# Pack&Ship WS

The Pack&Ship WS is mainly meant for small businesses. At this stage, it can only be combined with the usage of the kiala.xx shipment service, which features

* prepaid payment for the shipments via:
  + online prepayment using Ogone and PayPal (ES)
  + payment at the Kiala Point upon drop-off (BE/NL)
* generation of a PDF document with the to-be-printed labels that need to be glued on the corresponding parcel.

|  |  |
| --- | --- |
| C:\Users\Administrator\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\ZHLKJ2IK\MC900433883[1].png | If the parcel exceeds the maximum size and weight allowed by Kiala, we will not be able to handle your shipments! To know the exact restrictions, please contact Kiala. |

#### Obtaining a Logon

In order to use Pack&Ship WS it is required to open a profile/account on the Kiala.xx web site that corresponds with the country from which you will dispatch parcels. Registration can be started from the *“Send a Parcel”* menu on the front page, next to the Home button.

Once you have registered, go to your profile page where you will find a section *“Sender Profiles”.* If you haven’t made one, create one. For each sender profile you can generate the necessary credentials for the web service by clicking *“click here if you want to start using this sender profile via an e-commerce plugin”*. Result :



The sender id and hash key will serve as credentials in the web service.

#### Prerequisite

Each order needs a unique reference number, i.e. an Order ID.

#### Submitting Shipments

When calling the web service, it is mandatory to include your credentials. This is how you need to calculate your credentials:

1. Concatenate the **the sender id**  and **order reference**
2. Compute a [HMAC SHA-512 code](http://en.wikipedia.org/wiki/Hash-based_message_authentication_code) of that concatenated string. For this you need the **Hash Key**. This is a built-in algorithm for Java and PHP.
3. The generated hash should be encoded with [Base-64](http://en.wikipedia.org/wiki/Base64) after that.

Note : *PHP by default converts the output of the HMAC function to hexadecimal notation, while Java provides the raw binary (which is why we encode it as Base64, otherwise it's not readable), so setting the raw\_binary option on the hash\_hmac method should produce the correct result:*

*base64\_encode(hash\_hmac("sha512", "helloworld", "mykey", true))*

Documentation of how this can be done in PHP can be found here: http://be2.php.net/hash\_hmac.

A small tool can be provided by Kiala to check if you properly concatenated and hashed the data :



By means of the SOAP message, send a request to Kiala with your credentials and all the shipping information related to the order. The Kiala Point ID (retrieved from Locate&Select) is a key element.

In order to speed up the development process, you will need to use a SOAP library to call the web service.

#### Description of the Web Service

Please use the WSDL and XSD provided below for development.

Once the integration has been tested and approved by Kiala, you should replace <https://kportal-shippers-test.kiala.com/kportal/soap/OrderService?wsdl> with <https://kportal.kiala.com/kportal-1.0/soap/OrderService?wsdl> to allow it to work in production.

|  |  |
| --- | --- |
| **WSDL** | <?xml version='1.0' encoding='UTF-8'?>  <definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy" xmlns:wsp1\_2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://www.kiala.com/schemas/psws/1.0" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://www.kiala.com/schemas/psws/1.0" name="OrderService">  <types>  <xsd:schema>  <xsd:import namespace="http://www.kiala.com/schemas/psws/1.0" schemaLocation="http://kportal-shippers-test.kiala.com:80/kportal/soap/OrderService?xsd=1"/>  </xsd:schema>  </types>  <message name="createOrder">  <part name="createOrder" element="tns:createOrder"/>  </message>  <message name="createOrderResponse">  <part name="createOrderResponse" element="tns:createOrderResponse"/>  </message>  <message name="OrderServiceException">  <part name="fault" element="tns:orderFault"/>  </message>  <portType name="OrderService">  <operation name="createOrder">  <input wsam:Action="http://www.kiala.com/schemas/psws/1.0/OrderService/createOrderRequest" message="tns:createOrder"/>  <output wsam:Action="http://www.kiala.com/schemas/psws/1.0/OrderService/createOrderResponse" message="tns:createOrderResponse"/>  <fault message="tns:OrderServiceException" name="OrderServiceException" wsam:Action="http://www.kiala.com/schemas/psws/1.0/OrderService/createOrder/Fault/OrderServiceException"/>  </operation>  </portType>  <binding name="OrderServicePortBinding" type="tns:OrderService">  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>  <operation name="createOrder">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  <fault name="OrderServiceException">  <soap:fault name="OrderServiceException" use="literal"/>  </fault>  </operation>  </binding>  <service name="OrderService">  <port name="OrderServicePort" binding="tns:OrderServicePortBinding">  <soap:address location="http://kportal-shippers-test.kiala.com:80/kportal/soap/OrderService"/>  </port>  </service>  </definitions> |
| **XSD** | <?xml version='1.0' encoding='UTF-8'?>  <xs:schema xmlns:tns="http://www.kiala.com/schemas/psws/1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema" version="1.0" targetNamespace="http://www.kiala.com/schemas/psws/1.0">  <xs:element name="createOrder" nillable="true" type="tns:OrderRequest"/>  <xs:element name="createOrderResponse" nillable="true" type="tns:OrderConfirmation"/>  <xs:element name="orderFault" nillable="true" type="tns:OrderFault"/>  <xs:complexType name="OrderRequest">  <xs:sequence>  <xs:element name="identification" type="tns:Identification"/>  <xs:element name="delivery" type="tns:Delivery"/>  <xs:element name="parcel" type="tns:Parcel"/>  <xs:element name="sender" type="tns:Sender" minOccurs="0"/>  <xs:element name="receiver" type="tns:Receiver"/>  </xs:sequence>  <xs:attribute name="reference" type="xs:string" use="required"/>  </xs:complexType>  <xs:complexType name="Identification">  <xs:sequence>  <xs:element name="sender" type="xs:string"/>  <xs:element name="hash" type="xs:string"/>  <xs:element name="originator" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="Delivery">  <xs:sequence>  <xs:element name="from" type="tns:Node"/>  <xs:element name="to" type="tns:Node"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="Node">  <xs:sequence/>  <xs:attribute name="country" type="xs:string" use="required"/>  <xs:attribute name="node" type="xs:string"/>  </xs:complexType>  <xs:complexType name="Parcel">  <xs:sequence>  <xs:element name="description" type="xs:string"/>  <xs:element name="weight" type="xs:decimal"/>  <xs:element name="orderNumber" type="xs:string"/>  <xs:element name="orderDate" type="xs:date"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="Sender">  <xs:sequence>  <xs:element name="firstName" type="xs:string" minOccurs="0"/>  <xs:element name="surname" type="xs:string"/>  <xs:element name="address" type="tns:Address"/>  <xs:element name="email" type="xs:string"/>  <xs:element name="language" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="Address">  <xs:sequence>  <xs:element name="line1" type="xs:string" minOccurs="0"/>  <xs:element name="line2" type="xs:string" minOccurs="0"/>  <xs:element name="postalCode" type="xs:string" minOccurs="0"/>  <xs:element name="city" type="xs:string" minOccurs="0"/>  <xs:element name="country" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="Receiver">  <xs:sequence>  <xs:element name="firstName" type="xs:string"/>  <xs:element name="surname" type="xs:string"/>  <xs:element name="address" type="tns:Address"/>  <xs:element name="email" type="xs:string"/>  <xs:element name="language" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="OrderConfirmation">  <xs:sequence>  <xs:element name="trackingNumber" type="xs:string"/>  <xs:element name="parcelGuid" type="xs:string" minOccurs="0"/>  </xs:sequence>  <xs:attribute name="reference" type="xs:string"/>  </xs:complexType>  <xs:complexType name="OrderFault">  <xs:sequence>  <xs:element name="faultCode" type="tns:OrderFaultCode" minOccurs="0"/>  <xs:element name="message" type="xs:string" minOccurs="0"/>  <xs:element name="dataError" type="tns:DataError" nillable="true" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="DataError">  <xs:sequence>  <xs:element name="name" type="xs:string"/>  <xs:element name="errorCode" type="tns:ErrorCode"/>  <xs:element name="message" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:simpleType name="OrderFaultCode">  <xs:restriction base="xs:string">  <xs:enumeration value="AUTHENTICATION\_ERROR"/>  <xs:enumeration value="INVALID\_REQUEST"/>  <xs:enumeration value="VALIDATION\_ERROR"/>  <xs:enumeration value="SYSTEM\_ERROR"/>  </xs:restriction>  </xs:simpleType>  <xs:simpleType name="ErrorCode">  <xs:restriction base="xs:string">  <xs:enumeration value="MISSING\_REQUIRED\_ELEMENT"/>  <xs:enumeration value="PATTERN"/>  </xs:restriction>  </xs:simpleType>  </xs:schema> |

#### Description of Parameters

|  |  |
| --- | --- |
| **Identification** | |
| **Element** | | **Mandatory** | | **Type** | | | **Description** | | |
| sender | | Yes | | String (32) | | | ID of your sender profile, which you receive from the Kiala web site when registering as a user of the service. | | |
| reference | | Yes | | String (32) | | | Unique reference number for the order. Usually this is the “order number”, but if for example an order contains multiple physical parcels you will have to provide a unique reference for each parcel. | | |
| hash | | Yes | | String(128) | | | hash code of your Kiala Sender ID and reference as explained earlier in this document. | | |
| originator | | No | | String(32) | | | If applicable, the identifier of the e-commerce platform that’s hosting your site. Kiala has a list of e-commerce platforms it works with and can provide you with the correct originator ID to use.  Contact [kambrogi@kiala.com](mailto:kambrogi@kiala.com) to request your unique originator ID. | | |
| **Delivery** |
| **Element** | **Attribute** | | **Mandatory** | | **Type** | **Description** | | | |
| from |  | | Yes | | String (2) | The country from where the parcel will be sent. | | | |
| To |  | | Yes | | String(2) | The destination country. | | | |
| from | country | | Yes | | String(2) | First two letters of the origin country. | | | |
| from | node | | Yes | | String(7) | The locator ID of the destination Kiala Point. | | | |
| to | country | | Yes | | String(2) | First two letters of the destination country. | | | |
| to | node | | Yes | | String(7) | The locator ID of the destination Kiala Point. | | | |

There are some restrictions on the combinations of From and To values. Please contact Kiala to get the latest allowed combinations.

|  |
| --- |
| **Parcel** |
| **Element** | **Mandatory** | **Type** | | **Description** | | |
| description | No | String(300) | | Optional description of the parcel. | | |
| weight | Yes | Decimal format #0.000 | | The weight of the parcel in kg | | |
| orderNumber | No | String (32) | | Order number | | |
| orderDate |  | Date format yyyy-MM-dd | | Order date | | |
| **Receiver** | |
| **Element** | **Mandatory** | **Type** | **Description** | | | |
| firstName | No | String(100) | First name of the customer | | | |
| surname | Yes | String(100) | Surname of the customer | | | |
| email | Yes | String(320) | E-mail address the customer | | | |
| language | Yes | String(2) | Language of the customer | | | |
| line1 | No | String(200) | First line of the customer address | | | |
| line2 | No | String(200) | Second line of the customer address | | | |
| postalCode | Yes | String(20) | Postal code of the customer address | | | |
| city | No | String(100) | City of the customer address | | | |
| country | Yes | String(2) | Country of the customer address | | | |

The following language-country combination are allowed: es-ES, fr-FR, fr-BE, nl-BE, nl-NL

#### Example of a Request/Response

|  |  |
| --- | --- |
| **Request** | <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="http://www.kiala.com/schemas/psws/1.0">  <SOAP-ENV:Body>  <ns1:createOrder reference="12345">  <identification>  <sender>GiQSxBsSUoifJPnZ8GP6Wj5VFdzhYxAL</sender>  <hash>+qo7H7iB0drqyaWWxp … 61vTOfCAqzoZNBLEENbxg==</hash>  <originator>magento</originator>  </identification>  <delivery>  <from country="BE" node =""/>  <to country="BE" node="BE20183"/>  </delivery>  <parcel>  <description/>  <weight>4.536</weight>  <orderNumber>12345</orderNumber>  <orderDate>2014-05-30</orderDate>  </parcel>  <receiver>  <firstName>Jos</firstName>  <surname>SMOS</surname>  <address>  <line1>Paapsemlaan 22</line1>  <line2></line2>  <postalCode>1070</postalCode>  <city>Anderlecht</city>  <country>BE</country>  </address>  <email>jhanselaer@mailinator.com</email>  <language>nl</language>  </receiver>  </ns1:createOrder>  </SOAP-ENV:Body>  </SOAP-ENV:Envelope> |
| **Response** | <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">  <S:Body>  <ns2:createOrderResponse reference="12345" xmlns:ns2="http://www.kiala.com/schemas/psws/1.0">  <trackingNumber>0002TGZ</trackingNumber>  </ns2:createOrderResponse>  </S:Body>  </S:Envelope> |

#### Exceptions

In a number of cases the order can be refused and you will get an <error> element back instead of a <trackingNumber>. These cases are:

* + The values in the <identification> part don’t match up.
  + Invalid from/to combination.
  + The parcel weight is not specified or is lower than 0.000 kg
  + Missing receiver surname, e-mail, language, postal code or country.

The <error> element has an attribute named “code” and a list of all possible error codes can be obtained from Kiala upon request.