

## Documentación de la API clickline.com

Esta parte de la documentación incluye la descripción del protocolo y las muestras de herramientas que se utilizan para interactuar con el servidor de Clickline.com a través de la API.

El capítulo “Protocolo API-RPC” (2.1) establece los conceptos de tecnología. Explica lo que es la API, por qué es necesaria, como puede ser utilizada para la interacción entre Clickline y software de terceros, y los principios de interacción principal.

### 1. Prerrequisitos para el documento

Para entender este documento debe de estar familiarizado con:

1. Mensajes HTTP (types, structure). RFC 2616 (<http://www.ietf.org/rfc/rfc2616.txt>).
2. Tener una idea básica de que es y cómo se utilizan los ficheros XML (<http://www.w3schools.com/xml/default.asp>)

### 2. Protocolo API-RPC

Para apoyar la interacción entre Clickline.com y software de terceros, Clickline ofrece una API basada en XML, para facilitar dicha integración los paquetes XML se enviarán mediante protocolo HTTP cifrado con un formato y forma específica.

#### ¿Quién puede usar la API?

Por el momento, el uso de dicha API se le permite a clientes de Clickline y desarrolladores, teniendo estos últimos que registrarse para recibir su acceso.

No hay versiones específicas del API para Linux o Windows, ya que ambos sistemas operativos comparten la misma API.

La última versión del API de Clickline está establecida en 1.0.0.0

#### 2.1 Modelo Cliente - Servidor

La interacción entre Clickline y software de terceros se basa en el modelo cliente-servidor:

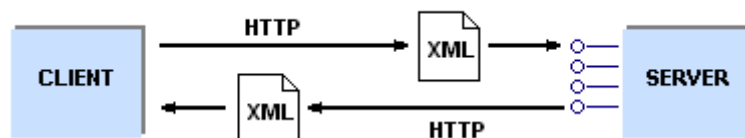


Figura 1: Interacción entre Clickline y software de terceros

La parte del servidor está representada por la API de Clickline (en adelante Servidor) que es una parte de Clickline.com. El software de terceros debe de ser desarrollado por una empresa terciaria (en adelante cliente).

La parte cliente envía un paquete de solicitud XML añadiendo una cabecera HTTP estándar, y la envía al servidor. Por el lado del servidor el paquete llega y es validado y procesado, el servidor realiza las operaciones solicitadas y responde con otro paquete XML que puede contener un resultado erróneo o correcto, dependiendo de lo sucedido en la ejecución.

*Ejemplo de envío de paquete:*

Código	Descripción
POST /includes/api.php HTTP/1.1	Se envía una petición POST al servidor localizado en /includes/api.php
Host: www.clickline.com Accept: */* HTTP_AUTH_LOGIN: usuario HTTP_AUTH_PASSWD: contraseña Pragma: no-cache Content-Length: 1398 Content-Type: text/xml	La versión es HTTP 1.1  Los encabezados HTTP_AUTH_LOGIN y HTTP_AUTH_PASSWD contendrán el usuario y contraseña que está intentando consultar. El content-type debe de ser "XML"
<?xml version="1.0" encoding="UTF-8" standalone="no"?>	Inicio del fichero XML
<packet version="1.0.0.0">	Indicamos en el atributo que versión del API vamos a utilizar, generalmente la última disponible.
<action> get_country_list</action>	Indica la acción a procesar por la API.
</packet>	Fin del paquete.

### *Ejemplo de respuesta del API:*

Código	Descripción
HTTP/1.1 200 OK Connection: close Content-Length: 161 Content-Type: text/xml Date: Mon, 3 Jul 2011 10:55:08 GMT Server: Clickline.com	Inicio del encabezado HTTP de respuesta.  El tipo MIME es "text/xml".
<?xml version="1.0" encoding="UTF-8" standalone="no"?>	Inicio del fichero XML
<packet version="1.0.0.0">	Indica en el atributo que versión del API está respondiendo, generalmente la última disponible.
<status> <result>success</result> </status> <country> <country_id>ES</country_id> <country_name>España</country_name> </country>	En "Status" nos indica si ha sido correcto o erróneo, si es diferente de "success" es erróneo.  Luego nos devuelve los datos consultados.
</packet>	Fin del paquete.

### *Envío de una petición en PHP:*

Para enviar una petición en PHP utilizaremos CURL, al simplificar el protocolo la acción no ocupará más de 20 líneas.

```

define ('HOST', 'www.clickline.com');
define ('PORT', 80);
define ('PATH', '/includes/api.php');
$url = 'https://' . HOST . ':' . PORT . '/' . PATH;
$headers = array(
    'HTTP_AUTH_LOGIN: usuario',
    'HTTP_AUTH_PASSWD: contraseña',
    'Content-Type: text/xml'
);
$ch = curl_init();
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
curl_setopt($ch, CURLOPT_URL, $url);
$packet = <<<EOP
<packet version="1.0.0.0">
<action>put_order</action>
</packet>
EOP;
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_exec($ch);
curl_close($ch);

```

### 3. Operaciones disponibles en la API

Hasta el momento, las operaciones disponibles son:

- **get\_balance\_inquiry**  
Consulta de saldo disponible para el cliente que consulta la API.
- **get\_shipping\_quote**  
Cotización de un transporte con pesos y medidas.
- **get\_shipping\_quote\_no\_measure**  
Cotización de un transporte sin medidas.
- **get\_carriers\_list**  
Lista los transportistas disponibles en el sistema.
- **get\_carriers\_list\_customer**  
Lista los transportistas asociados a la cuenta de usuario que consulta.
- **get\_orders**  
Trae un listado de pedidos realizados por el cliente.
- **get\_orders\_status\_id**  
Trae el status de un id de orden.
- **get\_order\_status**  
Trae un listado de los diferentes status de los pedidos.
- **get\_country\_list**  
Trae un listado de países válidos en el sistema.
- **put\_carriers**  
Graba los transportistas que usará el cliente.
- **put\_order**  
Graba un pedido al sistema, debemos comprobar primero si el cliente tiene saldo.

#### 3.1 Envíos y respuesta según función de la API

Detallaremos el contenido que debe de enviar el cliente y la respuesta según función descrita. Los valores en cursiva serán opciones y utilizados en forma de filtro.

## *Get\_balance\_inquiry*

### **El cliente envía:**

```
<?xml version="1.0"?>
<packet version="1.0.0.0">
  <action>get_balance_inquiry</action>
</packet>
```

### **El cliente recibe:**

```
<?xml version="1.0"?>
<packet version="1.0.0.0">
  <status>
    <result>success</result>
  </status>
  <balance>999</balance>
</packet>
```

Donde "balance" es el saldo a favor, siendo 0 un valor nulo y cualquier otro valor numérico su saldo real.

## *Get\_country\_list*

### **El cliente envía:**

```
<?xml version="1.0"?>
<packet version="1.0.0.0">
  <action>get_country_list</action>
</packet>
```

### **El cliente recibe:**

```
<?xml version="1.0"?>
<packet version="1.0.0.0">
  <status>
    <result>success</result>
  </status>
  <country>
    <country_id>ES</country_id>
    <country_name>España</country_name>
  </country>
  <country>
    <country_id>PT</country_id>
    <country_name>Portugal</country_name>
  </country>
</packet>
```

## *Get\_carriers\_list*

### **El cliente envía:**

```
<?xml version="1.0"?>
<packet version="1.0.0.0">
  <action>get_carriers_list</action>
</packet>
```

### **El cliente recibe:**

```
<?xml version="1.0"?>
<packet version="1.0.0.0">
  <status>
    <result>success</result>
```

```

    </status>
    <carriers>
      <carrier_id>1</carrier_id>
      <carrier>ASM</carrier>
    </carriers >
    <carriers>
      <carrier_id>2</carrier_id>
      <carrier>ASM Internacional</carrier>
    </carriers >
  </packet>

```

### *Get\_carriers\_list\_customer*

#### **El cliente envía:**

```

<?xml version="1.0"?>
<packet version="1.0.0.0">
  <action>get_carriers_list_customer</action>
</packet>

```

#### **El cliente recibe:**

```

<?xml version="1.0"?>
<packet version="1.0.0.0">
  <status>
    <result>success</result>
  </status>
  <carriers>
    <carrier_id>1</carrier_id>
    <carrier>ASM</carrier>
  </carriers >
  <carriers>
    <carrier_id>2</carrier_id>
    <carrier>ASM Internacional</carrier>
  </carriers >
</packet>

```

### *Get\_orders*

#### **El cliente envía:**

```

<?xml version="1.0"?>
<packet version="1.0.0.0">
  <action>get_orders</action>
  <order_status>processing</order_status>
  <start_date>2011-01-01</start_date>
  <end_date>2011-12-31</end_date>
</packet>

```

#### **El cliente recibe:**

```

<?xml version="1.0"?>
<packet version="1.0.0.0">
  <status>
    <result>success</result>
  </status>
  <orders>
    <order_id>1</order_id>
    <order_status>processing</order_status>
    <order_price>100.000</order_price>
    <order_date>2011-07-11 09:26:31</order_date>
    <order_shipping_name>Nombre cliente</order_shipping_name>
  </orders>
  <orders>
    <order_id>2</order_id>

```

```
    <order_status>processing</order_status>
    <order_price>120.000</order_price>
    <order_date>2011-07-11 10:40:13</order_date>
    <order_shipping_name>Nombre cliente</order_shipping_name>
  </orders>
</packet>
```

### *Get\_order\_status*

#### **El cliente envía:**

```
<?xml version="1.0"?>
<packet version="1.0.0.0">
  <action>get_order_status</action>
</packet>
```

#### **El cliente recibe:**

```
<?xml version="1.0"?>
<packet version="1.0.0.0">
  <status>
    <result>success</result>
  </status>
  <orders_status>
    <orders_status_id>canceled</orders_status_id>
    <order_status_name>Canceled</order_status_name>
  </orders_status>
  <orders_status>
    <orders_status_id>closed</orders_status_id>
    <order_status_name>Closed</order_status_name>
  </orders_status>
</packet>
```

### *Get\_orders\_status\_id*

#### **El cliente envía:**

```
<?xml version="1.0"?>
<packet version="1.0.0.0">
  <action>get_orders_status_id</action>
  <order_id>100</order_id>
</packet>
```

#### **El cliente recibe:**

```
<?xml version="1.0"?>
<packet version="1.0.0.0">
  <status>
    <result>success</result>
  </status>
  <orders>
    <order_id>100</order_id>
    <order_status>processing</order_status>
    <order_price>100.000</order_price>
    <order_date>2011-07-11 09:26:31</order_date>
    <order_shipping_name>Nombre cliente</order_shipping_name>
  </orders>
</packet>
```

### *Put\_carriers*

#### **El cliente envía:**

```
<?xml version="1.0"?>
<packet version="1.0.0.0">
  <action>put_carriers</action>
  <carriers>
    <carrier_id>3</carrier_id>
  </carriers>
  <carriers>
    <carrier_id>5</carrier_id>
  </carriers>
</packet>
```

#### El cliente recibe:

```
<?xml version="1.0"?>
<packet version="1.0.0.0">
  <status>
    <result>success</result>
    <total_carriers>2</total_carriers>
  </status>
</packet>
```

### *Get\_shipping\_quote*

#### El cliente envía:

```
<?xml version="1.0"?>
<packet version="1.0.0.0">
  <action>get_shipping_quote</action>
  <shipping_cp_from>08004</shipping_cp_from>
  <shipping_cp_to>28001</shipping_cp_to>
  <country_from>ES</country_from>
  <country_to>ES</country_to>
  <packages>1</packages>
  <shippings>
    <shipping_width>3</shipping_width>
    <shipping_height>4</shipping_height>
    <shipping_length>5</shipping_length>
    <shipping_weight>6</shipping_weight>
  </shippings>
  <shippings>
    <shipping_width>3</shipping_width>
    <shipping_height>4</shipping_height>
    <shipping_length>9</shipping_length>
    <shipping_weight>12</shipping_weight>
  </shippings>
</packet>
```

#### El cliente recibe:

```
<?xml version="1.0"?>
<packet version="1.0.0.0">
  <status>
    <result>success</result>
    <total_shippings>1</total_shippings>
  </status>
  <shippings>
    <shipping_id_carrier>6</shipping_id_carrier>
    <shipping_id_service>8</shipping_id_service>
    <shipping_logo>http://www.url.com/media/logos/tnt.png</shipping_logo>
    <shipping_charge>6.55</shipping_charge>
    <shipping_code_carrier>TNT24</shipping_code_carrier>
    <shipping_name_carrier>Tnt</shipping_name_carrier>
    <shipping_tax>18</shipping_tax>
    <shipping_saturday>0</shipping_saturday>
  </shippings>
</packet>
```

```
    </shippings>
</packet>
```

## *Get\_shipping\_quote\_no\_mesure*

### **El cliente envía:**

```
<?xml version="1.0"?>
<packet version="1.0.0.0">
  <action>get_shipping_quote_no_mesure</action>
  <shipping_cp_from>08004</shipping_cp_from>
  <shipping_cp_to>28001</shipping_cp_to>
  <country_from>ES</country_from>
  <country_to>ES</country_to>
  <weight>5</weight>
  <packages>ES</packages>
</packet>
```

### **El cliente recibe:**

```
<?xml version="1.0"?>
<packet version="1.0.0.0">
  <status>
    <result>success</result>
    <total_shippings>1</total_shippings>
  </status>
  <shippings>
    <shipping_id_carrier>6</shipping_id_carrier>
    <shipping_id_service>8</shipping_id_service>
    <shipping_logo>http://www.url.com/media/logos/tnt.png</shipping_logo>
    <shipping_charge>6.55</shipping_charge>
    <shipping_code_carrier>TNT24</shipping_code_carrier>
    <shipping_name_carrier>Tnt</shipping_name_carrier>
    <shipping_tax>18</shipping_tax>
    <shipping_saturday>0</shipping_saturday>
  </shippings>
</packet>
```

## *Put\_order*

Este método no dejará finalizar al no ser que el cliente disponga de saldo positivo en su cuenta, de igual modo, no es aconsejable recargar el saldo para realizar pruebas ya que las expediciones se cursarán en real.

Para poder enviar una petición de pruebas, añade la etiqueta “<testmode>1</testmode>” al XML y el sistema le enviará una respuesta de pruebas.

### **El cliente envía:**

```
<?xml version="1.0"?>
<packet version="1.0.0.0">
  <action>put_order</action>
  <testmode> 0</testmode>
  <transporter_id>1</transporter_id>
  <service_id>100</service_id>
  <service_name>ASM24</service_name>
  <hour_min_collection>4</hour_min_collection>
  <product_code>ASM</product_code>
  <payment_method>bolsapaymentmethod</payment_method>
  <client_ip>0.0.0.0</client_ip>
  <comments>Comentario del envío</comments>
```



```
<code_from>ES</code_from>
<code_to>ES</code_to>
<billing>
  <firstname_billing>Nombre</firstname_billing>
  <lastname_billing>Apellido 1</lastname_billing>
  <lastname2_billing>Apellido 2</lastname2_billing>
  <company_billing>Empresa</company_billing>
  <taxvat>18</taxvat>
  <road_type_billing>street</road_type_billing>
  <street_billing>Dirección</street_billing>
  <road_number_billing>6</road_number_billing>
  <portal_billing></portal_billing>
  <floor_billing>8</floor_billing>
  <door_billing>2</door_billing>
  <postcode_billing>08206</postcode_billing>
  <city_billing>Sabadell</city_billing>
  <country_billing>ES</country_billing>
  <telephone_billing>000000000</telephone_billing>
  <fax_billing>930000000</fax_billing>
  <email_billing>email@email.com</email_billing>
</billing>
<from>
  <firstname_from>Nombre</firstname_from>
  <lastname_from>Apellido 1</lastname_from>
  <lastname2_from>Apellido 2</lastname2_from>
  <company_from>Empresa</company_from>
  <road_type_from>street</road_type_from>
  <street_from>Dirección</street_from>
  <road_number_from>6</road_number_from>
  <portal_from></portal_from>
  <floor_from>8</floor_from>
  <door_from>2</door_from>
  <postcode_from>08004</postcode_from>
  <city_from>Sabadell</city_from>
  <country_from>ES</country_from>
  <telephone_from>000000000</telephone_from>
  <fax_from>930000000</fax_from>
  <email_from>email@email.com</email_from>
  <data_collection_from>6/9/2011</data_collection_from>
  <hour_range_from>1</hour_range_from>
  <merchandise_value>66,24</merchandise_value>
  <merchandise_desc></merchandise_desc>
  <merchandise_type></merchandise_type>
</from>
<to>
  <firstname_to>Nombre</firstname_to>
  <lastname_to>Apellido 1</lastname_to>
  <lastname2_to>Apellido 2</lastname2_to>
  <company_to></company_to>
  <road_type_to>street</road_type_to>
  <street_to>Dirección</street_to>
  <road_number_to></road_number_to>
  <portal_to></portal_to>
  <floor_to></floor_to>
  <door_to></door_to>
  <postcode_to>00000</postcode_to>
  <city_to>Ciudad</city_to>
  <country_to>ES</country_to>
  <telephone_to>000000000</telephone_to>
  <fax_to></fax_to>
  <email_to>email@email.com</email_to>
</to>
<packages>
  <width>21</width>
  <height>1</height>
  <length>3</length>
  <weight>4</weight>
  <charge>7.44</charge>
```

```
    <bultos>1</bultos>
    <taxes_percentage>18</taxes_percentage>
  </packages>
</packet>
```

**El cliente recibe:**

```
<?xml version="1.0"?>
<packet version="1.0.0.0">
  <status>
    <result>success</result>
    <order>1</order>
    <proforma_pdf>http://www.clickline.com/media/pdf/order/order_20
11-09-08_17-00-58.pdf</proforma_pdf>
    <consignment_pdf>http://www.clickline.com/media/pdf/order/order
_2011-09-08_17-00-58.pdf</consignment_pdf>
  </status>
</packet>
```