G. Kolly SA
Rte de La Gruyère 88
1724 Le Mouret

www.kolly.com

Phone +41 26 413 90 00

WEBSERVICE KOLLYgram DESCRIPTION

Date	Author	Revision	Comments
01.05.2012	SM	1.0	Creation of the document
08.05.2012	SM	1.1	Small modifications

Table of Contents

Introduction

Before to implement a service that will connect to the KOLLYgram webservice, we strongly advise you to read and understand this documentation.

You need to understand the relation between the tables in our database; otherwise, you will not use the functions properly.

Recommended Workshop

It's strongly recommended if you want to do more than just retrieve raw measures from the webservice that you follow a workshop of 3 to 6 hours (depending on the content) by G. Kolly SA.

General information about the fields

You must never fill fields with a Null value.

String fields have to receive at least an empty String.

The tables of the database

Users

The users are organized in a hierarchical manner, for example: the trucking Company "Alphatrans" can have 3 customers : NewYork, Washington, TheBigCompany. Under each of these customers there can be sub-customers. For example for NewYork : Manhattan, Bronx, Jersey, …

These customers can be either final customers or cities. The top parent is always a trucking company.

Chips

Each RFID chip (or bar code) has its own number.

ChipTypes

Each chip can be one of the following types:

1, 'Body Deister'
2, 'Chipnest Deister'
3, 'Datamars'
4, 'TIRIS'
5, 'Deister 125 Khz'
6, 'Barcode'
7, 'virtual Chip'

You can retrieve the list of chip types by using the function GetChipTypes.

GarbageTypes

Here is the list of the garbage types (please note that the garbage type id has nothing to do with the product number that can be found in the measure Record).

id	type_de_dechet	code	type_de_dechet_de	type_de_dechet_fr
1	Gewerbekehricht / Déchets industriels	DI	Gewerbekehricht	Déchets industriels
2	Kehricht / Ordures ménagères	OM	Kehricht	Ordures ménagères
3	Papier-Karton / Papier-Carton	PC	Papier-Karton	Papier-Carton
4	(unbekannt / inconnu)	?	(unbekannt)	(inconnu)
5	Plastic / Plastique	PL	Plastic	Plastique
6	Holz / Bois	BO	Holz	Bois
7	Papier-Karton + Plastic / Papier-Carton + Plastiques	PCPL	Papier-Karton + Plastic	Papier-Carton + Plastiques
8	Grüngut / Déchets verts	GR	Grüngut	Déchets verts
9	Frei / Libre	FREE	Frei	Libre
10	Grünglas / Verre vert	GG	Grünglas	Verre vert
11	Weissglas / Verre blanc	WG	Weissglas	Verre blanc
12	Braunglas / Verre brun	BG	Braunglas	Verre brun
13	Büchse/Alu / Conserves/Alu	BA	Büchse/Alu	Conserve / Alu
14	PET	PET	PET	PET

You can use GetGarbageTypes to get the latest list for these garbage types.

Containers

A container is materialized in the system by the creation of a link between these elements:

- A chip
- A customer =Id of the user who manages the container
- Place = The address's Id of the Address where is located the container
- From+To = The timespan of the validity of this link

ContainerTypes

Here is the list of the container types:

id	type_container	capacite_lt
1	Container 800 l.	800
2	Container 400 l.	400
3	Container 600 l.	600
4	Container 120 l.	120
5	Container 240 l.	240
6	Container 140 l.	140
7	Container 770 l.	770
8	Container 80 l.	80
9	Container 360 l.	360
10	Unbekannt/Inconnu	0
11	Container 1700 l.	1700
12	Container 1100 l.	1100
13	Container 2.5 m3	2500
14	Container 3.5 m3	3500
15	Container 4.5 m3	4500
16	Container 5 m3	5000
17	Toilette mobile	0
18	Benne de chantier	0
0	(Indéfini/Aucun)	0

You can retrieve the latest container types list by using the GetContainerTypes.

Addresses

Addresses contain

Vehicles

Each trucker can have many vehicles. Each vehicle has its own ID.

Webclients

To access a webservice, you need a password. The webcode contains both account and password.
Each webclient can access one to many trucker data (usually one).

The Webservice

The webservice is based on SOAP 1.1/1.2.

Demo video

Follow this link to see an 8 minutes Demo Video about how to create a simple program to download measures with the webservice. The sample demo is made with Visual Studio 2008, but the webservice is compatible with the major compatible programming languages.

Link to the demo video:
http://www.burningbox.com/divers/kollygramonlinedemos/webservicekollygramdemo1.html

URLs

The url of the webservice is:

https://data.kollygram.ch/ws/kollygramonline.asmx

The WSDL description can be found under:

https://data.kollygram.ch/ws/kollygramonline.asmx?WSDL

Functions

In most of the functions, to be authorized to use the function, you have to send the trucker id (transporteur id) and the webcode (CodeWebClient) that you have receveid.

There are more functions listed in the webservice, but in this documentation you will find only the most useful ones.

getMesure

```
Public Function getMesure(ByVal transporteurid As Long, ByVal vehic_id As Long, ByVal
CodeWebClient As String, ByVal mesure_id As Long) As mesure
```

Use this function to retrieve a measure.

Vehic_id correspond to the vehicle id. Mesure_id is the id of the measure wanted, not the reference number (Numéro de Quittance/Quittungsnummer).

The answer given will have these fields (the most important fields are described; you can ignore the supplementary fields for the usual applications):

Field	Type	Comment

id	**unsignedLong**	**Measure id**
vehicule	**unsignedLong**	**The vehicle id**
date_et_heure	dateTime	The date when the measure has been received by the server
data	string	Brute data, you don't need to use this field
numero	unsignedInt	you don't need to use this field
cs	unsignedInt	Deprecated field, you don't need to use it
cs_vhc	unsignedInt	Deprecated field, you don't need to use it
poids	**decimal**	**The weight in KG (or the other unit used by your system if specific)**
produit	**unsignedInt**	**The product number**
client	**int**	**The customer number (not the user id)**
cle	**string**	**The rfid chip key (number, not the id)**
cle2	string	The rfid chip key (number, not the id) of the second chip read (if there are two cheap to read on the antenna (specific to your hardware).
mode	unsignedInt	The type of the balance. 0=Standard, 1=Crane
debut	**dateTime**	**The start of the measure**
fin	**dateTime**	**The end of the measure**
km_debut	decimal	The km counter at the start of the measure (km counted from the last start of the balance hardware)
km_fin	decimal	The km counter at the end of the measure
quittance	**unsignedInt**	**The receipt number (Quittance/Quittungsnummer). This is very important to retrieve with the weight.**
nord	double	Swiss coordinates. You don't need to use this. Sometimes empty, even if the truck is equipped with a gps.
est	double	Swiss coordinates. You don't need to use this. Sometimes empty, even if the truck is equipped with a gps.
lo	**Double**	**Longitude. If the truck has a GPS and only if the location was acquired at the moment of the measure.**
la	**double**	**Latitude. If the truck has a GPS and only if the location was acquired at the moment of the measure.**
date_gps	dateTime	The date given by the GPS at the moment of the measure. Not always provided.
intervalle_secondes	decimal	You don't need to use this field.
type_	string	You don't need to use this field.

changements	string	List of changes made to this measure.
flag	unsignedShort	Information about the measure, see Flag in this documentation.
efface	**boolean**	**True is this measure has been deleted.**

getMesures

```
Public Function getMesures(ByVal transporteur_id As Long, ByVal vehic_id As Long,
ByVal CodeWebClient As String, ByVal from_ As Date, ByVal to_ As Date) As mesure()
```

Use this method to download measures. It is possible to set the vehicle id to 0, so all measures of the 'Transporteur' will be then downloaded (if too many measure are downloaded at once, there is a time out risk, so download measures day after day).

“From_” and “to_” filter the measure by the “start” (field “debut” datetime) of a measure.

getMesuresByQuittanceNr

```
Public Function getMesuresByQuittanceNr(ByVal transporteur_id As Long, ByVal vehic_id
As UInteger, _
                                        ByVal CodeWebClient As String, _
                                        ByVal from_ As Date, ByVal to_ As Date, _
                                        ByVal quittanceNo As Integer) As mesure()
```

Use this method to find measures by “Quittance” (QuittungsNr).

Vehicle id is mandatory.

“From_” and “to_” filter the measure by the “start” (field “debut” datetime) of a measure.

getMesuresByReceptionDate

```
Public Function getMesuresByReceptionDate(ByVal transporteur_id As Long, ByVal
vehicle_id As Long, ByVal CodeWebClient As String, _
                                  ByVal FromReceptionDate As Date, ByVal ToReceptionDate
As Date) As mesure()
```

With this function, the “debut” field (start of a measure) is ignored. But the “date_et_heure” field is taken for the filter, so the time when the server has received measures.

Use this method to download measures. The vehicle_id can be 0, if so: all measure from all vehicles will be sent. Time will be ignored in the “FromReceptionDate” and “ToReceptionDate” parameters, only the date part will be taken into account. Maximum 7 days of measures are sent.

AddChild2

```
Public Function AddChild2(ByVal transporteurid As Long, ByVal CodeWebClient As String, ByVal customer_number As String, ByVal customer_name As String, ByVal parentid As Long, ByVal type_utilisateur As String) As Long
```

Use this function to add a user in the hierarchy.

Provide the customer number and name; The parent id, therefore the “transporteur” id if you want to put the customer on top of the hierarchy. Otherwise the id that is the parent of this new customer. You have also to specify the type of the new customer: “k” for customer (Kunde/Client), “g” for a city (Gemeinde/Commune).

You will receive the ID of the newly customer created (customer, not an address). Please save this ID in your own database for further reference.

EditCustomer

```
Public Function EditCustomer(ByVal codeWebClient As String, ByVal transporteurid As Long, ByVal clientid As Long, ByVal name As String, number As String) As Boolean
```

Use this function to change the name or the number (not the id) of a customer. The number of the customer correspond of the number typed by the driver. It can be for example a ZIP (NPA/PLZ) of a city (Gemeinde/Commune).

Response : True if successful.

RenameCustomer

Rename customer is the same function, but you can only rename the name (not the number).

CanDeleteChild

```
Public Function CanDeleteChild(ByVal transporteurid As Long, ByVal CodeWebClient As String, ByVal child_id As Long) As Boolean
```

Use this function to detect if you can delete a customer in the hierarchy or not. The child id to provide is the user id of the customer to delete. The answer will be a Boolean; if true you can delete the customer. If no, it means that something is linked to this customer and it can’t be deleted.

DeleteChild

```
   Public Function DeleteChild(ByVal transporteurid As Long, ByVal CodeWebClient As String, ByVal child_id As Long) As Boolean
```

This function deletes a customer in the database. Child_id is the customer id to be deleted. True is sent back if successful.

ChangeParent

```
Public Function ChangeParent(ByVal transporteurid As Long, ByVal CodeWebClient As
String, ByVal child_id As Long, ByVal new_parent_id As Long) As Boolean
```

Use this function to move a customer from under one parent customer to another one. You have to provide the customer id of the child to be moved (child_id) and the customer id of the new parent (new_parent_id). The answer will be a Boolean; true if the operation has been successful.

getAllChildren

```
Public Function getAllChildren(ByVal transporteurid As Long, ByVal CodeWebClient As
String) As IdEtNom()
```

Retrieve all children under the carrier 'Transporteur_id'

Objects with only id and name are sent back (All children in the hierarchy, so it's recursive).

UpdateAddress

```
Public Function UpdateAddress(ByVal codeWebClient As String, ByVal transporteurid As
Long, _
                                        ByVal id As Long, _
                                        ByVal under_this_noclient As Long, _
                                        ByVal nom As String, _
                                        ByVal street As String, _
                                        ByVal street_number As String, _
                                        ByVal address_complement As String, _
                                        ByVal zip As String, _
                                        ByVal city As String, _
                                        ByVal lang As String, _
                                        ByVal email As String, _
                                        ByVal fax As String, _
                                        ByVal tel As String, _
                                        ByVal contact_person As String, _
                                        ByVal remark As String, _
                                        ByVal old_customer_number As String, _
                                        ByVal divers1 As String, _
                                        ByVal divers2 As String, _
                                        ByVal divers3 As String _
                                        ) As AddressResult
```

Use this function to create or update an address.

Please indicate 0 as the id to create an address. You will then retrieve in the response the new id of the address created. You will use then this id in the future to update the same address.

The result is the address or an error in the field “error_” (string).

The fields for an address are:

Field	Type (max len)	Comment
Id	**unsignedLong**	**Address ID**
client	**unsignedLong**	**The customer id where this address is placed (under which authority).**
numero	**String 10**	**The customer number (your own numbering, or for example ZIP number).**
complement	String 255	Address complement (Adresse Zusatz)
nom	**String 255**	**The name of the customer (company name, city name, customer name, …)**
fax	String 100	Fax number
email	String 100	Email
categorie_rabais	unsignedInt	The rebate category for this customer (leave 0)
tel	String 100	Phone number
personne_contact	String 255	Contact person
remarque	String 1000	Remarks and comments
ancien_numero_client	String 45	Old customer number (you can use this field to put your own ID)
rue_	String 45	The name of the street. Please put longer address information into the field “complement”.
npa	**String 10**	**ZIP (NPA/PLZ)**
localite	**String 45**	**City (Ort/Lieu)**
lang	**String 2**	**The customer language “fr” or “de”**
divers1	String 45	Additional custom field
divers2	String 45	Additional custom field
divers3	String 45	Additional custom field
categorie	String 1	Address category : “o” = place (ort), “r” = invoicing address (rechnung), b=both (beide). This is just indicative, it can also contain a “ “ (space). This field is not always provided / needed.

CreateNewPuce2

```
Public Function CreateNewPuce2(ByVal transporteurid As Long, ByVal codeWebClient As
String, ByVal new_chip_nr As String, ByVal new_chip_type As Integer,
ByVal chip_managed_by_client_id As Long) As ChipResponse
```

This function allows you to create a new chip. Fields to provide are: The chip number (new_chip_nr), the new_chip_type_id (see table under the chapter “ChipTypes”), and the id of the customer who can manage the chip (chip_managed_by_client_id).

The response will be the chip created (xml object) or the error (field “error_”).

Please note that you have to add “new_chip_nr” of 16 chars length. Please fill strings with 0 on the left. For exemple : Chip NR 123456, must be added as 0000000000123456.

GetPuceAccordingChipID

```
Public Function GetPuceAccordingChipID(ByVal codeWebClient As String, ByVal
transporteurid As Long, ByVal chip_id As Long) As puce
```

Returns a CHIP by using the database ID as search criteria.

The chip object contains the following fields :

Field	Type	Comment
Id	**unsignedLong**	**Chip ID**
puce_no	string	**The chip number**
type_de_puce	unsignedInt	**Chip type id**
client	unsignedLong	**id of the customer where this chip attached (this customer (and his parents) will be able to modify the chip).**
last_location	string	Full address of the last location of the chip
last_location_nom	string	The name of the last location of the chip
last_location_adresse	string	The address of the last location of the chip
last_location_complement	string	The address complement of the last location of the chip
last_location_npa	string	The ZIP of the last location of the chip
last_location_localite	string	The city of the last location of the chip
last_facturation	string	Full address of the last invoicing address of the chip
last_facturation_nom	string	The name of the last invoicing address of

		the chip
last_facturation_adresse	string	The address of the last invoicing address of the chip
last_facturation_complement	string	The address complement of the last invoicing address of the chip
last_facturation_npa	string	The ZIP of the last invoicing address of the chip
last_facturation_localite	string	The city of the last invoicing address of the chip
last_ts	dateTime	When was the chip modified last time

GetPuceAccordingChipNr

```
Public Function GetPuceAccordingChipNr(ByVal codeWebClient As String, ByVal
transporteurid As Long, ByVal chip_nr As String) As puce
```

The function is the same as the previous one, but you search with the chip number.

IsBlack

```
Public Function IsBlack(ByVal transporteurid As Long, ByVal CodeWebClient As String, _
                        ByVal puce_id As Long) As YesNoError
```

Use this function to know if the chip identified by the id (not the chip number) is black listed or not. If the check could not be made, the response will be Error (YesNoError is an Enum).

```
   Public Enum YesNoError
        No = 0
        Yes = 1
        Error_ = 2
    End Enum
```

GetBlackList

```
Public Function GetBlackList(ByVal transporteurid As Long, ByVal CodeWebClient As
String, ByVal reference_date As Date) As BlackList()
```

Use this function to get the black listed chips at a certain date (reference_date). Use the date of the current day to have the actual list.

SetBlack

```
Public Function SetBlack(ByVal transporteurid As Long, ByVal CodeWebClient As String,
ByVal puce_id As Long, ByVal black As Boolean) As String
```

Add/remove the chip with the id chip_id in the black list. The response is « OK » when success or contains the error (string).

For the parameter puce_id, please use the internal chip_id not the chip number.

To add a chip in the black list, parameter « black » has to be set to true.

To remove a chip from the black list, parameter « black » has to be set to false.

GetAddressesDeClient

```
Public Function GetAddressesDeClient(ByVal codeWebClient As String, ByVal noclient As Long, ByVal transporteurid As Long) As adresse()
```

Retrieves all addresses placed under a customer id (noclient).

DeleteAddress

```
Public Function DeleteAddress(ByVal transporteurid As Long, ByVal CodeWebClient As String, ByVal address_id As Long) As Boolean
```

You can delete an address (if this one is not used in the database). Please provide the address id. The answer is Boolean and will be True if the deletion will be successful.

GetContainerTypes

```
Public Function GetContainerTypes(ByVal codeWebClient As String, ByVal transporteurid As Long) As type_container_()
```

This function will send you the list of the container types.

Container type structure:

Field	Type (max len)	Comment
id	**unsignedInt**	**Container type id**
type_container	**string**	**Container type name**
capacite_lt	unsignedInt	Capacity in liters (0 if not know or not used)

GetGarbageTypes

```
Public Function GetGarbageTypes(ByVal transporteurid As Long, ByVal CodeWebClient As String) As type_dechet()
```

The list of garbage types will be sent. A garbage type is composed of these fields:

Field	Type (max len)	Comment
id	unsignedInt	Garbage type id
code	string	The short code for this type of garbage
type_de_dechet	String	The common garbage name
type_de_dechet_fr	String	Garbage name in French
type_de_dechet_de	String	Garbage name in German

LinkChipToAddress

```
Public Function LinkChipToAddress(ByVal codeWebClient As String,
ByVal transporteur_id As Long, ByVal puce_id As Long,
ByVal date_from As Date, ByVal date_to As Date, ByVal invoicing_address_id As Long,
ByVal location_address_id As Long, ByVal chip_manager_by_client_id As Long,
ByVal container_type As Integer, ByVal container_nr As String, ByVal garbage_id As
Integer) As LinkResponse
```

Once you have created a chip, created a location address and a invoicing address (which can be the same address), you can link all these entities with LinkChipToAddress.

Date_From and Date_To is the validity of these links. If you don't know when the date_to will be, then set it to 01.01.1800. It mean's then "the end date" is not known yet.

Chip_manager_by_client_id = is in fact "chip manage**d** by client id". Provide here the id of the customer of who will be in charge of this chip (most of the time the transporteur directly).

You can provide your own container number (`container_nr) (max 10 chars).`

The link response will be an object link including an object "parent" containing a chip information. If there was a problem during the process, an Error_ field will be sent (String).

The object link contains the following fields:

Field	Type	Comment
id	unsignedLong	Link id
puce	unsignedLong	Chip Id (not number)
client	unsignedLong	Customer Id
emplacement	unsignedLong	Location address Id
facturation	unsignedLong	Invoicing address Id
du	dateTime	Valid from

au	dateTime	Valid until (or 1.1.1800 if not unknown)
dernier_changement	dateTime	Last change made on
deleted	Boolean	True if this link has been deleted
dechet	unsignedInt	Garbage Type Id
container	unsignedInt	Container Type ID
container_nr	String 10	Container Number
cree_le	dateTime	Created on
cree_par	unsignedLong	Created by user id
dernier_changement_par	unsignedLong	Last change by user id
deleted_le	dateTime	Deleted on
deleted_par	unsignedLong	Deleted by user id
locked	boolean	Indicates if the link can be modified

MarkThisLinkAsDeleted

```
Public Function MarkThisLinkAsDeleted(ByVal codeWebClient As String, ByVal
transporteurid As Long, ByVal link_id As Long) As Boolean
```

Use this method to mark a link as deleted. It's not possible to really delete the record in database. We just mark it as deleted so we can keep track of the changes.

Please provide the link id to be deleted.

The response is True if the operation was successful.

MarkThisLinkAsFinishedOn

```
Public Function MarkThisLinkAsFinishedOn(ByVal codeWebClient As String,
ByVal transporteurid As Long, ByVal link_id As Long, ByVal date_to As String) As
String
```

Use this method to mark a link as finished. It works only with links having the date set to unknown (01.01.1800). The answer is either **"OK"** or the error **in a string**.

getChipTypes

```
Public Function getChipTypes() As IdEtNom()
```

Send back the description types for the chips (id and name fields).

getProducts

Allows you to download the full list of product for the carrier 'Transporteur_id'. Collection of id and name objects.

Appendix

Definition of the field “flag” (in a measure)

Bit 0-4 = Quality (0-7): 0 = very good … 7 very bad
10 = unstable
11 = max slope
12 = max load
Bit 6 : 0 = EndCustomer / 1 = Container (1 if the measure has been created following the emptying of a container, otherwise 0 if the measure has been created when the driver has hit the button „End customer“).