



# SMS FOXBOX

Hardware SMS Gateway

---

## API & INTEGRATION GUIDE

---

HTTP API • MCP • LDAP • SNMP • Email-to-SMS  
Developer & System Administrator Reference

Applies to all models: [FoxBox S1](#) • [FoxBox S2](#) • [FoxBox RS](#)

Version 1.0 – April 2026

[www.smsfoxbox.it](http://www.smsfoxbox.it)

# Table of Contents

---

1. Introduction .....	4
2. HTTP Web Services API .....	5
2.1 Base Configuration .....	5
2.2 Authentication .....	5
Obtaining the Webservices Token .....	5
Obtaining a Token via API .....	5
Rotating the Token.....	5
2.3 API Parameters Reference .....	6
Core Parameters.....	6
Messaging Parameters .....	6
Listing Filters (for ds, in, ix, sx).....	6
2.4 Operations Reference.....	7
Send SMS (op=pv).....	7
Delivery Status (op=ds).....	7
Incoming Messages (op=in) .....	7
Inbox (op=ix).....	7
Sandbox (op=sx) — Admin only.....	7
User Credit (op=cr) .....	7
Search Contacts (op=get_contact).....	7
Search Groups (op=get_contact_group) .....	7
Quick Info (op=query) .....	8
2.5 Admin Operations.....	9
Example: Add a New User .....	9
2.6 Error Codes .....	10
2.7 Best Practices .....	10
3. MCP — Model Context Protocol.....	11
3.1 Overview .....	11
3.2 Protocol Details .....	11
Supported MCP Methods.....	11
3.3 Available Tools .....	11
sms.send .....	11
sms.incoming.list.....	11
3.4 Use Cases.....	13
AI Agents Sending SMS Autonomously .....	13
AI-driven Customer Support.....	13
Enterprise Automation & DevOps .....	13
Multi-Agent Orchestration .....	13

4. Integration Templates .....	14
4.1 Script-Based Integration .....	14
Step 1: Test the API .....	14
Step 2: Create the Alert Script (foxbox-sms.sh).....	14
Step 3: Configure Alerting .....	14
4.2 Webhook / HTTP Integration .....	14
Direct Webhook URL .....	14
POST Bridge (for POST-only systems).....	14
4.3 Email-to-SMS Integration.....	16
Step 1: Create a Mail Alias.....	16
Step 2: Create the Conversion Script.....	16
Step 3: Configure the Device .....	16
5. LDAP Authentication .....	17
5.1 Overview .....	17
5.2 Prerequisites .....	17
5.3 Configuration .....	17
Connection Parameters .....	17
User Search Parameters.....	17
Automatic User Creation .....	18
5.4 Testing the Connection.....	18
5.5 Authentication Flow .....	18
6. SNMP Remote Monitoring.....	19
6.1 Overview .....	19
6.2 Available SNMP OIDs.....	19
Modem KPIs .....	19
Service Health .....	19
6.3 SNMP Configuration.....	20
6.4 NMS Integration Examples .....	20
Nagios / Icinga .....	20
Recommended Alert Thresholds.....	20
6.5 SNMPv3 Security .....	20
7. Appendix.....	22
7.1 Quick Reference — All API Operations .....	22
7.2 Support and Resources .....	22

# 1. Introduction

---

This guide covers all integration methods available on the SMS FoxBox hardware gateway platform. The software is identical across all FoxBox models (S1, S2, RS), meaning every API endpoint, protocol and configuration described in this document applies universally.

FoxBox supports the following integration methods:

- **HTTP Web Services API** – RESTful HTTP API for sending/receiving SMS and managing the system programmatically
- **MCP (Model Context Protocol)** – AI-ready protocol for direct integration with LLMs, agents and automation platforms
- **LDAP Authentication** – centralised user authentication against enterprise directories
- **SNMP Remote Monitoring** – remote monitoring of modem KPIs and service health
- **Email-to-SMS Gateway** – convert inbound emails into outbound SMS messages
- **Script, Webhook & Email Templates** – integration with Zabbix, Nagios, Checkmk, Uptime Kuma, n8n, Zapier and more

## 2. HTTP Web Services API

### 2.1 Base Configuration

The FoxBox HTTP API is accessible at the following base path:

```
https://<FOXBOX_HOST>/index.php?app=ws
```

For local/dev environments with self-signed certificates, use curl -k. In production, always use a trusted TLS certificate. The API returns JSON by default; append format=xml for XML output.

### 2.2 Authentication

Most operations require two query parameters:

Parameter	Description
u	Username of the FoxBox account
h	Webservices token (user-scoped, obtained from Settings)
p	Password (only used with op=get_token)

#### Obtaining the Webservices Token

1. Log in to the FoxBox web interface.
2. Click your username in the top-right corner and select Settings.
3. Set “Enable webservices” to Yes and save.
4. Copy the value shown in the “Webservices token” field.
5. (Optional) Configure “Webservices IP range” to restrict access to specific IPs.

#### Obtaining a Token via API

```
curl -kG "https://<HOST>/index.php" --data-urlencode "app=ws" \
  --data-urlencode "op=get_token" --data-urlencode "u=<USER>" \
  --data-urlencode "p=<PASSWORD>"
```

#### Rotating the Token

```
curl -k "https://<HOST>/index.php?app=ws&op=set_token&u=<USER>&h=<CURRENT_TOKEN>"
```

## 2.3 API Parameters Reference

All parameter values must be URL-encoded.

### Core Parameters

Parameter	Description
op	Operation to perform (see section 2.4)
format	Output format: json (default) or xml
query	Query string for search operations
login_key	Login key for ws_login operation

### Messaging Parameters

Parameter	Description
from	Sender ID (alphanumeric, if supported by carrier)
to	Destination phone number (with international prefix, no +)
msg	Message text (URL-encoded)
footer	Footer text appended to message
nofooter	Set to 1 to suppress footer
type	Message type
unicode	Set to 1 for Unicode encoding (non-GSM characters)
schedule	Scheduled send time (format: YYYY-MM-DD HH:MM:SS, space as %20)

### Listing Filters (for ds, in, ix, sx)

Parameter	Description
queue	Filter by queue code
src	Filter by source number
dst	Filter by destination number
dt	Filter by date
smslog_id	Filter by SMS log ID
last	Return last N records
c	Maximum number of results to return

## 2.4 Operations Reference

### Send SMS (op=pv)

Send an SMS message to one or more recipients:

```
curl -kG "https://<HOST>/index.php" --data-urlencode "app=ws" \  
  --data-urlencode "u=<USER>" --data-urlencode "h=<TOKEN>" \  
  --data-urlencode "op=pv" --data-urlencode "to=393401234567" \  
  --data-urlencode "msg=Hello from FoxBox"
```

Scheduled send example:

```
...&op=pv&to=393401234567&msg=Scheduled%20msg&schedule=2026-05-01%2009:00:00
```

### Delivery Status (op=ds)

Retrieve sent message status:

```
...&op=ds&c=50&last=1000
```

### Incoming Messages (op=in)

Retrieve received messages for the authenticated user:

```
...&op=in&c=20
```

### Inbox (op=ix)

Retrieve the user's inbox contents:

```
...&op=ix&c=50
```

### Sandbox (op=sx) — Admin only

Retrieve unhandled incoming messages from the default SMS database:

```
...&u=admin&h=<ADMIN_TOKEN>&op=sx&c=10
```

### User Credit (op=cr)

Check SMS credit balance:

```
...&op=cr
```

Response example:

```
{"status":"OK","error":"","credit":1000000,"error_string":null}
```

### Search Contacts (op=get\_contact)

```
...&op=get_contact&kwd=demo&c=25
```

### Search Groups (op=get\_contact\_group)

```
...&op=get_contact_group&kwd=marketing
```

## Quick Info (op=query)

```
...&op=query
```

## 2.5 Admin Operations

The following operations require administrator credentials:

Operation	Parameter	Description
Add user	accountadd	Create a new user account
Remove user	accountremove	Delete a user account
Ban user	accountban	Suspend a user account
Unban user	accountunban	Reactivate a suspended account
Set parent	parentset	Assign a parent user
Get parent	parentget	Retrieve the parent user
View credit	creditview	View a user's credit balance
Add credit	creditadd	Allocate credits to a user
Deduct credit	creditededuct	Remove credits from a user
Account prefs	accountpref	View/set user preferences
Account config	accountconf	View/set user configuration
Set login key	loginkeyset	Set the login key for ws_login
Inject SMS	inject	Inject an SMS into the send queue
Stoplist	stoplist	Manage the SMS stop list

### Example: Add a New User

```
curl -kG "https://<HOST>/index.php" --data-urlencode "app=ws" \  
  --data-urlencode "op=accountadd" --data-urlencode "u=admin" \  
  --data-urlencode "h=<ADMIN_TOKEN>" \  
  --data-urlencode "data_status=3" \  
  --data-urlencode "data_username=newuser" \  
  --data-urlencode "data_password=S3cureP@ss" \  
  --data-urlencode "data_name=New User" \  
  --data-urlencode "data_email=user@example.com"
```

## 2.6 Error Codes

Code	Meaning
ERR 100	Authentication failed
ERR 101	Account not found or disabled
ERR 102	Account suspended
ERR 103	Insufficient credit
ERR 104	Webservices not enabled for this user
ERR 105	Webservices token not set or invalid
ERR 106	Client IP not in the authorised range
ERR 200	Missing required send parameters (to/msg)
ERR 201	Send failed (gateway error)
ERR 400	Delivery status query returned no data
ERR 401	Invalid filter parameters
ERR 402	No matching records found
ERR 501	No data available
ERR 600+	Administrative operation errors

## 2.7 Best Practices

- Always URL-encode all parameter values, especially msg and schedule fields.
- Use unicode=1 for any message containing non-GSM characters (accented letters, emojis, CJK).
- Prefer HTTPS with trusted certificates in production; avoid -k outside development.
- Rotate tokens periodically using op=set\_token.
- Configure the Webservices IP range to limit API access to known client IPs.
- For high-throughput scenarios, batch messages and monitor the queue via op=ds.

## 3. MCP — Model Context Protocol

### 3.1 Overview

FoxBox is the first SMS hardware gateway to implement MCP (Model Context Protocol), an open standard that allows AI models, LLMs and automation platforms to discover and invoke tools in a structured, machine-readable way.

With MCP, FoxBox becomes a first-class tool in AI agent architectures — directly usable by language models, multi-agent orchestration systems and workflow engines without custom integration code.

### 3.2 Protocol Details

FoxBox exposes an MCP endpoint using JSON-RPC 2.0, fully compatible with any MCP client. Authentication is handled via Bearer token or MCP parameters.

#### Supported MCP Methods

Method	Description
initialize	Handshake and capability negotiation
tools/list	Discover available tools and their schemas
tools/call	Invoke a specific tool with parameters

### 3.3 Available Tools

#### sms.send

Send outbound SMS messages via FoxBox. Supports the following parameters:

Parameter	Description
destination	MSISDN of the recipient (international format)
message	Message body text
sender_id	Sender identity (alphanumeric)
schedule	Future send time
unicode	Enable Unicode mode for non-GSM characters
external_id	External message ID for tracking

#### sms.incoming.list

Retrieve incoming SMS messages. Supports filters:

Filter	Description
sender	Filter by sender number

receiver	Filter by receiver number
keyword	Filter by keyword match
datetime	Filter by date/time range
pagination	Incremental fetch with offset/limit

## 3.4 Use Cases

### AI Agents Sending SMS Autonomously

An AI agent can dynamically decide when, to whom and what to send — and invoke `sms.send` directly via MCP, without any custom SMS integration code. Use cases include fraud alerts, appointment reminders, emergency notifications and customer engagement flows.

### AI-driven Customer Support

An LLM can read incoming SMS via `sms.incoming.list`, understand customer intent, generate an appropriate response and reply via `sms.send` — all without exposing FoxBox directly to the internet.

### Enterprise Automation & DevOps

FoxBox can serve as an action endpoint inside automation platforms: monitoring systems, CI/CD pipelines, infrastructure alerts and incident response workflows. MCP allows these systems to discover SMS capabilities automatically.

### Multi-Agent Orchestration

In advanced architectures, one agent decides strategy, another generates content, and FoxBox executes delivery via MCP — making it a composable component in multi-agent systems.

## 4. Integration Templates

### 4.1 Script-Based Integration

For monitoring systems that support external script execution (Zabbix, Nagios, Checkmk, Centreon, LibreNMS, PandoraFMS, Axence, Motadata):

#### Step 1: Test the API

```
curl -k "https://<FOXBOX_HOST>/ws/sms?u=<USER>&h=<TOKEN>" \
  "&to=393401234567&msg=test%20alert"
```

#### Step 2: Create the Alert Script (foxbox-sms.sh)

```
#!/bin/bash
FOXBOX_URL="https://<FOXBOX_HOST>"
U="<USERNAME>"
H="<TOKEN>"
TO="$1"
MESSAGE="$2"
ENC_MSG=$(python3 -c "import urllib.parse,sys; \
  print(urllib.parse.quote(sys.argv[1]))" "$MESSAGE")
curl -sS "$FOXBOX_URL/ws/sms?u=${U}&h=${H}&to=${TO}&msg=${ENC_MSG}"
```

#### Step 3: Configure Alerting

Configure your monitoring system to call the script with the recipient number and alert message as arguments.

### 4.2 Webhook / HTTP Integration

For systems supporting HTTP requests or webhooks (Uptime Kuma, Graylog, Splunk, ServiceNow, n8n, Zapier, FortiSIEM, TheHive, Slack, Teams, Synology):

#### Direct Webhook URL

```
https://<FOXBOX_HOST>/ws/sms?u=<U>&h=<H>&to={{PHONE}}&msg={{MESSAGE}}
```

#### POST Bridge (for POST-only systems)

If the external system only supports HTTP POST, deploy a lightweight bridge script:

```
from flask import Flask, request
import requests, urllib.parse
app = Flask(__name__)
FOX="https://<FOXBOX_HOST>"
U="<USER>"; H="<TOKEN>"; PHONE="<DEFAULT_PHONE>"
@app.post("/alert")
```

```
def alert():
    body = request.json
    msg = urllib.parse.quote(body.get("message", "NoMessage"))
    requests.get(f"{FOX}/ws/sms?u={U}&h={H}&to={PHONE}&msg={msg}")
    return "OK"
```

## 4.3 Email-to-SMS Integration

For systems that only support outbound email alerts (APC UPS, APC NetBotz, Liebert, Danfoss SCADA, Sauter, Schneider Electric, Niagara 4):

### Step 1: Create a Mail Alias

Configure a Postfix or Exim alias that pipes incoming email to a Python script:

```
# /etc/aliases
alerts: |/usr/local/bin/mail_to_foxbox.py
```

### Step 2: Create the Conversion Script

```
#!/usr/bin/env python3
import sys, requests, urllib.parse
FOX="https://<FOXBOX_HOST>"
U="<USER>"; H="<TOKEN>"; TO="<DEST_PHONE>"
email = sys.stdin.read()
msg = urllib.parse.quote(email[:500])
requests.get(f"{FOX}/ws/sms?u={U}&h={H}&to={TO}&msg={msg}")
```

### Step 3: Configure the Device

Set the device's alert email destination to the alias created in Step 1. Trigger a test alarm and confirm SMS delivery.

## 5. LDAP Authentication

### 5.1 Overview

FoxBox supports LDAP authentication, allowing users to log in with their enterprise directory credentials. When enabled, the system first checks local credentials, then falls back to LDAP if the local login fails.

### 5.2 Prerequisites

- The php-ldap extension must be installed and active on the FoxBox.
- (Optional) A service account DN and password for directory searches.
- Existing local user accounts, unless automatic creation is enabled.

### 5.3 Configuration

Navigate to Settings → Main configuration → LDAP authentication tab. The available options are:

#### Connection Parameters

Option	Description
Enable LDAP authentication	Enables LDAP fallback after local auth fails (set to Yes)
LDAP server URL	Server address (e.g. ldap.example.org or ldaps://ldap.example.org)
LDAP server port	TCP port (blank = default based on schema)
LDAP encryption	None, LDAPS (SSL) or StartTLS
LDAP bind DN	Full DN of the service account for bind (blank = anonymous)
LDAP bind password	Password for the bind user

#### User Search Parameters

Option	Description
LDAP base DN	Starting point for search (e.g. ou=People,dc=example,dc=org)
LDAP user search filter	Filter with {{username}} placeholder (default: uid={{username}})
LDAP name attribute	Attribute mapped to Full Name (default: cn)
LDAP email attribute	Attribute mapped to email (default: mail)
LDAP mobile attribute	Attribute for mobile number (optional)
Default email domain	Suffix added if email is empty (e.g. @example.org)

## Automatic User Creation

When “Automatically create users from LDAP” is set to Yes, FoxBox creates a local user on successful LDAP login, populating name, email, mobile number and default parameters (ACL, credit, status). Each subsequent LDAP login updates the local profile automatically.

## 5.4 Testing the Connection

1. Enter all LDAP parameters in the configuration tab.
2. In the Test username / Test password section, enter LDAP credentials (or leave blank to test bind only).
3. Click “Test LDAP connection” to verify connectivity.

## 5.5 Authentication Flow

1. FoxBox checks local credentials first.
2. If local login fails and LDAP is enabled, the system performs an LDAP search and bind.
3. On successful LDAP authentication, the local user is created (if allowed) or updated.
4. The login is accepted and access is granted.

## 6. SNMP Remote Monitoring

### 6.1 Overview

FoxBox exposes modem KPIs and core service state via SNMP, enabling remote monitoring with platforms such as SolarWinds, PRTG, Nagios and Icinga.

### 6.2 Available SNMP OIDs

#### Modem KPIs

OID Name	Description	Values
modemState	Modem registration status	0=down, 1=registered
modemBars	Signal strength bars	0–4
modemRSRP	Reference Signal Received Power	dBm (e.g. -105)
modemRSRQ	Reference Signal Received Quality	dB
modemRSSI	Received Signal Strength Indicator	dBm
modemBand	Active LTE band	Band number
modemRAT	Radio Access Technology	LTE, 3G, etc.

#### Service Health

OID Name	Service	Values
svcNginx	Nginx web server	1=up, 0=down
svcKannelBear	Kannel bearerbox	1=up, 0=down
svcKannelSms	Kannel smsbox	1=up, 0=down
svcKannelQueue	Highwayd queue manager	1=up, 0=down
svcDockerFoxbox	FoxBox Docker container	1=up, 0=down
svcDockerNginx	Nginx proxy container	1=up, 0=down

## 6.3 SNMP Configuration

Edit /etc/snmp/snmpd.conf on the FoxBox:

```
rocommunity foxbox 192.168.0.0/24
```

Add extend entries for each OID:

```
extend modemState /usr/local/bin/snmp_modem_state.sh
extend modemBars /usr/local/bin/snmp_modemBars.sh
extend modemRSRP /usr/local/bin/snmp_modem_rsrp.sh
extend modemRSRQ /usr/local/bin/snmp_modem_rsrq.sh
extend modemRSSI /usr/local/bin/snmp_modem_rssi.sh
extend modemBand /usr/local/bin/snmp_modem_band.sh
extend modemRAT /usr/local/bin/snmp_modem_rat.sh
extend svcNginx /usr/local/bin/snmp_svc.sh nginx
extend svcKannelBear /usr/local/bin/snmp_svc.sh kannel-bearerbox
extend svcKannelSms /usr/local/bin/snmp_svc.sh kannel-smsbox
```

Restart SNMP daemon:

```
sudo systemctl restart snmpd
```

Test locally:

```
snmpwalk -v2c -c foxbox localhost 1.3.6.1.4.1.8072.1.3.2
```

## 6.4 NMS Integration Examples

### Nagios / Icinga

```
define service{
    host_name          foxbox
    service_description Modem State
    check_command      check_snmp_num!<OID>!1:1!0:0
}
```

### Recommended Alert Thresholds

Metric	Warning	Critical
modemState		= 0 (not registered)
modemBars	< 2	= 0
modemRSRP	≤ -110 dBm	≤ -115 dBm
svc* services		= 0 (service down)

## 6.5 SNMPv3 Security

For production environments, use SNMPv3 with authentication and encryption:

```
sudo net-snmp-create-v3-user -ro -a SHA -A 'AuthPass123' \
```

```
-x AES -X 'PrivPass123' snmpv3user
```

Configure your NMS with the same credentials (username, auth/priv algorithms and passwords). Restrict UDP/161 access via firewall to the NMS IP only.

## 7. Appendix

### 7.1 Quick Reference — All API Operations

Operation	op	Auth	Description
Send SMS	pv	user	Send message (supports scheduling, unicode)
Delivery status	ds	user	List sent messages with status
Incoming	in	user	List received messages
Inbox	ix	user	User inbox contents
Sandbox	sx	admin	Unhandled incoming messages
Credit	cr	user	Check credit balance
Get token	get_token	pass	Generate token using password
Set token	set_token	user	Rotate/regenerate token
Search contacts	get_contact	user	Search phonebook
Search groups	get_contact_group	user	Search contact groups
WS login	ws_login	key	Login via webservice login_key
Quick info	query	user	System status information
Add user	accountadd	admin	Create user account
Remove user	accountremove	admin	Delete user account
Ban user	accountban	admin	Suspend account
Unban user	accountunban	admin	Reactivate account
Add credit	creditadd	admin	Allocate SMS credits
Deduct credit	creditededuct	admin	Remove SMS credits

### 7.2 Support and Resources

Resource	URL
API Documentation	<a href="http://api.foxbox.it">api.foxbox.it</a>
Manuals & Guides	<a href="http://www.smsfoxbox.it/sms-api-manuals">www.smsfoxbox.it/sms-api-manuals</a>
AI Agent / MCP Info	<a href="http://www.smsfoxbox.it/ai-agent-ready-sms-api">www.smsfoxbox.it/ai-agent-ready-sms-api</a>
Support Tickets	<a href="http://www.smsfoxbox.it/support">www.smsfoxbox.it/support</a>
Old Legacy Documentation	<a href="http://legacy.smsfoxbox.it">legacy.smsfoxbox.it</a>
Email Support	<a href="mailto:sales@foxbox.it">sales@foxbox.it</a>