Complete API Reference - Agentic AI Hiring Platform

Base URL

Production: *https://agentic-v2-0.onrender.com*

Local: *http://localhost:8000*

Interactive Documentation

- **Swagger UI**: https://agentic-v2-0.onrender.com/docs

- **ReDoc**: https://agentic-v2-0.onrender.com/redoc

Complete API List

1 ROOT & HEALTH ENDPOINTS

Method	Endpoint	Description
`GET`	`/`	Welcome endpoint - API information
`GET`	`/health`	Health check - Service status

2 COMPANY ENDPOINTS

Method	Endpoint	Description
`POST`	`/company/`	Create new company

3 JOB ENDPOINTS

Method	Endpoint	Description
`POST`	`/job/create-with-company`	■ **RECOMMENDED** - Create company & job in one call
`POST`	`/job/`	Create job posting (company must exist)
`GET`	`/job/{job_id}`	Get job details by ID
`GET`	`/job/`	List all jobs with pagination
`GET`	`/job/company/{company_id}/applications`	Get all applications for a company
`GET`	`/job/{job_id}/applications`	Get applications for specific job
`GET`	`/job/{job_id}/applications/ranked`	Get ranked applications for a job

4 APPLICATION ENDPOINTS

Method	Endpoint	Description
`POST`	`/apply/{job_id}`	Submit job application with resume
`GET`	`/apply/{application_id}`	Get application details by ID
`GET`	`/apply/{application_id}/history`	Get application audit history

`GET`	`/apply/`	List all applications with pagination

5 CANDIDATE ENDPOINTS

Method	Endpoint	Description
`GET`	`/candidate/{candidate_id}`	Get candidate details
`GET`	`/candidate/{candidate_id}/applications`	Get all applications by candidate
`GET`	`/candidate/{candidate_id}/history`	Get candidate audit trail
`GET`	`/candidate/search/by-email?email={email}`	Search candidate by email
`GET`	`/candidate/`	List all candidates with pagination
`GET`	`/candidate/{candidate_id}/master`	■ **Master** - Complete candidate details
`GET`	`/candidate/master/all`	■ **Master** - All candidates complete details

6 ANALYTICS ENDPOINTS

Method	Endpoint	Description
`GET`	`/analytics/application/{application_id}/xai`	Get XAI (Explainable AI) explanation
`GET`	`/analytics/application/{application_id}/skill-gap`	Get skill gap analysis
`GET`	`/analytics/application/{application_id}/skill-graph`	Get skill evidence graph data
`GET`	`/analytics/job/{job_id}/rankings`	Get job application rankings
`GET`	`/analytics/fraud/{application_id}`	Get fraud detection analysis
`GET`	`/analytics/company/{company_id}/dashboard`	Get company analytics dashboard
`GET`	`/analytics/company/{company_id}/trends`	Get hiring trends over time

Most Important Endpoints

For Recruiters/HR:

1. *POST /job/create-with-company* - Create job posting (one-step)

2. *GET /job/{job_id}/applications/ranked* - View ranked applicants

3. *GET /analytics/application/{application_id}/xai* - Understand AI decisions

4. *GET /analytics/company/{company_id}/dashboard* - Company analytics

For Candidates:

1. *GET /job/* - Browse available jobs

2. *GET /job/{job_id}* - View job details

3. *POST /apply/{job_id}* - Apply to job

4. *GET /candidate/{candidate_id}/applications* - Track applications

For Admin/System:

1. *GET /candidate/master/all* - Export all candidate data

2. *GET /candidate/{candidate_id}/master* - Complete candidate profile

3. *GET /analytics/company/{company_id}/trends* - System analytics

Key Features by Endpoint Category

Job Management

- Create jobs with company in single API call
- Upload job description as PDF
- AI extracts skills from JD
- Track applications per job
- View ranked candidates

Application Processing

- Upload resume as PDF
- AI extracts candidate info & skills
- Automatic scoring (RFS, DCS, ELC)
- Fraud detection
- Automatic ranking
- Decision with explanation

Analytics & Insights

- Explainable AI decisions
- Skill gap analysis with learning roadmap
- Skill visualization graphs
- Company hiring trends
- Fraud detection reports
- Performance metrics

Candidate Tracking

- Complete candidate profiles
- Application history
- Multi-job application tracking
- Bulk candidate export
- Audit trails

Quick Start Examples

1. Create Job & Accept Applications

```
# Step 1: Create company and job
POST /job/create-with-company
Form Data:
  - company_name: "TechCorp"
  - company_description: "Leading tech company"
  - role: "Python Developer"
  - jd_pdf: [job_description.pdf]
  - required_experience: 3

# Step 2: View ranked applications
GET /job/{job_id}/applications/ranked
```

2. Candidate Applies to Job

```
# Step 1: Browse jobs
GET /job/?skip=0&limit=10

# Step 2: Apply
POST /apply/{job_id}
Form Data:
  - name: "John Doe"
  - email: "john@example.com"
  - experience: 5
  - resume_pdf: [resume.pdf]
```

3. Review Candidate

```
# Step 1: Get complete candidate details
GET /candidate/{candidate_id}/master

# Step 2: Understand AI decision
GET /analytics/application/{application_id}/xai

# Step 3: See skill gaps
GET /analytics/application/{application_id}/skill-gap
```

4. Export All Data

```
# Export all candidates with applications
GET /candidate/master/all?limit=500
```

Query Parameters Reference

Pagination (Most List Endpoints)

- *skip*: Number to skip (default: 0)
- *limit*: Maximum to return (varies by endpoint)

Filtering

- *status_filter*: Filter by application status
- *email*: Search by email address

Data Models

Job

- Company info, role, location, salary
- Employment type, experience required
- Job description text & PDF
- Extracted skills

Application

- Candidate & job references
- Scores: RFS, DCS, ELC, Composite
- Ranking position
- Decision & reasoning
- Fraud detection results
- Skill match analysis

Candidate

- Personal info (name, email, mobile, LinkedIn, GitHub)
- Years of experience
- Resume text & extracted skills
- Application history

File Upload Endpoints

Endpoint	Accepts	Purpose
`POST /job/create-with-company`	PDF	Job description
`POST /job/`	PDF	Job description
`POST /apply/{job_id}`	PDF	Resume

Supported Format: PDF only

Max File Size: Check server configuration

Response Formats

All endpoints return JSON responses with:

- **Success (200)**: Requested data

- **Created (201)**: For POST requests

- **Not Found (404)**: Resource doesn't exist

- **Error (500)**: Server error

Standard Error Format

```
{
  "detail": "Error message here"
}
```

Testing

Using Swagger UI (Recommended)

Visit: http://localhost:8000/docs
Expand any endpoint
Click "Try it out"
Fill parameters
Click "Execute"

Using cURL

```
# GET request
curl http://localhost:8000/job/1

# POST with JSON
curl -X POST http://localhost:8000/company/ \
  -H "Content-Type: application/json" \
  -d '{"name":"TechCorp","description":"Tech company"}'

# POST with file
curl -X POST http://localhost:8000/apply/1 \
  -F "name=John Doe" \
  -F "email=john@example.com" \
  -F "experience=5" \
  -F "resume_pdf=@resume.pdf"
```

Using Python

```
import requests

# GET
response = requests.get("http://localhost:8000/job/1")
data = response.json()

# POST with file
files = {"resume_pdf": open("resume.pdf", "rb")}
data = {"name": "John", "email": "john@example.com", "experience": 5}
response = requests.post("http://localhost:8000/apply/1", data=data, files=files)
```

Support & Documentation

- **Full Documentation**: See *API_ENDPOINTS.md*

- **Interactive API**: http://localhost:8000/docs

- **Master Endpoint Guide**: See *MASTER_ENDPOINT_GUIDE.md*

New Features

Master Endpoints (Recently Added)

- */candidate/{candidate_id}/master* - Complete single candidate details

- */candidate/master/all* - Complete all candidates details with pagination

These endpoints provide:

- Complete candidate profiles
- All applications with full details
- Job and company information
- All scores and rankings
- Decisions with explanations
- Fraud detection results
- Skill match analysis
- Summary statistics

Perfect for:

- HR dashboards
- Data export
- Reporting
- Analytics
- Audit trails