



**EBook Gratis**

# APRENDIZAJE stripe-payments

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#stripe-  
payments

# Tabla de contenido

Acerca de.....	1
<b>Capítulo 1: Comenzando con los pagos de banda.....</b>	<b>2</b>
Observaciones.....	2
Examples.....	2
Instalación o configuración.....	2
Modalidad de pago de banda incrustada.....	2
Hola Mundo en Python.....	4
Introducción a la API de Stripe.....	5
<b>Capítulo 2: Documentación de PHP-Stripe Connect para Symfony2.....</b>	<b>7</b>
Parámetros.....	7
Examples.....	7
Symfony2 - Ejemplo de integración de bandas.....	7
<b>Capítulo 3: Raya Agregar múltiples tarjetas al mismo usuario.....</b>	<b>9</b>
Examples.....	9
Crear cliente en banda.....	9
Cómo recuperar clientes y agregar tarjetas en Stripe.....	9
<b>Capítulo 4: Stripe.Net Introducción.....</b>	<b>10</b>
Sintaxis.....	10
Observaciones.....	10
Examples.....	10
Comenzando con Stripe.Net en ASP.Net Core 1.0.....	10
<b>Creditos.....</b>	<b>15</b>

---

## Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [stripe-payments](#)

It is an unofficial and free stripe-payments ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official stripe-payments.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Capítulo 1: Comenzando con los pagos de banda

## Observaciones

Esta sección proporciona una descripción general de qué es la franja de pagos y por qué un desarrollador puede querer usarla.

También debe mencionar cualquier tema importante dentro de la franja de pagos, y vincular a los temas relacionados. Dado que la Documentación para pagos en franjas es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

## Examples

### Instalación o configuración

Instrucciones detalladas para configurar o instalar los pagos de franjas.

### Modalidad de pago de banda incrustada

Registre una cuenta de producción / sandbox en <https://dashboard.stripe.com/register>

Inserte el código de abajo en su página web donde desea tener un botón de pago.

```
<form action="/charge" method="POST">
  <script
    src="https://checkout.stripe.com/checkout.js" class="stripe-button"
    data-key="pk_test_6pRNASCoBOKtIshFeQd4XMUh"
    data-amount="2000"
    data-name="Stripe.com"
    data-description="2 widgets"
    data-image="/img/documentation/checkout/marketplace.png"
    data-locale="auto">
  </script>
</form>
```

Resultado:

# Checkout Guide



## Stripe.com

2 widgets

 Email

 Card number

 MM / YY

 CVC

Remember me

**Pay \$20.00**

## Hola Mundo en Python

Un ejemplo de cómo ejecutar una banda fuera de la caja con wsgi desde un solo archivo.

Al principio, instale la API de la banda python, es decir, con pip:

```
pip install --user stripe
```

Cree `payment.py` que crea un servidor web WSGI en el puerto 8000 de forma inmediata

```
html = """
<html>
<body>
    <p>%(output)s</p>
</body>
</html>
"""

form = """
<form action="" method="POST">
    <script
        src="https://checkout.stripe.com/checkout.js" class="stripe-button"
        data-key="pk_test_6pRNASCoBOKtIshFeQd4XMUh"
        data-amount="999"
        data-name="Stripe.com"
        data-description="Hello World"
        data-locale="auto">
    </script>
</form>
"""

def application(environ, start_response):
    try:
        request_body_size = int(environ.get('CONTENT_LENGTH', 0))
    except (ValueError):
        request_body_size = 0
    request_body = environ['wsgi.input'].read(request_body_size)
    post = parse_qs(request_body)
    out = ''
    if post:
        print post
        token = post.get('stripeToken', [''])[0]
        token = escape(token)
        if token:
            import stripe
            stripe.api_key = "sk_test_BQokikJOvBiI2HlWgH4oIfQ2"
            try:
                charge = stripe.Charge.create(
                    amount="999",
                    currency="usd",
                    source=token,
                    description="Hello World",
                )
                out = '<pre>charge: %s</pre>' % (charge,)
            except Exception as e:
                print 'Exception %s' % (str(e),)
        else:
            out = 'missing in post: token'
```

```
else:
    out = form
    response_body = html % {
        'output': out,
    }
    status = '200 OK'
    response_headers = [('content-type', 'text/html;charset=utf-8')]
    start_response(status, response_headers)
    return [response_body]

from wsgiref.simple_server import make_server
from cgi import parse_qs, escape
httpd = make_server('', 8000, application)
httpd.serve_forever()
```

Tenga en cuenta:

- El formulario frontend contiene la **clave pública**.
- la parte de carga backend contiene la **clave secreta** .

Ejecuta el script

```
python payment.py
```

Navega con tu navegador para

```
http://localhost:8000/
```

Después de hacer clic en el botón Pagar e ingresar el número de la tarjeta de crédito (4242424242424242), el formulario se publica con el token. Por lo tanto, el pago podría procesarse y, finalmente, el objeto de `charge` se imprimirá en el navegador, que contiene:

```
...
"paid": true,
"description": "Hello World",
"status": "succeeded"
```

Recursos y lectura adicional:

- Publicación de WSGI: <http://wsgi.tutorial.codepoint.net/parsing-the-request-post>
- Formulario de frontend: <https://stripe.com/docs/checkout/tutorial>
- Cargo de backend: <https://stripe.com/docs/charges>

## Introducción a la API de Stripe

Un flujo de pago típico con Stripe se puede dividir en dos pasos:

1. Del lado del cliente, en su código de frontend (HTML + Javascript), recopila la información de pago del cliente mediante el formulario de [Checkout](#) creado previamente de Stripe o los campos de formulario de [elementos](#) . Esto devolverá un token que luego enviará a su servidor.

2. En el lado del servidor, en su código de fondo (en PHP, Python, Ruby o el lenguaje de programación del lado del servidor que prefiera), utiliza el token en una [solicitud de creación de cargo](#) para cargar la tarjeta.

El punto de este flujo de 2 pasos es que su servidor solo funciona con tokens de tarjeta y nunca con información en bruto de la tarjeta. Esto significa que nunca tendrá acceso a los números de tarjeta, lo que facilita en gran medida la carga del [cumplimiento de PCI](#) .

La [documentación](#) de Stripe es bastante extensa e incluye muchos ejemplos y tutoriales. ¡Asegúrate de revisarla!

Lea [Comenzando con los pagos de banda en línea](#): <https://riptutorial.com/es/stripe-payments/topic/1672/comenzando-con-los-pagos-de-banda>



# Capítulo 2: Documentación de PHP-Stripe Connect para Symfony2

## Parámetros

Parámetro	Detalles
cantidad	requerido: un entero positivo en la unidad monetaria más pequeña (por ejemplo, 100 centavos para cobrar \$ 1.00 o 100 para cobrar ¥ 100, una moneda de 0 decimales) que representa la cantidad a cobrar a la tarjeta. La cantidad mínima es de \$ 0.50 US o equivalente en la moneda a cargo.
moneda	requerido - código ISO de 3 letras para la moneda.
descripción	opcional, el valor predeterminado es nulo: una cadena arbitraria que puede adjuntar a un objeto de cargo. Se muestra cuando está en la interfaz web junto con el cargo. Tenga en cuenta que si usa Stripe para enviar recibos automáticos por correo electrónico a sus clientes, sus correos electrónicos de recibo incluirán la descripción de los cargos que están describiendo.
recibo de correo	la opción predeterminada es nula: la dirección de correo electrónico a la que se envía el recibo de este cargo. El recibo no será enviado hasta que se pague el cargo. Si este cargo es para un cliente, la dirección de correo electrónico especificada aquí anulará la dirección de correo electrónico del cliente. Los recibos no se enviarán por cargos de modo de prueba. Si se especifica <code>rece_email</code> para un cargo en modo en vivo, se enviará un recibo independientemente de su
exp_month	requerido - Número de dos dígitos que representa el mes de vencimiento de la tarjeta.
exp_year	requerido - Número de dos o cuatro dígitos que representa el año de vencimiento de la tarjeta.
número	requerido - El número de tarjeta, como una cadena sin separadores.
cvc	Generalmente se requiere -Card código de seguridad. Requerido a menos que su cuenta esté registrada en Australia, Canadá o los Estados Unidos. Muy recomendable incluir siempre este valor.

## Examples

### Symfony2 - Ejemplo de integración de bandas

Descargue la biblioteca API de Stripe y colóquela en la carpeta del proveedor

fuelle: [ <https://github.com/stripe/stripe-php>◆◆1]

incluye la biblioteca en tu controlador

```
use Stripe\BalanceTransaction;
use Stripe\Charge;
use Stripe\Stripe;
require_once('../vendor/stripe/init.php');
```

configurar la tecla de tira

```
\Stripe\Stripe::setApiKey('stripe_secret_key');
```

Llame a la función de carga para la transacción

```
$card = array(
    'number' => 'cardccn',
    'cvc' => 'cardcvc',
    'exp_month' => 'expMonth',
    'exp_year' => 'expYear',
);

$charge = Charge::create(
    array(
        'amount' => ('amount') * 100, // Amount will store in cent in Stripe Account
        'currency' => 'usd',
        'card' => $card,
        'description' => '$data['description',
        'receipt_email'=>'receipt_email'
    )
);
```

obtener los detalles del cargo

```
$data = Charge::retrieve('ch_%');
```

Lea Documentación de PHP-Stripe Connect para Symfony2 en línea:

<https://riptutorial.com/es/stripe-payments/topic/6633/documentacion-de-php-stripe-connect-para-symfony2>

# Capítulo 3: Raya Agregar múltiples tarjetas al mismo usuario

## Examples

### Crear cliente en banda

```
public function createCustomer($data , $token)//pass form data and token id
{
    $customer=Customer::create(array(
        "email"=>$data['email'],
        "description" => $data['name'],
        "source" => $token // obtained with Stripe.js
    ));
    return $customer['id'];
}
```

Para más información siga este [enlace](#).

### Cómo recuperar clientes y agregar tarjetas en Stripe

```
public function addCard($cust_id, $token)
{
    $retriveResult=Customer::retrieve($cust_id);
    $tokendata = Token::retrieve($token);
    $newcard = $tokendata['card'];
    $flag = 1;

    foreach ($retriveResult['sources']['data'] as $card) {
        if($card['fingerprint'] === $newcard['fingerprint'])
        {
            $cardid = $card['id'];
            $flag = 0;
            break;
        }
    }

    if($flag)
    {
        $savecard = $retriveResult->sources->create(array("source" =>$token));
        $cardid = $savecard['id'];
    }
    return $cardid;
}
```

Lea Raya Agregar múltiples tarjetas al mismo usuario en línea: <https://riptutorial.com/es/stripe-payments/topic/9547/raya-agregar-multiples-tarjetas-al-mismo-usuario>

# Capítulo 4: Stripe.Net Introducción

## Sintaxis

- `var stripeSubscriptionOptions = new StripeSubscriptionCreateOptions ();`

`// crear una variable para mantener el objeto de opciones`

```
stripeSubscriptionOptions.Quantity = model.update;
```

`// Ejemplo de opción de cantidad de asientos para una suscripción.`

```
var subscriptionService = new StripeSubscriptionService();
```

`// crear un servicio para hacer la llamada a la API`

```
var stripeSubscription = subscriptionService.Create(user.CustomerIdentifier,  
planId,
```

`stripeSubscriptionOptions);`

`// service.create (string CustID, string PlanID, Object SubscriptionOptions)`

`// El ID del cliente debe guardarse de su base de datos, puede recuperar planID de la banda usando un PlanService y crear el objeto de opciones como se muestra arriba. Si NuGet the Stripe.Net intellisense funciona para estos también.`

## Observaciones

En algún lugar al principio de su controlador debe llamar

```
StripeConfiguration.SetApiKey(YOUR_SECRET_KEY VAR);
```

y para la seguridad de los datos, debe ser un valor oculto como secreto en los ajustes

Si no configura la clave API, no podrá modificar suscripciones ni crear clientes.

## Examples

### Comenzando con Stripe.Net en ASP.Net Core 1.0

<https://github.com/jaymedavis/stripe.net> es un gran punto de partida.

Suponiendo que está utilizando MVC w / Razor, necesita tener algunas cosas en su página Ver

```
<script type="text/javascript" src="https://js.stripe.com/v2/"></script>
```

Este script llama a stripe.js para manejar la creación de un token.

```
<script type="text/javascript">
    Stripe.setPublishableKey('YOUR STRIPE PUBLIC KEY');

var stripeResponseHandler = function (status, response) {
    var $form = $('#payment-form');

    if (response.error) {
        // Show the errors on the form
        $form.find('.payment-errors').text(response.error.message);
        $form.find('button').prop('disabled', false);
    } else {
        // token contains id, last4, and card type
        var token = response.id;
        // Insert the token into the form so it gets submitted to the server
        $form.append($('
```

Para agregar el token a un modelo, asegúrese de cambiar

```
$form.append($('
```

para reflejar su modelo. La forma debe parecerse a esta.

```
<form asp-action="confirm" method="POST" id="payment-form">
    <span class="payment-errors"></span>

    <div class="row">
        <label>
            <span>Card Number</span>
            <input type="text" data-stripe="number" value="4242424242424242">
        </label>
    </div>

    <div class="row">
        <label>
            <span>CVC</span>
            <input type="text" data-stripe="cvc" value="123">
        </label>
    </div>
</form>
```

```

</div>

<div class="row">
  <label>
    <span>Expiration (MM/YYYY)</span>
    <input type="text" data-stripe="exp-month" value="12">
  </label>
  <input type="text" data-stripe="exp-year" value="2020">
</div>

  <button type="submit">Buy Now</button>
</form>

```

El controlador toma a uno de los usuarios y busca un identificador de cliente que es la identificación del cliente dada por franja. Si esto no está guardado en la base de datos, entonces funciona creando un cliente para ellos

```

public async Task<IActionResult> Index(OrderViewModel model) //HOME PAGE beginning of
managing subs
{
  //get the user and their ID
  var user = await GetCurrentUserAsync();
  var userId = user?.Id;
  // If they have a customer Identifier use it
  if (!string.IsNullOrEmpty(user.CustomerIdentifier)) //eventually if user has a saved
card as well
  {
    //Create the API call to subscription and put its response in a list
    var subscriptionService = new StripeSubscriptionService();
    IEnumerable<StripeSubscription> response =
subscriptionService.List(user.CustomerIdentifier);

    ViewBag.Subscription = response;

    ViewBag.Customer = user.CustomerIdentifier;

  }
  ModelState.Clear();
  return View(model);
}
public async Task<IActionResult> Confirm(OrderViewModel model, string stripeToken) // CREATE
CHARGE NO CARD/ NO CUSTOMER
{
  model.stripeToken = stripeToken;
  //get the user and their ID
  var user = await GetCurrentUserAsync();
  var userId = user?.Id;
  var planId = "YOUR PLAN ID HERE"; //plan ID only 1 atm but will need to be a dynamic
plan ID list later
  // If they have a customer Identifier use it
  if (!string.IsNullOrEmpty(user.CustomerIdentifier))
  {
    //Create the API call to subscription and put its response in a list
    //Use the subscription options to apply a quantity to the initial subscription for
seats
    var stripeSubscriptionOptions = new StripeSubscriptionCreateOptions();
    stripeSubscriptionOptions.Quantity = model.update;
    var subscriptionService = new StripeSubscriptionService();

```

```

        var stripeSubscription = subscriptionService.Create(user.CustomerIdentifier,
planId, stripeSubscriptionOptions);
        //save Subscriptions data here

ModelState.Clear();
//await SaveSubscription(stripeSubscription, user);
await _userManager.UpdateAsync(user);
}
else // Customer is new and doesn't have an ID
{
    //Create API options
var customer = new StripeCustomerCreateOptions();

    // Add option values
customer.Email = $"{user.Email}";
customer.Description = $"{user.Email} [{user.Id}]";
customer.PlanId = planId;

customer.SourceToken = model.stripeToken;
//Make the call to create the customer with the creation options
var customerService = new StripeCustomerService();
StripeCustomer stripeCustomer = customerService.Create(customer);

//save the customer ID
user.CustomerIdentifier = stripeCustomer.Id;

//create card update options and add billing info
var cardOptions = new StripeCardUpdateOptions();
cardOptions.AddressLine1 = model.BillingInfo.AddressL1;
cardOptions.AddressLine2 = model.BillingInfo.AddressL2;
cardOptions.AddressCountry = model.BillingInfo.Country;
cardOptions.AddressCity = model.BillingInfo.City;
cardOptions.AddressState = model.BillingInfo.State;
cardOptions.AddressZip = model.BillingInfo.Zip;
cardOptions.Name = model.BillingInfo.Name;
var cardUpdate = new StripeCardService();
// get the customer card ID and then update the card info
StripeCustomer customerCardGet = customerService.Get(user.CustomerIdentifier);
var cardId = customerCardGet.DefaultSourceId;
StripeCard Card = cardUpdate.Update(user.CustomerIdentifier, cardId, cardOptions);

//save Subscriptions data here
//user.ConcurrentUsers = Stripe Quantity

ModelState.Clear();

await _userManager.UpdateAsync(user);

}
ViewBag.Success = "confirm";
return View("Success");
}

```

Este ejemplo particular incluye algunos de los extras, como tomar información de facturación en el controlador. Si desea esto, simplemente agréguelo al formulario en la vista y cree un modelo para mantenerlo.

Lea [Stripe.Net Introducción en línea](https://riptutorial.com/es/stripe-payments/topic/4328/stripe-): <https://riptutorial.com/es/stripe-payments/topic/4328/stripe->





---

# Creditos

S. No	Capítulos	Contributors
1	Comenzando con los pagos de banda	<a href="#">andpei</a> , <a href="#">Community</a> , <a href="#">Yumiko</a> , <a href="#">Ywain</a>
2	Documentación de PHP-Stripe Connect para Symfony2	<a href="#">Mohammad Fareed</a>
3	Raya Agregar múltiples tarjetas al mismo usuario	<a href="#">ashish bansal</a>
4	Stripe.Net Introducción	<a href="#">Alex Rohr</a>