

# FinSMS API Documentation

## v1.0

---

Last Updated: January 2025

API Version: v1.0

Base URL: <https://platform.finsms.io/api/v1/sms>

---

## Table of Contents

---

1. [Overview](#)
2. [Authentication](#)
3. [Endpoints](#)
  - [Send SMS Simple](#)
  - [Send SMS](#)
  - [Send SMS MULTI](#)
  - [SMS Report](#)
4. [Error Codes](#)
5. [Credit System](#)
6. [Phone Number Formats](#)

7. [SMS States](#)
  8. [Network Operators](#)
  9. [Example Usage](#)
  10. [Security](#)
  11. [Support](#)
- 

## Overview

---

FinSMS API provides SMS sending and reporting services via RESTful endpoints. The API follows a similar structure to CepSMS API for easy integration.

**Base URL:** `https://platform.finsms.io/api/v1/sms`

**Content-Type:** `application/json`

**Method:** All endpoints use `POST` method

---

## Authentication

---

All API requests require `User` (API Key) and `Pass` (API Secret) parameters in the request body.

## How to Get API Credentials

Contact your FinSMS administrator to obtain your API Key and API Secret. These credentials are provided by the administrator through the admin panel.

## Authentication Format

Include `User` and `Pass` in every request body:

```
{  
  "User": "your_api_key",  
  "Pass": "your_api_secret",  
  ...  
}
```

## Endpoints

### 1. Send SMS Simple

**Endpoint:** `POST /api/v1/sms/send`

**Description:** Send a simple SMS message to one or more phone numbers.

**Request Body:**

```
{  
  "User": "API_KEY",  
  "Pass": "API_SECRET",  
  "Message": "selam test",  
  "Numbers": ["905321234567"]  
}
```

## Parameters:

- `User` (required): Your API Key
- `Pass` (required): Your API Secret
- `Message` (required): SMS message text (max 180 characters per credit)
- `Numbers` (required): Array of phone numbers (at least one number required)

## Response (Success):

```
{  
  "MessageId": "42367313232",  
  "Status": "OK"  
}
```

## Response (Error):

```
{  
  "MessageId": 0,  
  "Status": "Error"  
}
```

## Example cURL:

```
curl -X POST https://platform.finsms.io/api/v1/sms/send  
\  
-H "Content-Type: application/json" \  
-d '{  
  "User": "your_api_key",  
  "Pass": "your_api_secret",  
  "Message": "Test message",  
  "Numbers": ["905321234567"]  
}'
```

## Example JavaScript:

```
const response = await fetch('https://platform.finsms.io/api/v1/sms/send', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
  },
  body: JSON.stringify({
    User: 'your_api_key',
    Pass: 'your_api_secret',
    Message: 'Hello, this is a test message.',
    Numbers: ['905321234567']
  })
});

const result = await response.json();

if (result.Status === 'OK') {
  console.log('SMS sent:', result.MessageId);
} else {
  console.error('Error:', result.Status);
}
```

## 2. Send SMS

**Endpoint:** POST /api/v1/sms/send-advanced

**Description:** Send an advanced SMS message with custom sender name, coding, start date, and validity period.

## Request Body:

```
{  
    "From": "Baslik",  
    "User": "API_KEY",  
    "Pass": "API_SECRET",  
    "Message": "selam test",  
    "Coding": "default",  
    "StartDate": null,  
    "ValidityPeriod": 1140,  
    "Numbers": ["905321234567"]  
}
```

## Parameters:

- `User` (required): Your API Key
- `Pass` (required): Your API Secret
- `Message` (required): SMS message text
- `Numbers` (required): Array of phone numbers
- `From` (optional): Sender name (max 11 characters)
- `Coding` (optional): Message coding - `"default"` or `"turkish"` (default: `"default"`)
- `StartDate` (optional): Send date in ISO format (null = send immediately)
- `ValidityPeriod` (optional): Validity period in minutes (default: 1140)

## Response:

```
{  
  "MessageId": "42367313232",  
  "Status": "OK"  
}
```

## Example cURL:

```
curl -X POST https://platform.finsms.io/api/v1/sms/send-  
advanced \  
-H "Content-Type: application/json" \  
-d ' {  
  "From": "FinSMS",  
  "User": "your_api_key",  
  "Pass": "your_api_secret",  
  "Message": "Test message",  
  "Coding": "turkish",  
  "Numbers": ["905321234567"]  
}'
```

## 3. Send SMS MULTI

**Endpoint:** `POST /api/v1/sms/send-multi`

**Description:** Send multiple distinct SMS messages to different phone numbers in a single request.

**Request Body:**

```
{
    "From": "Baslik",
    "User": "API_KEY",
    "Pass": "API_SECRET",
    "Coding": "default",
    "StartDate": null,
    "ValidityPeriod": 1440,
    "Messages": [
        {
            "Message": "test mesaj 1",
            "GSM": "905321234567"
        },
        {
            "Message": "test mesaj 2",
            "GSM": "905441234567"
        }
    ]
}
```

## Parameters:

- **User** (required): Your API Key
- **Pass** (required): Your API Secret
- **Messages** (required): Array of message objects, each containing:
  - **Message** (required): SMS message text
  - **GSM** (required): Phone number

- `From` (optional): Sender name
- `Coding` (optional): Message coding - `"default"` or `"turkish"`
- `StartDate` (optional): Send date in ISO format
- `ValidityPeriod` (optional): Validity period in minutes

### Response (Single Message):

```
{  
  "MessageId": "42367313232",  
  "Status": "OK"  
}
```

### Response (Multiple Messages):

```
{  
  "MessageIds": ["42367313232", "42367313233"],  
  "Status": "OK",  
  "SuccessCount": 2,  
  "FailedCount": 0  
}
```

### Example cURL:

```
curl -X POST https://platform.finsms.io/api/v1/sms/send-
multi \
-H "Content-Type: application/json" \
-d '{
  "User": "your_api_key",
  "Pass": "your_api_secret",
  "Messages": [
    {
      "Message": "Message 1",
      "GSM": "905321234567"
    },
    {
      "Message": "Message 2",
      "GSM": "905441234567"
    }
  ]
}'
```

---

## 4. SMS Report

**Endpoint:** `POST /api/v1/sms/report`

**Description:** Get delivery report for a sent SMS message.

**Request Body:**

```
{  
  "User": "API_KEY",  
  "Pass": "API_SECRET",  
  "MessageId": "42367313232"  
}
```

## Parameters:

- `User` (required): Your API Key
- `Pass` (required): Your API Secret
- `MessageId` (required): Message ID returned from send endpoints

## Response:

```
{  
    "Status": "OK",  
    "Report": [  
        {  
            "GSM": "905321234567",  
            "State": "İletildi",  
            "Network": "Turkcell"  
        },  
        {  
            "GSM": "905323214567",  
            "State": "İletildi",  
            "Network": "Turkcell"  
        }  
    ]  
}
```

## Response (Error):

```
{  
    "Status": "Error",  
    "Error": "SMS mesajı bulunamadı",  
    "Report": []  
}
```

## Example cURL:

```
curl -X POST  
https://platform.finsms.io/api/v1/sms/report \  
-H "Content-Type: application/json" \  
-d '{  
    "User": "your_api_key",  
    "Pass": "your_api_secret",  
    "MessageId": "42367313232"  
}'
```

## Example JavaScript:

```

const response = await
fetch('https://platform.finsms.io/api/v1/sms/report', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
  },
  body: JSON.stringify({
    User: 'your_api_key',
    Pass: 'your_api_secret',
    MessageId: '42367313232'
  })
}) ;

const result = await response.json();
if (result.Status === 'OK') {
  result.Report.forEach(report => {
    console.log(`GSM: ${report.GSM}, State: ${report.State}, Network: ${report.Network}`);
  });
}

```

## Error Codes

HTTP Status	Description
200	Success

400	Bad Request (missing parameters, insufficient credit, invalid phone number, etc.)
401	Unauthorized (invalid API Key/Secret)
404	Not Found (message not found in report endpoint)
500	Internal Server Error

## Common Error Responses:

```
{  
  "MessageId": 0,  
  "Status": "Error"  
}
```

```
{  
  "Status": "Error",  
  "Error": "Error message description",  
  "Report": []  
}
```

## Credit System

- 180 characters = 1 credit
- Credit is calculated based on message length for each SMS

- For example:
    - 1-180 characters = 1 credit
    - 181-360 characters = 2 credits
    - 361-540 characters = 3 credits
  - Returns **400** error if insufficient credit
  - Failed SMS credits are automatically refunded after 48 hours
  - Admin users have unlimited credits (no deduction)
- 

## Phone Number Formats

---

Accepted formats:

- **905321234567** (12 digits, starts with 90)
- **05321234567** (11 digits, starts with 0)
- **5321234567** (10 digits, starts with 5)

**Note:** All formats are automatically normalized to **905xxxxxxxx** format internally.

---

## SMS States

---

The following states are returned in SMS reports:



State	Description
Rapor Bekliyor	SMS sent, waiting for delivery report
İletildi	SMS delivered successfully
İletilmmedi	SMS not delivered
Zaman Aşımı	SMS timeout (delivery timeout)

---

## Network Operators

---

The following network operators are supported:

- [TTMobile](#)
  - [Turkcell](#)
  - [Vodafone](#)
  - [KKTCell](#)
  - [Telsim](#)
  - [Şebeke Dışı](#) (Network Out of Range)
- 

## Example Usage

---

Complete Example: Send and Check Status

```
// Step 1: Send SMS

const sendResponse = await
fetch('https://platform.finsms.io/api/v1/sms/send', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
  },
  body: JSON.stringify({
    User: 'your_api_key',
    Pass: 'your_api_secret',
    Message: 'Hello, this is a test message.',
    Numbers: ['905321234567']
  })
}) ;

const sendResult = await sendResponse.json();

if (sendResult.Status === 'OK') {
  console.log('SMS sent successfully! MessageId:', sendResult.MessageId);

  // Step 2: Wait a few seconds for processing
  await new Promise(resolve => setTimeout(resolve, 5000));
}

// Step 3: Check SMS status
const reportResponse = await
fetch('https://platform.finsms.io/api/v1/sms/report', {
  method: 'POST',
```

```

headers: {
    'Content-Type': 'application/json',
},
body: JSON.stringify({
    User: 'your_api_key',
    Pass: 'your_api_secret',
    MessageId: sendResult.MessageId
})
);

const reportResult = await reportResponse.json();

if (reportResult.Status === 'OK') {
    reportResult.Report.forEach(report => {
        console.log(`Phone: ${report.GSM}`);
        console.log(`Status: ${report.State}`);
        console.log(`Network: ${report.Network}`);
    });
}
} else {
    console.error('SMS send failed:', sendResult);
}

```

## Python Example

```

import requests
import time

API_BASE_URL = "https://platform.finsms.io/api/v1/sms"

```

```
API_KEY = "your_api_key"

API_SECRET = "your_api_secret"

# Send SMS

send_data = {

    "User": API_KEY,
    "Pass": API_SECRET,
    "Message": "Test message from Python",
    "Numbers": ["905321234567"]
}

response = requests.post(f"{API_BASE_URL}/send",
    json=send_data)
result = response.json()

if result.get("Status") == "OK":

    message_id = result.get("MessageId")
    print(f"SMS sent! MessageId: {message_id}")

    # Wait for processing
    time.sleep(5)

    # Check status
    report_data = {

        "User": API_KEY,
        "Pass": API_SECRET,
        "MessageId": message_id
    }

    report_response = requests.post(f"
```

```
{API_BASE_URL}/report", json=report_data)

report_result = report_response.json()

if report_result.get("Status") == "OK":

    for report in report_result.get("Report", []):

        print(f"Phone: {report['GSM']}, Status:
{report['State']}, Network: {report['Network']}")

else:

    print(f"Error: {result}")
```

---

## Security

---

### 1. Keep API Credentials Secure

- Never use API Key and Secret in public code or client-side applications
- Store Secret only on server-side
- Use environment variables to store credentials
- Rotate API keys regularly

### 2. Use HTTPS

- All API requests must be made over HTTPS
- Never send credentials over unencrypted connections

### 3. Rate Limiting

- Rate limiting is applied to API requests
- Returns `429 Too Many Requests` error for excessive requests
- Contact support if you need higher rate limits

## 4. Best Practices

- Validate phone numbers before sending
  - Check credit balance before sending large batches
  - Implement retry logic for failed requests
  - Log all API requests for audit purposes
- 

## Support

---

For questions, issues, or support:

- Email: [support@finsms.io](mailto:support@finsms.io)
  - Documentation: <https://docs.finsms.io>
  - Status Page: <https://status.finsms.io>
- 

## Changelog

---

v1.0 (January 2025)

- Initial release
  - Send SMS Simple endpoint
  - Send SMS Advanced endpoint
  - Send SMS MULTI endpoint
  - SMS Report endpoint
  - API Key authentication
  - Automatic refund system for failed SMS
- 

© 2025 FinSMS. All rights reserved.