERATO LIMITE FREQUENZA
IGFS_125	CARTA NON ATTIVA
IGFS_129	SOSPETTA FRODE SU CARTA
IGFS_160	CARTA PERSA
IGFS_164	DATA ANTEC. A BLOCCO CARTA
IGFS_180	DATI ERRATI
IGFS_181	DATI SENSIBILI ERRATI
IGFS_1921	3DS: IMPOSSIBILE AUTENTICARE CARTA (PARES=U)
IGFS_1922	3DS: AUTENTICAZIONE NON AVVENUTA (PARES=N)
IGFS_1923	3DS: IMPOSSIBILE VERIFICARE ISCRIZIONE CARTA (VERES=U)
IGFS_20000	DATI MANCANTI
IGFS_20001	CODICE OPERAZIONE NON VALIDO
IGFS_20007	STATO ORDINE NON VALIDO
IGFS_20010	URL INVIO RISPOSTA NON VALIDO

CODICE	DESCRIZIONE
IGFS_20011	URL INVIO ERRORE NON VALIDO
IGFS_20012	SHOPID NON VALIDO
IGFS_20013	CODICE LINGUA NON VALIDO
IGFS_20014	CAMPO AGGIUNTIVO NON VALIDO
IGFS_20018	CVV2 NON VALIDO
IGFS_20019	SHOPID NON VALIDO
IGFS_20020	CAMPO ADDIZIONALE NON VALIDO
IGFS_20021	CAMPO API VERSION NON VALIDO
IGFS_20022	CAMPO SIGNATURE NON VALIDO
IGFS_20023	CAMPO PAYMENT ID NON VALIDO
IGFS_20024	CODICE AUTORIZZAZIONE MANCANTE
IGFS_20025	CAMPO REFERENCE DATA NON VALIDO
IGFS_20026	SHOP ID DUPLICATO
IGFS_20027	RICHIESTA BATCH NON VALIDA
IGFS_20028	DATI BATCH MANCANTI
IGFS_20029	DATI BATCH NON VALIDI
IGFS_20030	DIRECTORY DATI BATCH NON VALIDA
IGFS_20031	DATI BATCH DUPLICATI
IGFS_20032	NOME BATCH FILE NON VALIDO
IGFS_20033	DATI BATCH NON TROVATI
IGFS_20034	BATCH SHOPID NON VALIDO
IGFS_20035	ID ORDINE NON VALIDO
IGFS_20036	PAN NON VALIDO
IGFS_20037	CVV2 NON VALIDO
IGFS_20038	DATA SCADENZA ERRATA
IGFS_20044	DESCRIZIONE PAGAMENTO NON VALIDA
IGFS_20090	TRANSAZIONE CANCELLATA DALL'UTENTE
IGFS_20100	ERRORE NOTIFICA MERCHANT
IGFS_400	STORNO OK
IGFS_800	TERMINALE NON ABILITATO
IGFS_801	BANCA SELEZIONATA ERRATA
IGFS_802	TENTATIVI PIN ESAURITI
IGFS_803	CODICE TERMINALE ERRATO
IGFS_804	CHIAVE DISALLINEATA
IGFS_805	ERRORE CIFRATURA
IGFS_807	TERMINALE CHIUSO
IGFS_808	TERMINALE NON CHIUSO
IGFS_809	ERRORE SEQUENZA
IGFS_810	TERMINALE NON RICONOSCIUTO
IGFS_811	TERMINALE BLOCCATO

CODICE	DESCRIZIONE
IGFS_812	TERMINALE CHIUSO FORZATAMENTE
IGFS_813	OPERAZIONE NON PERMESSA
IGFS_814	TRANSAZIONE IN CORSO
IGFS_815	CARTA BLOCCATA
IGFS_90000	DATABASE ERROR
IGFS_90005	TIMESTAMP ERRATO
IGFS_902	TRANSAZIONE NON VALIDA
IGFS_903	REINVIARE TRANSAZIONE
IGFS_907	EMITTENTE NON ADERENTE
IGFS_908	DESTINAZIONE NON TROVATA
IGFS_909	ERRORE DI SISTEMA
IGFS_910	SISTEMA ISSUER NON ATTIVO
IGFS_911	TIME OUT
IGFS_912	ISSUER NON RAGGIUNGIBILE
IGFS_913	TRANSAZIONE DUPLICATA
IGFS_990	STRUMENTO PAGAMENTO NON ATTIVO