



Aggiornamento dell'integrazione per essere compatibile con la PSD2

Premessa

Lo scopo di questa documentazione è di condividere con te tutti gli elementi necessari per aggiornare la tua integrazione API di PayPlug al fine di renderla compatibile con la PSD2.

Troverai una serie di cambiamenti, ma come potrai notare non andrai affatto incontro a grosse difficoltà. Tutti gli esempi principali sono in PHP, ma [il nostro Team è a tua disposizione](#) se hai domande o se hai bisogno di aiuto per adattarli in un altro linguaggio.

Per maggiori informazioni, non esitare a consultare la [nostra documentazione online](#).

Indice

Premessa	2
Indice	2
Aggiornamento dell'API PayPlug	3
Integrazione delle modifiche	4
Le informazioni di fatturazione	4
Esempio completo	6
Utilizzo delle carte salvate	7
Il pagamento in 1-click	7
Gestione delle scadenze di un abbonamento o di un pagamento rateizzato	9
Conclusione	10

Aggiornamento dell'API PayPlug

Per iniziare devi ottenere l'ultima versione della libreria PayPlug disponibile su Github e sostituire la versione che stai usando attualmente.

- PHP : <https://github.com/payplug/payplug-php/releases>
- Python : <https://github.com/payplug/payplug-python/releases>

Se non stai usando le librerie PHP o Python di PayPlug, le ultime versioni della nostra API richiedono di dichiarare un numero di versione nell'intestazione HTTP che viene inviata. La versione che devi usare è: 2019-08-06.

Ad esempio, per chiamare l'API prima utilizzavi una chiamata di questo tipo:

```
$ curl -X GET https://api.payplug.com/v1/payments \  
-H "Authorization: Bearer sk_live_43b7e007298f57f732800a52"
```

D'ora in avanti dovrai inviare:

```
$ curl -X GET https://api.payplug.com/v1/payments \  
-H "Authorization: Bearer sk_live_43b7e007298f57f732800a52" \  
-H "PayPlug-Version: 2019-08-06"
```

Con le nostre librerie PHP e Python questa parte è già gestita, quindi se già le usi non devi occupartene.

Integrazione delle modifiche

Nel processo di creazione di un pagamento, alcune chiavi sono state sostituite ed altre aggiunte. Devi quindi necessariamente fare delle modifiche per evitare errori quando l'API viene chiamata.

Le informazioni di fatturazione

Nella versione precedente dell'API le informazioni di fatturazione erano dichiarate sotto la chiave **"customer"**, quest'ultima è stata sostituita dalle chiavi **"billing"** e **"shipping"**.

Nella sezione **"fatturazione"**, le seguenti chiavi sono raccomandate per l'autenticazione senza attrito:

first_name	Il nome dell'acquirente (100 caratteri max).
last_name	Il cognome dell'acquirente (100 caratteri max).
email	L'indirizzo e-mail dell'acquirente (255 caratteri max).
address1	L'indirizzo postale dell'acquirente (255 caratteri max).
postcode	Il codice postale dell'acquirente (16 caratteri max).
city	La città dell'acquirente (100 caratteri max).
country	Il Paese dell'acquirente, basato sulla norma ISO 3166 .
mobile_phone_number	Il numero di cellulare del destinatario o dell'acquirente (format international standard E.164). Ad esempio, un numero di cellulare italiano sarà sotto la forma +393331289456.
landline_phone_number	Il numero di telefono fisso del destinatario o dell'acquirente (format international standard E.164). Ad esempio, un numero fisso italiano sarà sotto la forma +390212894567.
language	La lingua del destinatario o dell'acquirente in 2 lettere nel formato ISO 639-1 code . Lingue supportate: it, en, fr.

Nella sezione **"shipping"**, le seguenti chiavi sono raccomandate per supportare l'autenticazione senza attrito:

first_name	Il nome dell'acquirente (100 caratteri max).
last_name	Il cognome dell'acquirente (100 caratteri max).
email	L'indirizzo e-mail dell'acquirente (255 caratteri max).
address1	L'indirizzo postale dell'acquirente (255 caratteri max).

postcode	Il codice postale dell'acquirente (16 caratteri max).														
city	La città dell'acquirente (100 caratteri max).														
country	Il Paese dell'acquirente, basato sulla norma ISO 3166 .														
mobile_phone_number	Il numero di cellulare del destinatario o dell'acquirente (format international standard E.164). Ad esempio, un numero di cellulare italiano sarà sotto la forma +393331289456.														
landline_phone_number	Il numero di telefono fisso del destinatario o dell'acquirente (format international standard E.164). Ad esempio, un numero fisso italiano sarà sotto la forma +390212894567.														
language	La lingua del destinatario o dell'acquirente in 2 lettere nel formato ISO 639-1 code . Lingue supportate: it, en, fr.														
delivery_type	<p>La modalità di consegna. Per questa chiave puoi scegliere uno dei seguenti valori:</p> <table border="1"> <tr> <td>BILLING</td> <td>La spedizione viene effettuata verso l'indirizzo di fatturazione dell'acquirente.</td> </tr> <tr> <td>VERIFIED</td> <td>La spedizione viene effettuata verso un indirizzo verificato dal venditore.</td> </tr> <tr> <td>NEW</td> <td>La spedizione viene effettuata ad un indirizzo diverso da quello di fatturazione.</td> </tr> <tr> <td>SHIP_TO_STORE</td> <td>Il ritiro avviene in un punto di consegna (l'indirizzo deve essere inserito nella variabile indirizzo della chiave shipping).</td> </tr> <tr> <td>DIGITAL_GOODS</td> <td>Tutti i beni digitali (senza spedizione).</td> </tr> <tr> <td>TRAVEL_OR_EVENT</td> <td>Vendita di biglietti per eventi o viaggi (nessuna spedizione).</td> </tr> <tr> <td>OTHER</td> <td>Altro (tutti i servizi online senza spedizione).</td> </tr> </table>	BILLING	La spedizione viene effettuata verso l'indirizzo di fatturazione dell'acquirente.	VERIFIED	La spedizione viene effettuata verso un indirizzo verificato dal venditore.	NEW	La spedizione viene effettuata ad un indirizzo diverso da quello di fatturazione.	SHIP_TO_STORE	Il ritiro avviene in un punto di consegna (l'indirizzo deve essere inserito nella variabile indirizzo della chiave shipping).	DIGITAL_GOODS	Tutti i beni digitali (senza spedizione).	TRAVEL_OR_EVENT	Vendita di biglietti per eventi o viaggi (nessuna spedizione).	OTHER	Altro (tutti i servizi online senza spedizione).
BILLING	La spedizione viene effettuata verso l'indirizzo di fatturazione dell'acquirente.														
VERIFIED	La spedizione viene effettuata verso un indirizzo verificato dal venditore.														
NEW	La spedizione viene effettuata ad un indirizzo diverso da quello di fatturazione.														
SHIP_TO_STORE	Il ritiro avviene in un punto di consegna (l'indirizzo deve essere inserito nella variabile indirizzo della chiave shipping).														
DIGITAL_GOODS	Tutti i beni digitali (senza spedizione).														
TRAVEL_OR_EVENT	Vendita di biglietti per eventi o viaggi (nessuna spedizione).														
OTHER	Altro (tutti i servizi online senza spedizione).														

Le chiavi di cui sopra non sono obbligatorie ma sono fortemente raccomandate per favorire l'autenticazione senza attrito (senza 3-D Secure). Maggiori dettagli ed informazioni sulle chiavi disponibili sono elencati nella [documentazione API](#).

Esempio completo

Più concretamente, ecco un esempio che permette di creare un pagamento tenendo conto di queste modifiche:

```
<?php
$payment = \Payplug\Payment::create(array(
    'amount'           => 3300,
    'currency'         => 'EUR',
    'save_card'        => false,
    'billing'          => array(
        'title'         => 'miss',
        'first_name'    => 'Eva',
        'last_name'     => 'Kant',
        'mobile_phone_number' => '+390212893467',
        'email'         => 'eva.kant@example.net',
        'address1'      => 'Viale Monte Nero 8',
        'postcode'      => '20135',
        'city'          => 'Milano',
        'country'       => 'IT',
        'language'      => 'it'
    ),
    'shipping'         => array(
        'title'         => 'miss',
        'first_name'    => 'Eva',
        'last_name'     => 'Kant',
        'mobile_phone_number' => '+390212893467',
        'email'         => 'eva.kant@example.net',
        'address1'      => 'Viale Monte Nero 8',
        'postcode'      => '20135',
        'city'          => 'Milano',
        'country'       => 'IT',
        'language'      => 'it',
        'delivery_type' => 'BILLING'
    ),
    'hosted_payment'  => array(
        'return_url'    => 'https://example.net/success?id=42',
        'cancel_url'    => 'https://example.net/cancel?id=42'
    ),
    'notification_url' => 'https://example.net/notifications?id=42',
    'metadata'         => array(
        'customer_id' => 42
    )
));
```

```
$payment_url = $payment->hosted_payment->payment_url;  
$payment_id = $payment->id;
```

Utilizzo delle carte salvate

L'insieme delle azioni da eseguire quando si salva una carta non cambia, dovrai solo adattare il tuo codice per integrare le informazioni di fatturazione come indicato sopra.

Tutte le informazioni e gli esempi sul salvataggio di una carta sono disponibili nella [nostra documentazione API](#).

I cambiamenti principali avvengono quando si crea un pagamento che utilizza una carta registrata. Quando si crea questo pagamento è necessario utilizzare la chiave **"initiator"** con uno dei seguenti valori:

PAYER	L'acquirente è presente sul sito al momento dell'acquisto ed il pagamento viene effettuato su sua iniziativa. Questo è il caso del pagamento 1-click, ad esempio.
MERCHANT	L'acquirente non è presente quando viene effettuato il pagamento. Questo è il caso di un pagamento tramite abbonamento o della rata di un pagamento rateizzato.

Questa chiave permette di indicare alla banca dell'acquirente chi è l'autore del pagamento. Questa informazione è importante perché, a seconda di ciò che viene dichiarato, potresti trovarti a gestire il 3-D Secure. Vediamo nel dettaglio i due possibili scenari.

Il pagamento in 1-click

Nel caso di un pagamento 1-click, per la chiave **"initiator"** devi usare la variabile **"PAYER"**, in quanto l'acquirente è presente sul sito del negozio. Nel caso di un pagamento 1-click, la banca emittente della carta può decidere di ricorrere al 3-D Secure, quindi devi essere in grado di gestire il reindirizzamento verso la pagina di autenticazione.

Ecco, di seguito, un esempio per gestire un caso come questo:

```
<?php  
  
// Creazione del pagamento utilizzando un token della carta.  
$payment = \Payplug\Payment::create(array(  
    'amount' => 1000,
```

```
'currency'          => 'EUR',
'payment_method'   => 'card_e7133426b8de947b37161dfba1897dd1',
'initiator'        => 'PAYER',
'billing'           => array(
    'title'          => 'miss',
    'first_name'     => 'Eva',
    'last_name'      => 'Kant',
    'email'          => 'eva.kant@example.net',
    'address1'       => 'Viale Monte Nero 8',
    'postcode'       => '20135',
    'city'           => 'Milano',
    'country'        => 'IT',
    'language'       => 'it'
),
'shipping'          => array(
    'title'          => 'miss',
    'first_name'     => 'Eva',
    'last_name'      => 'Kant',
    'email'          => 'eva.kant@example.net',
    'address1'       => 'Viale Monte Nero 8',
    'postcode'       => '20135',
    'city'           => 'Milano',
    'country'        => 'IT',
    'language'       => 'it',
    'delivery_type' => 'BILLING'
),
'hosted_payment'   => array(
    'return_url'     => 'https://www.example.net/success?id=42'
),
'notification_url' => 'https://example.net/notifications?id=42',
'metadata'         => array(
    'customer_id'   => 42
)
));

// Se il pagamento ha avuto successo, il cliente viene reindirizzato
// verso la pagina di atterraggio. Non è necessario reindirizzare, si può
// anche elaborare il pagamento riuscito in questa tappa.
if ($payment->is_paid) {
    header('Location: '.$payment->hosted_payment.return_url);
}
// In caso di rifiuto durante la creazione del pagamento, viene
// restituito un errore
else if ($payment->failure !== null ) {
```



```
// Visualizzare / Registrare l'errore proveniente da failure.message
}
// In caso di 3-D Secure viene restituito un URL, è quindi necessario
reindirizzare il cliente verso di esso
else if ($payment->hosted_payment.payment_url !== null) {
    header('Location: '.$payment->hosted_payment.payment_url);
}
// Infine, in caso di problemi, viene restituito un errore
else {
    // Visualizzare / Registrare un errore
}
}
```

Gestione delle scadenze di un abbonamento o di un pagamento rateizzato

Hai utilizzato l'API di PayPlug per impostare una gestione dell'abbonamento o del pagamento rateizzato. Il cliente ha già fatto un primo pagamento e poi, basandoti su uno scadenziario, hai impostato uno script per addebitare automaticamente la carta.

Al momento del pagamento, l'acquirente non è dunque presente e non compie alcuna azione per attivare il pagamento. I pagamenti ricorrenti o rateizzati con un importo fisso sono [esenti dall'autenticazione](#) (una volta effettuato il 1° pagamento) in quanto sono considerati come "avviati dal commerciante". Se sei in questo caso, la chiave `"initiator"` deve usare la variabile `"MERCHANT"`.

Ecco, di seguito, un esempio per gestire un caso come questo:

```
<?php
// Creazione del pagamento utilizzando un token della carta.
$payment = \Payplug\Payment::create(array(
    'amount'          => 1000,
    'currency'        => 'EUR',
    'payment_method' => 'card_e7133426b8de947b37161dfba1897dd1',
    'initiator'       => 'MERCHANT',
    'billing'         => array(
        'title'        => 'miss',
        'first_name'   => 'Eva',
        'last_name'    => 'Kant',
        'email'        => 'eva.kant@example.net',
        'address1'     => 'Viale Monte Nero 8',
        'postcode'     => '20135',
        'city'         => 'Milano',
    ),
),
```

```
'country'    => 'IT',
'language'   => 'it'
),
'shipping'   => array(
  'title'     => 'miss',
  'first_name' => 'Eva',
  'last_name'  => 'Kant',
  'email'      => 'eva.kant@example.net',
  'address1'   => 'Viale Monte Nero 8',
  'postcode'   => '20135',
  'city'       => 'Milano',
  'country'    => 'IT',
  'language'   => 'it',
  'delivery_type' => 'BILLING'
),
'hosted_payment' => array(
  'return_url' => 'https://www.example.net/success?id=42'
),
'notification_url' => 'https://example.net/notifications?id=42',
'metadata'         => array(
  'customer_id' => 42
)
));
```

Conclusione

Come abbiamo visto, le modifiche per rendere la tua integrazione compatibile con la PSD2 non sono così complesse. Se incontri delle difficoltà o se hai delle domande, il nostro Team è a tua [disposizione per assisterti](#).

Tieni presente che l'implementazione della PSD2 mira a permettere alle banche di garantire che una persona che paga su un sito di e-commerce sia effettivamente il proprietario della carta utilizzata. Quindi, più informazioni si forniscono al momento della creazione dei pagamenti, più aumentano le possibilità di beneficiare di un pagamento senza attrito.

Scopri di più sulla PSD2:

- [Cosa bisogna sapere sulla PSD 2](#)