

## GLS ONLINE API

GLS online SOAP webservice for printing services

### WSDL Description

**Style:** rpc

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** printlabel

**Input:** printlabelRequest (soap:body, use = encoded)

<b>*username</b> type <i>string</i>	- user name – request it from GLS
<b>*password</b> type <i>string</i>	- password – request it from GLS
<b>*senderid</b> type <i>string</i>	- GLS client number – request it from GLS
<b>*sender_name</b> type <i>string</i>	- sender name
<b>*sender_address</b> type <i>string</i>	- sender address
<b>*sender_city</b> type <i>string</i>	- sender city
<b>*sender_zipcode</b> type <i>string</i>	- sender zip code
<b>*sender_country</b> type <i>string</i>	- sender country code or country name
<b>sender_contact</b> type <i>string</i>	- sender contact person
<b>sender_phone</b> type <i>string</i>	- sender phone
<b>sender_email</b> type <i>string</i>	- sender email address
<b>*consig_name</b> type <i>string</i>	- consignee name
<b>*consig_address</b> type <i>string</i>	- consignee address
<b>*consig_city</b> type <i>string</i>	- consignee city
<b>*consig_zipcode</b> type <i>string</i>	- consignee zip code
<b>*consig_country</b> type <i>string</i>	- consignee country code or country name
<b>consig_contact</b> type <i>string</i>	- consignee contact person
<b>consig_phone</b> type <i>string</i>	- consignee phone or SMS number for services
<b>consig_email</b> type <i>string</i>	- consignee email address – also used for services
<b>*pcount</b> type <i>int</i>	- count of the parcels / labels to print
<b>*pickupdate</b> type <i>string</i>	- pickup date in format yyyy-MM-dd
<b>content</b> type <i>string</i>	- content of the parcel – info printed on label
<b>clientref</b> type <i>string</i>	- client reference
<b>codamount</b> type <i>decimal</i>	- COD amount

**coderef** type *string* - COD reference – used if COD amount is set

**services** type *svcDataArray* - array of type *svcData* - array of services

- **code** type *string* - 3 letter service code, please see list of services in Appendix A
- **info** type *string* - parameter for service

**\*printertemplate** type *string* - type of the printer – list in Appendix B

**\*printit** type *Boolean* - true, if label has to be printed, false if stored in the list

**\*timestamp** type *string* - timestamp of the request in format yyyyMMddHHmmss

**\*hash** type *string* - hash code of the request – Appendix C

- \*Mandatory fields marked with asterisk
- do not pass null for optional values, empty string is required
  - as **services** pass at least an empty array
- to avoid spamming of the service, same request can be sent max. 5 times within 5 minutes. In case it is required to print more labels with same data, use **pcount** variable to pass count

### Output:

printlabelResponse (soap:body, use = encoded)

**return** type *printlabel\_result*

- **pcls** type *ArrayOfString* - array of type *string* – return list of the parcel numbers printed
- **pdfdata** type *string* - pdf data to print or display pdf. Encoded with base64
- **successfull** type *Boolean* - if false, *errcode* and *errdesc* will be set
- **errcode** type *int* - numeric error code, please list in Appendix D
- **errdesc** type *string* - error description

- **pcls** and **pdfdata** will be returned only if **printit** is *true* and **successfull** is *true*
- sample code for NuSOAP client in Appendix E
- sample code for handling pdfdata from result in c# in Appendix F

## Appendix A

### List of services

Service code	Service name	Parameter
T12	Express Service	
PSS	Pick&Ship Service	Pickup data in format yyyy-MM-dd
PRS	Pick&Return Service	Pickup data in format yyyy-MM-dd
XS	Exchange Service	<ul style="list-style-type: none"> <li>it is necessary to print second label for return parcel</li> </ul>
SZL	DocumentReturn Service	Document number – string, max. 15 char
INS	DeclaredValueInsurance Service	Value of the parcel
SBS	Standby Service	
DDS	DayDefinite Service	Date of delivery in format yyyy-MM-dd
SDS	ScheduledDelivery Service	Time range of delivery in format HH:mm-HH:mm
SAT	Saturday Service	
AOS	AddresseeOnly Service	Name of the recipient / contact person can be used
24H	Guaranteed24 Service	
EXW	ExWorks Service	
SM1	SMS Service	SMS Phone number and SMS text in format "phone nr in international format sms text". Variable #ParcelNr# can be used for parcel number.
SM2	PreAdvice Service	SMS Phone number in international format
CS1	Contact Service	Recipient phone number / contact phone number can be used
TGS	ThinkGreen Service	
FDS	FlexDelivery Service	Email address
FSS	FlexDeivery SMS Service	SMS phone number in international format <ul style="list-style-type: none"> <li>not available without FDS</li> </ul>
PSD	ShopDelivery Service	DropOffPoint ID
DPV	DeclaredParcelValue	Used in case of HR, 20xxx zip codes, to declare value of the parcel

- not all services are available in each country and each area
- in case of service PSS/PRS label is not printed, just service ordered

## Appendix B

### Types of the printer template

Type	Description
A5	A5 format, blank label
A5_PP	A5 format, preprinted label
A5_ONA4	A5 format, printed on A4
A4_2x2	A4 format, 4 labels on layout 2x2
A4_4x1	A4 format, 4 labels on layout 4x1

## Appendix C

### Hash code

- hash code is used to verify integrity of the sent request
- **calculation:** join following fields content into string and use sha1() algorithm:  
*username, password, senderid, sender\_name, sender\_address, sender\_city, sender\_zipcode, sender\_country, sender\_contact, sender\_phone, sender\_email, consig\_name, consig\_address, consig\_city, consig\_zipcode, consig\_country, consig\_contact, consig\_phone, consig\_email, pcount, pickupdate, content, clientref, codamout, coderef*  
(each input field without *services, printertemplate, printit, timestamp, hashcode*)

#### Sample PHP function:

```
functiongetHash($data) {  
    $hashBase = '';  
    foreach($data as $key => $value) {  
        if ($key != 'services'  
        && $key != 'hash'  
        && $key != 'timestamp'  
        && $key != 'printit'  
        && $key != 'printertemplate') {  
            $hashBase .= $value;  
        }  
    }  
  
    return sha1($hashBase);  
}
```

#### Sample c# function:

```
usingSystem.Security.Cryptography;  
privatestring getHash()  
{  
    stringexpression =  
        string.Format("{0}{1}{2}{3}{4}{5}{6}{7}{8}{9}{10}{11}{12}{13}{14}{15}{16}  
        {17}{18}{19}{20}{21}{22}{23}{24}", user, pwd, senderid, senderName,  
        senderAddress, senderCity, senderZipcode, senderCountry, senderContact,  
        senderPhone, senderEmail, consigName, consigAddress, consigCity,  
        consigZipcode, consigCountry, consigContact, consigPhone, consigEmail,  
        pcount, pickupdate, content, clientref, codamount, coderef));  
  
    SHA1 sha1 =SHA1.Create();  
    stringhash =sha1.ComputeHash(System.Text.Encoding.UTF8.GetBytes(expression));  
    return BitConverter.ToString(hash).Replace("-", "").ToLower();  
}
```

## Appendix D

### Error codes

Type	Description
0	OK
1	Authentication failed
2	Invalid hash
3	Unable to store data please try again later
4	Invalid printer template please check documentation
5	Missing parameters:
6	Invalid timestamp
7	Invalid sender country
8	Invalid consignee country
9	Invalid sender zipcode
10	Invalid consignee zipcode
11	Invalid pickup date
12	Parcel count must be between 1 and 99
13	Missing contact person for export parcel
14	COD is not allowed to this export country
15	Max value for COD is:
16	Invalid COD rounding the smallest fraction is
17	Invalid service code(s):
18	Invalid service combination(s):
19	Service(s) not available in pickup country:
20	Service(s) not available between sender and consignee country:
21	Service(s) not available on consignee country/zipcode:
22	Invalid / missing parameters for service(s):
23	FSS service is valid only with ordered FDS service
24	For parcels to HR with zipcode 20xxx please use DPV parameter info to send declared parcel value for parcel
25	Same request sent 5 times within last 5 minutes!

## Appendix E

### NuSOAP client sample code in PHP

```
<?php
require_once('lib/nusoap.php');
$_HTTP = !empty($_SERVER['HTTPS']) ? 'https://' : 'http://';
$wsdl_path = $_HTTP.$_SERVER['HTTP_HOST'].'/webservices/soap_server.php?wsdl';
$client = new nusoap_client($wsdl_path,'wsdl');
$client->soap_defencoding = 'UTF-8';
$client->decode_utf8 = false;

$in = array(
    'username' => 'username',
    'password' => 'password',
    'senderid' => '0000000000',
    'sender_name' => 'sendername',
    'sender_address' => 'senderaddress',
    'sender_city' => 'sendercity',
    'sender_zipcode' => '2351',
    'sender_country' => 'HU',
    'sender_contact' => 'sendercontact',
    'sender_phone' => 'senderphone',
    'sender_email' => 'valaki@valaki.sk',
    'consig_name' => 'consigname',
    'consig_address' => 'consigaddress',
    'consig_city' => 'consigcity',
    'consig_zipcode' => '2351',
    'consig_country' => 'HU',
    'consig_contact' => 'consigcontact',
    'consig_phone' => 'consigphone',
    'consig_email' => 'valaki@valaki.sk',
    'pcount' => 1,
    'pickupdate' => '2014-01-29',
    'content' => 'tartalom',
    'clientref' => 'clientref',
    'codamount' => '1526',
    'codref' => 'codref',
    'services' => array(),
    'printertemplate' => 'A4_2x2',
    'printit' => true,
    'timestamp' => '20140129150000',
    'hash' => 'xsd:string' );

$in['hash'] = getHash($in);

$return = $client->call('printlabel', $in);
if ($return) {
    if ($return['successfull']) {
        header('Content-type: application/pdf');
        die(base64_decode($return['pdfdata']));
    } else {
        var_dump($return);
    }
}
```



```
functiongetHash($data) {  
    $hashBase = '';  
    foreach($data as $key => $value) {  
        if ($key != 'services'  
            && $key != 'hash'  
            && $key != 'timestamp'  
            && $key != 'printit'  
            && $key != 'printertemplate') {  
                $hashBase .= $value;  
            }  
        }  
    }  
  
    return sha1($hashBase);  
}  
  
?>
```

## Appendix F

### Handling pdfdata from result in c#

```
byte[] data = Convert.FromBase64String(result.pdfdata);  
System.IO.File.WriteAllBytes(@"c:\temp\label.pdf", data);
```