	FlexServer REST webservices	Created:	09/07/2009
		Updated:	06/02/2020
		Revision:	Rev. 67
		Owner:	Poul Møller Hansen

FlexServer http REST webservices

Copyright ©2020 Flextrack ApS - Poul Møller Hansen <pmh@flextrack.dk>

6 Feb 2020

| | **FlexServer REST webservices** | Created:
Updated:
Revision:
Owner: | 09/07/2009
06/02/2020
Rev. 67
Poul Møller Hansen |

Contents

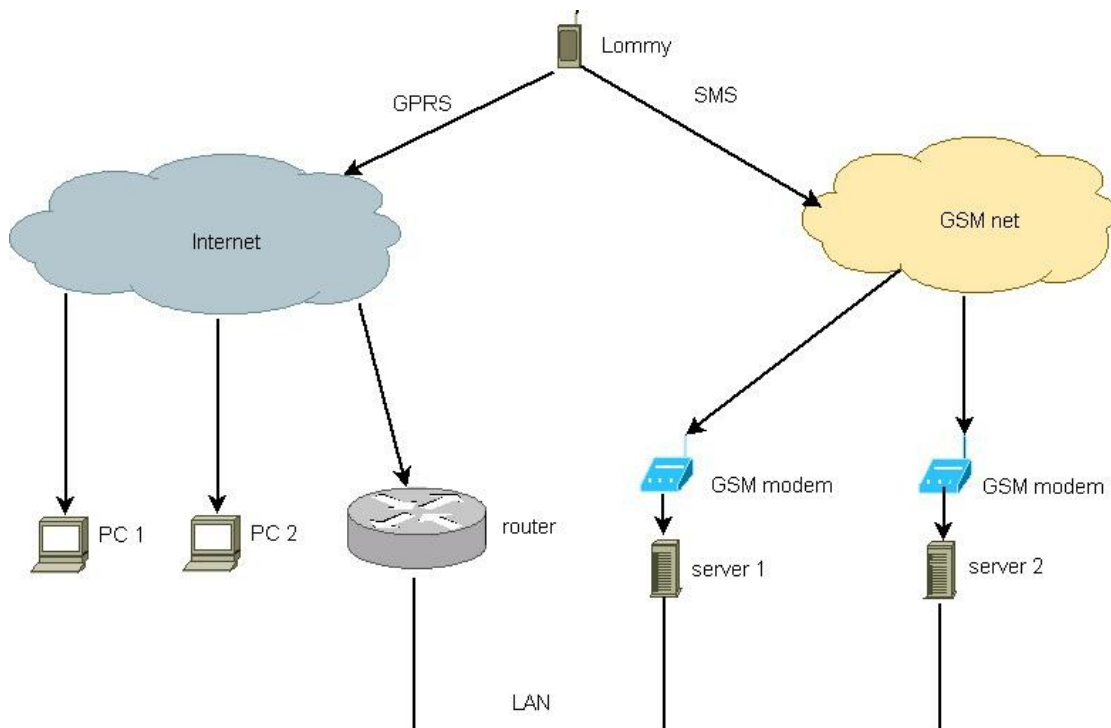| | | Created: | 09/07/2009 |
		Updated:	06/02/2020
	FlexServer REST webservices	Revision:	Rev. 67
		Owner:	Poul Møller Hansen

1 Introduction

The FlexServer http[1] REST[2] webservices interface is an interface being used to get and store data in the FlexServer backend. The FlexServer are handling the protocol and traffic between the server and the numerous Lommy® GPS units, and it contains the UI (User Interface) application.
http REST is chosen because of simplicity, to minimize overhead, and then there is a good chance the UI will be web based, a reason is also, that the browsers have natively http REST support.
Method GET is used to retrieve data either from the database backend, frontend or the connected units via the frontend. Data are returned in JSON[3] format. POST are used when data are updated, PUT when data are inserted and DELETE when data are deleted. When the GET/POST/PUT/DELETE was successful, response code 200 are returned.
Please refer to section "http response codes"[4] for possible failure conditions.
The REST webservices are available on http://flexws.appserver.dk

2   http response codes

In case a failure code is returned, the reason is written in the body entity.

Table 1: http response codes used.

code	description
200	OK
202	Accepted
204	No Content
301	Moved Permanently
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
409	Conflict
500	Internal Server Error
503	Service Unavailable

3   URN tree

/flexws/

4 Interfaces description

Common parameters for all GET requests are guid, ts and mask.

guid is a unique string that belongs to a unique Lommy® unit. This string can be found in the web interface. Select "advanced" in Profile → User data → "User level for technical settings" and the string can be found in the bottom of the page. Ex. 3UzWiMAruRv0QH00000000

ts is an incrementing integer with the epoch timestamp of when the data was received on the server. This number is used as a input pointer, so you can decide from what row your export should start. On the initial export you should set this number to 0 or calculate the timestamp of when you want to start from, using the epochcalc interface.

mask is a hexadecimal number setting bits (fields) of interest. Each bit represents a field, so unused fields aren't exported to save bandwidth in both ends. This is an optional field, default is to export all fields.
Ex. to export the fields id, date, tine, and tag id, but but rssi, the bit pattern should be 1111 and the value for mask is F

Maximum number of rows in one request are 1000 if no end_ts are given. If end_ts are given, the range between ts and end_ts may not exceed 1 month. The interface "begin timestamp" can be used to find first row of data, if all data are to be retrieved.
There must be at least 1 minute of pause between requests or the response will be delayed.

1. begin timestamp /flexws/begintimestamp/

1.a This interface can be used to find the timestamp of the first message of a given message type are received from the given unit. This is very usable, so you don't have to query every single month from january 1. 1970 where data surely are not present. It will give you ts to start with, if you want to fetch all data.

Table 4.1.1: /flexws/begintimestamp/ GET, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
msgtype	20	*	Message type, see table 4.1.2 for valid options
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"
callback	100		Callback function for JSONP requests

Table 4.1.2: Message types

msgtype	for interface
EVENT	event
GDATA	gdata
OPT-STATUS	optstatus
OWRH-REPORT	owrhreport
OWTEMP-REPORT	owtempreport
SUTMESSAGE	sutmessage
UNITSTAT	unitstat
WOLF-ALIVE	wolf-alive
WOLF-HEARD	wolf-heard
WOLF-REMOVED	wolf-removed

Table 4.1.3: /flexws/begintimestamp/ GET, return values.

parameter	datatype	description
ts	int32	epoch timestamp of first data received of given message type

1.b Test form for 1.a /flexws/begintimestamp/form/

2. /flexws/calculation/batteryconsumption/:

2.a This interface can be used to calculate the expected remaining capacity on battery powered devices, after it has sent a report.

Table 4.2.1: /flexws/calculation/batteryconsumption/ POST, input parameters.

parameter	max length	mandatory	description
model	10	*	Model of the device
initial_capacity	20	*	What was the capacity of the battery before the report in mAh.
vstart	5	*	What was the supply voltage when starting to report. vstart, vend and vdelay are from the previous report, and is for that reason not available in the latest report.
vend	5	*	What was the supply voltage after the report has been sent.
vdelay	5	*	What was the supply voltage a few minutes af the report.
ton	5	*	Time in seconds the device was turnen on while reporting.
gps	1	*	0 if GPS was not active in the report, 1 if it was.
temp	3		Temperature measured by the device
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"

Table 4.2.2: /flexws/calculation/batteryconsumption/ POST, return values.

name	datatype	description
remaining_capacity	double	Calculated remaining capacity of the battery. Additionally also these daily values, depending on the device configuration, must be subtracted. If temperature is under 5°C add 0.35mAh per day. If temperature is abone 50°C add 1.5mAh per day. Battery self-discharge: 9B2 model, add 1mAt per day

name	datatype	description
		9B6 model, add 1.5mAh per day.
		Light sensor active, add 0.05mAh per day.
		G-sensor active, add 0.1mAh per day.
		EchoTag active, add 0.05mAh per day.
		Beacon active, add 0.35mAh per day.

Possible returned error codes and messages:

400, "Bad Request"

400, "Blank Values Given"

400, "Value 'value' > 'length'"

2.b Test form for 2.a /flexws/calculation/batteryconsumption/form/

3. command /flexws/command/

3.a Use this interface to send commands, configuration parameters or any other kind of data, that should be queued for delivery to the unit.

Table 4.3.1: /flexws/command/ input data.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
parameter	20	*	Command or parameter
value	2048		Value for the command or parameter
addqueue	1	*	0/1 Add to queue or do only deliver if unit is connected to the server
update	1	*	0/1 Update existing command/parameter in queue, if already there
persistent	1	*	0/1 Keep in queue to send command/parameter to unit every time it connects to the server. Else it is deleted from queue when delivered.
onlyvalue	1		Normally parameter and value separated with a comma, are being sent to the device. If set to 1, only the content of value are being sent. Parameter are only used as a key then, when updating queue content. Default onlyvalue are 0.

Possible return codes and messages:
202, "Accepted"
400, "Bad Request"
400, "Blank Values Given"
400, "Value 'value' > 'length'"
401, "Unauthorized"
500, "Internal Server Error"

3.b Test form for 3.a /flexws/command/form/

4. Devicelist /flexws/devicelist/

 This interface returns a list of device members on the manager account given.

Table 4.4.1: /flexws/devicelist/ POST, input parameters.

parameter	max length	mandatory	description
username	50	*	Username for manager account
password	50	*	Password for manager account
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"

Table 4.4.2: /flexws/devicelist/ POST, return values.

name		description
nodeid	string(10)	Serialnumber of device
model	string(10)	Model of device
unitname	string(50)	Name of the device
unitcomment	strin(100)	Comment set on device
fwversion	int(32)	Firmware of device
appversion	Int(32)	Userapp version of device
uap	string(30)	UAP type and version
guid	string(40)	Unique identifier of device for FlexWS
msncountrycode	string(5)	Mobile countrycode of msn
msn	string(20)	Mobile subscription number of SIM-card in device
tachograph	string(20)	Tachograph information
tachograph_epoch	int(32)	Epoch timestamp of tachograph information updated.
corrected	int(32)	Epoch timestamp of latest correction of settings
latest_position	int(32)	Epoch timestamp of latest position from device

Possible returned error codes and messages:
301, "Moved Permanently"
400, "Blank Values Given"
400, "Value 'value' > 'length'"
401, "Unauthorized"
404, "Not Found"
500, "Internal Server Error"

5. epochcalc /flexws/epochcalc/

5.a This interface will not return any data from the unit, but is used to calculate the value of "ts", the epoch timestamp value. "ts" is a timestamp of when the data was received in the server backend, and epoch is the elapsed number of seconds since 01-01-1970.

Table 4.5.1: /flexws/epochcalc/ epoch to datetime, input parameters.

parameter	max length	mandatory	description
epoch	10	*	Epoch value (seconds since 01-01-1970

Table 4.5.2: /flexws/epochcalc/ epoch to datetime, return values.

parameter	description
year	Year of epoch value (YYYY)
month	Month of epoch value ((M)M)
day	Day of epoch value ((D)D)
hours	Hours of epoch value ((H)H)
minutes	Minutes of epoch value ((m)m)
seconds	Seconds of epoch value ((s)s)

Table 4.5.3: /flexws/epochcalc/ datetime to epoch, input parameters.

parameter	max length	mandatory	description
year	4	*	Year part of datetime (YYYY)
month	2	*	Month part of datetime ((M)M)
day	2	*	Day part of datetime ((D)D)
hours	2	*	Hours part of datetime ((H)H)
minutes	2	*	Minutes part of datetime ((m)m)
seconds	2	*	Seconds part of datetime ((s)s)

Table 4.5.4: /flexws/epochcalc/ datetime to epoch, return values.

parameter	description
epoch	Epoch value of datetime

6. event /flexws/event/:

6.a This interface is used to get events logged on the system

Table 4.6.1: /flexws/event/ GET, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
ts	10	*	Last timestamp or start of period. If end_ts are not given max. 1000 rows can be retrieved at a time.
end_ts	10		End timestamp. span between ts and end_ts may not exceed 1 month or error 400 "Timestan To Large" are received.
tz	40		Valid timezone for date/time values. Ex.Europe/Copenhagen. Default is UTC
mask	8		Hexadecimal number with bits (fields) of interest
typemask	40		Bit mask with event types of interest. Se table 4.6.2 for pattern.
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"
callback	100		Callback function for JSONP requests

Table 4.6.2: mask for event types.

name	description	bit
allalarmsoff	All alarms, except SYS are disabled	0
allalarmson	All alarms are enabled	1
auto_subscribe	Auto report interval has been subscribed	2
battlow	Battery level low	3
battlow1h	Battery level low, 1 hour delay	4
battlow5m	Battery level low, 5 min. delay	5
button1	Button 1 has been pressed	6
button2	Button 2 has been pressed	7
charger	Charger has been removed	8
charger10	Charger has been removed, 10 min, delay	9
charger5m	Charger has been removed, 5 min. delay	10

name	description	bit
charging	Charger has been connected	11
charging1h	Charger has been connected, 1 hour delay	12
charging5m	Charger has been connected, 5 min. delay	13
config	Configuration of unit from webservice	14
data6	Data hold for 6 months has been subscribed	15
dbook_subscribe	Driver book functionality has been subscribed	16
deadman	Deadman event	17
ecentre	Subscribed to emergency centre	18
edgeiohigh	I/O port had a leading edge	19
edgeiolow	I/O port had a tailing edge	20
freemap	Free map opened	21
ftnlogoff	Terminal user logoff	22
ftnlogon	Terminal user logon	23
ftnpstart	Terminal pause start	24
ftnpstop	Terminal pause stop	25
ftnsstart	Terminal salary start	26
ftnsstop	Terminal salary stop	27
func_subscribe	Function button subscription	28
funcpress	Function button has been pressed	29
geofence_subscribe	Geofence functionality subscription	30
geofenceoff	Inside geofence zone again. Zone defined in unit	31
geofenceon	Outside geofence zone. Zone defined in unit	32
geofenceon1h	Outside geofence zone, 1 hour delay. Zone defined in unit	33
geofenceon5m	Outside geofence zone, 5 min. delay. Zone defined in unit	34
gle	G-limit has been exceeded	35
gpsfail	GPS failed, unit has been restarted	36
greatcircle1h	Outside geofence zone, 1 hour delay. Zone defined on server	37
greatcircle5m	Outside geofence zone, 5 min. delay. Zone defined on server	38
greatcircleoff	Inside geofence zone again. Zone defined on server	39
greatcircleon	Outside geofence zone. Zone defined on server	40
i1high	Input 1 level is high	41
i1high1h	Input 1 level has been high for 1 hour	42
i1high5m	Input 1 level has been high for 5 min.	43
i1low	Input 1 level is low	44
i1low1h	Input 1 level has been low for 1 hour	45
i1low5m	Input 1 level has been low for 5 min.	46
i2high	Input 2 level is high	47
i2high1h	Input 2 level has been high for 1 hour	48
i2high5m	Input 2 level has been high for 5 min.	49
i2low	Input 2 level is low	50

name	description	bit
i2low1h	Input 2 level has been low for 1 hour	51
i2low5m	Input 2 level has been low for 5 min.	52
i3high	Input 3 level is high	53
i3high1h	Input 3 level has been high for 1 hour	54
i3high5m	Input 3 level has been high for 5 min.	55
i3low	Input 3 level is low	56
i3low1h	Input 3 level has been low for 1 hour	57
i3low5m	Input 3 level has been low for 5 min.	58
i4high	Input 4 level is high	59
i4high1h	Input 4 level has been high for 1 hour	60
i4high5m	Input 4 level has been high for 5 min.	61
i4low	Input 4 level is low	62
i4low1h	Input 4 level has been low for 1 hour	63
i4low5m	Input 4 level has been low for 5 min.	64
idleoff	Vehicle not idling anymore	65
idleon	Vehicle is idling	66
igncounter	Minute counter for ignition active has been exceeded	67
iohigh	I/O port level is high	68
iohigh1h	I/O port level has been high for 1 hour	69
iohigh5m	I/O port level has been high for 5 min.	70
iolow	I/O port level is low	71
iolow1h	I/O port level has been low for 1 hour	72
iolow5m	I/O port level has been low for 5 min.	73
ioport_subscribe	I/O port functionality subscription	74
logoff	Unit has logged off on the server	75
logon	Unit has logged on to the server	76
lost	Alarm event for uni not connected to the server	77
lost14d	Alarm event for unit hasn't been connected to the server for 14 days	78
map	Map opened	79
maxspeed	Max speed has been exceeded	80
maxspeed5m	Max speed has been exceeded for 5 min.	81
mcounter	Distance counter has been exceeded	82
monthlyfee	Monthly free drawn	83
movement	Event for unit that hasn't been moved for a given period of time	84
movement1h	Event for unit that hasn't been moved for a given period of time, 1 hour delay	85
movement5m	Event for unit that hasn't been moved for a given period of time, 5 min delay	86
moving	Unit has been moved	87
moving1h	Unit has been moved for 1 hour	88
moving5m	Unit has been moved for 5 min.	89

name	description	bit
newpossms	New position requested using a text message SMS)	90
nofix	Unit has not GPS fix for a number of reports	91
nolost	Unit reconnected to the server after being reported as lost	92
notmoving	Unit has now been moved for a period of time	93
notmoving1h	Unit has now been moved for a period of time, 1 hour delay	94
notmoving5m	Unit has now been moved for a period of time, 5 min. delay	95
onemap	Active subscription on maps, and one map opened.	96
owdid	1-Wire driver ID has been read	97
owrh	Measured humidity out of bounds	98
owtemp	Measured temperature out of bounds	99
payment	Payment has been done	100
pdamap	Map has been opened on a PDA	101
plus	PLUS package subscription	102
point	Point	103
pos1_subscribe	Subscribed to 1 position/hour	104
pos10	Subscribed to 10 position/hour	105
pos15	Subscribed to 15 position/hour	106
pos15_subscribe	Subscribed to 15 position/hour	107
pos30	Subscribed to 30 position/hour	108
pos6	Subscribed to 6 position/hour	109
pos6_subscribe	Subscribed to 6 position/hour	110
pos60	Subscribed to 60 position/hour	111
powerdown	Unit powered down for a period of time from the server	112
poweroff	Unit has been powered off from the server	113
reset	Unit has been reset from the server	114
roam_subscribe	Roaming subscription	115
roaming	Unit is roaming	116
settlement	Settlement of balance	117
sleep_subscribe	Sleep mode subscription	118
smsalarmalloff	All alarms, except SYS are disabled. Command received as text message (SMS)	119
smsalarmallon	All alarms enabled. Command received as text message (SMS)	120
smsalarmoff	Alarms with same mobile number as sender are disabled. Command received as text message (SMS)	121
smsalarmon	Alarms with same mobile number as sender are enabled. Command received as text message (SMS)	122
smsgfoff	Geofence alarm disabled. Command received as text message (SMS)	123
smsgfon	Geofence alarm enabled. Command received as text message (SMS)	124
smspos	Position requested in text message (SMS)	125
smswarn	Unit is in SMS mode instead of GPRS	126

name	description	bit
stopstealth	Stop stealtmode	127
suddenaraise	Sudden raise in analog measurement	128
tilt	Unit has been tilted	129
tilt_subscribe	Tils subscription	130
zonein	Unit has entered a defined zone	131
zoneout	Unit has left a defined zone.	132
ainhigh	Analog measurement above limit	
ainlow	Analog measurement below limit	
alarmactive	Alarm user active	
alarmactiveschedule	Alarm user scheduled active	
analogdeclining	Analog measurement are declining	
charger1h	Charger has been connected for 1 hour	
cooler_devconn_conn	Cooler equipment turned on	
cooler_devconn_disc	Cooler equiment turned off	
cooler_data_lost	Cooler data lost	
cooler_data_nolost	Cooler data back	
dooropen	Door opened	
doorclosed	Door closed	
maxspeed1h	Max speed has been exceeded for 1 hour	
owrh1h	Measured humidity out of bounds for 1 hour	
owrh5m	Measured humidity out of bounds for 5 min.	
owtempsensorhigh	Fault on temperature sensor	
owtempsensorlow	Temperature sensor are OK again	
seeemail	More instructions in email.	
sms	Text message sent.	
suddenadrop	Sudden drop in analog measuments	
suddenaraise	Sudden raise in analog measurements	
tag_keepalive_lost	Keepalive signal from tag are lost	
tag_keepalive_nolost	Keepalive signal from tag are heard again	
tag_lowbatt_set	Tag battery level low	
tag_lowbatt_unset	Tag battery level is OK again	
temp_inside_range	Temperature inside range again	
temp_outside_range	Temperature outside range	
vpfuel_level	Low fuel level	
vpfuel_sudden_drop	Sudden drop in fuel level	
wolfheard	Wolf tag heard	
wolfremoved	Wolf tag can no longer be heard	

Table 4.6.3: /flexws/event/ GET, return values.

name	datatype	description	bit
ts	int32	Server timestamp when data was received on the server (UTC) This is the number to use as parameter "ts" to get data that has been stored since last request.	0
userid	string(20)	Who has caused the event	1
type	string(30)	Type of event. See table 4.6.2 for possible values	2
price	double	Price of the event, negative values are drawn	3
action	string(10)	Action of the event. Possible values are "", continue, email, log, payment, sms, start, stop, url, urlresp	4
currency	string(3)	Three letter currency code in ISO 4217	5
eventtext	text	Text of the event	6
recipient	string(50)	recipient of the event	7
geozoneid	int32	Geozone id that caused the event	8
uid	string(50)	uid of invoice	9
userlog	boolean	Syslog or userlog (0/1)	10
date	date	GPS date of event (YYYY-MM-DD)	11
time	time	GPS time of the event (HH:MM:SS)	12
flexserverid	int16	Id of the flexserver, data are coming from	13
id	int64	Row id, unique per flexserverid	14
unitstat_id	int64	Id from unitstat that triggered the event, if triggered from that.	15
nodeid	string(10)	nodeid (serialnumber) of the unit	16

Possible returned error codes and messages:

400, "Blank Values Given"

400, "Unknown TimeZone"

401, "Unauthorized"

500, "Internal Server Error"

6.b Test form for 6.a /flexws/event/form/

7. eventhandler /flexws/eventhandler/:

7.a This interface is used to maintain event handlers that will be triggered on specific events. The handler can send email, text messages and make GET/POST requests to a webservice.

Body for the POST method is an json array of objects with some, or all of these elements

Root element	Type	Description
active	number	1 for active and 0 for inactive alarm
activefrom	number	epoch value when eventhandler will become active. Default null
Activeto	number	epoch value when eventhandler will become inactive. Default null
weekdays	number	bitmask for weekdays eventhandler is active. Default 0 and 127 are all weekdays. bit0=mon, bit1=tue … bit6=sun.
periodfrom	string	(HHMMSS) UTC time of weekday eventhandler will become active. Default null
periodto	string	(HHMMSS) UTC time of weekday eventhandler will become inactive. Default null
type	number	event type described in table 4.7.2
text	string	Alarm string to send. Default strings are used if value is blank. Depending on context, different [XX] tags can be used, to be replaced with actual values from event. String tags are described in table 4.7.3. The same strings can be used as http get/post parameter values.
high	number	1 for alarm high type trigger
low	number	1 for alarm low type trigger
delay	number	Delay in seconds before the event is checked again, and trigger is fired if event persists.
min_temp	number	minimum temperature in decimal degrees for temperature alarm.
max_temp	number	maximum temperature in decimal for temperature alarm.
moved_distance	number	distance in meters for "moved" alarm.
min_distance	number	Distance alarm if value is below this minimum.

max_distance	number	Distance alarm if value is above this maximum.
measure_port	number	Port distance alarm is measuring on.
calendar_schedule	number	epoch value when calendar event should happen. Default null
delete_after_use	number	1 delete eventhandler after it has been used once, default null
locale	string	Locale used in event messages. Default is unit locale.
filter	String	Ignore filter
eventHandlerHasCallbackUrl	object	
eventHandlerHasEmail	object	
eventHandlerHasSms	object	

eventHandlerHasCallbackUrl element	Type	Description
scheme	string	Scheme, default http
username	string	Optional username for basic authentication
password	string	Optional password for basic authentication
host	string	Webhost
port	number	Port if not port 80
path	string	Path part of URI
query	string	Query parameters.
post	number	1 if using POST method else GET.
body	string	Body content for POST method.

eventHandlerHasEmail element	Type	Description
subject	string	Subject of email
recipient	string	Email address

eventHandlerHasSms element	Type	Description
msisdn	string	Mobile number with countrycode.

Table 4.7.1: /flexws/eventhandler/ GET, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"
callback	100		Callback function for JSONP requests

Table 4.7.2: /flexws/eventhandler/ POST, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
body		*	JSON object, see class description and appendix A.
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"

Table 4.7.3: /flexws/eventhandler/ DELETE, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
uuid	36		UUID of the eventhandler to delete. If blank all are deleted.
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and

parameter	max length	mandatory	description
			"message"

Table 4.7.4: /flexws/eventhandler/ PUT, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
uuid	36		UUID of the eventhandler to update. If blank all are updated with active value.
body		*	JSON object. Currently the following fields can be updated: active activefrom activeto periodfrom periodto weekdays high low eventHandlerHasEmail eventHandlerHasSms recipient_old & recipient_new can be used to change an email recipient. If eventHandlerHasEmail or eventHandlerHasSms are updated, current array are deleted prior to update.
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"

Table 4.7.5: Event types

Type	description
1	Charger removed
2	Battery level low
3	AUX port high
4	Function button pressed
5	Trailing edge on AUX port
6	Leading edge on AUX port
9	Button 1 has been pressed

Type	description
10	Button 2 has been pressed
11	Movement alarm
12	Deadman alarm
13	Outside geofence
15	Unit is in motion
16	Input port 1 active
17	Input port 2 active
18	Input port 3 active
19	Input port 4 active
23	1-Wire driver id connected
28	Wolf tag heard
30	G-limit has been exceeded
31	1-Wire humidity alarm
32	Max speed alarm (unit defined)
35	Lightsensor alarm
36	Button 3 has been pressed
37	Button 4 has been pressed
64	Outside geofence (server defined)
65	Max speed alarm (server defined)
66	Tilt alarm
67	Unit is not connected to the server
68	Time counter alarm
69	Inside geozone
70	Idling
71	Unit connected again after being reported disconnected (type 67)
72	Ignition counter alarm
73	Unit is not connected to the server (14 days period)
74	Meter counter event
75	Sudden raise in analog measurements
76	No GPS fix in latest 250 positions
77	Analog values outside limits
78	Accumulated input port active exceeded limit
79	Analog values are slowly declining
80	Temperature sensor faulty
81	Temperature level alarm
82	Sudden drop in analog measurements
83	Cooler device connected
84	Cooler data lost
85	Sudden drop in fuel level
86	Fuel level below 20%

Type	description
87	RF tag keepalive signal lost
88	RF tag battery level low
89	Door opened
90	Clima tag temperature level alarm
91	Clima tag humidity level alarm
92	Clima tag battery level low
93	Vport temperature above or below limits
94	Moved more than distance given since last position reported
95	Echo tag heard or lost
96	When entered another country since last position reported
97	Account to expire. Event will fire 14, 7, 1 & 0 days before expiration
98	Information on account low balance
99	Unit connect and disconnect from server
100	No keepalive / keepalive again from unit
101	Unit has no GPS coverage / has GPS coverage again
102	Driving with no chauffeur-card
103	Ultrasonic distance below limit
104	Ultrasonic distance above limit
105	Sensor battery low
106	Calendar event has occured
107	Missing equipment from tag collection
108	Refind event for lost equipment

Table 4.7.6: Bracket [] tags

Type	description
[ADDRESS]	Nearest postal address
[COMMENT]	Zone comment
[DIR]	Direction 0-360
[DIST]	Distance from current position to geofence (server) center
[GUID]	Unique id of the unit for webservices
[LAT]	Latitude in degrees, minutes and seconds
[LATDEC]	Latitude in decimal degrees
[LINK]	The login link
[LOCATION]	Zone location
[LON]	Longtitude in degrees, minutes and seconds
[LONDEC]	Longtitude in decimal degrees
[MSN]	Telephone number of the unit
[NAME]	Name on the alarm receiver id
[NODEID]	Nodeid of the unit
[RADIUS]	Radius stored for the geofence (server) setup

Type	description
[SATS]	Number of satellites seen
[SPEED]	Actual speed in km/h
[TNO]	Transaction number
[TYPE]	Alarmtype
[TIME]	Datetime, corrected to local timezone
[UNITNAME]	The name of the unit given
[UNITSTAT]	The complete UNITSTAT message
[UNITSTATE]	The unit state bit pattern from the UNITSTAT
[VALUE1]	value 1
[VALUE2]	value 2
[VALUE3]	value 3
[VOLTAGE]	Battery voltage in mV

7.b Test form for 7.a /flexws/eventhandler/form/

8. gdata /flexws/gdata/:

8.a This interface is used to either send (POST) or receive (GET) data of the message type GDATA[5]. When sending data the client must support redirection, as FlexWS will redirect your post data to the server having the unit account (return code 301) or back to FlexWS if the unit have no longer an account here.

Table 4.8.1: /flexws/gdata/ GET, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
ts	10	*	Last timestamp received or start of period. If end_ts are not given max. 1000 rows can be received per request.
end_ts	10		End timestamp. Span between ts and end_ts may not exceed 1 month or error 400 "Timespan To Large" are received.
in	1		If set only data sent from unit to backend are shown
tz	40		Valid timezone for date/time values. Ex.Europe/Copenhagen. Default is UTC
mask	8		Hexadecimal number with bits (fields) of interest
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"
callback	100		Callback function for JSONP requests

Table 4.8.2: /flexws/gdata/ GET, return values.

name	datatype	description	bit
ts	int32	Server timestamp when data was received on the server (UTC) This is the number to use as parameter "ts" to get data that has been stored since last request.	0
date	date	GPS date of event (YYYY-MM-DD)	1
time	time	GPS time of event (HH:MM:SS)	2
nodeid	string(10)	nodeid (serialnumber) of the unit	3
slot	int(26)	Every data source has a unique slot number. Currently only slot 1 is used, data to/from terminal	4
in	boolean	Set to 1 if it is data received from unit or 0 if sent to unit	5

FlexServer REST webservices

Created:	09/07/2009	
Updated:	06/02/2020	
Revision:	Rev. 67	
Owner:	Poul Møller Hansen	

name	datatype	description	bit
msg	text	text content	6

Possible returned error codes and messages:
400, "Blank Values Given"
400, "Timespan To Large"
400, "Unknown TimeZone"
400, "Value 'value' > 'length'"
401, "Unauthorized"
500, "Internal Server Error"

Table 4.8.3: /flexws/gdata/ POST, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
slot	10	*	Slot number is message usage. Currently only 1 is used (terminal communication)
msg	1400		Text content of data
tz	40		Valid timezone for date/time values. Ex.Europe/Copenhagen. Default is UTC. Used when parsing slot 1 messages (terminal)
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"

Possible return codes and messages:
202, "Accepted"
301, "Moved Permanently"
400, "Bad Request"
400, "Blank Values Given"
400, "Value 'value' > 'length'"
401, "Unauthorized"
500, "Internal Server Error"

8.b Test form for 8.a /flexws/gdata/form/

9. geozone /flexws/geozone/:

9.a This interface is used to maintain circular geozones. When a unit enters or leaves one or more of the defined zones, it can depending of the alarm receiver setup, either send an email, text message (SMS) or call a REST webservice. In any case the event will be logged and can be retrieved using the /flexws/event interface.

Table 4.9.1: /flexws/geozone/ GET, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
id	10		id of the zone. If omitted the complete list will be returned.
posformat	1		Position format 0. degrees, minutes, decimal minutes (default) ex. N55.30.1351 1. decimal degrees ex 55.502252 2. degrees, minutes, seconds ex. N55°30'9''
mask	8		Hexadecimal number with bits (fields) of interest
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"
callback	100		Callback function for JSONP requests

Table 4.9.2: /flexws/geozone/ GET, return values.

name	datatype	description	bit
id	int(32)	Numeric unique value of the zone	0
location	string(50)	Description of the zone	1
lat	double	Latitude of center	2
lon	double	Longitude of center	3
activefrom	time	Zone is active from this time of day (HHMMSS)	4
activeto	time	Zone is active to this time of day (HHMMSS)	5
comment	text	Comment of the zone	6
radius	int(32)	Radius of the zone	7
created	int(32)	Timestamp of when the zone was created	8
corrected	int(32)	Timestamp of when the zone was changed	9
polygon	OGC WKT[8]	Geometry of the zone	10

name	datatype	description	bit
timelimit	time	Must leave or enter this zone before timelimit (HHMMSS)	11
type	int(32)	1=toll road, 2=go zone, 3=nogo zone, 4=fence zone, 5=report zone	12

Possible returned error codes and messages:

400, "Blank Values Given"

401, "Unauthorized"

500, "Internal Server Error"

Table 4.9.3: /flexws/geozone/ POST, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
location	50		Description of the zone, used as a part of the event.
lat	12	(*)	Latitude of the center
lon	12	(*)	Longitude of the center
activefrom	6		Time of day zone is active from (HHMMSS) default 000000
activeto	6		Time of day zone is active to (HHMMSS) default 235959
timelimit	6		Must leave or enter this zone before timelimit (HHMMSS)
comment	256		Comment for the zone
radius	6	(*)	Radius of zone in meters
polygon	OGC WKT[8]	(*)	Geometry of the zone
type	1		Default 4, 1=toll road, 2=go zone, 3=nogo zone, 4=fence zone, 5=report zone, 6=zone for reluctant door opened/closed event
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"

(*) Either lat/lon/radius og polygon must be given

Table 4.9.4: /flexws/geozone/ POST, return values.

parameter	datatype	description
id	int(32)	Numerical value of the id created or updated

Possible return codes and messages:
301, "Moved Permanently"
400, "Bad Request"
400, "Blank Values Given"
400, "Unknown TimeZone"
400, "Value 'value' > 'length'"
401, "Unauthorized"
500, "Internal Server Error"

Table 4.9.5: /flexws/geozone/ DELETE, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
id	10	*	id of geozone to delete
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"

Possible return codes and messages:
301, "Moved Permanently"
400, "Bad Request"
400, "Blank Values Given"
400, "Value 'value' > 'length'"
401, "Unauthorized"
404, "Not Found"
500, "Internal Server Error"

9.b Test form for 9.a /flexws/geozone/form/

10. latestoptstatus /flexws/latestoptstatus/:

10.a This interface will return the latest data received of message type OPT-STATUS[5] analog measurements.

Table 4.10.1: /flexws/latestoptstatus/ GET, input parameters.

parameter	max length	mandatory	description
guid	36	(*)	Unit identifier
tz	40		Valid timezone for date/time values. Ex.Europe/Copenhagen. Default is UTC.
mask	8		Hexadecimal number with bits (fields) of interest.
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"
callback	100		Callback function for JSONP requests

Table 4.10.2: /flexws/latestoptstatus/ GET, return values.

name	datatype	description	bit
ts	int(32)	Server timestamp when data was received on the server (UTC)	0
date	date	GPS date of event (YYYY-MM-DD)	1
time	time	GPS time of event (HH:MM:SS)	2
nodeid	string(10)	nodeid (serialnumber) of the unit	3
ain1	int(16)	Voltage on input port 1 in decivolts	4
ain2	int(16)	Voltage on input port 2 in decivolts	5
ain3	int(16)	Voltage on input port 3 in decivolts	6
ain4	int(16)	Voltage on input port 4 in decivolts	7
aval1	int(32)	Number of minutes (accumulated) input port 1 has been logic high	8
aval2	int(32)	Number of minutes (accumulated) input port 2 has been logic high	9
aval3	int(32)	Number of minutes (accumulated) input port 3 has been logic high	10
aval4	int(32)	Number of minutes (accumulated) input port 4 has been logic high	11

Possible returned error codes and messages:

400, "Bad Request"
400, "Blank Values Given"
400, "Value 'value' > 'length'"
401, "Unauthorized"
500, "Internal Server Error"

10.b Test form for 10.a /flexws/latestoptstatus/form/

11. latestowtempreport /flexws/latestowtempreport/:

11.a This interface will return the latest data received of message type OWTEMP-REPORT[5] temperature measured by 1-Wire sensors.

Table 4.11.1: /flexws/latestowtempreport/ GET, input parameters.

parameter	max length	mandatory	description
guid	36	(*)	Unit identifier
tz	40		Valid timezone for date/time values. Ex.Europe/Copenhagen. Default is UTC.
mask	8		Hexadecimal number with bits (fields) of interest.
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"
callback	100		Callback function for JSONP requests

Table 4.11.2: /flexws/latestowtempreport/ GET, return values. Data are returned as a JSON array, one element for each temperature sensor.

name	datatype	description	bit
ts	int(32)	Server timestamp when data was received on the server (UTC)	0
date	date	GPS date of event (YYYY-MM-DD)	1
time	time	GPS time of event (HH:MM:SS)	2
nodeid	string(10)	nodeid (serialnumber) of the unit	3
hwaddress	string(20)	Hardware address of temperature sensor	4
temp	int(16)	Temperature in decidegrees (Celcius)	5
status	int(16)	Status of sensor. bit 0: Is 1 if device is responding bit 1: Is 1 if low temperature alarm bit 2: Is 1 if high temperature alarm	6
comment	string(100)	Comment/name given to sensor	7

Possible returned error codes and messages:
400, "Bad Request"
400, "Blank Values Given"
400, "Value 'value' > 'length'"
401, "Unauthorized"
500, "Internal Server Error"

11.b Test form for 11.a /flexws/latestowtempreport/form/

12. latestsutmessage /flexws/latestsutmessage/:

12.a This interface will return the latest data received from devices using the SUT protocol [6].

Table 4.12.1: /flexws/latestsutmessage/ GET, input parameters.

parameter	max length	mandatory	description
guid	36	(*)	Unit identifier
tz	40		Valid timezone for date/time values. Ex.Europe/Copenhagen. Default is UTC.
mask	8		Hexadecimal number with bits (fields) of interest.
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"
callback	100		Callback function for JSONP requests

Table 4.12.2: /flexws/latestsutmessage/ GET, return values. Data are returned as a JSON object named sutmessage.

name	datatype	description	bit
ts	int(32)	Server timestamp when data was received on the server (UTC)	0
date	date	GPS date of event (YYYY-MM-DD), only available with GPS position	1
time	time	GPS time of event (HH:MM:SS), only available with GPS position	2
nodeid		nodeid (serialnumber) of the unit	3
tno	string(10)	Transaction number of message, starting af -1 at unit startup and decrementing	4
iccid	string(20)	ICCID code of SIM-card	5
us	int(64)	Unit state[5]	6
ta	int(32)	GSM timing advance	7
temp	int(16)	Temperature of unit	8
version	int(16)	Program version of unit	9
fwversion	int(16)	Firmware version of unit	10
nodeid	string(10)	Nodeid of EchoTag	11
status	int(16)	Status of EchoTag	12
lat	double	Latitude of position	13
lon	double	Longitude of position	14
speed	int(32)	Speed measured	15
ttf	int(32)	Time to GPS fix	16

name	datatype	description	bit
radius	int(32)	Radius of estimated position, if positioned purely from GSM data	17
cellid	int(32)	GSM cell id	18
rssi	int(32)	Radio signal strength indication	19
lac	int(32)	Location are code	20
mcc	int(16)	Mobile country code	21
mnc	int(16)	Mobile network code	22
dir	Int(16)	Direction of position	23
hourvalue	int(32)	Input counter	24

Possible returned error codes and messages:
400, "Bad Request"
400, "Blank Values Given"
400, "Value 'value' > 'length'"
401, "Unauthorized"
500, "Internal Server Error"

12.b Test form for 12.a /flexws/latestsutmessage/form/

13. latestunitstat /flexws/latestunitstat/:

13.a This interface will return, either the latest or the latest with a valid postion, message of type UNITSTAT[5]

Table 4.13.1: /flexws/latestunitstat/ GET, input parameters.

parameter	max length	mandatory	description
guid	36	(*)	Unit identifier
groupguid	36	(*)	Group uuid. Either guid or groupguid must be set.
posformat	1		Position format 0. degrees, minutes, decimal minutes (default) ex. N55.30.1351 1. decimal degrees ex 55.502252 2. degrees, minutes, seconds ex. N55°30'9''
good	1		If set to 1, the latest UNITSTAT with a valid postion are returned, which is not necessarily the latest data received.
tz	40		Valid timezone for date/time values. Ex.Europe/Copenhagen. Default is UTC.
mask	8		Hexadecimal number with bits (fields) of interest. If unset and you are calling this interface with groupguid and unitstat_mask was set when created the logingroup, that value will be used.
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"
callback	100		Callback function for JSONP requests

Table 4.13.2: /flexws/latestunitstat/ GET, return values.

name	datatype	description	bit
ts	int(32)	Server timestamp when data was received on the server (UTC)	0
date	date	GPS date of event (YYYY-MM-DD)	1
time	time	GPS time of event (HH:MM:SS)	2
nodeid	string(10)	nodeid (serialnumber) of the unit	3
lat	double	Latitude of position	4

name	datatype	description	bit
lon	double	Longitude of position	5
speed	int(16)	Speed in km/h	6
dir	int(16)	Direction of movement	7
sv	int(16)	Number of satellites in view	8
volt	int(16)	Battery voltage in mV	9
rssi	int(16)	Signal strength og GSM signal in dBm	10
us	int(64)	Unit state[5] 1)	11
type	Int(16)	Transport method of the data -1 unknown, 0 tcp, 1 udp, 2 SMS, 3 http (FLEETMAN[5])	12
net	string(10)	Which GSM network is used MCC+MNC	13
height	int(16)	Height of position relative to MSL	14
hdop	int(16)	Horizontal dilution of position	15
cellid	string(5)	Cell ID in the GSM network	16
dd	double	Delta distance since previous report	17
ttf	int(16)	Time to fix in seconds	18
lac	string(5)	Location Area Code in the GSM network	19
odo	int(32)	Odometer value	20
gle	int(32)	Number G-limit has been exceeded	21

Possible returned error codes and messages:

204, "No Content"

400, "Bad Request"

400, "Blank Values Given"

400, "Value 'value' > 'length'"

401, "Unauthorized"

500, "Internal Server Error"

1) us is a hexadecimal representation of a bit pattern showing state of the unit, at the time when the position was reported. This can be charger connected, function button pressed, unit is moving etc. Please see table "Unit State field" in [5]

Here is an example code in Java, checking if input port 1 (ignition) is high. According to the table port 1 is bit 16.

```
String us = "18003";
long ius = Long.parseInt(us, 16);
if (((ius >> 16) & 0x1) > 0) {
   // Ignition on
} else {
   // Ignition off
}
```

13.b Test form for 13.a /flexws/latestunitstat/form/

14. logingroup /flexws/logingroup/

14.a This interface is used to maintain a logical group of units. The group guid is authorization to all devices contained by the group and auth guid authenticates manipulating the group and it's members.

Table 4.14.1: /flexws/logingroup/ GET, input parameters. Returns properties og the group itself.

parameter	max length	mandatory	description
authguid	36	*	The auth uuid returned when the group was created
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"
callback	100		Callback function for JSONP requests

Table 4.14.2: /flexws/logingroup/ GET, return values.

name	datatype	description
groupguid	string(40)	The unique group identifier
username	string(50)	username to the group
password	string(50)	password to the group
qos	int(16)	MQTT QoS value
unitstat_mask	int(32)	mask for fields included in push data

Possible returned error codes and messages:
400, "Blank Values Given"
400, "Value 'value' > 'length'"
404, "Not Found"
500, "Internal Server Error"

Table 4.14.3: /flexws/logingroup/ POST, input parameters. Sets username and password etc. qos and *_mask fields are only used with push data.

parameter	max length	mandatory	description
authguid	36	*	The auth uuid returned when the group was created
username	50	*	Username must be at least 6 characters. It is case insensitive.
password	50	*	Password must be at least 6 characters. It is case insensitive.

parameter	max length	mandatory	description
qos	1		MQTT Qos field 0-2, 0 is default
unitstat_mask	8		Which unitstat fields to include in the data. See 4.33.2 for fields. Ex. to get date, time, nodeid, lat, lon, speed and sv set unitstat_mask to 17E
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"

Possible returned error codes and messages:
400, "Blank Values Given"
400, "Password < 6 characters"
400, "Username < 6 characters"
400, "Username already in use"
400, "Value 'value' > 'length'"
401, "Unauthorized"
500, "Internal Server Error"

Table 4.14.4: /flexws/logingroup/ DELETE, input parameters. Deletes the login group and members

parameter	max length	mandatory	description
authguid	36	*	The auth uuid returned when the group was created
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"

Possible returned error codes and messages:
400, "Blank Values Given"
400, "Value 'value' > 'length'"
401, "Unauthorized"
500, "Internal Server Error"

Table 4.14.5: /flexws/logingroup/ PUT, input parameters. Create a new group. qos and *_mask fields

are only used with push data.

parameter	max length	mandatory	description
username	50	*	The username to authenticate to the group It must be at least 6 characters and it is case insensitive.
password	50	*	The password to authenticate to the group It must be at least 6 characters and it is case insensitive.
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"

Table 4.14.2: /flexws/logingroup/ PUT, return values.

name	datatype	description
authguid	string(40)	The unique auth identifier
groupguid	string(40)	The unique group identifier

Possible returned error codes and messages:

400, "Blank Values Given"

400, "Value 'value' > 'length'"

400, "Username already in use"

500, "Internal Server Error"

14.b Test form for 14.a /flexws/logingroup/form/

15. keepalive /flexws/logingroup/keepalive/

15.a Keepalive must be used with push services. data feed will start, when this interface has been called, and it must be called at least every 30 minutes to maintain the data flow. The groupguid value can be used for getting data from more devices in one request.

Table 4.15.1: /flexws/logingroup/keepalive/ POST, input parameters. Updates the keepalive timestamp

parameter	max length	mandatory	description
username	50	*	username for the group
password	50	*	password for the group
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"

Table 4.15.2: /flexws/logingroup/keepalive/ POST, return values.

name	datatype	description
groupguid	string(40)	The unique group identifier

Possible returned error codes and messages:
400, "Blank Values Given"
400, "Value 'value' > 'length'"
404, "Not Found"
500, "Internal Server Error"

15.b Test form for 15.a /flexws/logingroup/form/

16. member /flexws/logingroup/member/

16.a This interface is used to add and remove members to a login group

Table 4.16.1: /flexws/logingroup/member/ POST, input parameters. Add member to group

parameter	max length	mandatory	description
authguid	36	*	The auth uuid returned when the group was created
guid	36	*	guid of the unit you want to add to the group
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"

Possible returned error codes and messages:
400, "Blank Values Given"
400, "Value 'value' > 'length'"
401, "Unauthorized"
404, "Not Found"
409, "Member already exists in logingroup"
500, "Internal Server Error"

Table 4.16.2: /flexws/logingroup/member/ DELETE, input parameters. Remove member from group

parameter	max length	mandatory	description
authguid	36	*	The auth uuid returned when the group was created
guid	36	*	guid of the unit you want to remove from the group
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"

Possible returned error codes and messages:
400, "Blank Values Given"

400, "Value 'value' > 'length'"
401, "Unauthorized"
404, "Not Found"
500, "Internal Server Error"

16.b Test form for 16.a /flexws/logingroup/member/form/

17. members /flexws/logingroup/members

17.a This interface is used to retrieve all members of a login group

Table 4.17.1: /flexws/logingroup/ GET, input parameters. Return members of a login group given

parameter	max length	mandatory	description
authguid	36	*	The auth uuid returned when the group was created
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"
callback	100		Callback function for JSONP requests

Table 4.17.2: /flexws/logingroup/ GET, return values.

name	datatype	description
groupguid	string(40)	The unique group identifier
members ->		
nodeid	string(10)	nodeid (serial number) of unit 1
unitname	string(50)	name of unit 1
nodeid	string(10)	nodeid (serial number) of unit 2
unitname	string(50)	name of unit 2
...		...

Possible returned error codes and messages:
400, "Blank Values Given"
400, "Value 'value' > 'length'"
404, "Not Found"
500, "Internal Server Error"

17.b Test form for 17.a /flexws/logingroup/members/form/

18. lommy /flexws/lommy/

18.a This interface will return metadata of the given unit

Table 4.18.1: /flexws/lommy/ GET, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
mask	12		Hexadecimal number with bits (fields) of interest
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"
callback	100		Callback function for JSONP requests

Table 4.18.2: /flexws/lommy/ GET, return values.

name	datatype	description	bit
nodeid	string(10)	nodeid (serialnumber) of the unit	0
custno	string(50)	Customer number to unit	1
active	boolean	Is account active (can unit logon to the server)	2
comment	string(100)	Comment to the unit	3
msncountrycode	string(5)	Countrycode of SIM card	4
msn	string(20)	Telephone number of SIM card	5
fleetman	int(16)	Connected to this fleetmanagement server	6
fleetuser	string(30)	Username on fleetmanagement server	7
fleetpass	string(30)	Password on fleetmanagement server	8
salesdate_dealer	date	Salesdate to dealer (YYYY-MM-DD)	9
dealer	string(50)	Dealer	10
histdata	boolean	Is historically data visible (Subscriptions where normally not enabled)	11
model	string(10)	Model of unit	12
network	string(5)	GSM network from where latest position were reported in a text message	13
blocked	boolean	Web account is disabled	14
subscription	int(16)	Subscription type	15
salesdate_enduser	date	Salesdate to enduser (YYYY-MM-DD)	16
demo_to	date	Account active to this date (YYYY-MM-DD)	17
manual_config	boolean	Unit is configured manually and default configuration template	18

name	datatype	description	bit
		will not be used	
posurl	text	Send every position reported from the unit, to this address	19
posfailed	int(16)	Number of failed http pos	20
posemail	string(100)	Send email to this address if max. number of failed http pos attempts has been reached	21
gid	int(32)	Group ID of manager login	22
unitname	string(50)	Name given to the unit from manager login	23
alwayson	boolean	Report if unit are not logged on to the server	24
imei	string(15)	IMEI of unit	25
unitcomment	strint(100)	Comment to unit	26
hwrevision	int(32)	Hardware revision of unit	27
altconftemplate	int(32)	Alternative configuration template in use	28
ignition_counter	int(32)	Elapsed number of seconds with ignition active (input port 1)	29
ecentre	string(10)	Subscribed to this emergency center	30
swupdate_attempts	int(16)	Number of software updates attempted	31
swupdate_date	date	Date of last software update attempt (YYYY-MM-DD)	32
mcounter	int(64)	Moved distance in meters. Calculated on reported positions	33
iccid	string(32)	ICCID number of SIM card	34
iccid_account	boolean	Is ICCID number used as member id in fleet management	35
variant	int(16)	Variant of model	36
fuelmeter	int(16)	Decimal value of fuelmeter	37
fueltanksize	int(16)	Fuel tanksize in liter	38
fleet_statemask	int(64)	Decimal value to AND with unit state to see if position should be shown in fleet management.	39
owdidtoggle	int(16)	How many seconds are the output active when an iButton has been read	40
nofix	int(32)	How many reports (UNITSTAT) in a row, without GPS fix.	41
appversion	int(32)	Application version in 1/10	42
fwversion	int(32)	Firmware version in 1/10	43
optversion	int(32)	OPT2 version in 1/10	44

Possible returned error codes and messages:
400, "Blank Values Given"
400, "Value 'value' > 'length'"
401, "Unauthorized"
500, "Internal Server Error"

18.b Test form for 18.a /flexws/lommy/form/

19 add member in Mobile Workforce /flexws/mw/member/add/

19.a This interface will create a member in Mobile Workforce fleet management in the specified fleet, on the specified position and with an optional icon.

Table 4.19.1: /flexws/mw/member/add/ POST, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
fleet_id	10	*	The id of the fleet where the member should be created. You can get this number from support.
lat	12	*	Latitude of position
lon	12	*	Longitude of position
name	50		Name of the member
icon_id	10		Id of the icon to be used when showing the member. You can get the available id's by contacting support. If undefined then default for the fleet will be used.
global	1		Member is visible for all users in the fleet
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"

Table 4.19.2: /flexws/mw/member/add/ POST, return values.

name	datatype	description
member_uuid	string(40)	A uuid that must be used if deleting the member again using webservice /flexws/mw/member/delete/

Possible returned error codes and messages:
400, "Blank Values Given"
400, "Value 'value' > 'length'"
401, "Unauthorized"
500, "Internal Server Error"

19.b Test form for 19.a /flexws/mw/member/add/

20 Delete a member in Mobile Workforce /flexws/mw/member/delete/

20.a This interface will delete the member in Mobile Workforce fleet management, having the member_uuid as control number.

Table 4.20.1: /flexws/mw/member/delete/ POST, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
member_uuid	36	*	UUID of the member to delete
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"

Possible returned error codes and messages:
400, "Blank Values Given"
400, "Value 'value' > 'length'"
401, "Unauthorized"
500, "Internal Server Error"

20.b Test form for 20.a /flexws/mw/member/delete/

21. optstatus /flexws/optstatus/

21.a This interface will return data received of the message type OPT-STATUS[5]

Table 4.21.1: /flexws/optstatus/ GET, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
ts	10	*	Last timestamp received or start of period. If end_ts are not given max. 1000 rows can be received per request.
end_ts	10		End timestamp. Span between ts and end_ts may not exceed 1 month or error 400 "Timespan To Large" are received.
tz	40		Valid timezone for date/time values. Ex. Europe/Copenhagen. Default is UTC.
mask	8		Hexadecimal number with bits (fields) of interest
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"
callback	100		Callback function for JSONP requests

Table 4.21.2: /flexws/optstatus/ GET, return values.

name	datatype	description	bit
ts	int(32)	Server timestamp when data was received on the server (UTC) This is the number to use as parameter "ts" to get data that has been stored since last request.	0
date	date	GPS date of event (YYYY-MM-DD)	1
time	time	GPS time of event (HH:MM:SS)	2
nodeid	string(10)	nodeid (serialnumber) of the unit	3
ain1	int(16)	Analog value on input port 1	4
ain2	int(16)	Analog value on input port 2	5
ain3	int(16)	Analog value on input port 3	6
ain4	int(16)	Analog value on input port 4	7
aval1	int(32)	Accumulated input time port 1 has been high (min.)	8
aval2	int(32)	Accumulated input time port 2 has been high (min.)	9
aval3	int(32)	Accumulated input time port 3 has been high (min.)	10

name	datatype	description	bit
aval4	int(32)	Accumulated time input port 4 has been high (min.)	11

Possible returned error codes and messages:

400, "Blank Values Given"

400, "Timespan To Large"

400, "Unknown TimeZone"

400, "Value 'value' > 'length'"

401, "Unauthorized"

500, "Internal Server Error"

21.b Test form for 21.a /flexws/optstatus/form/

22. owrhreport /flexws/owrhreport/

22.a This interface will return data received of the message type OWRH-REPORT[5], humidity measurements

Table 4.22.1: /flexws/owrhreport/ GET, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
ts	10	*	Last timestamp received or start of period. If end_ts are not given max. 1000 rows can be received per request.
end_ts	10		End timestamp. Span between ts and end_ts may not exceed 1 month or error 400 "Timespan To Large" are received.
tz	40		Valid timezone for date/time values. Ex. Europe/Copenhagen. Default is UTC.
mask	8		Hexadecimal number with bits (fields) of interest
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"
callback	100		Callback function for JSONP requests

Table 4.22.2: /flexws/owrhreport/ GET, return values.

name	datatype	description	bit
ts	int(32)	Server timestamp when data was received on the server (UTC) This is the number to use as parameter "ts" to get data that has been stored since last request.	0
report	int(64)	Report number. There can be more rows from different sensors, but reported at the same time	1
date	date	GPS date of event (YYYY-MM-DD)	2
time	time	GPS time of event (HH:MM:SS)	3
nodeid	string(10)	nodeid (serialnumber) of the unit	4
hwaddress	string(20)	Unique hardware address of temperature sensor	5
humidity	int(16)	Humidity in % RH	6
status	int(16)	Status of sensor[5]	7
comment	string(100)	Comment to sensor	8

Possible returned error codes and messages:

400, "Blank Values Given"

400, "Timespan To Large"

400, "Unknown TimeZone"

400, "Value 'value' > 'length'"

401, "Unauthorized"

500, "Internal Server Error"

22.b Test form for 22.a /flexws/owrhreport/form/

23. owtempreport /flexws/owtempreport/

23.a This interface will return data received of the message type OWTEMP-REPORT[5]

Table 4.23.1: /flexws/owtempreport/ GET, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
ts	10	*	Last timestamp received or start of period. If end_ts are not given max. 1000 rows can be received per request.
end_ts	10		End timestamp. Span between ts and end_ts may not exceed 1 month or error 400 "Timespan To Large" are received.
tz	40		Valid timezone for date/time values. Ex. Europe/Copenhagen. Default is UTC.
mask	8		Hexadecimal number with bits (fields) of interest
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"
callback	100		Callback function for JSONP requests

Table 4.23.2: /flexws/owtempreport/ GET, return values.

name	datatype	description	bit
ts	int(32)	Server timestamp when data was received on the server (UTC) This is the number to use as parameter "ts" to get data that has been stored since last request.	0
report	int(64)	Report number. There can be more rows from different sensors, but reported at the same time	1
date	date	GPS date of event (YYYY-MM-DD)	2
time	time	GPS time of event (HH:MM:SS)	3
nodeid	string(10)	nodeid (serialnumber) of the unit	4
hwaddress	string(20)	Unique hardware address of temperature sensor	5
temp	int(16)	Temperature in 1/10 (Celsius)	6
status	int(16)	Status of sensor[5]	7
comment	string(100)	Comment to sensor	8

Possible returned error codes and messages:

400, "Blank Values Given"
400, "Timespan To Large"
400, "Unknown TimeZone"
400, "Value 'value' > 'length'"
401, "Unauthorized"
500, "Internal Server Error"

23.b Test form for 23.a /flexws/owtempreport/form/

24. report /flexws/report/

24.a This interface will return the report specified. At the time of writing the reports available are unitstat, start-stop, temperature and start-stop-acc and the possible output formats, depending on the reports, are:
CSV, Comma Separated Values
HTML, TyperText Markup Language
ODF, OpenDocumentFormat
PDF, Portable Document Format
RTF, Rich Text Format
XLS, Microsoft Office Excel
XML, Extensible Markup Language
JSON, JavaScript Object Notation
KML, Google Earth
KMZ, Google Earth compressed format
A list of reports and possible export formats can be retrieved using the interface 25 /flexws/reportlist/

Table 4.24.1: /flexws/report/ GET, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
locale	5		Locale for texts in report. Ex. da_DK, de_DE, en_IE. Default is da_DK
ts_from	10	*	Timestamp of start of data (epoch)
ts_to	10	*	Timestamp of end of data (epoch)
tz	40		Valid timezone for date/time values. Ex.Europe/Copenhagen. Default is UTC.
report_id	10	*	Id of report, can be found with the /flexws/reportlist/ interface
report_type_id	10	*	Id of export format, can be found with the /flexws/reportlist/ interface
parameters	500		Ekstra parameters for the report as key/values. Ex. p1=v1&p2=v2&p3=v3
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"

The data are returned as a binary stream with an appropriate header for the export format.

Possible returned error codes and messages:
400, "Bad Request"
400, "Blank Values Given"
400, "Unknown TimeZone"
400, "Value 'value' > 'length'"
401, "Unauthorized"
500, "Internal Server Error"
24.b Test form for 24.a /flexws/report/form/

25. reportlist /flexws/reportlist/

25.a This interface will return all possible reports and their export formats in JSON format.

Table 4.25.1: /flexws/reportlist/ GET, input parameters.

parameter	max length	mandatory	description
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"
callback	100		Callback function for JSONP requests

Possible returned error codes and messages:
500, "Internal Server Error"

25.b Test form for 25.a /flexws/reportlist/form/

26, sensors /flexws/sensors/:

26.a This interface is used to maintain different sensors on the system, like humidity and temperature sensors. The values are used as thresholds, to make eventhandler events, if they are exceeded in values reported from the device.

Http methods and url's used are:
GET – To get a list of sensors of the specified type
/flexws/sensors/<type>/<guid>

DELETE – To delete a sensor of the specified sensor_id and type
/flexws/sensors/<type>/<guid>/<sensor_id>

POST – To create new sensors
/flexws/sensors/<type>/<guid>
Sensor data must be in body of the message in JSON format.

Type "climatag_temperature" and "owtemp"
```
[
  {
    "sensor_id": String,              - ID of the sensor
    "sensor_description": String,     - Description of the sensor
    "temp_min": Number,               - Minimum temperature in decimal degrees
    "temp_max": Number                - Maximum temperature in decimal degrees
  }
]
```

Type "climatag_humidity"
```
[
  {
    "sensor_id": String,              - ID of the sensor
    "sensor_descrription": String,    - Description of the sensor
    "humid_min": Number,              - Minimum humidity in % RH
    "humid_max": Number               - Maximum humidity in % RH
  }
]
```

Type "vport"
```
[
  {
    "sensor_id": String,              - ID of the sensor
    "sensor_descrription": String,    - Description of the sensor
    "min0": Number,           - Minimum temperature in decimal degrees
```

```
    "max0": Number          - Maximum temperature in decimal degrees
  }
]
```

Table 4.26.1: valid input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"
callback	100		Callback function for JSONP requests

Table 4.26.1: Valid values for <type>

Type	description
climatag_humidity	Clima tag humidity values
climatag_temperature	Clima tag temperature values
owtemp	1-Wire temperature sensors
vport	Vport sensor

26.b Test form for 26.a /flexws/sensors/form/

27. SUT message /flexws/sutmessage/:

27.a This interface will return data received from devices using the SUT protocol [6].

Table 4.27.1: /flexws/sutmessage/ GET, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
ts	10	*	Last timestamp received or start of period. If end_ts are not given max. 1000 rows can be received per request.
end_ts	10		End timestamp. Span between ts and end_ts may not exceed 1 month or error 400 "Timespan To Large" are received.
tz	40		Valid timezone for date/time values. Ex.Europe/Copenhagen. Default is UTC.
mask	8		Hexadecimal number with bits (fields) of interest
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"
callback	100		Callback function for JSONP requests

Table 4.27.2: /flexws/sutmessage/ GET, return values. Data are returned as a JSON object named sutmessage.

name	datatype	description	bit
ts	int(32)	Server timestamp when data was received on the server (UTC)	0
date	date	GPS date of event (YYYY-MM-DD), only available with GPS position	1
time	time	GPS time of event (HH:MM:SS), only available with GPS position	2
nodeid	string(10)	nodeid (serialnumber) of the unit	3
tno	int(32)	Transaction number of message, starting af -1 at unit startup and decrementing	4
iccid	string(20)	ICCID code of SIM-card	5
us	int(64)	Unit state[5]	6
ta	int(32)	GSM timing advance	7
temp	int(16)	Temperature of unit	8
version	int(16)	Program version of unit	9

name	datatype	description	bit
fwversion	int(32)	Firmware version of unit	10
vstart	int(16)	Start voltage from last report	11
vend	int(16)	End voltage from last report	12
vdelay	int(16)	Voltage measured 100ms after turning of the GSM module	13
nodeid	string(10)	Nodeid of EchoTag	14
status	int(16)	Status of EchoTag	15
lat	double	Latitude of position	16
lon	double	Longitude of position	17
speed	int(32)	Speed measured	18
ttf	int(32)	Time to GPS fix	19
radius	int(32)	Radius of estimated position, if positioned purely from GSM data	20
cellid	int(32)	GSM cell id	21
rssi	int(32)	Radio signal strength indication	22
lac	int(32)	Location are code	23
mcc	int(16)	Mobile country code	24
mnc	int(16)	Mobile network code	25
ton	int(32)	Time on before sleeping again	26
dir	int(32)	Direction of position (gps only)	27
hourvalue	int(32)	Input counter	28

Possible returned error codes and messages:
400, "Blank Values Given"
400, "Timespan To Large"
400, "Unknown TimeZone"
400, "Value 'value' > 'length'"
401, "Unauthorized"
500, "Internal Server Error"

27.b Test form for 27.a /flexws/sutmessage/form/

28. Configuration of units using the SUT protocol /flexws/sutmessage/configuration/

This interface is used to get and set configuration parameters in devices using the SUT protocol

Table 4.28.1: /flexws/sutmessage/configuration/ GET, input parameters. Returns the configuration of the device.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"
callback	100		Callback function for JSONP requests

Table 4.28.2: /flexws/sutmessage/configuration/ GET, return values.

name		description
ack_ts	timestamp	Timestamp of last acknowledge from the device
act	int(32)	Hexadecimal number with actions enabled in the device.
act_ts	int(32)	Timestamp of act configuration change. If newer than ack_ts the change is still in queue to be sent to the device.
act_revert_value	int(32)	ACT value to revert to
act_revert_after_positions	int(32)	Revert ACT after this number of positions
act_revert_after_ts	timestamp	Revert ACT after this UTC timestamp
echolist	text	List of echo tags the device is accepting
echolist_ts	int(32)	Timestamp of echolist configuration change. If newer than ack_ts the change is still in queue to be sent to the device.
week	string(30)	Week calender in the device
week_ts	int(32)	Timestamp of week configuration change. If newer than ack_ts the change is still in queue to be sent to the device.
sun	string(30)	Sunday calender in the device
sun_ts	int(32)	Timestamp of sun configuration change. If newer than ack_ts the change is still in queue to be sent to the device.
param	string	JSON array with other configuration parameters. The object contains the members file, parameter, ts, revert_value, revert_after_positions and revert_after_ts values.
cmd	String	JSON array with command for the unit

Possible returned error codes and messages:
400, "Blank Values Given"
400, "Value 'value' > 'length'"
401, "Unauthorized"
404, "Not Found"
500, "Internal Server Error"

Table 4.28.3: /flexws/sutmessage/configuration/ POST, input parameters. Sets and changes configuration values.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
act	10		Actions enabled in the device
act_revert_after_positions	10		Revert to previous act value after this number of positions
act_revert_after_ts	10		Revert to previous act value after this epoch timestamp
week	31		Week calendar
sun	31		Sunday calendar

param	256		JSON array of objects with the other configuration parameters.
			The object contains the members file, parameter, ts and value.
			Example:
			[{
			    "file": "par",
			    "param": "report-interval",
			    "value": "30"
			}, {
			    "file": "par",
			    "param": "moving-gps",
			    "value": "1"
			}]

cmd	512		JSON array of commands to the unit.
			Example:
			[
			  {
			    "cmd": "reset",
			    "value": ""
			  }
			]

format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"

Possible returned error codes and messages:
301, "Moved Permanently"

400, "Blank Values Given"
400, "Value 'value' > 'length'"
401, "Unauthorized"
500, "Internal Server Error"

28.b Test form for 28.a /flexws/sutmessage/configuration/form/

29. Inserting position data for units using the SUT protocol /flexws/sutmessage/gpsinfo/

 This interface is used to set the position to the device report, if the position is not already known.

Table 4.29.1: /flexws/sutmessage/gpsinfo/ POST, input parameters. Sets the position of the device.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
id	10	*	Row id to insert data into
lat	12	*	Latitue of position in decimal degrees
lon	12	*	Longitude of position in decimal degrees
radius	10		Radius of estimated position
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"

Possible returned error codes and messages:
301, "Moved Permanently"
400, "Blank Values Given"
400, "Value 'value' > 'length'"
401, "Unauthorized"
409, "Conflict"
500, "Internal Server Error"

29.b Test form for 29.a /flexws/sutmessage/gpsinfo/form/

30. terminal message /flexws/terminal/clear/:

30.a Use this interface to clear specified type of data in the terminal

Table 4.30.1: /flexws/terminal/clear/ POST, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
type	1	*	Type of data to clear. (0 = messages, 1 = tasks)
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"

Possible return codes and messages:
202, "Accepted"
301, "Moved Permanently"
400, "Bad Request"
400, "Blank Values Given"
400, "Value 'value' > 'length'"
401, "Unauthorized"
500, "Internal Server Error"

30.b Test form for 30.a /flexws/terminal/clear/

31. terminal message /flexws/terminal/message/:

31.a Use this interface to send a message to a terminal.

Table 4.31.1: /flexws/terminal/message/ POST, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
sender	50	*	The numeric id of the office user, or a valid email address.
recipient	50	*	The numeric id of the driver recipient or set to 0 if it should be seen by any user logged on.
priority	1		Priority of the message. Same values as used in e-mail header X-Priority. 1 (Highest) 2 (High) 3 (Normal) 4 (Low) 5 (Lowest)
subject	50	*	Subject of message
msg	1024		Message body
receipt	1		If set to 1 a receipt message will appear in GDATA[5] when the message has been read.
lat	12		Optional latitude for navigation.
lon	12		Optional longitude for navigation.
tz	40		Valid timezone for date/time values. Ex.Europe/Copenhagen. Default is UTC.
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"

Table 4.31.2: /flexws/terminal/message/ POST, return values.

name	datatype	description
uuid	string(40)	Unique id RFC4122, of message to link the reply with the original message

Possible return codes and messages:
202, uuid value

301, "Moved Permanently"
400, "Bad Request"
400, "Blank Values Given"
400, "Value 'value' > 'length'"
401, "Unauthorized"
500, "Internal Server Error"

31.b Test form for 31.a /flexws/terminal/message/form/

32. terminal task /flexws/terminal/task/:

32.a Use this interface to send a task to a terminal

Table 4.32.1: /flexws/terminal/task/ POST, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
priority	1		Priority of the message. Same values as used in e-mail header X-Priority. 1 (Highest) 2 (High) 3 (Normal) 4 (Low) 5 (Lowest)
uuid	36	(*)	Mandatory for update and delete, action 1 & 2
ordernumber	10	*	Numeric order number
ordername	50	*	Subject of task
description	1024		Task description
note	512		Editable note for the task. The edited note will be sent from the terminal as message type 51
location	128		Postal address where the task must be carried out
date (start)	8		Date in format YYYYMMDD of when the task should be carried out (start). Will be set to current date if not given.
time (start)	6		Time in format HHMMSS of when the task should be carried out (start). Will be set to current time if not given.
end_date	8		Date in format YYYYMMDD of when the task should be carried out (end). Will be set to current date if not given.
end_time	6		Time in format HHMMSS of when the task should be carried out (end). Will be set to current time if not given.
action	1	*	0 = new, 1 = update existing order number, 2 = delete order number
receipt	1		If set to 1 a receipt message will appear in GDATA[5] when the message has been read.
lat	12		Optional latitude for navigation.
lon	12		Optional longitude for navigation.

parameter	max length	mandatory	description
associated_nodeid	10		Optional nodeid associated with the task
tz	40		Valid timezone for date/time values. Ex.Europe/Copenhagen. Default is UTC.
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"

Table 4.10.2: /flexws/terminal/task/ POST, return values.

name	datatype	description
uuid	string(40)	Unique id RFC4122, of task to be used if updating or deleting the task

Possible return codes and messages:
202, uuid value
301, "Moved Permanently"
400, "Bad Request"
400, "Blank Values Given"
400, "Value 'value' > 'length'"
401, "Unauthorized"
500, "Internal Server Error"

32.b Test form for 32.a /flexws/terminal/task/form/

33. unitstat /flexws/unitstat/:

33.a This interface will return data received of the message type UNITSTAT[5]

Table 4.33.1: /flexws/unitstat/ GET, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
ts	10	*	Last timestamp received or start of period. If end_ts are not given max. 1000 rows can be received per request.
end_ts	10		End timestamp. Span between ts and end_ts may not exceed 1 month or error 400 "Timespan To Large" are received.
posformat	1		Position format 0. degrees, minutes, decimal minutes (default) ex. N55.30.1351 1. decimal degrees ex 55.502252 2. degrees, minutes, seconds ex. N55°30'9''
tz	40		Valid timezone for date/time values. Ex.Europe/Copenhagen. Default is UTC.
mask	8		Hexadecimal number with bits (fields) of interest
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"
callback	100		Callback function for JSONP requests

Table 4.33.2: /flexws/unitstat/ GET, return values.

name	datatype	description	bit
ts	int(32)	Server timestamp when data was received on the server (UTC) This is the number to use as parameter "ts" to get data that has been stored since last request.	0
date	date	GPS date of event (YYYY-MM-DD)	1
time	time	GPS time of event (HH:MM:SS)	2
nodeid	string(10)	nodeid (serialnumber) of the unit	3
lat	double	Latitude of position	4
lon	double	Longitude of position	5
speed	int(16)	Speed in km/h	6

name	datatype	description	bit
dir	int(16)	Direction of movement	7
sv	int(16)	Number of satellites in view	8
volt	int(16)	Battery voltage in mV	9
rssi	int(16)	Signal strength og GSM signal in dBm	10
us	int(64)	Unit state[5] 1)	11
type	int(16)	Transport method of the data -1 unknown, 0 tcp, 1 udp, 2 SMS, 3 http (FLEETMAN[5])	12
net	string(10)	Which GSM network is used MCC+MNC	13
height	int(16)	Height of position relative to MSL	14
hdop	int(16)	Horizontal dilution of position	15
cellid	string(5)	Cell ID in the GSM network	16
dd	double	Delta distance since previous report	17
ttf	int(16)	Time to fix in seconds	18
lac	string(5)	Location Area Code in the GSM network	19
odo	int(32)	Odometer value	20
gle	int(32)	Number of times G-limit has been exceeded	21
flexserverid	int(16)	Id of the flexserver, data are coming from	22
id	int(64)	Row id, unique per flexserverid	23

Possible returned error codes and messages:

400, "Blank Values Given"

400, "Timespan To Large"

400, "Unknown TimeZone"

400, "Value 'value' > 'length'"

401, "Unauthorized"

500, "Internal Server Error"

1) us is a hexadecimal representation of a bit pattern showing state of the unit, at the time when the position was reported. This can be charger connected, function button pressed, unit is moving etc. Please see table "Unit State field" in [5]

Here is an example code in Java, checking if input port 1 (ignition) is high. According to the table port 1 is bit 16.

```java
String us = "18003";
int ius = Integer.parseInt(us, 16);
if (((ius >> 16) & 0x1) > 0) {
   // Ignition on
} else {
   // Ignition off
}
```

33.b Test form for 33.a /flexws/unitstat/form/

34. wolf-alive /flexws/wolf-alive/:

34.a When a wolf tag are heard, it's reported in the WOLF-HEARD[5] message. Tags still in range are reported every WOLF-REPORT-INTERVAL[5] seconds. This interface will return data received of the message type WOLF-ALIVE[5].

Table 4.34.1: /flexws/wolf-alive/ GET, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
ts	10	*	Last timestamp received or start of period. If end_ts are not given max. 1000 rows can be received per request.
end_ts	10		End timestamp. Span between ts and end_ts may not exceed 1 month or error 400 "Timespan To Large" are received.
tz	40		Valid timezone for date/time values. Ex.Europe/Copenhagen. Default is UTC.
mask	8		Hexadecimal number with bits (fields) of interest
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"
callback	100		Callback function for JSONP requests

Table 4.34.2: /flexws/wolf-alive/ GET, return values.

name	datatype	description	bit
ts	int(32)	Server timestamp when data was received on the server (UTC) This is the number to use as parameter "ts" to get data that has been stored since last request.	0
date	date	GPS date of event (YYYY-MM-DD)	1
time	time	GPS time of event (HH:MM:SS)	2
nodeid	string(10)	nodeid (serialnumber) of the unit	3
tag id	string(10)	id of the tag heard	4
rssi	int(16)	RSSI (Radio Signal Strength Indication) of the signal from the tag	5
status	int(16)	Status bit bit 7: 1 => tag is a listed unconnected tag, 0=>"normal" unconnected tag Bit 1/0: battery level	6

name	datatype	description	bit
		00=> Battery <20% 01=> Battery <40% 10=>Battery<60%, 11=> Battery >=60%	

Possible returned error codes and messages:

400, "Blank Values Given"

400, "Timespan To Large"

400, "Unknown TimeZone"

400, "Value 'value' > 'length'"

401, "Unauthorized"

500, "Internal Server Error"

34.b Test form for 34.a /flexws/wolf-alive/form/

35. wolf-heard /flexws/wolf-heard/:

35.a This interface will return data received of the message type WOLF-HEARD[5]

Table 4.35.1: /flexws/wolf-heard/ GET, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
ts	10	*	Last timestamp received or start of period. If end_ts are not given max. 1000 rows can be received per request.
end_ts	10		End timestamp. Span between ts and end_ts may not exceed 1 month or error 400 "Timespan To Large" are received.
tz	40		Valid timezone for date/time values. Ex.Europe/Copenhagen. Default is UTC.
mask	8		Hexadecimal number with bits (fields) of interest
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"
callback	100		Callback function for JSONP requests

Table 4.35.2: /flexws/wolf-heard/ GET, return values.

name	datatype	description	bit
ts	int(32)	Server timestamp when data was received on the server (UTC) This is the number to use as parameter "ts" to get data that has been stored since last request.	0
date	date	GPS date of event (YYYY-MM-DD)	1
time	time	GPS time of event (HH:MM:SS)	2
nodeid	string(10)	nodeid (serialnumber) of the unit	3
tag id	string(10)	id of the tag heard	4
rssi	int(16)	RSSI (Radio Signal Strength Indication) of the signal from the tag	5
comment	string(100)	Comment of the tag	6
status	int(16)	Status bit bit 7: 1 => tag is a listed unconnected tag, 0=>"normal" unconnected tag Bit 1/0: battery level 00=> Battery <20%	7

name	datatype	description	bit
		01=> Battery <40% 10=>Battery<60%, 11=> Battery >=60%	

Possible returned error codes and messages:

400, "Blank Values Given"

400, "Timespan To Large"

400, "Unknown TimeZone"

400, "Value 'value' > 'length'"

401, "Unauthorized"

500, "Internal Server Error"

35.b Test form for 35.a /flexws/wolf-heard/form/

36. wolf-removed /flexws/wolf-removed/:

36.a This interface will return data received of the message type WOLF-REMOVED[5]

Table 4.36.1: /flexws/wolf-removed/ GET, input parameters.

parameter	max length	mandatory	description
guid	36	*	Unit identifier
ts	10	*	Last timestamp received or start of period. If end_ts are not given max. 1000 rows can be received per request.
end_ts	10		End timestamp. Span between ts and end_ts may not exceed 1 month or error 400 "Timespan To Large" are received.
tz	40		Valid timezone for date/time values. Ex.Europe/Copenhagen. Default is UTC.
mask	8		Hexadecimal number with bits (fields) of interest
format	enum:		format of the error messages
	"" (blank)		Response code are returned in header and text error message in body
	json		Response code are returned in header and response code and message are returned in a JSON array as the keys "code" and "message"
callback	100		Callback function for JSONP requests

Table 4.36.2: /flexws/wolf-removed/ GET, return values.

name	datatype	description	bit
ts	int(32)	Server timestamp when data was received on the server (UTC) This is the number to use as parameter "ts" to get data that has been stored since last request.	0
date	date	GPS date of event (YYYY-MM-DD)	1
time	time	GPS time of event (HH:MM:SS)	2
nodeid	string(10)	nodeid (serialnumber) of the unit	3
tag id	string(10)	id of the tag heard	4
comment	string(100)	Comment of the tag	5

Possible returned error codes and messages:
400, "Blank Values Given"
400, "Timespan To Large"
400, "Unknown TimeZone"
400, "Value 'value' > 'length'"

401, "Unauthorized"
500, "Internal Server Error"

36.b Test form for 36.a /flexws/wolf-removed/form/

# 5	Push data

Instead of polling contantly for new data, you can let the client connect to a message queue, and have the data delivered as it arrives. This can as example be used on mobile devices to save system ressources, lower data transferred and lower power consumption. The message queue protocol used is MQTT. It is specially designed for telemetry devices. You can read more on the subject here http://mqtt.org/ and http://www.ibm.com/developerworks/webservices/library/ws-mqtt/index.html

FlexWS services for push data are in its initially phase and only UNITSTAT (33) data are currently available. Queue name is flexws/<groupguid>. See logingroup interfaces on how to do the setup and start data feed. MQTT "Client Identifier" must start with "flexws-"

6   Appendix A

Http body sample of eventhandler POST endpoint

```
[
    {
        "delay": 0,
        "high": 1,
        "low": 0,
        "type": 4,
        "text": "The function button has been activated",
        "eventHandlerHasCallbackUrl": [
            {
                "scheme": "http",
                "username": "",
                "password": "",
                "host": "host.server.com",
                "port": "",
                "path": "/get_handler/",
                "query": "k1=v1",
                "post": 0,
                "text": ""
            },
            {
                "scheme": "http",
                "username": "",
                "password": "",
                "host": "host.server.com",
                "port": "",
                "path": "/post_handler/",
                "query": "k1=v1",
                "post": 1,
                "text": ""
            }
        ],
        "eventHandlerHasEmail": [
            {
                "subject": "",
                "recipient": "me@server.com"
            }
        ],
        "eventHandlerHasSms": [
            {
                "msisdn": "+4500000000"
            }
        ]
    }
]
```

7   References

[1] http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
[2] http://en.wikipedia.org/wiki/Representational_State_Transfer
[3] http://json.org
[4] http://en.wikipedia.org/wiki/List_of_HTTP_status_codes
[5] http://www.flextrack.dk/files/FlextrackStandardTracker.pdf
[6] http://cdn.appserver.dk/files/FlextrackMiniTracker.pdf
[7] https://en.wikipedia.org/wiki/Well-known_text