

REST API Documentation v2.4

Overview

This document provides comprehensive documentation for the Company Platform REST API. The API enables developers to integrate with our platform, manage resources, and build custom applications.

Base URL: `https://api.company.com/v2`

Authentication: OAuth 2.0 or API Key

Authentication

API Key Authentication

Include your API key in the request header:

`Authorization: Bearer YOUR_API_KEY`

OAuth 2.0 Flow

1. Redirect user to authorization endpoint
2. User grants permission
3. Exchange authorization code for access token
4. Use access token in API requests

Authorization Endpoint: `https://auth.company.com/oauth/authorize`
Token Endpoint: `https://auth.company.com/oauth/token`

Rate Limiting

- Standard tier: 1,000 requests per hour
- Premium tier: 10,000 requests per hour
- Enterprise tier: Custom limits

Rate limit information is included in response headers:

`X-RateLimit-Limit: 1000`
`X-RateLimit-Remaining: 999`
`X-RateLimit-Reset: 1609459200`

Common Response Codes

- 200 OK - Request successful
- 201 Created - Resource created successfully
- 400 Bad Request - Invalid request parameters
- 401 Unauthorized - Authentication failed
- 403 Forbidden - Insufficient permissions
- 404 Not Found - Resource not found

- 429 Too Many Requests - Rate limit exceeded
- 500 Internal Server Error - Server error

Users API

Get User Profile

Retrieve information about the authenticated user.

Endpoint: GET /users/me

Response:

```
{  
  "id": "usr_123456",  
  "email": "user@example.com",  
  "name": "John Doe",  
  "role": "developer",  
  "created_at": "2024-01-15T10:30:00Z",  
  "verified": true  
}
```

Update User Profile

Update the authenticated user's profile information.

Endpoint: PATCH /users/me

Request Body:

```
{  
  "name": "Jane Doe",  
  "avatar_url": "https://example.com/avatar.jpg"  
}
```

List Users

Retrieve a paginated list of users (admin only).

Endpoint: GET /users

Query Parameters: - `page` (integer) - Page number (default: 1) - `per_page` (integer) - Results per page (default: 20, max: 100) - `role` (string) - Filter by role

Projects API

Create Project

Create a new project.

Endpoint: POST /projects

Request Body:

```
{  
    "name": "My New Project",  
    "description": "Project description",  
    "visibility": "private",  
    "tags": ["web", "api"]  
}
```

Response: 201 Created

```
{  
    "id": "prj_789012",  
    "name": "My New Project",  
    "owner_id": "usr_123456",  
    "created_at": "2024-12-12T14:30:00Z",  
    "status": "active"  
}
```

Get Project

Retrieve project details.

Endpoint: GET /projects/{project_id}

Response:

```
{  
    "id": "prj_789012",  
    "name": "My Project",  
    "description": "Description here",  
    "owner": {  
        "id": "usr_123456",  
        "name": "John Doe"  
    },  
    "visibility": "private",  
    "members": 5,  
    "created_at": "2024-12-01T10:00:00Z",  
    "updated_at": "2024-12-12T14:30:00Z"  
}
```

Update Project

Update project settings.

Endpoint: PATCH /projects/{project_id}

Request Body:

```
{  
    "name": "Updated Project Name",
```

```
        "description": "New description",
        "visibility": "public"
    }
```

Delete Project

Permanently delete a project.

Endpoint: DELETE /projects/{project_id}

Response: 204 No Content

List Projects

Retrieve projects for the authenticated user.

Endpoint: GET /projects

Query Parameters: - `page` (integer) - Page number - `per_page` (integer) - Results per page - `status` (string) - Filter by status: `active`, `archived`, `deleted` - `sort` (string) - Sort by: `created_at`, `updated_at`, `name`

Webhooks API

Create Webhook

Register a webhook endpoint.

Endpoint: POST /webhooks

Request Body:

```
{
  "url": "https://example.com/webhook",
  "events": ["project.created", "project.updated"],
  "secret": "webhook_secret_key"
}
```

List Webhooks

Get all registered webhooks.

Endpoint: GET /webhooks

Delete Webhook

Remove a webhook subscription.

Endpoint: DELETE /webhooks/{webhook_id}

File Upload API

Upload File

Upload a file to the platform.

Endpoint: POST /files/upload

Request: multipart/form-data

Parameters: - `file` (file) - The file to upload - `project_id` (string) - Associated project ID - `visibility` (string) - public or private

Response:

```
{  
  "id": "file_345678",  
  "name": "document.pdf",  
  "size": 1048576,  
  "mime_type": "application/pdf",  
  "url": "https://cdn.company.com/files/document.pdf",  
  "created_at": "2024-12-12T15:00:00Z"  
}
```

Get File

Retrieve file metadata.

Endpoint: GET /files/{file_id}

Delete File

Delete a file permanently.

Endpoint: DELETE /files/{file_id}

Analytics API

Get Usage Statistics

Retrieve API usage statistics.

Endpoint: GET /analytics/usage

Query Parameters: - `start_date` (date) - Start date (ISO 8601) - `end_date` (date) - End date (ISO 8601) - `granularity` (string) - hour, day, week, month

Response:

```
{  
  "total_requests": 15234,  
  "successful_requests": 15100,  
  "failed_requests": 134,
```

```

    "average_response_time": 145,
    "data_points": [
        {
            "timestamp": "2024-12-12T00:00:00Z",
            "requests": 1250
        }
    ]
}

```

SDK Libraries

Official SDK libraries are available for:

- Python: `pip install company-sdk`
- JavaScript/Node.js: `npm install @company/sdk`
- Ruby: `gem install company-sdk`
- Go: `go get github.com/company/sdk-go`

Code Examples

Python Example

```

from company_sdk import Client

client = Client(api_key='YOUR_API_KEY')

# Create a project
project = client.projects.create(
    name='My Project',
    description='A test project',
    visibility='private'
)

print(f"Project created: {project.id}")

# List projects
projects = client.projects.list(status='active')
for proj in projects:
    print(f"- {proj.name}")

```

JavaScript Example

```

const Company = require('@company/sdk');

const client = new Company.Client({
    apiKey: 'YOUR_API_KEY'
});

// Get user profile
async function getProfile() {

```

```
const user = await client.users.me();
console.log(`Welcome, ${user.name}`);
}

// Upload file
async function uploadFile(filePath) {
  const file = await client.files.upload({
    file: filePath,
    visibility: 'public'
  });
  console.log(`File URL: ${file.url}`);
}
```

Versioning

The API uses semantic versioning. Current version: **v2.4.0**

- Major version changes may include breaking changes
- Minor version changes add new features (backward compatible)
- Patch version changes include bug fixes

Support

- Documentation: <https://docs.company.com>
 - Developer Forum: <https://forum.company.com>
 - Email Support: api-support@company.com
 - Status Page: <https://status.company.com>
-

Last updated: December 12, 2024