

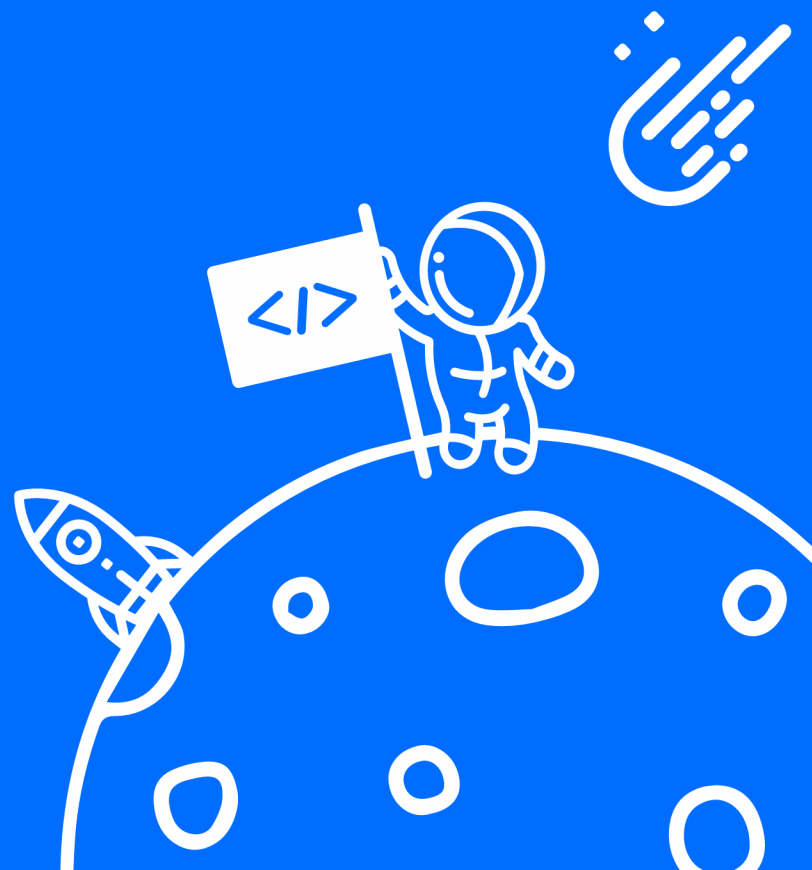
PAYCOMET

## Proxy YieldPlanet v1.4

24-02-2020

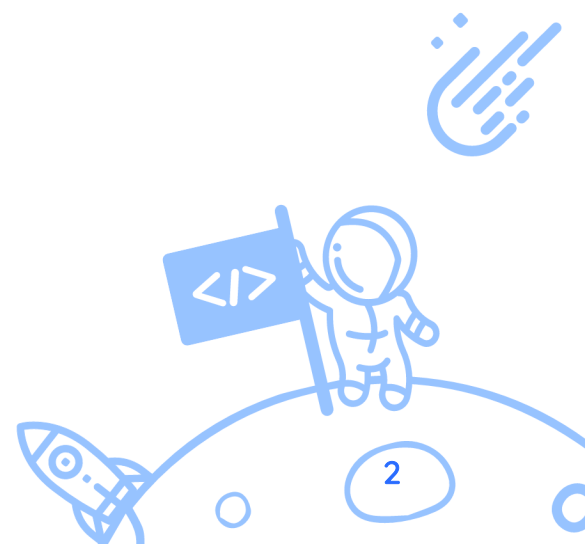
<https://www.paycomet.com>  
[info@paycomet.com](mailto:info@paycomet.com)

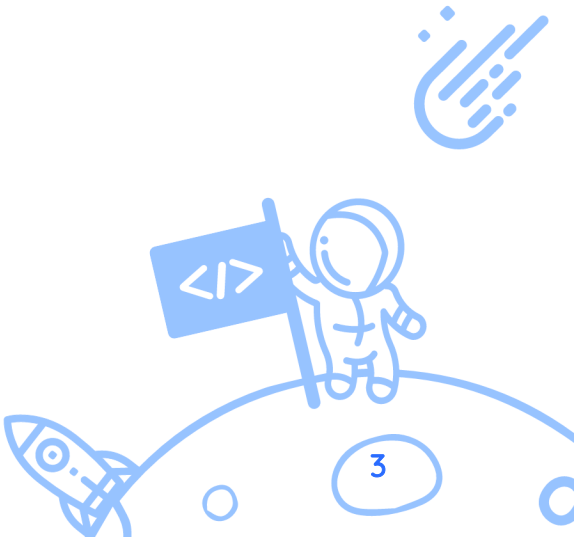
Paycomet © 2020



## Control de versiones

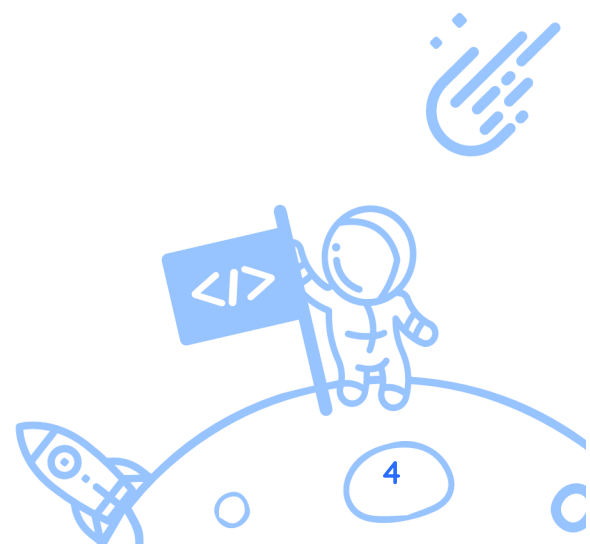
| Autor             | Fecha      | Versión | Modificaciones  |
|-------------------|------------|---------|---|
| Juan Sánchez      | 07/09/2017 | V 1.0   | Versión inicial   |
| W.E.D             | 20/09/2017 | V 1.0.1 | Corrección etiqueta de cierre username en el ejemplo del apartado Endpoint y ejemplo de llamada |
| W.E.D             | 02/03/2013 | V 1.1   | En tarjetas ya tokenizadas, se devolverá únicamente el elemento iduser                          |
| Juan Sánchez      | 12/03/2018 | V 1.2   | Se permite tokenizar tarjetas sin cvc.  |
| Jose Ramón García | 26/06/2019 | V 1.3   | Cambio Marca y Firma por SHA512   |
| Diego Marcos      | 24/02/2020 | V 1.4   | Cambio EndPoints y API-KEYS   |





## Contenido

|  |    |
|--|----|
| Control de versiones .....                                     | 2  |
| Introducción .....   | 5  |
| Tipos de reservas.....   | 5  |
| Tokenización de la información de tarjeta de una reserva ..... | 8  |
| Transformación de los datos de tarjeta .....                   | 8  |
| Endpoint y datos de la llamada .....                           | 10 |
| Respuesta error.....   | 11 |
| Envío de API-KEYS.....   | 12 |
| Entorno de desarrollo.....                                     | 13 |
| Contacto .....   | 13 |

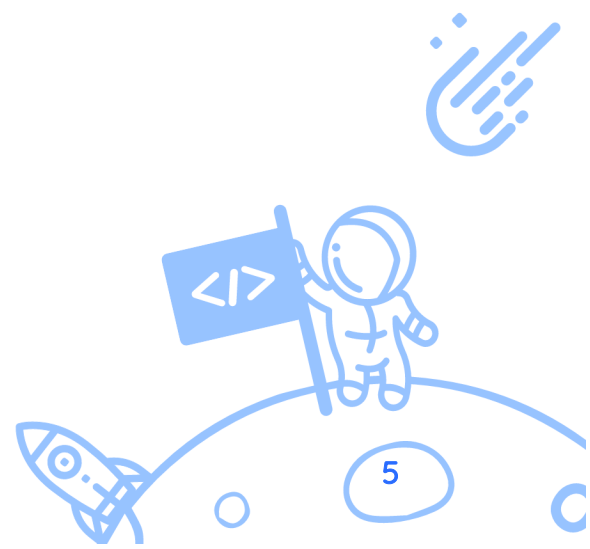


## Introducción

El servicio Proxy de PAYCOMET con YieldPlanet permite tokenizar las tarjetas registradas en las reservas hoteleras y verificar que el formato de los datos de tarjeta recibidos en las mismas son correctos. Para poder acceder a este tipo de integración, se deberá contar previamente con la integración de la API de YieldPlanet (en el momento de generar esta versión XML Services - YieldPlanet API 3.0), en la que se envían los datos de la reserva ya que la respuesta incluirá todos los datos de la reserva seleccionada y se añadirán y/o transformarán los relativos a la tarjeta.

## Tipos de reservas

Los tipos de reservas que se tratan en esta integración son aquellas que pueden recuperarse mediante identificador de hotel (`hotel_id`) y aquellas con cambios de estado desde una fecha (`status_changed_since`).



## Ejemplo de Información contenida en una reserva

```

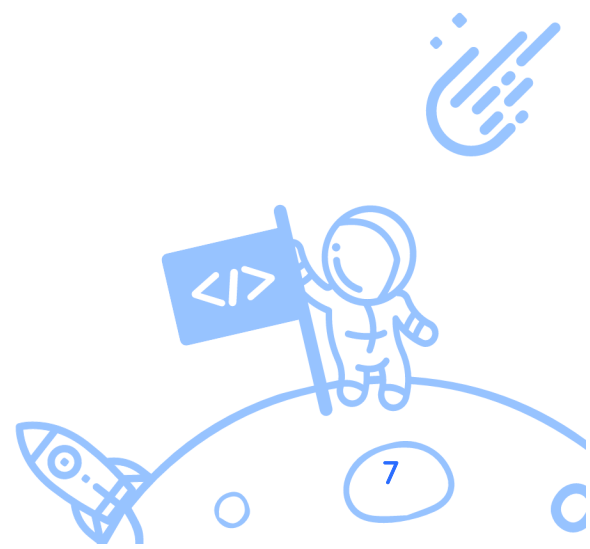
<?xml version="1.0" encoding="utf-8"?>
<Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <GetOrdersResponse xmlns="XmlServices">
    <GetOrdersResult>
      <Status>
        <Success>true</Success>
        <ErrorId>0</ErrorId>
        <TimeStamp>2017-09-07T14:05:52.0162772+02:00</TimeStamp>
      </Status>
      <Orders>
        <Order>
          <Id>34192783</Id>
          <HotelId>3</HotelId>
          <PartnerId>719</PartnerId>
          <PartnerName>Booking.com XML</PartnerName>
          <Code>test05091</Code>
          <OrderDate>2017-09-05T12:55:50+02:00</OrderDate>
          <Arrival>2017-09-11T00:00:00</Arrival>
          <Departure>2017-09-12T00:00:00</Departure>
          <CustomerFirstName>Michał</CustomerFirstName>
          <CustomerName>Krasicki</CustomerName>
          <CustomerCountryCode>PL</CustomerCountryCode>
          <CustomerAddress>Witunia          ul.          Złotowska
14</CustomerAddress>

          <CustomerCity>Więcbork</CustomerCity>
          <CustomerZip/>
          <CustomerEmail>mail@mail.com</CustomerEmail>
          <CustomerPhone>503087111</CustomerPhone>
          <CustomerComments>
            Breakfast at 6 BED PREFERENCE:Double or Twin
Room: 2 singles You have a booker that would like free parking. (based on availability)
          </CustomerComments>
          <ExtraInfo>
            rate_id:    918508,    rewritten_from_id:    0,
rewritten_from_name: Secret rate
          </ExtraInfo>
          <CommissionInfo/>
          <CancellationPolicy/>
          <ccType>VISA</ccType>
          <ccNumber>4111111111111111</ccNumber>
          <ccExpireDate>2020-08-01T00:00:00</ccExpireDate>
          <ccCVV>111</ccCVV>
          <ccHolderName>tes test</ccHolderName>
          <TotalPrice>209.53</TotalPrice>
          <Currency>USD</Currency>
          <PaymentType>Unspecified</PaymentType>
          <Status>Confirmed</Status>
          <LastStatusChanged>2017-09-
05T14:30:43+02:00</LastStatusChanged>
        </Order>
      </Orders>
    </GetOrdersResult>
  </GetOrdersResponse>
</Envelope>

```

```
Room</Name>
<NumRooms>1</NumRooms>
<NumAdults>2</NumAdults>
<NumChildren>0</NumChildren>
<MealType>Breakfast</MealType>
<GuestNames/>
<Sequence>1</Sequence>
<Prices>
  <OrderRoomPrice>
    <Date>2017-09-
11T00:00:00</Date>
    <Price>209.53</Price>
  </OrderRoomPrice>
</Prices>
<AddOns/>
</OrderRoom>
</Rooms>
<AddOns/>
<NumAdults xsi:nil="true"/>
</Order>
</Orders>
</GetOrdersResult>
</GetOrdersResponse>
</Envelope>
```

Los procesos que se describen a continuación se realizan para cada una de las reservas contenidas en el nodo Orders



## Tokenización de la información de tarjeta de una reserva

Al recibir una reserva, si esta contiene datos de tarjeta (existen reservas que no pueden tratarse ya que no los contienen), se intenta realizar una tokenización de los mismos. Si todos los datos necesarios están presentes y se consigue realizar correctamente, al nodo <Order> se le añadirán los elementos

- iduser
- tokenuser

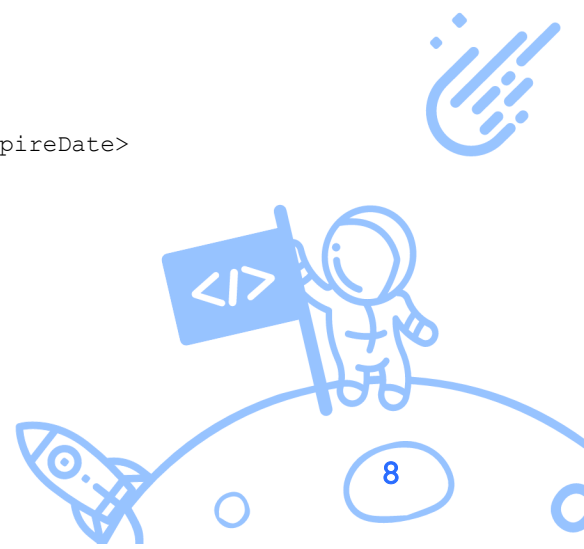
que representan la tarjeta tokenizada y que serán necesarios para operaciones posteriores. A partir de la versión 1.2 se permite tokenizar aunque no llegue cvc. Por favor, solícitanos la habilitación de esta opción.

En los casos en los que una tarjeta ya se haya tokenizado previamente y se haya devuelto el par iduser/tokenuser con anterioridad, solamente se añadirá el elemento iduser al nodo Order, que coincidirá con el que se asignó y devolvió en el momento de realizar la primera tokenización de esa tarjeta.

## Transformación de los datos de tarjeta

En la respuesta devuelta, la tarjeta aparecerá enmascarada, 411111\*\*\*\*\*1111. Le fecha de caducidad se devolverá tal y como aparece en la reserva y el código CVC2 se eliminará y no llegará. Por lo tanto, en la reserva anterior, una vez tokenizada la información de la tarjeta, el nodo <Order> quedará de la siguiente manera:

```
<Order>
  ...
  <ccType>VISA</ccType>
  <ccNumber>411111*****1111</ccNumber>
  <ccExpireDate>2020-08-01T00:00:00</ccExpireDate>
  <ccHolderName>tes test</ccHolderName>
  ...
  <iduser>99999999</iduser>
  <tokenuser>XxXxXxXxXxXxX</tokenuser>
</Order>
```





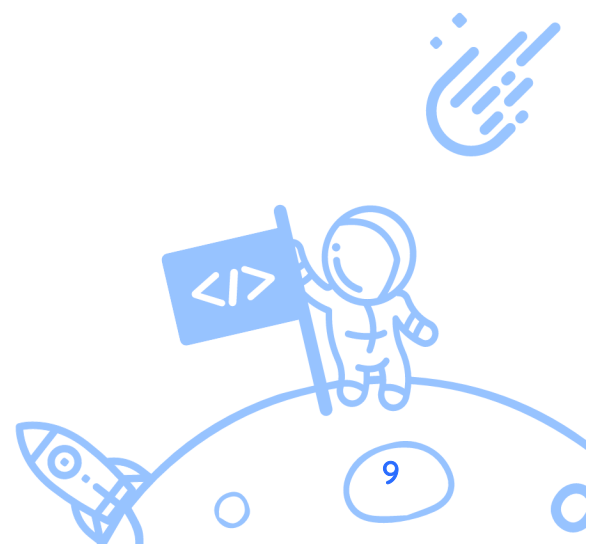
En caso de error en la tokenización, no aparecerán los elementos iduser / tokenuser y se añadirá un elemento ds\_error\_id

```
<Order>
  ...
  <ccType>VISA</ccType>
  <ccNumber>411111*****1111</ccNumber>
  <ccExpireDate>2020-08-01T00:00:00</ccExpireDate>
  <ccHolderName>tes test</ccHolderName>
  ...
  <ds_error_id>9999</iduser>
</Order>
```

Se informará también dentro del nodo Order, en el caso de que existan datos de tarjeta, de si la tarjeta pertenece a la categoría B2B; posibles valores 1/0

```
<Order>
  ...
  <NumAdults xsi:nil="true"/>
  <b2b>0</b2b>
  <iduser>99999999</iduser>
  <tokenuser> XxXxXxXxXxXxX </tokenuser>
</Order>
```

En el caso de que una reserva no contenga datos de tarjeta, se devolverá tal y como fue recibida.



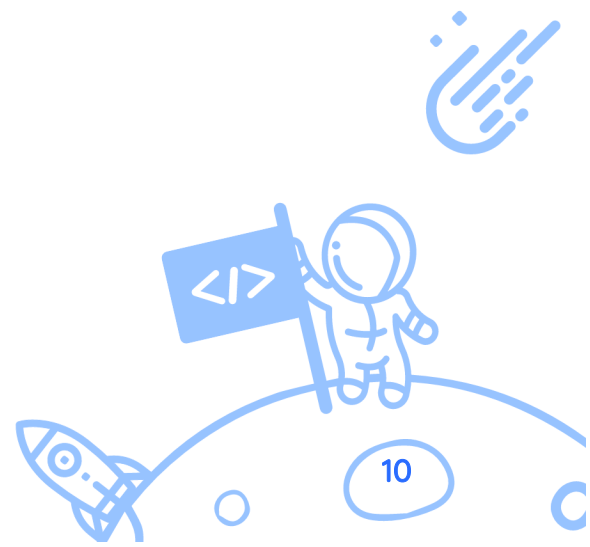
## Endpoint y datos de la llamada

El script al que hay que realizar la petición es:

- <https://eclipse.paycomet.com/api/v1/yieldplanet>

El tipo de petición que espera es un post con un parámetro 'xml' con la siguiente estructura:

```
<root>
  <booking>
    <xml>
      <request>
        <user_name>vuestro usuario YieldPlanet</user_name>
        <password>vuestra password YieldPlanet</password>
        <hotel_id>id del Hotel</hotel_id>
        <status_changed_since>cambio de estado desde</ status_changed_since>
      </request>
    </xml>
    <url>URL YieldPlanet a la que se debe realizar la llamada</url>
  </booking>
  <paytpv>
    <ds_merchant_code>código de comercio en PAYTPV</ds_merchant_code>
    <ds_merchant_terminal>número de terminal en
PAYTPV</ds_merchant_terminal>
  </paytpv>
</root>
```



## Respuesta error

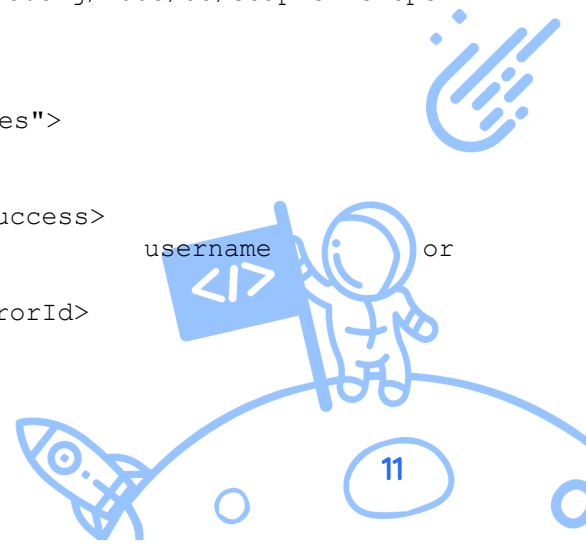
Si se produce un error en los datos de la petición o al acceder a YieldPlanet, se enviará un mensaje de error igual al que devuelve YieldPlanet actualmente, con la consideración siguiente: si el error se produce en la llamada a PAYCOMET (credenciales, formato) o en la llamada desde PAYCOMET a YieldPlanet (timeout, sin respuesta, etc) el elemento ErrorId se sustituye por ErrorId-PAYTPV, indicando que no ha habido establecimiento de conexión con YieldPlanet. Una vez superado ese punto, si se produce un error se devolverá exactamente lo devuelto por YieldPlanet (con elemento ErrorId) :

### Error PAYCOMET anterior a la llamada a YieldPlanet

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetOrdersResponse xmlns="XmlServices">
      <GetOrdersResult>
        <Status>
          <Success>>false</Success>
          <ErrorMsg>Invalid signature. Please check your
configuration</ErrorMsg>
          <ErrorId-PAYTPV>1123</ErrorId-PAYTPV>
          <TimeStamp>2017-09-
07T13:32:56.7964590+02:00</TimeStamp>
        </Status>
      </GetOrdersResult>
    </GetOrdersResponse>
  </soap:Body>
</soap:Envelope>
```

### Error devuelto por YieldPlanet

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetOrdersResponse xmlns="XmlServices">
      <GetOrdersResult>
        <Status>
          <Success>>false</Success>
          <ErrorMsg>Invalid
password.</ErrorMsg>
          <ErrorId>1001</ErrorId>
```



```
                <TimeStamp>2017-09-  
07T13:53:23.2633149+02:00</TimeStamp>  
                </Status>  
            </GetOrdersResult>  
        </GetOrdersResponse>  
    </soap:Body>  
</soap:Envelope>
```

## Envío de API-KEYS

En el nuevo Endpoint el método de autenticación ha sido actualizado de una restricción por IP al envío una cabecera específica con el API-KEY generado para el producto/productos en cuestión.

La cabecera HTTP debe tener el siguiente nombre:

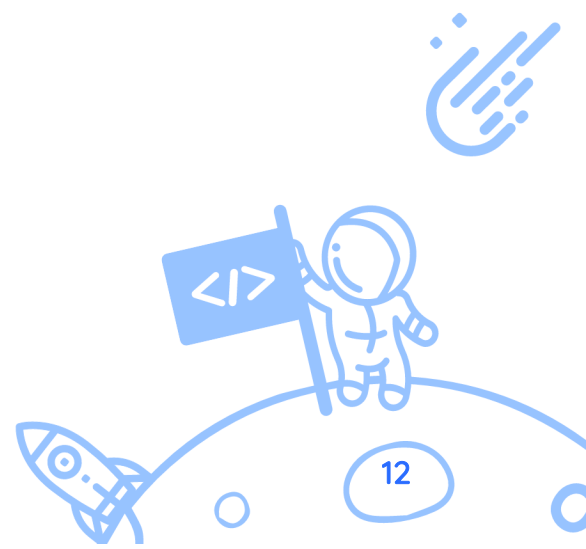
**PAYCOMET-API-TOKEN**

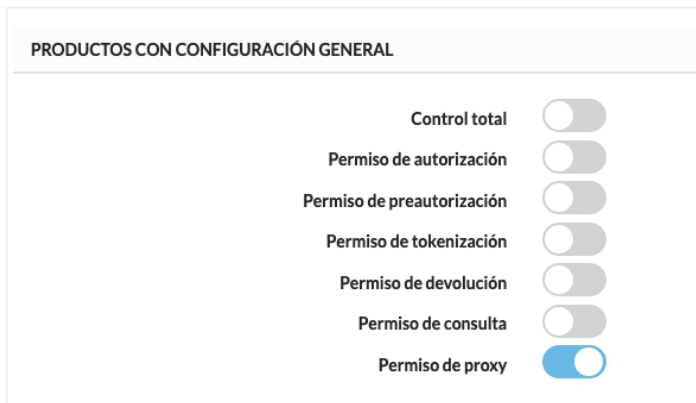
El valor de la cabecera debe ser la clave API-KEY generada a través del panel de control de cliente: "*Mis productos*" -> "*API Keys*".

Una vez en el panel de gestión de API Keys debe crearse una nueva "API KEY" en el botón superior derecho y asignarle un nombre.

En el siguiente panel debe copiarse la API KEY ya que solo aparecerá en una ocasión. Por motivos de seguridad, no volverá a mostrarse.

Los permisos de las API KEYS pueden asignarse a uno o varios productos, e incluso a todos los de una cuenta mediante los selectores:





Para el producto que vaya a utilizarse para presentar las reservas de la OTA (Online Travel Agency) y donde quedarán tokenizadas las tarjetas será necesario asignarle permisos específicos de "**Permiso de proxy**".

Una vez finalizadas las gestiones de API Keys, el valor generado a la API Key deberá enviarse en la cabecera como método de autenticación.

## Entorno de desarrollo

Este desarrollo no cuenta con un sandbox en el que puedan realizarse llamadas con datos de prueba. Las llamadas al script son todas reales. Lo que debe hacerse para poder probar la integración es crear en YieldPlanet hoteles y reservas de prueba que permitan verificar el proceso completo.

## Contacto

Para ampliar información o resolver cualquier duda acerca de esta integración, puedes ponerte en contacto con nosotros en esta dirección de correo [tecnico@paycomet.com](mailto:tecnico@paycomet.com)

