

# Voclio System - API Documentation

Complete API Reference Documentation

Version 1.0.0 | Generated on 2024-01-15

**Version:** 1.0.0

**Base URL:** <https://api.voclio.com/api>

**Authentication:** Bearer Token (JWT)

---

## Table of Contents

1. Authentication Endpoints
  2. Voice & Notes Endpoints
  3. Task Management Endpoints
  4. Calendar Sync Endpoints
  5. Tag Endpoints
  6. Focus Mode Endpoints
  7. Productivity Endpoints
  8. Backup & Restore Endpoints
  9. Reminder Endpoints
  10. Notification Endpoints
  11. Settings Endpoints
  12. Admin Endpoints
- 

## Authentication Endpoints

**POST** /api/auth/register

Register a new user account.

**Request Body:**

```
{
  "email": "user@example.com",
  "password": "SecurePassword123!",
  "name": "John Doe",
  "phone_number": "+1234567890"
}
```

**Response (201 Created):**

```
{
  "success": true,
  "message": "User registered successfully",
  "data": {
    "user_id": 123,
    "email": "user@example.com",
    "name": "John Doe",
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
    "created_at": "2024-01-15T10:30:00Z"
  }
}
```

---

**POST /api/auth/login**

Login with email and password.

**Request Body:**

```
{
  "email": "user@example.com",
  "password": "SecurePassword123!"
}
```

**Response (200 OK):**

```
{
  "success": true,
  "message": "Login successful",
  "data": {
    "user_id": 123,
    "email": "user@example.com",
    "name": "John Doe",
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
    "refresh_token": "refresh_token_here",
  }
}
```

```
    "expires_in": 3600
  }
}
```

---

## **POST** /api/auth/logout

Logout the current user.

### **Headers:**

```
Authorization: Bearer {token}
```

### **Response (200 OK):**

```
{
  "success": true,
  "message": "Logout successful"
}
```

---

## **POST** /api/auth/refresh-token

Refresh the access token.

### **Request Body:**

```
{
  "refresh_token": "refresh_token_here"
}
```

### **Response (200 OK):**

```
{
  "success": true,
  "data": {
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
    "expires_in": 3600
  }
}
```

---

## GET /api/auth/profile

Get current user profile.

### Headers:

```
Authorization: Bearer {token}
```

### Response (200 OK):

```
{
  "success": true,
  "data": {
    "user_id": 123,
    "email": "user@example.com",
    "name": "John Doe",
    "phone_number": "+1234567890",
    "subscription_tier": "pro",
    "theme": "dark",
    "language": "en",
    "created_at": "2024-01-15T10:30:00Z",
    "updated_at": "2024-01-15T10:30:00Z"
  }
}
```

---

## PUT /api/auth/profile

Update user profile.

### Headers:

```
Authorization: Bearer {token}
```

### Request Body:

```
{
  "name": "John Smith",
  "phone_number": "+1234567890"
}
```

### Response (200 OK):

```
{
  "success": true,
  "message": "Profile updated successfully",
}
```

```
"data": {  
  "user_id": 123,  
  "name": "John Smith",  
  "phone_number": "+1234567890",  
  "updated_at": "2024-01-15T11:00:00Z"  
}
```

---

## **POST** /api/auth/forgot-password

Request password reset.

### **Request Body:**

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

### **Response (200 OK):**

```
{  
  "success": true,  
  "message": "Password reset email sent"  
}
```

---

## **POST** /api/auth/reset-password

Reset password with token.

### **Request Body:**

```
{  
  "token": "reset_token_here",  
  "new_password": "NewSecurePassword123!",  
  "confirm_password": "NewSecurePassword123!"  
}
```

### **Response (200 OK):**

```
{  
  "success": true,  
  "message": "Password reset successful"
```

```
}
```

---

## **POST** /api/auth/google

Authenticate with Google.

### **Request Body:**

```
{
  "id_token": "google_id_token_here",
  "access_token": "google_access_token_here"
}
```

### **Response (200 OK):**

```
{
  "success": true,
  "data": {
    "user_id": 123,
    "email": "user@gmail.com",
    "name": "John Doe",
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
    "is_new_user": false
  }
}
```

---

## **POST** /api/auth/apple

Authenticate with Apple.

### **Request Body:**

```
{
  "identity_token": "apple_identity_token_here",
  "authorization_code": "apple_auth_code_here"
}
```

### **Response (200 OK):**

```
{
  "success": true,
  "data": {
```

```
    "user_id": 123,
    "email": "user@privaterelay.appleid.com",
    "name": "John Doe",
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
    "is_new_user": false
  }
}
```

---

## **POST** /api/auth/send-otp

Send OTP code to email or phone.

### **Request Body:**

```
{
  "email": "user@example.com",
  "type": "login"
}
```

**Types:** login, register, reset

### **Response (200 OK):**

```
{
  "success": true,
  "message": "OTP sent successfully",
  "expires_in": 300,
  "otp_id": "otp_123456"
}
```

---

## **POST** /api/auth/verify-otp

Verify OTP code.

### **Request Body:**

```
{
  "email": "user@example.com",
  "otp_code": "123456",
  "type": "login",
  "otp_id": "otp_123456"
}
```



**Response (200 OK):**

```
{
  "success": true,
  "message": "OTP verified successfully",
  "data": {
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
    "user": {
      "user_id": 123,
      "email": "user@example.com",
      "name": "John Doe"
    }
  }
}
```

---

**POST /api/auth/resend-otp**

Resend OTP code.

**Request Body:**

```
{
  "email": "user@example.com",
  "type": "login",
  "otp_id": "otp_123456"
}
```

**Response (200 OK):**

```
{
  "success": true,
  "message": "OTP resent successfully",
  "expires_in": 300
}
```

---

## Voice & Notes Endpoints

**POST /api/voice/upload**

Upload voice recording file.

**Headers:**

```
Authorization: Bearer {token}
Content-Type: multipart/form-data
```

**Request Body (Form Data):**

```
file: (audio file)
duration: 120
```

**Response (200 OK):**

```
{
  "success": true,
  "data": {
    "recording_id": 456,
    "file_url": "https://storage.voclio.com/audio/123.mp3",
    "duration": 120,
    "uploaded_at": "2024-01-15T10:30:00Z"
  }
}
```

---

**POST /api/voice/transcribe**

Transcribe voice recording.

**Headers:**

```
Authorization: Bearer {token}
```

**Request Body:**

```
{
  "recording_id": 456,
  "language": "en"
}
```

**Response (200 OK):**

```
{
  "success": true,
  "data": {
    "transcription_id": 789,
    "text": "This is the transcribed text from the voice recording..."
  }
}
```

```
    "confidence": 0.95,  
    "language": "en",  
    "created_at": "2024-01-15T10:35:00Z"  
  }  
}
```

---

## GET /api/voice/:id

Get voice recording details.

### Headers:

```
Authorization: Bearer {token}
```

### Response (200 OK):

```
{  
  "success": true,  
  "data": {  
    "recording_id": 456,  
    "file_url": "https://storage.voclio.com/audio/123.mp3",  
    "duration": 120,  
    "file_size": 2048000,  
    "format": "mp3",  
    "transcription_status": "completed",  
    "created_at": "2024-01-15T10:30:00Z"  
  }  
}
```

---

## GET /api/notes

Get all notes with pagination and filters.

### Headers:

```
Authorization: Bearer {token}
```

### Query Parameters:

- **page** (integer, default: 1)

- **limit** (integer, default: 20)
- **search** (string)
- **tags** (string, comma-separated IDs)
- **sort** (string: **created\_at**, **updated\_at**, **title**)
- **order** (string: **asc**, **desc**)

#### Response (200 OK):

```
{
  "success": true,
  "data": [
    {
      "note_id": 123,
      "title": "Meeting Notes",
      "content": "Meeting content...",
      "transcription_text": "Transcribed text...",
      "summary_text": "Summary...",
      "view_mode": "list",
      "created_at": "2024-01-15T10:30:00Z",
      "updated_at": "2024-01-15T10:30:00Z",
      "tags": [1, 2, 3]
    }
  ],
  "pagination": {
    "page": 1,
    "limit": 20,
    "total": 150,
    "pages": 8
  }
}
```

### GET /api/notes/:id

Get specific note details.

#### Headers:

```
Authorization: Bearer {token}
```

#### Response (200 OK):

```
{
```

```
"success": true,
"data": {
  "note_id": 123,
  "title": "Meeting Notes",
  "content": "Full note content...",
  "transcription_text": "Transcribed text...",
  "summary_text": "Summary...",
  "view_mode": "list",
  "is_summarized": true,
  "voice_recording": {
    "recording_id": 456,
    "file_url": "https://storage.voclio.com/audio/123.mp3",
    "duration": 120
  },
  "tags": [
    {
      "tag_id": 1,
      "name": "work",
      "color": "#FF5733"
    }
  ],
  "created_at": "2024-01-15T10:30:00Z",
  "updated_at": "2024-01-15T10:30:00Z"
}
}
```

---

## POST /api/notes

Create a new note.

### Headers:

```
Authorization: Bearer {token}
```

### Request Body:

```
{
  "title": "Meeting Notes",
  "content": "Note content...",
  "transcription_text": "Transcribed text...",
  "voice_recording_id": 456,
  "tags": [1, 2, 3]
}
```

### Response (201 Created):

```
{
  "success": true,
```

```
"message": "Note created successfully",
"data": {
  "note_id": 123,
  "title": "Meeting Notes",
  "content": "Note content...",
  "created_at": "2024-01-15T10:30:00Z"
}
```

---

## **PUT** /api/notes/:id

Update a note.

### Headers:

```
Authorization: Bearer {token}
```

### Request Body:

```
{
  "title": "Updated Meeting Notes",
  "content": "Updated content...",
  "tags": [1, 2, 4]
}
```

### Response (200 OK):

```
{
  "success": true,
  "message": "Note updated successfully",
  "data": {
    "note_id": 123,
    "title": "Updated Meeting Notes",
    "content": "Updated content...",
    "updated_at": "2024-01-15T11:00:00Z"
  }
}
```

---

## **DELETE** /api/notes/:id

Delete a note.

**Headers:**

```
Authorization: Bearer {token}
```

**Response (200 OK):**

```
{
  "success": true,
  "message": "Note deleted successfully"
}
```

---

**GET /api/notes/search**

Search notes with full-text search.

**Headers:**

```
Authorization: Bearer {token}
```

**Query Parameters:**

- **q** (string, required) - Search query
- **page** (integer, default: 1)
- **limit** (integer, default: 20)

**Response (200 OK):**

```
{
  "success": true,
  "data": [
    {
      "note_id": 123,
      "title": "Meeting Notes",
      "content": "Found content...",
      "relevance_score": 0.85,
      "created_at": "2024-01-15T10:30:00Z"
    }
  ],
  "pagination": {
    "page": 1,
    "limit": 20,
    "total": 25,
    "pages": 2
  }
}
```

```
}
```

---

## **POST** /api/notes/:id/summarize

Generate AI summary for a note.

### Headers:

```
Authorization: Bearer {token}
```

### Response (200 OK):

```
{
  "success": true,
  "data": {
    "summary": "This note discusses the main points from the meeting including project timelines",
    "summary_generated_at": "2024-01-15T10:35:00Z",
    "llm_model_used": "gpt-5"
  }
}
```

---

## **POST** /api/notes/:id/extract-tasks

Extract tasks from a note using AI.

### Headers:

```
Authorization: Bearer {token}
```

### Response (200 OK):

```
{
  "success": true,
  "data": {
    "tasks": [
      {
        "id": 789,
        "title": "Complete project documentation",
        "description": "Finish the project documentation",
        "due_date": "2024-01-18",
        "priority": "high"
      }
    ],
  },
}
```



```
{
  "id": 790,
  "title": "Review code changes",
  "description": "Review all code changes",
  "due_date": "2024-01-22",
  "priority": "medium"
}
]
```

---

## GET /api/notes/:id/tags

Get tags for a note.

### Headers:

Authorization: Bearer {token}

### Response (200 OK):

```
{
  "success": true,
  "data": [
    {
      "tag_id": 1,
      "name": "work",
      "color": "#FF5733"
    },
    {
      "tag_id": 2,
      "name": "important",
      "color": "#33FF57"
    }
  ]
}
```

---

## POST /api/notes/:id/tags

Add tags to a note.

### Headers:

```
Authorization: Bearer {token}
```

#### Request Body:

```
{
  "tags": [1, 2, 3]
}
```

#### Response (200 OK):

```
{
  "success": true,
  "message": "Tags added successfully",
  "data": {
    "note_id": 123,
    "tags": [1, 2, 3]
  }
}
```

---

## Task Management Endpoints

### GET /api/tasks

Get all tasks with filters.

#### Headers:

```
Authorization: Bearer {token}
```

#### Query Parameters:

- **page** (integer, default: 1)
- **limit** (integer, default: 20)
- **status** (string: **pending**, **in\_progress**, **completed**, **cancelled**)
- **priority** (string: **low**, **medium**, **high**, **urgent**)
- **category\_id** (integer)
- **due\_date\_from** (date)

- **due\_date\_to** (date)

### Response (200 OK):

```
{
  "success": true,
  "data": [
    {
      "task_id": 789,
      "title": "Complete project",
      "description": "Finish the task...",
      "status": "pending",
      "priority": "high",
      "due_date": "2024-01-20T10:00:00Z",
      "completed_at": null,
      "category": {
        "category_id": 1,
        "name": "Work",
        "color": "#FF5733"
      },
      "created_at": "2024-01-15T10:30:00Z",
      "updated_at": "2024-01-15T10:30:00Z"
    }
  ],
  "pagination": {
    "page": 1,
    "limit": 20,
    "total": 50,
    "pages": 3
  }
}
```

## GET /api/tasks/:id

Get specific task details.

### Headers:

Authorization: Bearer {token}

### Response (200 OK):

```
{
  "success": true,
  "data": {
    "task_id": 789,
    "title": "Complete project",
    "description": "Finish the task...",

```

```
    "notes": "Additional notes...",
    "status": "pending",
    "priority": "high",
    "due_date": "2024-01-20T10:00:00Z",
    "completed_at": null,
    "creation_method": "manual",
    "source_note_id": 123,
    "calendar_event_id": "evt_123",
    "is_synced_to_calendar": true,
    "category": {
      "category_id": 1,
      "name": "Work",
      "color": "#FF5733"
    },
    "created_at": "2024-01-15T10:30:00Z",
    "updated_at": "2024-01-15T10:30:00Z"
  }
}
```

---

## POST /api/tasks

Create a new task.

### Headers:

```
Authorization: Bearer {token}
```

### Request Body:

```
{
  "title": "Complete project",
  "description": "Finish the task...",
  "due_date": "2024-01-20",
  "priority": "high",
  "category_id": 1,
  "note_id": 456
}
```

### Response (201 Created):

```
{
  "success": true,
  "message": "Task created successfully",
  "data": {
    "task_id": 789,
    "title": "Complete project",
    "status": "pending",
    "priority": "high",
```

```
    "due_date": "2024-01-20T10:00:00Z",
    "calendar_event_id": "evt_123",
    "created_at": "2024-01-15T10:30:00Z"
  }
}
```

---

## **PUT** /api/tasks/:id

Update a task.

### Headers:

```
Authorization: Bearer {token}
```

### Request Body:

```
{
  "title": "Updated Task",
  "description": "Updated description...",
  "status": "in_progress",
  "priority": "urgent",
  "due_date": "2024-01-25"
}
```

### Response (200 OK):

```
{
  "success": true,
  "message": "Task updated successfully",
  "data": {
    "task_id": 789,
    "title": "Updated Task",
    "status": "in_progress",
    "updated_at": "2024-01-15T11:00:00Z"
  }
}
```

---

## **DELETE** /api/tasks/:id

Delete a task.

### Headers:

Authorization: Bearer {token}

#### Response (200 OK):

```
{
  "success": true,
  "message": "Task deleted successfully"
}
```

---

### PUT /api/tasks/:id/complete

Mark task as completed.

#### Headers:

Authorization: Bearer {token}

#### Response (200 OK):

```
{
  "success": true,
  "message": "Task marked as completed",
  "data": {
    "task_id": 789,
    "status": "completed",
    "completed_at": "2024-01-15T11:00:00Z"
  }
}
```

---

### GET /api/tasks/stats

Get task statistics.

#### Headers:

Authorization: Bearer {token}

#### Query Parameters:

- **start\_date** (date)

- **end\_date** (date)

#### Response (200 OK):

```
{
  "success": true,
  "data": {
    "total": 50,
    "pending": 20,
    "in_progress": 15,
    "completed": 10,
    "cancelled": 5,
    "completion_rate": 0.20,
    "by_priority": {
      "high": 15,
      "medium": 25,
      "low": 10
    }
  }
}
```

### GET /api/tasks/by-date

Get tasks grouped by date.

#### Headers:

Authorization: Bearer {token}

#### Query Parameters:

- **start\_date** (date, required)

- **end\_date** (date, required)

#### Response (200 OK):

```
{
  "success": true,
  "data": {
    "2024-01-15": [
      {
        "task_id": 789,
        "title": "Task 1",
        "status": "pending",
        "priority": "high"
      }
    ]
  }
}
```

```
    }
  ],
  "2024-01-16": [
    {
      "task_id": 790,
      "title": "Task 2",
      "status": "in_progress",
      "priority": "medium"
    }
  ]
}
}
```

---

## GET /api/tasks/by-category

Get tasks grouped by category.

### Headers:

Authorization: Bearer {token}

### Response (200 OK):

```
{
  "success": true,
  "data": {
    "1": {
      "category": {
        "category_id": 1,
        "name": "Work",
        "color": "#FF5733"
      },
      "tasks": [
        {
          "task_id": 789,
          "title": "Task 1",
          "status": "pending"
        }
      ],
      "count": 15
    }
  }
}
```

---



## POST /api/tasks/bulk

Create multiple tasks at once.

### Headers:

```
Authorization: Bearer {token}
```

### Request Body:

```
{
  "tasks": [
    {
      "title": "Task 1",
      "description": "Description 1",
      "due_date": "2024-01-20",
      "priority": "high"
    },
    {
      "title": "Task 2",
      "description": "Description 2",
      "due_date": "2024-01-22",
      "priority": "medium"
    }
  ]
}
```

### Response (201 Created):

```
{
  "success": true,
  "message": "Tasks created successfully",
  "data": {
    "created": 2,
    "tasks": [
      {
        "task_id": 789,
        "title": "Task 1"
      },
      {
        "task_id": 790,
        "title": "Task 2"
      }
    ]
  }
}
```

---

## Calendar Sync Endpoints

## GET /api/calendar/sync

Get calendar sync status.

### Headers:

Authorization: Bearer {token}

### Response (200 OK):

```
{
  "success": true,
  "data": {
    "is_active": true,
    "calendar_service": "google",
    "sync_direction": "bidirectional",
    "sync_tasks": true,
    "sync_reminders": true,
    "last_synced_at": "2024-01-15T10:30:00Z",
    "last_sync_status": "success"
  }
}
```

---

## POST /api/calendar/sync

Enable calendar sync.

### Headers:

Authorization: Bearer {token}

### Request Body:

```
{
  "calendar_service": "google",
  "access_token": "access_token_here",
  "refresh_token": "refresh_token_here",
  "sync_tasks": true,
  "sync_reminders": true,
  "sync_direction": "bidirectional"
}
```

### Response (201 Created):

```
{
  "success": true,
  "message": "Calendar sync enabled",
  "data": {
    "sync_id": 123,
    "calendar_service": "google",
    "is_active": true
  }
}
```

---

## **PUT** /api/calendar/sync

Update calendar sync settings.

### **Headers:**

```
Authorization: Bearer {token}
```

### **Request Body:**

```
{
  "sync_tasks": false,
  "sync_reminders": true,
  "sync_direction": "one_way_out"
}
```

### **Response (200 OK):**

```
{
  "success": true,
  "message": "Calendar sync updated",
  "data": {
    "sync_id": 123,
    "sync_tasks": false,
    "sync_reminders": true
  }
}
```

---

## **DELETE** /api/calendar/sync

Disable calendar sync.

**Headers:**

```
Authorization: Bearer {token}
```

**Response (200 OK):**

```
{
  "success": true,
  "message": "Calendar sync disabled"
}
```

---

**POST /api/calendar/sync/now**

Trigger manual calendar sync.

**Headers:**

```
Authorization: Bearer {token}
```

**Response (200 OK):**

```
{
  "success": true,
  "message": "Sync initiated",
  "data": {
    "sync_status": "in_progress",
    "estimated_completion": "2024-01-15T10:35:00Z"
  }
}
```

---

**GET /api/calendar/events**

Get calendar events.

**Headers:**

```
Authorization: Bearer {token}
```

**Query Parameters:**

- **start\_date** (date)

- **end\_date** (date)

#### Response (200 OK):

```
{
  "success": true,
  "data": [
    {
      "event_id": "evt_123",
      "title": "Meeting",
      "description": "Team meeting",
      "start_time": "2024-01-20T10:00:00Z",
      "end_time": "2024-01-20T11:00:00Z",
      "task_id": 789,
      "calendar_service": "google"
    }
  ]
}
```

### **POST** /api/calendar/events

Create calendar event.

#### Headers:

```
Authorization: Bearer {token}
```

#### Request Body:

```
{
  "title": "Meeting",
  "description": "Team meeting",
  "start_time": "2024-01-20T10:00:00Z",
  "end_time": "2024-01-20T11:00:00Z",
  "task_id": 789
}
```

#### Response (201 Created):

```
{
  "success": true,
  "data": {
    "event_id": "evt_123",
    "title": "Meeting",
    "calendar_service": "google",
    "created_at": "2024-01-15T10:30:00Z"
  }
}
```

```
}  
}
```

---

## Tag Endpoints

### GET /api/tags

Get all tags.

#### Headers:

```
Authorization: Bearer {token}
```

#### Response (200 OK):

```
{  
  "success": true,  
  "data": [  
    {  
      "tag_id": 1,  
      "name": "work",  
      "color": "#FF5733",  
      "description": "Work related",  
      "created_at": "2024-01-15T10:30:00Z"  
    }  
  ]  
}
```

---

### POST /api/tags

Create a new tag.

#### Headers:

```
Authorization: Bearer {token}
```

#### Request Body:

```
{
```

```
    "name": "work",
    "color": "#FF5733",
    "description": "Work related"
  }
```

#### Response (201 Created):

```
{
  "success": true,
  "message": "Tag created successfully",
  "data": {
    "tag_id": 1,
    "name": "work",
    "color": "#FF5733",
    "created_at": "2024-01-15T10:30:00Z"
  }
}
```

---

### PUT /api/tags/:id

Update a tag.

#### Headers:

```
Authorization: Bearer {token}
```

#### Request Body:

```
{
  "name": "work-important",
  "color": "#FF5733",
  "description": "Important work related"
}
```

#### Response (200 OK):

```
{
  "success": true,
  "message": "Tag updated successfully",
  "data": {
    "tag_id": 1,
    "name": "work-important",
    "color": "#FF5733",
    "updated_at": "2024-01-15T11:00:00Z"
  }
}
```

---

## **DELETE** /api/tags/:id

Delete a tag.

### **Headers:**

```
Authorization: Bearer {token}
```

### **Response (200 OK):**

```
{
  "success": true,
  "message": "Tag deleted successfully"
}
```

---

## **Focus Mode Endpoints**

## **POST** /api/focus-sessions

Start a focus session.

### **Headers:**

```
Authorization: Bearer {token}
```

### **Request Body:**

```
{
  "timer_duration": 25,
  "ambient_sound": "rain",
  "sound_volume": 50
}
```

### **Response (201 Created):**

```
{
  "success": true,
  "message": "Focus session started",
  "data": {
```



```
    "session_id": 123,  
    "start_time": "2024-01-15T10:30:00Z",  
    "timer_duration": 25,  
    "status": "active"  
  }  
}
```

---

## **PUT** /api/focus-sessions/:id

Update focus session (pause/resume).

### Headers:

```
Authorization: Bearer {token}
```

### Request Body:

```
{  
  "status": "paused"  
}
```

### Response (200 OK):

```
{  
  "success": true,  
  "message": "Focus session updated",  
  "data": {  
    "session_id": 123,  
    "status": "paused",  
    "updated_at": "2024-01-15T10:35:00Z"  
  }  
}
```

---

## **DELETE** /api/focus-sessions/:id

End focus session.

### Headers:

```
Authorization: Bearer {token}
```

### Response (200 OK):

```
{
  "success": true,
  "message": "Focus session ended",
  "data": {
    "session_id": 123,
    "duration": 1500,
    "status": "completed",
    "end_time": "2024-01-15T10:55:00Z"
  }
}
```

---

## GET /api/focus-sessions

Get focus sessions.

### Headers:

```
Authorization: Bearer {token}
```

### Query Parameters:

- **page** (integer, default: 1)
- **limit** (integer, default: 20)
- **start\_date** (date)
- **end\_date** (date)

### Response (200 OK):

```
{
  "success": true,
  "data": [
    {
      "session_id": 123,
      "start_time": "2024-01-15T10:30:00Z",
      "end_time": "2024-01-15T10:55:00Z",
      "duration": 1500,
      "ambient_sound": "rain",
      "status": "completed"
    }
  ],
  "pagination": {
```

```
    "page": 1,  
    "limit": 20,  
    "total": 50,  
    "pages": 3  
  }  
}
```

---

## GET /api/focus-sessions/stats

Get focus session statistics.

### Headers:

```
Authorization: Bearer {token}
```

### Query Parameters:

- **start\_date** (date)

- **end\_date** (date)

### Response (200 OK):

```
{  
  "success": true,  
  "data": {  
    "total_sessions": 50,  
    "total_duration": 75000,  
    "average_duration": 1500,  
    "longest_session": 3600,  
    "by_date": {  
      "2024-01-15": 5,  
      "2024-01-16": 3  
    }  
  }  
}
```

---

## Productivity Endpoints

### GET /api/productivity/streak

Get productivity streak information.

**Headers:**

```
Authorization: Bearer {token}
```

**Response (200 OK):**

```
{
  "success": true,
  "data": {
    "current_streak": 5,
    "longest_streak": 10,
    "last_activity_date": "2024-01-15",
    "streak_type": "general"
  }
}
```

---

## GET /api/productivity/achievements

Get user achievements.

**Headers:**

```
Authorization: Bearer {token}
```

**Response (200 OK):**

```
{
  "success": true,
  "data": {
    "earned": [
      {
        "achievement_id": 1,
        "name": "First Task",
        "description": "Complete your first task",
        "icon": "trophy",
        "earned_date": "2024-01-10T10:30:00Z"
      }
    ],
    "available": [
      {
        "achievement_id": 2,
        "name": "Week Warrior",
        "description": "Complete 7 tasks in a week",
        "icon": "medal",

```

```
        "progress": 5,
        "required_value": 7
      }
    ]
  }
}
```

---

## **GET** /api/productivity/summary

Get daily/weekly summary.

### Headers:

```
Authorization: Bearer {token}
```

### Query Parameters:

- **period** (string: **daily**, **weekly**, default: **daily**)
- **date** (date)

### Response (200 OK):

```
{
  "success": true,
  "data": {
    "period": "daily",
    "date": "2024-01-15",
    "tasks_completed": 5,
    "tasks_created": 3,
    "notes_created": 2,
    "focus_sessions": 3,
    "focus_duration": 4500,
    "ai_summary": "Today you completed 5 tasks and created 3 new ones..."
  }
}
```

---

## **GET** /api/productivity/suggestions

Get AI-powered productivity suggestions.

**Headers:**

```
Authorization: Bearer {token}
```

**Response (200 OK):**

```
{
  "success": true,
  "data": {
    "suggestions": [
      {
        "type": "task_optimization",
        "message": "You have 3 high priority tasks due tomorrow. Consider breaking them into smaller tasks.",
        "priority": "high"
      },
      {
        "type": "time_management",
        "message": "Your focus sessions are most productive in the morning. Schedule important tasks during this time.",
        "priority": "medium"
      }
    ]
  }
}
```

---

## Backup & Restore Endpoints

### **POST** /api/backup

Create a backup of user data.

**Headers:**

```
Authorization: Bearer {token}
```

**Request Body:**

```
{
  "backup_type": "full"
}
```

**Response (201 Created):**

```
{
  "success": true,
```

```
"message": "Backup created successfully",
"data": {
  "backup_id": 123,
  "backup_date": "2024-01-15T10:30:00Z",
  "backup_type": "full",
  "status": "creating",
  "backup_file_url": null
}
```

---

## GET /api/backup

List all backups.

### Headers:

```
Authorization: Bearer {token}
```

### Response (200 OK):

```
{
  "success": true,
  "data": [
    {
      "backup_id": 123,
      "backup_date": "2024-01-15T10:30:00Z",
      "backup_type": "full",
      "backup_size": 5242880,
      "status": "completed",
      "backup_file_url": "https://storage.voclio.com/backups/123.zip"
    }
  ]
}
```

---

## GET /api/backup/:id

Get backup details.

### Headers:

```
Authorization: Bearer {token}
```

**Response (200 OK):**

```
{
  "success": true,
  "data": {
    "backup_id": 123,
    "backup_date": "2024-01-15T10:30:00Z",
    "backup_type": "full",
    "backup_size": 5242880,
    "status": "completed",
    "backup_file_url": "https://storage.voclio.com/backups/123.zip",
    "created_at": "2024-01-15T10:30:00Z"
  }
}
```

---

**POST /api/backup/:id/restore**

Restore from backup.

**Headers:**

```
Authorization: Bearer {token}
```

**Response (200 OK):**

```
{
  "success": true,
  "message": "Restore initiated",
  "data": {
    "restore_status": "in_progress",
    "estimated_completion": "2024-01-15T10:35:00Z"
  }
}
```

---

**DELETE /api/backup/:id**

Delete a backup.

**Headers:**

```
Authorization: Bearer {token}
```



### Response (200 OK):

```
{
  "success": true,
  "message": "Backup deleted successfully"
}
```

---

## Reminder Endpoints

### GET /api/reminders

Get all reminders.

#### Headers:

```
Authorization: Bearer {token}
```

#### Query Parameters:

- **page** (integer, default: 1)
- **limit** (integer, default: 20)
- **status** (string: **pending**, **sent**, **dismissed**)
- **task\_id** (integer)

### Response (200 OK):

```
{
  "success": true,
  "data": [
    {
      "reminder_id": 123,
      "task_id": 789,
      "task": {
        "task_id": 789,
        "title": "Complete project",
        "due_date": "2024-01-20T10:00:00Z"
      },
      "reminder_time": "2024-01-20T09:00:00Z",
      "reminder_type": "notification",
      "is_sent": false,
    }
  ]
}
```

```
        "sent_at": null,
        "created_at": "2024-01-15T10:30:00Z"
    }
],
"pagination": {
    "page": 1,
    "limit": 20,
    "total": 25,
    "pages": 2
}
}
```

---

## **GET** /api/reminders/:id

Get specific reminder.

### Headers:

```
Authorization: Bearer {token}
```

### Response (200 OK):

```
{
  "success": true,
  "data": {
    "reminder_id": 123,
    "task_id": 789,
    "reminder_time": "2024-01-20T09:00:00Z",
    "reminder_type": "both",
    "is_sent": false,
    "is_delivered": false,
    "sent_at": null,
    "created_at": "2024-01-15T10:30:00Z"
  }
}
```

---

## **POST** /api/reminders

Create a reminder.

### Headers:

```
Authorization: Bearer {token}
```

**Request Body:**

```
{
  "task_id": 789,
  "reminder_time": "2024-01-20T09:00:00Z",
  "reminder_type": "both",
  "notification_types": ["email", "whatsapp", "push"]
}
```

**Response (201 Created):**

```
{
  "success": true,
  "message": "Reminder created successfully",
  "data": {
    "reminder_id": 123,
    "task_id": 789,
    "reminder_time": "2024-01-20T09:00:00Z",
    "reminder_type": "both",
    "status": "pending",
    "created_at": "2024-01-15T10:30:00Z"
  }
}
```

---

**PUT /api/reminders/:id**

Update a reminder.

**Headers:**

```
Authorization: Bearer {token}
```

**Request Body:**

```
{
  "reminder_time": "2024-01-20T08:00:00Z",
  "reminder_type": "email"
}
```

**Response (200 OK):**

```
{
  "success": true,
  "message": "Reminder updated successfully",
  "data": {
    "reminder_id": 123,

```

```
    "reminder_time": "2024-01-20T08:00:00Z",  
    "updated_at": "2024-01-15T11:00:00Z"  
  }  
}
```

---

## **DELETE** /api/reminders/:id

Delete a reminder.

### Headers:

```
Authorization: Bearer {token}
```

### Response (200 OK):

```
{  
  "success": true,  
  "message": "Reminder deleted successfully"  
}
```

---

## **PUT** /api/reminders/:id/snooze

Snooze a reminder.

### Headers:

```
Authorization: Bearer {token}
```

### Request Body:

```
{  
  "snooze_minutes": 15  
}
```

### Response (200 OK):

```
{  
  "success": true,  
  "message": "Reminder snoozed",  
  "data": {  
    "reminder_id": 123,  
  }  
}
```

```
    "new_reminder_time": "2024-01-20T09:15:00Z"
  }
}
```

---

## **PUT** /api/reminders/:id/dismiss

Dismiss a reminder.

### Headers:

```
Authorization: Bearer {token}
```

### Response (200 OK):

```
{
  "success": true,
  "message": "Reminder dismissed",
  "data": {
    "reminder_id": 123,
    "status": "dismissed"
  }
}
```

---

## **GET** /api/reminders/upcoming

Get upcoming reminders.

### Headers:

```
Authorization: Bearer {token}
```

### Query Parameters:

- **hours** (integer, default: 24) - Hours ahead to look

### Response (200 OK):

```
{
  "success": true,
  "data": [
    {
```

```
    "reminder_id": 123,  
    "task_id": 789,  
    "task_title": "Complete project",  
    "reminder_time": "2024-01-20T09:00:00Z",  
    "reminder_type": "both"  
  }  
]  
}
```

---

## Notification Endpoints

### **POST** /api/notifications/email

Send email notification.

#### Headers:

```
Authorization: Bearer {token}
```

#### Request Body:

```
{  
  "to": "user@example.com",  
  "subject": "Task Reminder",  
  "body": "You have a task due soon...",  
  "template_id": "reminder_email"  
}
```

#### Response (200 OK):

```
{  
  "success": true,  
  "message": "Email sent successfully",  
  "data": {  
    "message_id": "msg_123",  
    "sent_at": "2024-01-15T10:30:00Z"  
  }  
}
```

---

### **POST** /api/notifications/whatsapp

Send WhatsApp notification.

**Headers:**

```
Authorization: Bearer {token}
```

**Request Body:**

```
{
  "to": "+1234567890",
  "message": "You have a task due soon...",
  "template_name": "task_reminder"
}
```

**Response (200 OK):**

```
{
  "success": true,
  "message": "WhatsApp message sent successfully",
  "data": {
    "message_id": "wa_123",
    "sent_at": "2024-01-15T10:30:00Z"
  }
}
```

---

**POST /api/notifications/push**

Send push notification.

**Headers:**

```
Authorization: Bearer {token}
```

**Request Body:**

```
{
  "title": "Task Reminder",
  "body": "You have a task due soon...",
  "data": {
    "task_id": 789,
    "type": "reminder"
  }
}
```

**Response (200 OK):**

```
{
  "success": true,
  "message": "Push notification sent successfully",
  "data": {
    "message_id": "push_123",
    "sent_at": "2024-01-15T10:30:00Z"
  }
}
```

---

## GET /api/notifications

Get all notifications.

### Headers:

```
Authorization: Bearer {token}
```

### Query Parameters:

- **page** (integer, default: 1)
- **limit** (integer, default: 20)
- **read** (boolean)
- **type** (string: **email**, **whatsapp**, **push**)

### Response (200 OK):

```
{
  "success": true,
  "data": [
    {
      "notification_id": 123,
      "type": "email",
      "title": "Task Reminder",
      "message": "You have a task due soon...",
      "is_read": false,
      "read_at": null,
      "created_at": "2024-01-15T10:30:00Z"
    }
  ],
  "pagination": {
    "page": 1,
    "limit": 20,
    "total": 50,
  }
}
```



```
    "pages": 3
  }
}
```

---

## **PUT** /api/notifications/:id/read

Mark notification as read.

### **Headers:**

```
Authorization: Bearer {token}
```

### **Response (200 OK):**

```
{
  "success": true,
  "message": "Notification marked as read",
  "data": {
    "notification_id": 123,
    "is_read": true,
    "read_at": "2024-01-15T11:00:00Z"
  }
}
```

---

## **DELETE** /api/notifications/:id

Delete a notification.

### **Headers:**

```
Authorization: Bearer {token}
```

### **Response (200 OK):**

```
{
  "success": true,
  "message": "Notification deleted successfully"
}
```

---

## **PUT** /api/notifications/preferences

Update notification preferences.

### **Headers:**

```
Authorization: Bearer {token}
```

### **Request Body:**

```
{
  "email_enabled": true,
  "whatsapp_enabled": false,
  "push_enabled": true,
  "email_for_reminders": true,
  "email_for_tasks": true,
  "whatsapp_for_reminders": false,
  "push_for_reminders": true
}
```

### **Response (200 OK):**

```
{
  "success": true,
  "message": "Notification preferences updated",
  "data": {
    "email_enabled": true,
    "whatsapp_enabled": false,
    "push_enabled": true
  }
}
```

---

## **GET** /api/notifications/preferences

Get notification preferences.

### **Headers:**

```
Authorization: Bearer {token}
```

### **Response (200 OK):**

```
{
  "success": true,
  "data": {
```

```
    "email_enabled": true,  
    "whatsapp_enabled": false,  
    "push_enabled": true,  
    "email_for_reminders": true,  
    "email_for_tasks": true,  
    "whatsapp_for_reminders": false,  
    "push_for_reminders": true  
  }  
}
```

---

## Settings Endpoints

### GET /api/settings

Get all user settings.

#### Headers:

```
Authorization: Bearer {token}
```

#### Response (200 OK):

```
{  
  "success": true,  
  "data": {  
    "theme": "dark",  
    "language": "en",  
    "timezone": "UTC",  
    "currency": "USD",  
    "date_format": "YYYY-MM-DD",  
    "time_format": "24h",  
    "biometric_enabled": true,  
    "notification_enabled": true  
  }  
}
```

---

### PUT /api/settings

Update user settings.

**Headers:**

```
Authorization: Bearer {token}
```

**Request Body:**

```
{
  "theme": "light",
  "language": "ar",
  "timezone": "Asia/Riyadh",
  "date_format": "DD-MM-YYYY"
}
```

**Response (200 OK):**

```
{
  "success": true,
  "message": "Settings updated successfully",
  "data": {
    "theme": "light",
    "language": "ar",
    "updated_at": "2024-01-15T11:00:00Z"
  }
}
```

---

**PUT /api/settings/theme**

Update theme setting.

**Headers:**

```
Authorization: Bearer {token}
```

**Request Body:**

```
{
  "theme": "dark"
}
```

**Response (200 OK):**

```
{
  "success": true,
  "message": "Theme updated",
  "data": {
    "theme": "dark"
  }
}
```

```
}  
}
```

---

## **PUT** /api/settings/language

Update language setting.

### **Headers:**

```
Authorization: Bearer {token}
```

### **Request Body:**

```
{  
  "language": "ar"  
}
```

### **Response (200 OK):**

```
{  
  "success": true,  
  "message": "Language updated",  
  "data": {  
    "language": "ar"  
  }  
}
```

---

## **PUT** /api/settings/subscription

Update subscription tier.

### **Headers:**

```
Authorization: Bearer {token}
```

### **Request Body:**

```
{  
  "subscription_tier": "pro"  
}
```

**Response (200 OK):**

```
{
  "success": true,
  "message": "Subscription updated",
  "data": {
    "subscription_tier": "pro",
    "subscription_start_date": "2024-01-15",
    "subscription_end_date": "2024-02-15"
  }
}
```

---

**PUT /api/settings/email**

Update email address.

**Headers:**

```
Authorization: Bearer {token}
```

**Request Body:**

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

**Response (200 OK):**

```
{
  "success": true,
  "message": "Email updated successfully",
  "data": {
    "email": "newemail@example.com",
    "updated_at": "2024-01-15T11:00:00Z"
  }
}
```

---

**PUT /api/settings/phone**

Update phone number.

**Headers:**

Authorization: Bearer {token}

#### Request Body:

```
{
  "phone_number": "+9876543210"
}
```

#### Response (200 OK):

```
{
  "success": true,
  "message": "Phone number updated successfully",
  "data": {
    "phone_number": "+9876543210",
    "updated_at": "2024-01-15T11:00:00Z"
  }
}
```

---

### GET /api/settings/notifications

Get notification settings.

#### Headers:

Authorization: Bearer {token}

#### Response (200 OK):

```
{
  "success": true,
  "data": {
    "email_enabled": true,
    "whatsapp_enabled": false,
    "push_enabled": true,
    "email_for_reminders": true,
    "email_for_tasks": true,
    "whatsapp_for_reminders": false,
    "push_for_reminders": true
  }
}
```

---

### PUT /api/settings/notifications

Update notification settings.

**Headers:**

```
Authorization: Bearer {token}
```

**Request Body:**

```
{
  "email_enabled": true,
  "whatsapp_enabled": true,
  "push_enabled": true,
  "email_for_reminders": true,
  "whatsapp_for_reminders": true,
  "push_for_reminders": true
}
```

**Response (200 OK):**

```
{
  "success": true,
  "message": "Notification settings updated",
  "data": {
    "email_enabled": true,
    "whatsapp_enabled": true,
    "push_enabled": true
  }
}
```

---

## Admin Endpoints

### **GET** /api/admin/users

Get all users (Admin only).

**Headers:**

```
Authorization: Bearer {admin_token}
```

**Query Parameters:**

- **page** (integer, default: 1)



- **limit** (integer, default: 20)
- **search** (string)
- **subscription\_tier** (string)
- **is\_active** (boolean)

#### Response (200 OK):

```
{
  "success": true,
  "data": [
    {
      "user_id": 123,
      "email": "user@example.com",
      "name": "John Doe",
      "subscription_tier": "pro",
      "is_active": true,
      "created_at": "2024-01-15T10:30:00Z"
    }
  ],
  "pagination": {
    "page": 1,
    "limit": 20,
    "total": 1000,
    "pages": 50
  }
}
```

---

### GET /api/admin/users/:id

Get specific user details (Admin only).

#### Headers:

Authorization: Bearer {admin\_token}

#### Response (200 OK):

```
{
  "success": true,
  "data": {
    "user_id": 123,
    "email": "user@example.com",
    "name": "John Doe",
  }
}
```

```
    "phone_number": "+1234567890",
    "subscription_tier": "pro",
    "is_active": true,
    "created_at": "2024-01-15T10:30:00Z",
    "last_login_at": "2024-01-15T12:00:00Z"
  }
}
```

---

## **PUT** /api/admin/users/:id

Update user (Admin only).

### Headers:

```
Authorization: Bearer {admin_token}
```

### Request Body:

```
{
  "subscription_tier": "free",
  "is_active": false
}
```

### Response (200 OK):

```
{
  "success": true,
  "message": "User updated successfully",
  "data": {
    "user_id": 123,
    "subscription_tier": "free",
    "is_active": false,
    "updated_at": "2024-01-15T11:00:00Z"
  }
}
```

---

## **DELETE** /api/admin/users/:id

Delete user (Admin only).

### Headers:

Authorization: Bearer {admin\_token}

**Response (200 OK):**

```
{
  "success": true,
  "message": "User deleted successfully"
}
```

---

## **POST /api/admin/users/:id/reset-password**

Reset user password (Admin only).

**Headers:**

Authorization: Bearer {admin\_token}

**Request Body:**

```
{
  "new_password": "NewPassword123!"
}
```

**Response (200 OK):**

```
{
  "success": true,
  "message": "Password reset successfully"
}
```

---

## **GET /api/admin/api-usage**

Get API usage statistics (Admin only).

**Headers:**

Authorization: Bearer {admin\_token}

**Query Parameters:**

- **start\_date** (date)
- **end\_date** (date)
- **api\_type** (string: **whisper**, **gpt**, **calendar**)

#### Response (200 OK):

```
{
  "success": true,
  "data": {
    "total_requests": 10000,
    "total_errors": 50,
    "success_rate": 0.995,
    "by_api_type": {
      "whisper": {
        "requests": 5000,
        "errors": 20,
        "total_cost": 25.50
      },
      "gpt": {
        "requests": 4000,
        "errors": 25,
        "total_cost": 45.75
      },
      "calendar": {
        "requests": 1000,
        "errors": 5,
        "total_cost": 0
      }
    },
    "by_date": {
      "2024-01-15": 500,
      "2024-01-16": 600
    }
  }
}
```

---

### **GET** /api/admin/api-keys

Get all API keys (Admin only).

#### Headers:

```
Authorization: Bearer {admin_token}
```

#### Response (200 OK):

```
{
  "success": true,
  "data": [
    {
      "key_id": 1,
      "api_type": "whisper",
      "provider": "openai",
      "is_active": true,
      "rate_limit": 1000,
      "created_at": "2024-01-15T10:30:00Z",
      "last_used_at": "2024-01-15T12:00:00Z"
    }
  ]
}
```

---

## **POST** /api/admin/api-keys

Create new API key (Admin only).

### Headers:

```
Authorization: Bearer {admin_token}
```

### Request Body:

```
{
  "api_type": "whisper",
  "access_token": "sk-...",
  "provider": "openai",
  "rate_limit": 1000
}
```

### Response (201 Created):

```
{
  "success": true,
  "message": "API key created successfully",
  "data": {
    "key_id": 1,
    "api_type": "whisper",
    "is_active": true,
    "created_at": "2024-01-15T10:30:00Z"
  }
}
```

---

## **PUT** /api/admin/api-keys/:id

Update API key (Admin only).

### **Headers:**

```
Authorization: Bearer {admin_token}
```

### **Request Body:**

```
{
  "is_active": false,
  "rate_limit": 2000
}
```

### **Response (200 OK):**

```
{
  "success": true,
  "message": "API key updated successfully",
  "data": {
    "key_id": 1,
    "is_active": false,
    "updated_at": "2024-01-15T11:00:00Z"
  }
}
```

---

## **DELETE** /api/admin/api-keys/:id

Delete API key (Admin only).

### **Headers:**

```
Authorization: Bearer {admin_token}
```

### **Response (200 OK):**

```
{
  "success": true,
  "message": "API key deleted successfully"
}
```

---

## GET /api/admin/logs

Get activity logs (Admin only).

### Headers:

Authorization: Bearer {admin\_token}

### Query Parameters:

- **page** (integer, default: 1)
- **limit** (integer, default: 20)
- **activity\_type** (string)
- **severity** (string: **info**, **warning**, **error**, **critical**)
- **start\_date** (date)
- **end\_date** (date)

### Response (200 OK):

```
{
  "success": true,
  "data": [
    {
      "log_id": 123,
      "activity_type": "user_login",
      "activity_category": "authentication",
      "user_id": 456,
      "admin_id": null,
      "severity": "info",
      "ip_address": "192.168.1.1",
      "timestamp": "2024-01-15T10:30:00Z",
      "details": {
        "method": "POST",
        "endpoint": "/api/auth/login"
      }
    }
  ],
  "pagination": {
    "page": 1,
    "limit": 20,
    "total": 5000,
    "pages": 250
  }
}
```

---

## GET /api/admin/config

Get app configuration (Admin only).

### Headers:

```
Authorization: Bearer {admin_token}
```

### Response (200 OK):

```
{
  "success": true,
  "data": [
    {
      "config_id": 1,
      "config_key": "focus_mode_enabled",
      "config_value": "true",
      "config_type": "boolean",
      "category": "feature",
      "description": "Enable/disable focus mode feature"
    },
    {
      "config_id": 2,
      "config_key": "ai_summary_enabled",
      "config_value": "true",
      "config_type": "boolean",
      "category": "feature",
      "description": "Enable/disable AI summary feature"
    }
  ]
}
```

---

## PUT /api/admin/config

Update app configuration (Admin only).

### Headers:

```
Authorization: Bearer {admin_token}
```

### Request Body:



```
{
  "config_key": "focus_mode_enabled",
  "config_value": "false"
}
```

#### Response (200 OK):

```
{
  "success": true,
  "message": "Configuration updated successfully",
  "data": {
    "config_id": 1,
    "config_key": "focus_mode_enabled",
    "config_value": "false",
    "updated_at": "2024-01-15T11:00:00Z"
  }
}
```

---

## Error Responses

All endpoints may return the following error responses:

### 400 Bad Request

```
{
  "success": false,
  "error": {
    "code": "VALIDATION_ERROR",
    "message": "Invalid request data",
    "details": {
      "email": "Email is required"
    }
  }
}
```

### 401 Unauthorized

```
{
  "success": false,
  "error": {
    "code": "UNAUTHORIZED",
    "message": "Invalid or expired token"
  }
}
```

## 403 Forbidden

```
{
  "success": false,
  "error": {
    "code": "FORBIDDEN",
    "message": "Insufficient permissions"
  }
}
```

## 404 Not Found

```
{
  "success": false,
  "error": {
    "code": "NOT_FOUND",
    "message": "Resource not found"
  }
}
```

## 500 Internal Server Error

```
{
  "success": false,
  "error": {
    "code": "INTERNAL_ERROR",
    "message": "An internal error occurred"
  }
}
```

---

**Documentation Version:** 1.0.0

**Last Updated:** 2024-01-15

**Contact:** [api-support@voclio.com](mailto:api-support@voclio.com)