Email Verification API Documentation

March 26, 2025

Contents

1 Introduction

This document provides comprehensive documentation for the Email Verification API, including authentication methods, endpoint specifications, and usage examples.

2 Authentication

The API uses JWT (JSON Web Token) authentication:

$$\text{Authentication} = \text{Bearer } + \text{JWT Token}$$

$$\text{Headers} = \{\text{"Authorization": "Bearer <token>"}\}$$

3 API Endpoints

`Endpoint`	Method	Description	Auth Required
Auth Endpoints			
`/api/auth/login`	POST	Authenticate user and get tokens	No
`/api/auth/register`	POST	Register a new user	No
`/api/auth/refresh`	POST	Refresh access token	Yes
`/api/auth/me`	GET	Get current user information	Yes
`/api/auth/users`	GET	Get all users (admin only)	Yes (Admin)
`/api/auth/users`	POST	Create a new user (admin only)	Yes (Admin)
`/api/auth/users/<username>`	PUT	Update a user (admin only)	Yes (Admin)
`/api/auth/users/<username>`	DELETE	Delete a user (admin only)	Yes (Admin)
`/api/auth/change-password`	POST	Change current user's password	Yes
`/api/auth/permissions`	GET	Get current user's permissions	Yes
`/api/auth/permissions/<username>`	PUT	Update a user's permissions (admin only)	Yes (Admin)
Verification Endpoints			
`/api/verification/single`	POST	Verify a single email address	Yes
`/api/verification/batch`	POST	Start a batch verification process	Yes
`/api/verification/batch/<job_id>`	GET	Get status of a batch verification job	Yes
`/api/verification/batch/<job_id>/results`	GET	Get results of a batch verification job	Yes
`/api/verification/batch`	GET	Get all batch verification jobs	Yes
`/api/verification/batch/<job_id>`	DELETE	Delete a batch verification job	Yes
`/api/verification/terminal`	POST	Run terminal controller for batch verification	Yes
`/api/verification/terminal/<job_id>`	GET	Get status of a terminal controller job	Yes
`/api/verification/terminal`	GET	Get all terminal controller jobs	Yes
`/api/verification/terminal/<job_id>`	DELETE	Delete a terminal controller job	Yes
`/api/verification/upload`	POST	Upload a CSV file for verification	Yes
`/api/verification/download`	GET	Download verification results by category	Yes
Statistics Endpoints			
`/api/statistics/summary`	GET	Get verification statistics summary	Yes
`/api/statistics/domain`	GET	Get domain statistics	Yes
`/api/statistics/reason`	GET	Get reason statistics	Yes
`/api/statistics/reasons-from-csv`	GET	Get reasons from CSV files	Yes
`/api/statistics/history`	GET	Get verification history for an email	Yes
`/api/statistics/category-history`	GET	Get verification history for a category	Yes
Admin Endpoints			
`/api/admin/settings`	GET	Get application settings	Yes (Admin)
`/api/admin/settings`	POST	Update application settings	Yes (Admin)

4 Request and Response Examples

4.1 Authentication

```
POST /api/auth/login
Content-Type: application/json

{
  "username": "user@example.com",
  "password": "password123"
}
```

Listing 1: Login Request

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIs...",
  "refresh_token": "eyJhbGciOiJIUzI1NiIs...",
  "user": {
    "id": "123e4567-e89b-12d3-a456-426614174000",
    "username": "user",
    "email": "user@example.com",
    "role": "user"
  }
}
```

Listing 2: Login Response

4.2 Email Verification

```
POST /api/verification/single
Authorization: Bearer eyJhbGciOiJIUzI1NiIs...
Content-Type: application/json

{
  "email": "test@example.com"
}
```

Listing 3: Single Verification Request

```
{
  "email": "test@example.com",
  "is_valid": true,
  "status": "deliverable",
  "verification_details": {
    "smtp_check": true,
    "domain_check": true,
    "disposable": false
  },
  "timestamp": "2025-03-26T14:30:45Z"
}
```

Listing 4: Verification Response

5 API Improvements

5.1 API Versioning

Implement API versioning to ensure backward compatibility:

/api/v1/verification/single

5.2 Rate Limiting

- 100 requests/minute for authenticated users
- 10 requests/minute for unauthenticated requests
- 5 requests/second for batch operations

5.3 Webhook Support

```
POST /api/webhooks/register
Content-Type: application/json

{
  "url": "https://example.com/webhook",
  "events": ["verification.completed", "batch.completed"]
}
```

Listing 5: Webhook Registration

6 Implementation Guide

6.1 External API Access

1. Enable CORS in Flask:

```python
from flask_cors import CORS

def create_app():
    app = Flask(__name__)
    CORS(app, resources={r"/api/*": {"origins": "*"}})
    return app
```

2. Add API key authentication:
   - Create API key model and endpoints
   - Implement key-based rate limiting
3. Document with Swagger:
   - Install Flask-RESTX or flasgger
   - Add endpoint annotations
   - Host at `/api/docs`