

Pasos previos:

PAGINA WEB: <http://185.207.145.237/> o <http://bit.do/pasarelaIW>

1º Solicitar el acceso a la plataforma y rellenar el formulario de inscripción prestando atención a los siguientes campos:

NICK: Va a ser el identificador del comercio. Se recomienda que coincida con el APP_NAME del fichero .env y se utilice este cuando se necesite identificar la web

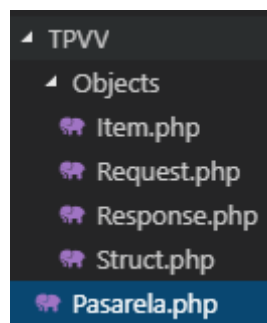
APP_NAME=FruteriaMartinez

KEY: Una clave que se va a emplear para realizar las tareas de cifrado de la información y que sea totalmente segura frente a posibles sabotajes. Se recomienda ponerla también en el fichero .env

TPVV_KEY=masterPasarela

ENDPOINT: Es el punto de escucha a la respuesta de nuestro servidor. Cuando el pago se complete, la respuesta será enviada a ese punto. Se ha de crear una ruta (fichero routes/web.php) especifica que espere un post.

2º Descargar los ficheros e introducirlos en el proyecto en el lugar que deseéis.



3º Comprobar la simulación de tarjetas.

45XX	Visa	TODO OK
51XX	MM	TODO OK
46XX	Visa	TABLA
52XX	MM	TABLA

Tarjeta desactivada	00-19
Error autenticacion titular	20-39
Intentos PIN excedidos	40-59
CVV incorrecto	60-79
Saldo insuficiente	80-99
Caducada (check)	TODOS

El resto dan un error con mensaje genérico.

NOTA: En todos los casos 46XX y 51XX se comprobará la caducidad de la tarjeta, siguiendo las condiciones lógicas de caducidad (meses entre 1 y 12). El año máximo valido es el 2030.

Generar un pago

El controlador asociado al pago llamará a la clase **PASARELA** y generará el objeto, pasando 2 variables a la vista asociada (Los argumentos son el **Nick** y el **idPedido**).

```
$tpvv = new Pasarela('Fran',1); //web, idpedido

$tpvv->AnadirProducto('smartphone',1); //producto,cantidad
$tpvv->AnadirProducto('bateria_externa',1); //producto,cantidad
$tpvv->AnadirProducto('funda',1); //producto,cantidad
$tpvv->AsignarPrecioFinal(267.38); //precio

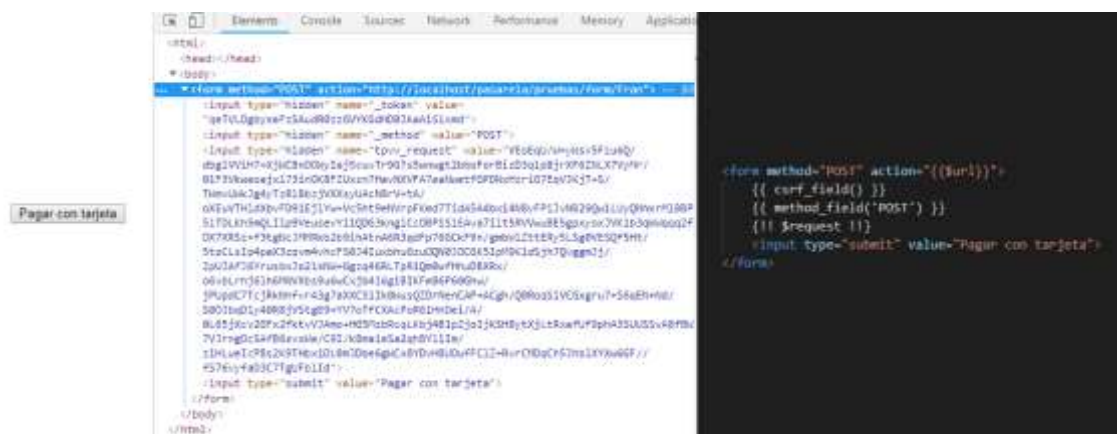
return view('pago/form',['request'=>$tpvv->getREQUEST(),'url'=>$tpvv->getURL()]);
```

Estas dos variables son:

Request: imprimirá un input oculto con la transacción cifrada.

URL: Dirección a la que va a hacer el post (nuestro servidor)

La vista asociada debe incluir un formulario (del diseño que se desee) que deberá incluir los campos generados por la clase Pasarela. La imagen de la derecha es un ejemplo funcional.



Al pagar se redirigirá a la pasarela de pago, donde habrá que rellenar el formulario con todos los datos. Los números que aparecen son los caracteres de la validación básica.

<p>Introducir tarjeta de credito</p>  <p>4581 9231 2837 4383</p> <p>NOMBRE APELLIDOS 12/2020</p> <p>10-19</p> <p>4581 9231 2837 4383</p> <p>12 / 2020 7</p> <p>Nombre Apellidos 3</p> <p>1238 3-4</p>	<p>Lista de productos asociados</p> <p>smartphone - x1</p> <p>bateria_externa - x1</p> <p>funda - x1</p> <p>Precio</p> <p>267.38€</p> <p>PAGAR!</p>
--	---

Tras proceder al pago se realizará la simulación de la tarjeta tal y como se explica en el [paso 3 de los pasos previos](#). Tras ello, mostrará un mensaje informativo de éxito o de error automáticamente se os redirige a la pagina web mediante el **ENDPOINT** configurado durante el registro. Si falla la validación básica no se realizará ninguna redirección.

Pago realizado correctamente. Redirigiendo automáticamente a la web. [Volver](#)

Comprobar un pago

En el controlador se deberá de crear un método que atienda la petición **POST** del **ENDPOINT**. Se deberá crear un objeto **Pasarela** con los valores de la web (**NICK**), el idPedido que será NULL y la clave (**KEY**) para el cifrado. Posteriormente se llamará al método **SetRESPONSE** que se encargará de rellenar de datos de la respuesta.

```
public function endpoint(Request $response){  
  
    $tpvv = new Pasarela('Fran',NULL,'cl12347'); //web, idPedido, clave  
    $tpvv->SetRESPONSE($response);  
    $datos = $tpvv->ValidateResponse(); //Se obtiene un array  
  
}
```

Por último, al llamar al método **ValidateResponse** obtendremos un array de 4 campos, como el que se reflejan a continuación. En caso de que la validación sea errónea se devolverá false.

```
array:4 ▾  
  "web" => "Fran"  
  "idPedido" => 1001  
  "estado" => "aceptado"  
  "fecha" => "2019-01-20 22:52:11"  
]
```

generado
aceptado
rechazado
devolucion

<- Estados posibles

Para evitar el error **419** debido a la verificación **csrf** hay que indicar al middleware que no realice dicha verificación en la ruta **endpoint**. Para ello hay que poner como excepción la misma cadena que se encuentra en el fichero **routes/web.php**. A continuación, un ejemplo:

```
class VerifyCsrfToken extends Middleware  
{  
    protected $addHttpCookie = true;  
  
    protected $except = [  
        'generar/*'  
    ];  
}  
//TPVV  
Route::post('generar/{web}', "PasarelaController@pform");
```