

SyncV API Documentation

This document provides details on all available API endpoints in the SyncV application, including authentication requirements, request parameters, and response formats.

Table of Contents

1. [Authentication](#)
2. [User Management](#)
3. [CV Management](#)
4. [Job Description Management](#)
5. [Matching](#)
6. [Messages](#)

Authentication

All endpoints except the authentication endpoints require a JWT token to be provided in the Authorization header as a Bearer token.

Register

Register a new user in the system.

- **URL:** `/api/auth/register`
- **Method:** `POST`
- **Authentication:** Not required
- **Request Body:**

json

```
{  
  "name": "User Name",  
  "email": "user@example.com",  
  "password": "password"  
}
```

- **Response:**
 - Success (200): "User registered successfully!"
 - Error (400): "Error: Email is already in use!"

Login

Authenticate a user and retrieve a JWT token.

- **URL:** `/api/auth/login`
- **Method:** `POST`
- **Authentication:** Not required
- **Request Body:**

json

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

- **Response:**
 - Success (200):

json

```
{  
  "token": "JWT_TOKEN_STRING",  
  "type": "Bearer"  
}
```

User Management

Get Current User

Retrieve information about the currently authenticated user.

- **URL:** `/api/users/me`
- **Method:** `GET`
- **Authentication:** Required (JWT Token)
- **Response:**
 - Success (200):

json

```
{  
  "id": 1,  
  "name": "User Name",  
  "email": "user@example.com"  
}
```

- Error (403): Access denied message
- Error (500): Internal server error message

CV Management

Upload Single CV

Upload a single CV file for the authenticated user.

- **URL:** `/api/cvs/upload`
- **Method:** `POST`
- **Content-Type:** `multipart/form-data`
- **Authentication:** Required (JWT Token)
- **Request Parameters:**
 - `file`: The CV file (PDF or DOCX, max 5MB)
- **Response:**
 - Success (201):

json

```
{
  "id": 1,
  "userId": 1,
  "userName": "User Name",
  "name": "filename.pdf",
  "size": 12345,
  "type": "application/pdf",
  "uploadedAt": "2025-04-24T10:30:00"
}
```

- Error (400): File validation error message
- Error (500): Internal server error message

Upload Multiple CVs

Upload multiple CV files for the authenticated user.

- **URL:** `/api/cvs/upload-multiple`
- **Method:** `POST`
- **Content-Type:** `multipart/form-data`
- **Authentication:** Required (JWT Token)
- **Request Parameters:**
 - `files`: Array of CV files (PDF or DOCX, max 5MB each)
- **Response:**
 - Success (201): Array of CV DTOs
 - Error (400): File validation error message

- Error (500): Internal server error message

Get CV Information

Retrieve information about a specific CV.

- **URL:** `/api/cvs/{id}`
- **Method:** `GET`
- **Authentication:** Required (JWT Token)
- **Path Variables:**
 - `{id}`: ID of the CV
- **Response:**
 - Success (200): CV DTO
 - Error (404): Not found

Download CV

Download a specific CV file.

- **URL:** `/api/cvs/download/{id}`
- **Method:** `GET`
- **Authentication:** Required (JWT Token)
- **Path Variables:**
 - `{id}`: ID of the CV
- **Response:**
 - Success (200): Binary file data with appropriate content type
 - Error (404): Not found

Get All User's CVs

Retrieve all CVs belonging to the authenticated user.

- **URL:** `/api/cvs`
- **Method:** `GET`
- **Authentication:** Required (JWT Token)
- **Response:**
 - Success (200): Array of CV DTOs
 - Error (500): Internal server error message

Delete CV

Delete a specific CV.

- **URL:** `/api/cvs/{id}`
- **Method:** `DELETE`
- **Authentication:** Required (JWT Token)
- **Path Variables:**
 - `{id}`: ID of the CV
- **Response:**
 - Success (200): Success message
 - Error (403): Permission denied message
 - Error (500): Internal server error message

Delete All CVs

Delete all CVs belonging to the authenticated user.

- **URL:** `/api/cvs`
- **Method:** `DELETE`
- **Authentication:** Required (JWT Token)
- **Response:**
 - Success (200): "All CVs are deleted successfully!"
 - Error (403): Permission denied message
 - Error (500): Internal server error message

Job Description Management

Upload Single Job Description

Upload a single job description file for the authenticated user.

- **URL:** `/api/jds/upload`
- **Method:** `POST`
- **Content-Type:** `multipart/form-data`
- **Authentication:** Required (JWT Token)
- **Request Parameters:**
 - `file`: The job description file (PDF or DOCX, max 5MB)
- **Response:**
 - Success (201): Job Description DTO
 - Error (400): File validation error message

- Error (500): Internal server error message

Upload Multiple Job Descriptions

Upload multiple job description files for the authenticated user.

- **URL:** `/api/jds/upload-multiple`
- **Method:** `POST`
- **Content-Type:** `multipart/form-data`
- **Authentication:** Required (JWT Token)
- **Request Parameters:**
 - `files`: Array of job description files (PDF or DOCX, max 5MB each)
- **Response:**
 - Success (201): Array of Job Description DTOs
 - Error (400): File validation error message
 - Error (500): Internal server error message

Get Job Description Information

Retrieve information about a specific job description.

- **URL:** `/api/jds/{id}`
- **Method:** `GET`
- **Authentication:** Required (JWT Token)
- **Path Variables:**
 - `id`: ID of the job description
- **Response:**
 - Success (200): Job Description DTO
 - Error (404): Not found

Download Job Description

Download a specific job description file.

- **URL:** `/api/jds/download/{id}`
- **Method:** `GET`
- **Authentication:** Required (JWT Token)
- **Path Variables:**
 - `id`: ID of the job description

- **Response:**
 - Success (200): Binary file data with appropriate content type
 - Error (404): Not found

Get All User's Job Descriptions

Retrieve all job descriptions belonging to the authenticated user.

- **URL:** `/api/jds`
- **Method:** `GET`
- **Authentication:** Required (JWT Token)
- **Response:**
 - Success (200): Array of Job Description DTOs
 - Error (500): Internal server error message

Delete Job Description

Delete a specific job description.

- **URL:** `/api/jds/{id}`
- **Method:** `DELETE`
- **Authentication:** Required (JWT Token)
- **Path Variables:**
 - `id`: ID of the job description
- **Response:**
 - Success (200): Success message
 - Error (403): Permission denied message
 - Error (500): Internal server error message

Delete All Job Descriptions

Delete all job descriptions belonging to the authenticated user.

- **URL:** `/api/jds`
- **Method:** `DELETE`
- **Authentication:** Required (JWT Token)
- **Response:**
 - Success (200): "All JDs are deleted successfully!"
 - Error (403): Permission denied message

- Error (500): Internal server error message

Matching

Match CVs with Job Descriptions

Find job descriptions that match a specific CV.

- **URL:** `/api/match/cvs/{id}`
- **Method:** `POST`
- **Authentication:** Required (JWT Token)
- **Path Variables:**
 - `{id}`: ID of the CV
- **Request Body:**

json

```
[
  {
    "skill": "Java",
    "weight": 0.8
  },
  {
    "skill": "Python",
    "weight": 0.5
  }
]
```

- **Response:**
 - Success (200): Array of JD responses with similarity scores

json

```
[
  {
    "id": 1,
    "userId": 1,
    "jdName": "Senior Developer.pdf",
    "score": 0.85
  },
  {
    "id": 2,
    "userId": 1,
    "jdName": "Java Engineer.pdf",
    "score": 0.75
  }
]
```

- Error (500): Internal server error message

Match Job Descriptions with CVs

Find CVs that match a specific job description.

- **URL:** `/api/match/jds/{id}`
- **Method:** `GET`
- **Authentication:** Required (JWT Token)
- **Path Variables:**
 - `{id}`: ID of the job description
- **Response:**
 - Success (200): Array of CV responses with similarity scores

json

```
[
  {
    "id": 1,
    "userId": 1,
    "cvName": "John_Doe_Resume.pdf",
    "score": 0.92
  },
  {
    "id": 3,
    "userId": 1,
    "cvName": "Jane_Smith_CV.pdf",
    "score": 0.78
  }
]
```

- Error (404): Not found
- Error (500): Internal server error message

Messages

Publish Test Message

A test endpoint to publish a message to the Redis channel.

- **URL:** `/api/test/publish`
- **Method:** `GET`
- **Authentication:** Required (JWT Token)
- **Request Parameters:**
 - `message`: The message to publish
- **Response:**
 - Success (200): "Message published"

Important Notes

1. **File Size Limits:** Both CV and job description files are limited to 5MB.
2. **Supported File Types:** Only PDF and DOCX files are supported.
3. **Authentication:** All non-auth endpoints require a valid JWT token in the Authorization header.
4. **Redis Channel:** The application uses Redis for publishing messages about uploaded files for further processing.

5. **ML Service Integration:** The match endpoints interface with a separate ML service to provide similarity scores between CVs and job descriptions.

Sample Authentication Usage

```
# Register a new user
POST /api/auth/register
Content-Type: application/json

{
  "name": "John Doe",
  "email": "john.doe@example.com",
  "password": "securePassword123"
}

# Login to get JWT token
POST /api/auth/login
Content-Type: application/json

{
  "email": "john.doe@example.com",
  "password": "securePassword123"
}

# Use the token for authenticated requests
GET /api/users/me
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Sample File Upload Usage

```
# Upload a CV
POST /api/cvs/upload
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: multipart/form-data
```

Form data:

- file: (binary file content)

```
# Upload multiple Job Descriptions
POST /api/jds/upload-multiple
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: multipart/form-data
```

Form data:

- files[0]: (binary file content)
- files[1]: (binary file content)

Sample Matching Usage

```
# Find matching JDs for a CV with custom skill weights
POST /api/match/cvs/5
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json
```

```
[
  {
    "skill": "Java",
    "weight": 0.9
  },
  {
    "skill": "Spring Boot",
    "weight": 0.8
  },
  {
    "skill": "Docker",
    "weight": 0.6
  }
]
```

```
# Find matching CVs for a JD
GET /api/match/jds/3
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```