

INSCRIBEN

REST API Documentation

Version 1.0 | Base URL: <https://inscriben.com>

Generated: October 25, 2026 | NXFIN TECHNOLOGIES PRIVATE LIMITED, India

This document provides the complete reference for the Inscriben REST API. Use this API to integrate electronic signature workflows into your applications — send documents for signature, manage signatories, track document status, and verify signed documents.

Table of Contents

1. Authentication
2. Security Features
3. API Endpoints
4. Error Codes
5. Rate Limits
6. Integration Guide

1. Authentication

All API requests require a Bearer token in the Authorization header. Generate API keys from your API Keys settings page in the Inscriben dashboard.

Request Header

```
Authorization: Bearer ss_YOUR_API_KEY
```

API Key Prefix

All Inscriben API keys begin with the prefix `ss_` followed by a unique token. Copy your key immediately after creation — it will not be shown again.

API Key Scopes

Full Access

All operations: create documents, send reminders, void documents (GET, POST, DELETE).

Read Only

Read-only access to documents, credits, verification, and audit trails (GET only).

2. Security Features

- **IP Allowlisting**
Restrict API access to specific IP addresses or CIDR ranges
- **Domain Restrictions**
Limit API key usage to specific referring domains
- **Key Expiration**
Set automatic expiration dates on API keys
- **Rate Limiting**
100 requests per minute per API key
- **AES-256 Encryption**
All documents are encrypted at rest with per-document initialization vectors
- **Audit Trail**
Every action is logged with timestamps, IP addresses, and user agents

3. API Endpoints

Base URL: <https://inscriben.com>

POST /api/v1/documents/send-for-signature

Full Access

Upload a document and add signatories for e-signature. Supports three signature stamp placement modes: 'default' (bottom grid, no action required), 'manual' (caller specifies exact coordinates per signatory), or 'auto' (server scans PDF for signature markers and places stamps automatically, falling back to bottom-right corner). Placement coordinates are percentages (0–100) of the page dimensions: { page, x, y, width, height }.

Form Parameters

file	File	Required	The PDF document to upload
name	string	Required	Display name for the document
securityLevel	string	Optional	Security level: 'Standard' (default) or 'Advanced'
signingMode	string	Optional	'sequential' (default for 2+ signatories) or 'parallel'
placementMode	string	Optional	'default' 'manual' 'auto'. Controls where signature stamps appear. Default: 'default' (bottom grid). 'manual': provide placements[] per signatory. 'auto': server detects signature markers in PDF text; if no markers are found, stamps fall back to a bottom-right-first grid (single signatory lands at x"H59%, additional signatories fill leftward). If fewer markers are found than signatories, unmatched signatories get the standard bottom-left-first grid.
signatories	JSON array	Required	Array of { name, email, order?, placements? }. 'placements' is required when placementMode=manual. Each placement: { page ("e1), x (0–100), y (0–100), width (0–100), height (0–100) } — all values are % of page dimensions.
sendNotifications	string	Optional	'true' (default) to email signing invitations, 'false' to suppress

Response

```
{
  "documentId": "RqUAhY",
  "name": "Service Agreement",
  "hash": "sha256:a1b2c3...",
  "signingMode": "sequential",
  "placementMode": "manual",
  "signatories": [
    {
      "name": "John Doe",
      "email": "john@example.com",
      "signingUrl": "https://inscriben.com/sign/tok_xyz789",
      "effectivePlacements": [
        {
          "page": 1,
          "x": 60,
          "y": 80,
          "width": 35,
          "height": 8
        }
      ]
    }
  ]
}
```

cURL Example

```
# Mode 1 – default (bottom grid, no extra params needed):
curl -X POST https://inscriben.com/api/v1/documents/send-for-signature \
  -H "Authorization: Bearer ss_YOUR_API_KEY" \
  -F "file=@contract.pdf" \
  -F 'name=Service Agreement' \
  -F 'signatories=[{"name":"John Doe","email":"john@example.com","order":1}]'
```

```
# Mode 2 – manual (specify exact stamp coordinates per signatory):
curl -X POST https://inscriben.com/api/v1/documents/send-for-signature \
  -H "Authorization: Bearer ss_YOUR_API_KEY" \
  -F "file=@contract.pdf" \
  -F 'name=Service Agreement' \
  -F 'placementMode=manual' \
  -F 'signatories=[{"name":"John","email":"john@example.com","order":1,"placements":
[{"page":1,"x":60,"y":80,"width":35,"height":8}]]]'
```

```
# Mode 3 – auto (server finds signature markers in PDF text):
curl -X POST https://inscriben.com/api/v1/documents/send-for-signature \
  -H "Authorization: Bearer ss_YOUR_API_KEY" \
  -F "file=@contract.pdf" \
  -F 'name=Service Agreement' \
  -F 'placementMode=auto' \
  -F 'signatories=[{"name":"John","email":"john@example.com","order":1}]'
```

GET /api/v1/documents

Read Only

List all documents with pagination and optional status filter

Query Parameters

status	string	Optional	Filter by status: 'pending', 'completed', 'voided'
page	integer	Optional	Page number (default: 1)
limit	integer	Optional	Items per page (default: 20, max: 100)

Response

```
{
  "documents": [
    {
      "id": "doc_abc123",
      "name": "Service Agreement",
      "status": "pending",
      "createdAt": "2026-02-13T10:00:00Z"
    }
  ],
  "total": 42,
  "page": 1,
  "limit": 20
}
```

cURL Example

```
curl -X GET \
  "https://inscriben.com/api/v1/documents?status=pending&page=1&limit=20" \
  -H "Authorization: Bearer ss_YOUR_API_KEY"
```

GET /api/v1/documents/:documentId

Read Only

Get detailed information about a specific document

Response

```
{
  "id": "doc_abc123",
  "name": "Service Agreement",
  "status": "pending",
  "hash": "sha256:a1b2c3d4e5f6...",
  "createdAt": "2026-02-13T10:00:00Z",
  "signatories": [
    {
      "name": "John Doe",
      "email": "john@example.com",
      "status": "pending",
      "signedAt": null
    }
  ],
  "auditSummary": [
    {
      "action": "DOCUMENT_CREATED",
      "timestamp": "2026-02-13T10:00:00Z"
    }
  ]
}
```

cURL Example

```
curl -X GET \
  "https://inscriben.com/api/v1/documents/doc_abc123" \
  -H "Authorization: Bearer ss_YOUR_API_KEY"
```

GET /api/v1/documents/:documentId/download

Read Only

Download the original uploaded document

Response

Binary file stream (Content-Type: application/pdf)

cURL Example

```
curl -X GET \  
  "https://inscriben.com/api/v1/documents/doc_abc123/download" \  
  -H "Authorization: Bearer ss_YOUR_API_KEY" \  
  -o document.pdf
```

GET /api/v1/documents/:documentId/signed

Read Only

Download the signed PDF version of the document. Returns 404 if signing is not yet complete.

Response

Binary file stream (Content-Type: application/pdf). Returns 404 if signed version is not yet available.

cURL Example

```
curl -X GET \  
  "https://inscriben.com/api/v1/documents/doc_abc123/signed" \  
  -H "Authorization: Bearer ss_YOUR_API_KEY" \  
  -o document-signed.pdf
```

GET /api/v1/documents/:documentId/audit

Read Only

Get the full audit trail for a document

Response

```
{
  "documentId": "doc_abc123",
  "documentName": "Service Agreement",
  "auditTrail": [
    {
      "action": "DOCUMENT_CREATED",
      "details": "Document uploaded via API",
      "timestamp": "2026-02-13T10:00:00Z",
      "ipAddress": "192.168.1.1"
    },
    {
      "action": "SIGNATURE_REQUESTED",
      "details": "Signature request sent to john@example.com",
      "timestamp": "2026-02-13T10:00:01Z",
      "ipAddress": "192.168.1.1"
    }
  ]
}
```

cURL Example

```
curl -X GET \
  "https://inscriben.com/api/v1/documents/doc_abc123/audit" \
  -H "Authorization: Bearer ss_YOUR_API_KEY"
```

POST /api/v1/documents/:documentId/reminders

Full Access

Send signing reminders to pending signatories. Optionally target a specific signatory by email.

Request Body

<code>signatoryEmail</code>	<code>string</code>	Optional	Send reminder to a specific signatory. If omitted, all pending signatories receive a reminder.
-----------------------------	---------------------	----------	--

Response

```
{
  "sent": 1,
  "signatories": [
    {
      "name": "John Doe",
      "email": "john@example.com",
      "status": "reminder_sent"
    }
  ]
}
```

cURL Example

```
curl -X POST \
  "https://inscriben.com/api/v1/documents/doc_abc123/reminders" \
  -H "Authorization: Bearer ss_YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"signatoryEmail": "john@example.com"}'
```

DELETE /api/v1/documents/:documentId

Full Access

Void/cancel a pending document. Only works on documents that are not yet fully signed.

Response

```
{
  "message": "Document voided successfully",
  "documentId": "doc_abc123"
}
```

cURL Example

```
curl -X DELETE \
  "https://inscriben.com/api/v1/documents/doc_abc123" \
  -H "Authorization: Bearer ss_YOUR_API_KEY"
```

GET /api/v1/credits

Read Only

Check your current credit balance and available packs

Response

```
{
  "freeCreditsRemaining": 8,
  "paidCreditsAvailable": 50,
  "packs": [
    {
      "id": "pack_1",
      "credits": 50,
      "purchasedAt": "2026-01-15T00:00:00Z"
    }
  ]
}
```

cURL Example

```
curl -X GET \
  "https://inscriben.com/api/v1/credits" \
  -H "Authorization: Bearer ss_YOUR_API_KEY"
```

GET /api/v1/verify/:documentId

Read Only

Verify the authenticity and integrity of a signed document

Response

```
{
  "verified": true,
  "documentId": "doc_abc123",
  "name": "Service Agreement",
  "status": "completed",
  "hash": "sha256:a1b2c3d4e5f6...",
  "signatories": [
    {
      "name": "John Doe",
      "email": "john@example.com",
      "status": "signed",
      "signedAt": "2026-02-13T12:00:00Z"
    }
  ]
}
```

cURL Example

```
curl -X GET \
  "https://inscriben.com/api/v1/verify/doc_abc123" \
  -H "Authorization: Bearer ss_YOUR_API_KEY"
```

4. Error Codes

The API uses standard HTTP status codes. All error responses include a JSON body with a "message" field.

Error Response Format

```
{
  "message": "Description of what went wrong"
}
```

Code	Status	Description
401	Unauthorized	Invalid or missing API key
403	Forbidden	Insufficient scope, IP/domain restriction, or access denied
404	Not Found	Resource not found
402	Payment Required	Insufficient credits
429	Too Many Requests	Rate limit exceeded (100 requests/minute)
500	Internal Server Error	An unexpected error occurred on the server

5. Rate Limits

API requests are rate limited to 100 requests per minute per API key. If you exceed this limit, you will receive a 429 Too Many Requests response.

Rate Limit Headers

```
X-RateLimit-Limit: 100
X-RateLimit-Remaining: 95
X-RateLimit-Reset: 1707800400
```

6. Integration Guide

Step 1: Create an Organization

Register on Inscriben and create or join an organization from your dashboard.

Step 2: Generate an API Key

Navigate to API Keys in the sidebar. Create a key with the appropriate scope (Full Access or Read Only). Copy the key immediately.

Step 3: Purchase Credits

Go to Billing and purchase a signature pack (Starter: 100 credits for \$20, or Business: 1,000 credits for \$150). New organizations receive free starter credits.

Step 4: Send Your First Document

Use the POST `/api/v1/documents/send-for-signature` endpoint to upload a document and define signatories.

Step 5: Track Signing Status

Poll GET `/api/v1/documents/:documentId` to check the document status. Or configure webhooks in your dashboard for real-time notifications.

Step 6: Download Signed Documents

Once all signatories have signed, use GET `/api/v1/documents/:documentId/signed` to download the completed PDF with embedded signatures.

Step 7: Verify Document Authenticity

Use GET `/api/v1/verify/:documentId` to verify the integrity and authenticity of any signed document.

Webhook Events

Configure webhooks from your dashboard to receive real-time notifications for the following events:

- `document.created` — A new document has been created and sent for signature
- `document.completed` — All signatories have signed the document
- `signatory.signed` — A signatory has completed their signature
- `signatory.rejected` — A signatory has rejected/declined to sign

Webhook Payload Example

```
{
  "event": "document.completed",
  "documentId": "doc_abc123",
  "timestamp": "2026-02-13T12:00:00Z",
  "data": {
    "name": "Service Agreement",
    "status": "signed"
  }
}
```


