Apna API Integration

This guide covers the required setup, available fields, and data format for the API integration.

The API integration is a **push-based** system where Apna sends candidate details to client's API endpoint as soon as candidates apply to any of your job listings on apna portal.

Note- There is no ready API for posting a job on Apna, it is currently in development and we will support semi automated ways to post jobs on apna. API exists to send all the application details to the ATS or any system of client whenever an applicant applies to the job already posted on Apna.

The core service of this project is the one-time implementation of the ATS integration. While we're happy to provide occasional support for minor post-integration issues, ongoing maintenance and regular reporting are not included. Successful implementation will be verified through email confirmation from the end-user.

Endpoint Configuration

- **Endpoint URL**: Provided by the enterprise.
- **HTTP Method**: `POST`
- **Content-Type**: `application/json`
- **Authentication**: Basic or Token-based (configured as per enterprise requirements)

Request Structure

Apna will send candidate details in a structured JSON format. Below is a list of available fields, along with a sample payload.

Payload Fields

Candidate Information

Field	Type	Description
`full_name`	String	Full name of the candidate
`gender`	String	Gender (`m` or `f`)

`date_of_birth`	String	Candidate's date of birth in ISO format
`current_salary`	Integer	Current monthly salary in local currency
`current_salary_v2`	Integer	Annualized current salary
`notice_period`	Integer	Notice period in days
`english_language_marker`	String	Proficiency level (e.g., `basic`)

Location Information

Field	Type	Description
`user_location.display_name`	String	General area (e.g., `New Delhi, DL`)
`user_location.city`	String	City name
`user_location.derived_location.lat`	Float	Latitude of candidate's location
`user_location.derived_location.lng`	Float	Longitude of candidate's location

Education Background

Field	Type	Description
`educations[].education_level`	String	Highest education level (e.g., `Graduate`)
`educations[].degree`	String	Degree name (e.g., `B.Tech`)
`educations[].education_institute`	String	Institute name
`educations[].specialization`	String	Area of specialization

Employment History

Field	Type	Description
`experiences[].organization`	String	Organization name

`experiences[].job_title`	String	Job title
`experiences[].employment_type`	String	Type (e.g., `full-time`)
`experiences[].industry`	String	Industry sector
`experiences[].department`	String	Department
`experiences[].start`	String	Start date in ISO format
`experiences[].end`	String	End date in ISO format, or `null` if ongoing

Skills & Languages

Field	Type	Description
`known_languages[]`	Array	Languages known (e.g., `Hindi`, `English`)
`skills[]`	Array	Skills (e.g., `JavaScript`, `SQL`)

Contact Information

Field	Type	Description
`contact_info.email`	String	Candidate's email address
`contact_info.phone_number`	String	Candidate's phone number

Sample Payload

Here’s an example payload Apna might push to your endpoint:

json
Copy code

```
{
  "full_name": "Sarthak Birla",
  "gender": "m",
  "date_of_birth": "1996-06-22T18:30:00Z",
```

```
  "current_salary": 40000,
  "current_salary_v2": 350000,
  "notice_period": 60,
  "english_language_marker": "basic",
  "user_location": {
    "display_name": "New Delhi, DL",
    "city": "New Delhi",
    "derived_location": { "lat": 28.6081686133823, "lng":
77.12277925645002 }
  },
  "educations": [
    {
      "education_level": "Graduate" ,
      "degree": "B.Tech in Computer Engineering" ,
      "education_institute": "Jaypee University of Information
Technology" ,
      "specialization": "Computer" ,
      "start": "2014-07-31T18:30:00Z",
      "end": "2018-04-30T18:30:00Z",
      "ongoing": false
    }
  ],
  "experiences": [
    {
      "organization": "Apnatime Tech",
      "job_title": "Lead Software Engineer",
      "employment_type": "full-time",
      "industry": "Software Product",
      "department": "Software Engineering",
      "start": "2021-06-30T18:30:00Z",
      "ongoing": true
    }
  ],
  "known_languages": ["Hindi","English"],
  "skills": ["JavaScript","SQL"],
  "contact_info": { "email": "sarthak23birla@gmail.com",
"phone_number": "9999999999" }
}
```

Retry Mechanism

If the initial request to the client's endpoint does not receive a response or returns an error, Apna will automatically **retry once after 5 seconds.** This mechanism ensures resilience in case of temporary network or server issues on the client side.

- **Retry Count**: 1
- **Retry Delay**: 5 seconds

This retry mechanism minimizes data loss and maximizes successful lead delivery.

Throughput Requirements

For optimal performance and uninterrupted real-time lead delivery:

- **Expected Throughput**: Clients should ensure their systems can handle a minimum throughput of **10 requests per second (RPS)** without throttling or blocking.

Given the real-time nature of the integration, high throughput should not generally be a concern, but this requirement is highlighted to maintain clarity and confidence for clients expecting a consistent and immediate data push.

Error Codes & Responses

Success

- **Status Code**: 200
- **Response**: `{ "status": "success", "message": "Lead data received successfully." }`

Error Scenarios

Status Code	Description
400	Invalid Payload (e.g., missing required fields)
401	Unauthorized (e.g., invalid API key)

Steps for Integration

1. Agreement on the approach of 1 way API integration or 2 way semi automated integration
2. Client has to share the following
    a. API documentation of the ATS or their system where the leads data need to be pushed to
    b. Candidate profile fields confirmation based on the parameters shared above
    c. UAT/testing platform access to validate the integration is working correctly before we launch it on the production servers
3. Validation of a test job data and then making it go-live for all jobs post successful integration