

LOCCO

Server Soap Interface

Project name : Locco Soap Service
Project number : 200901
Developer : Engels – HLC
Document title : LOCCO Server Soap Interface
Document version : v1.02
Author : B. Koreman
Date : 11-03-2013

Inhoud

1 COMMUNICATION INTERFACE	4
1.1 PROCEDURE	4
2 INTERFACE	5
2.1 LogIn	5
2.1.1 Function	6
2.1.2 Parameters	6
2.1.4 Errors	6
2.2 LogOut	6
2.2.1 Function	6
2.2.3 Returns	6
2.2.4 Errors	6
2.3 SelectRegion	6
2.3.1 Function	7
2.3.3 Returns	7
2.3.4 Errors	7
2.4 GetStateView	7
2.4.1 Function	7
2.4.3 Returns	7
2.4.4 Errors	8
2.5 GetState	9
2.5.1 Function	9
2.5.2 Parameters	9
2.5.3 Returns	9
2.4.4 Errors	10
2.6 GetDumps	11
2.6.1 Function	11
2.6.2 Parameters	11
2.6.3 Returns	11
2.6.4 Errors	12
2.7 ConfirmDumps	13
2.7.1 Function	13
2.7.2 Parameters	13
2.7.3 Returns	13

2.7.4 Errors	13
2.9 ConfirmEvents	14
2.9.1 Function	14
2.9.2 Parameters	14
2.9.3 Returns	15
2.9.4 Errors	15
2.10 GetWhiteList	15
2.10.1 Function	15
2.10.3 Returns	15
2.10.4 Errors	15
2.11 SendWhitelist	16
2.11.1 Function	16
2.11.2 Parameters	16
2.11.3 Returns	16
2.11.4 Errors	16
2.12 GetBlacklist	17
2.13 SendBlacklist	17
2.14 GetConfigByUnitID	18
2.14.1 Command	18
2.14.2 Parameters	18
2.14.3 Returns	18
2.14.4 Errors	19
2.15 SendConfig	20
2.15.1 Function	20
2.15.2 Parameters	20
2.15.3 Return	21
2.15.4 Errors	21
A Appendix	22
A.1 Error codes	22
A.2 Eventtypes	23

1 COMMUNICATION INTERFACE

This document describes the communication interface for the client program of the HLCBSCOM-server (<http://soapv2.hlcbs.nl/page=soap.wsdl> version 1.02).

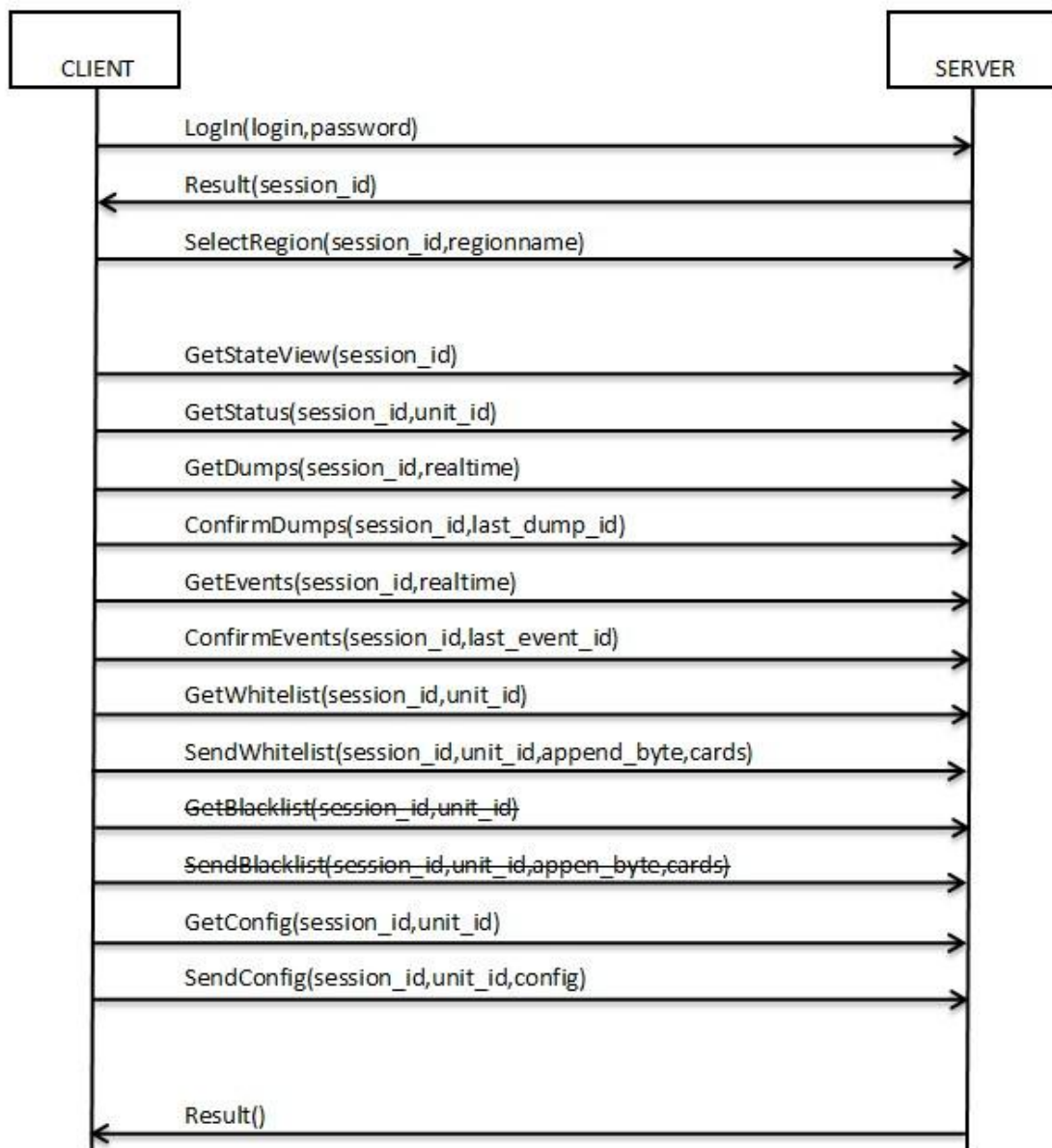
1.1 PROCEDURE

The interface forms a client-server protocol based on SOAP (Simple Object Access Protocol). This protocol is schematically displayed in figure 1.1.

The protocol is straight forward. A session always starts with the function 'Login', this will return a 'session key'. To close the session the function 'Logout' needs to be called. All other functions can be called in random order. Between a few SOAP calls there is a dependency. For instance retrieving the dumps (first the dumps, if the number of dumps is larger than zero, the dumps need to be confirmed by the call 'Confirm Dumps').

There are a number of errors that can occur. For instance a user can try to select a wrong city, this will result in an error. This is a user error. There will be three types of errors, user errors, syntax errors and database errors.

The error string always starts with a #-character. In case of a user error this character is followed by '[uniquenumber]'. The client can attach a comment to this error. A short description of the error is also included. In all other cases of errors the returnvalue is '# - Server Error – functionname'.



figuur 1.1: SOAP commands

2 INTERFACE

The interface contains a series of SOAP calls which implement the previously mentioned 'protocol'.

2.1 Login

The first function which is called is de login-function. This functions logs a user on and starts a session.

2.1.1 Function

```
Login( login As String,  
       Password As String) As String
```

2.1.2 Parameters

Login loginname of the user. Password
Password of the user.

2.1.3 Returns session_id =

'd7ad8f23ba7c9a2b6738e5ec02e3706f'

The returned value is the session_id which needs to be used for the rest of the SOAP functions.

2.1.4 Errors

"#[00000] Server Error – Login"

"#[10000] Error – unknown username or password"

"#[10010] Error – problem retrieving session_id"

2.2 LogOut

This function is called when closing the clientsoftware and closes the session.

2.2.1 Function

```
Logout( session_id As String) As Boolean
```

2.2.2 Parameters session_id

The obtained session_id

2.2.3 Returns

True, if the session is successfully closed.

False, when the session cannot be closed.

2.2.4 Errors

'False' if an error occurs. Error is logged in the logfile on the server.

2.3 SelectRegion

This function will select the provided Region. This region should be assigned to the logged in user.

2.3.1 Function

`SelectRegion(session_id As String) as Boolean`

2.3.2 Parameters session_id

The obtained session_id

2.3.3 Returns

True, if region is successfully selected. False, if Region could not be selected.

2.3.4 Errors

'False' if an error occurs. Error is logged in the logfile on the server.

2.4 GetStateView

This function will show all the information available for the containers available for a region.

2.4.1 Function

`GetStateView(session_id As String) As String`

2.4.2 Parameters session_id

The obtained session_id

2.4.3 Returns

An XML recordset with unit_id, location_nr, activated, deactivated, status, error_code, dumpcount, maxdumps, watermark, current_date, voltage, bw_status, bw_date.

unit_id (int):

The unique number assigned to an access control after production.

The unit_id is assigned to a location or container in the client database.

location_nr (string):

Not used.

activated (int):

Unix Timestamp (number of seconds after 1-1-1970) when the containers is emptied for the last time.

deactivated (int):

Unix Timestamp when the access control is removed from the location.

status (string):

error_code(int):

bit[15...0]; Appendix A.1

dumpcount(int):

Current number of dumps in the container

maxdumps (int):

Percentage of the fillrate of the container. 100 is full 0 is empty.

watermark (int):

Percentage of the watermark reached. 100 is watermark reached.

current_date(int):

Unix Timestamp of the last time the container had communicated.

Voltage:

Voltage of the battery in the container. Normal values are between 8 and 12 volt. '0' in case it's not a battery system.

bw_status(string):

bw_date(int):

Unix_timestamp of the last time the blacklist/whitelist is updated.

2.4.4 Errors

"#[00000] Server Error- Container State View"

"#[11010] Invalid session_id – Container State View"

"#[20030] No containers found for Region"

2.5 GetState

This function will show all the information available for the selected.

2.5.1 Function

```
GetStateView( session_id As String,  
              unit_id As Int) As String
```

2.5.2 Parameters

session_id The obtained session_id
unit_id The unit_id of the access control

2.5.3 Returns

An XML recordset with unit_id, location_nr, activated, deactivated, status, error_code, dumpcount, maxdumps, watermark, current_date, voltage, bw_status, bw_date.

unit_id (int):

The unique number assigned to an access control after production.

The unit_id is assigned to a location or container in the client database.

location_nr (string):

Not used.

activated (int):

Unix Timestamp (number of seconds after 1-1-1970) when the containers is emptied for the last time.

deactivated (int):

Unix Timestamp when the access control is removed from the location.

status (string):

error_code(int):

bit[15...0];

dumpcount(int):

Current number of dumps in the container

maxdumps (int):

Percentage of the fillrate of the container. 100 is full 0 is empty.

watermark (int):

Percentage of the watermark reached. 100 is watermark reached.

current_date(int):

Unix Timestamp of the last time the container had communicated.

Voltage:

Voltage of the battery in the container. Normal values are between 8 and 12 volt. '0' in case it's not a battery system.

bw_status(string):

bw_date(int):

Unix_timestamp of the last time the blacklist/whitelist is updated.

2.4.4 Errors

"#[00000] Server Error- GetState"

"#[11010] Invalid session_id – GetState"

"#[20040] Container not found"

2.6 GetDumps

This function returns all the unconfirmed dumps.

2.6.1 Function

```
GetDumps (session_id As String,  
          Realtime As boolean) As String
```

2.6.2 Parameters

session_id The obtained session_id realtime realtime
communication with access control.

2.6.3 Returns

An XML recordset with all the unconfirmed dumps. The XML record contains the following datafields: dump_id, location_nr, unit_id, card_nr, timestamp, type, status

Dump_id (int):

The unique dump identity number.

Location_nr (int):

Not used yet.

unit_id (int):

The unit_id of the access control

Card_nr (String):

The cardnumber used.

Timestamp (Int):

Unix_timestamp of the moment the dump has taken place.

Type (String):

DumpType gives extra information about the dump

0: invalid transaction

1: valid transaction

2: valid transaction with servicecard

- 3: blacklisted card transaction invalid.
- 4: Valid transaction with PrePay card
- 5: Valid transaction with Button-Only
- 6: Valid transaction with Coin-Validator
- 7: Out-of-Order card used to put container Out of Order
- 8: Out-of-Order card used to put container in Order
- 9: Empty Container card used

Status(String):

Not used yet

2.6.4 Errors

“#[00000] Server Error – Get Dump Record”

“#[11010] Invalid Session ID – Get Dump Record”

2.7 ConfirmDumps

This function is used to confirm the dumps retrieved and successfully processed by the client. After confirming the last_dump_id all the dumps with a dump_id smaller than the last_dump_id will no longer be send with the GetDumps function.

2.7.1 Function

```
ConfirmDumps( session_id As String,  
              last_dump_id As Int) As String
```

2.7.2 Parameters

session_id The obtained session_id last_dump_id The dump_id of the last successfully processed dump by the client.

2.7.3 Returns

'true' after successfully processed confirm request. Else 'false'.

2.7.4 Errors

'False' if an error occurs. Error is logged in logfile on server.

2.8 GetEvents

This function will return the events (appendix A.2). An event is information send by the access controller to the server.

2.8.1 Parameters

```
GetEvents(session_id as String,  
          realtime as Boolean) As String
```

2.8.2 Parameters session_id The obtained session_id
realtime realtime communication with access control.

2.8.3 Returns

An XML recordset with all the unconfirmed events. The XML record contains the following datafields: event_id,location_nr,unit_id,eventtype,timestamp,code,value.

event_id (int):

The unique event identity number.

Location_nr (int):

Not used yet.

unit_id (int):

The unit_id of the access control

Eventtype (String):

Type of event. Appendix A.2

Timestamp (Int):

Unix_timestamp of the moment the event has taken place on the container.

code (Int):

Errorcode as described in Appendix A.1

Value (String):

Additional information of the event.

2.8.4 Errors

“#[00000] Server Error – Get Event Record”

“#[11010] Invalid Session ID – Get Event Record”

2.9 ConfirmEvents

This function is used to confirm the events retrieved and successfully processed by the client. After confirming the last_event_id all the events with a event_id smaller than the last_event_id will no longer be send with the GetEvents function.

2.9.1 Function

```
ConfirmEventss(session_id As String,  
last_event_id As Int) As String
```

2.9.2 Parameters

session_id The obtained session_id last_event_id The event_id of the last successfully processed event by the client.

2.9.3 Returns

'true' after successfully processed confirm request. Else 'false'.

2.9.4 Errors

'False' if an error occurs. Error is logged in logfile on server.

2.10 GetWhiteList

This function will return the whitelist on a container.

2.10.1 Function

```
GetWhitelist( session_id As String,  
unit_id As Int) As String
```

2.10.2 Parameters session_id The obtained
session_id unit_id The unit_id of the access control

2.10.3 Returns

An XML recordset with all the cards authorized on the container. The XML record contains the following datafields: card_nr, cardtype.

card_nr (String):

The cardnumber authorized on the container.

Cardtype(String): Type
of card:

"1" usercard

"2" resetcard

"3" servicecard

2.10.4 Errors

"#[00000] Server Error – GetWhitelist"

"#[11010] Invalid Session ID – GetWhitelist"

"#[20040] unit not found"

2.11 SendWhitelist

This function is used to send the whitelist to the container. A whitelist contains all authorized cards that are allowed to use the container.

2.11.1 Function

```
SendWhitelist( session_id As String,  
               unit_id As Int,  
               cards As String,          appendbyte As  
Int) As String
```

2.11.2 Parameters

session_id	The obtained session_id
unit_id	The unit_id of the access control
cards	Array with with card_nr and cardtype . <u>card_nr</u> : Cardnumber of the card authorized to use the container <u>cardtype</u> : type of card. "1" usercard "2" resetcard "3" servicecard
appendbyte	The appendbyte is used to reduce datacommunication to the container. Appendbytes: 0: replace whitelist on the container with the send cards 1: append the send cards to the whitelist 2: remove the send cards from the whitelist.

2.11.3 Returns

String "OK" if whitelist is successfully replaced/updated

2.11.4 Errors

"#[11010] Invalid Session ID – SendWhitelist"
"#[00002] Validation Whitelist failed"
"#[00003] Empty whitelist (No Whitelist has been send)"
"#[00005] unknown unit_id"
"#[00007] invalid card specification"

2.12 GetBlacklist

Not available yet.

2.13 SendBlacklist

Not available yet.

2.14 GetConfigByUnitID

This function will retrieve the current configuration set on the unit on a container. The configuration describes the behavior of the container.

2.14.1 Command

```
GetConfigByUnitID( session_id As String,  
                  Unit_id As Int) As String
```

2.14.2 Parameters

session_id The obtained session_id unit_id
The unit_id of the access control

2.14.3 Returns

An XML recordset with all the configuration parameters on the container. The XML record contains the following datafields:

location_nr,unit_id,slottimeout,maxdumps,watermark,dumpdays,dumpweeks,dumptime)

Location_nr (int):

Not used yet.

unit_id (int):

The unit_id of the access control

Slottimeout(int):

Time in seconds that the lock needs to stay open after a valid card is used. Default:
10

Maxdumps (int):

Maximum number of dumps that the container can contain. When the dumpcount reaches this value the fillrate will be 100%.

Default: 120

Watermark (int):

A number of dumps before a certain level is reached. For instance number of dumps one day before container reaches his maximum number of dumps.

Dumpdays (String):

Communication parameter. Describes on what day the container should communicate. The parameter contains 7 positions. Each position is a day of the week in the following order: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday. The value of the character can be 1 (communicate) or 0 (do not communicate). Example: 1111111 (communicate every day of the week).

1111100 (communicate on Monday till Friday).

Dumpweeks(String):

Not used yet.

Dumptime(Int):

Number of seconds after midnight the container should start to communicate.

2.14.4 Errors

“#[00000] Server Error – GetConfigByUnitID”

“#[11010] Invalid Session ID – GetConfigByUnitID”

“#[20040] unit not found”

2.15 SendConfig

Send the configuration to a unit. The configuration defines the behavior of the container.

2.15.1 Function

```
SendConfig( session_id As String,  
            location_nr As String,  
            config As String) As String
```

2.15.2 Parameters

session_id	The obtained session_id
location_nr	not used yet. Leave empty.
Config	array with the configuration parameters.

Location_nr (int):
Not used yet. unit_id (int):
The unit_id of the access control

Slottimeout(int):
Time in seconds that the lock needs to stay open after a valid card is used.
Default: 10

Maxdumps (int):
Maximum number of dumps that the container can contain. When the dumpcount reaches this value the fillrate will be 100%.
Default: 120 Watermark (int):
A number of dumps before a certain level is reached. For instance number of dumps one day before container reaches his maximum number of dumps.
Default: 100

Dumpdays (String):
Communication parameter. Describes on what day the container should communicate.
The parameter contains 7 positions. Each position is a day of the week in the following order: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday. The value of the character can be 1 (communicate) or 0 (do not communicate).
Example: 1111111 (communicate every day of the week).
 1111100 (communicatie on Monday till Friday).
Default: 1111111

Dumpweeks(String):

Not used yet.

Dumptime(Int):

Number of seconds after midnight the container should start to communicate.

2.15.3 Return

String "OK" if config is successfully send

2.15.4 Errors

"#[00000] Server Error – SendConfig"

"#[11010] Invalid session ID – SendConfig"

"#[16010] Unit ID not available for Region"

A Appendix

A.1 Error codes

errorcode	description
1	Error cardreader
2	Batterij low
4	Real time clock error
8	Memory full
16	Whitelist memory full
32	Whitelist incomplete
64	3 times failed dump
128	communication error
512	sensor broken
1024	invalid datum/time during startup

A.2 Eventtypes

eventtype	description
1	Power up
2	Power down
11	Unknown card
13	Blacklisted card
21	valid transaction wl card
22	invalid transaction wl card
23	valid transaction all access card
24	invalid transaction all access card
25	resetcard used
26	servicecard used
31	error detected
32	Battery low
41	Watermark reached
42	Container Full
43	container emptied
60	tambour opened by application server
61	tambour closed by application server
70	container removed
71	container placed
72	container in service
73	container out service
74	gps latitude
75	gps longitude