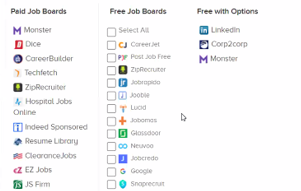
# Job Boards INDEX

Here’s a direct list of API documentation links and resources for automating job postings on popular job boards:



Here is the list of names from the image:

# Paid Job Boards:

## Monster(Foundit) (US)

[foundit.in](https://recruiter.foundit.in/)

Foundit.in does not have a public API available for posting jobs. I contacted the Monster/Foundit team but did not receive a response.

## Dice (US)

Dice does not have a public API available for posting jobs. I contacted the Dice team but did not receive a response.

[www.dice.com](https://www.dice.com/hiring/pricing)

699$/month

Unlimited job posts

Jobs promoted across other channels

Unlimited users, 100 shared candidate views/month

Branded company profile

Branded & featured jobs

ATS integrations

Onboarding, training & support

## CareerBuilder (US)

CareerBuilder does not have a public API available for posting jobs.

They requested these details when contacted via mail:

* **Company name -**
* **Main company telephone number**

· **Business physical address –**

* **Company web site**
* **What state is the company registered for business -**
* **Your full name and current title with the company –**
* **Attach a copy of the business license, or articles of incorporation, or assumed name filing**
* **How many jobs do you need to post?**
* **How many licenses need access?**
* **January Start Date:**

<https://hiring.careerbuilder.com/post-jobs>

$299 / MONTH

Promoted job postings

Unlimited applicants

Published to our sites, mobile apps, and our extensive partner network

Automatically sent to relevant candidates

Access resume database of millions of active candidates3

Proactively message qualified candidates

Ad-free experience

Unlimited user access

Monthly credits to promote jobs or proactively search and engage candidates 4.5

## TechFetch (US)

TechFetch does not have a public API available for posting jobs. They told they will provide Integration details only after paying $2500.

<https://www.techfetch.com/subscriptions.htm>

PackagePlatinum

Diamond

Total Resume Views 10,000 12,000

Total Jobs (Post or Repost) 500 600

Validity 12 Months 12 Months

Number of Users 1 1

Price $1799 $2499

## ZipRecruiter (US)

<https://support.ziprecruiter.com/s/contactsupport>

how to authenticate?

To authenticate your requests to the ZipRecruiter API, you will need to use **Basic Authentication**. Here’s how you can do it:

1. **API Key**: First, you need to get your API key from ZipRecruiter.
2. **Authorization Header**: You will include this API key in the Authorization header of your HTTP requests.

Here are examples of how to set this up in different tools and languages:

Using cURL

In cURL, you can set the Authorization header like this:

Job API ([Job API – ZipRecruiter Partner Platform Documentation](https://www.ziprecruiter.com/partner/documentation/#job-api))

The Job API is used by ZipRecruiter partners to post jobs to our network, and later, to close them.

One endpoint, <https://api.ziprecruiter.com/partner/v0/job> can be used for basic CRUD operations:

* POST to create jobs
* GET to retrieve a job
* PUT to update a job
* DELETE to close a job

The body of the request must contain a json representation of a job, [described below](https://www.ziprecruiter.com/partner/documentation/?python#the-job-model).

**Unicode in job\_ids**

In the Job API's GET, PUT, and DELETE operations, a job\_id is included in the URI.   
If this job\_id has any characters that are not in the *unreserved set*, they must be UTF-8 encoded, then the bytes URL-encoded (percent-encoded) [Wikipedia](https://en.wikipedia.org/wiki/Percent-encoding#Percent-encoding_in_a_URI). (curl does this automatically, perhaps your library of choice does as well.)

**Authentication**

Sample authorization. Make sure to replace meowmeowmeow in all examples with your API key.

# With shell, you can just pass the correct header with each request   
curl<https://api.ziprecruiter.com/partner/v0/job> \   
 -H "Authorization: Basic meowmeowmeow" ...

Partner API uses [Basic authentication](https://en.wikipedia.org/wiki/Basic_access_authentication) to allow access to all endpoints. You will be provided an API key; please keep it secret.

If your client's Basic Auth library requires separate `username` and `password` fields, simply decode the key we provide.

**Job API Operations**

**Post a job**

The job is an json-encoded object in the content. The job model is [described below](https://www.ziprecruiter.com/partner/documentation/?python#the-job-model). The job (identified by job\_id) cannot already exist.

**HTTP Request**

POST<https://api.ziprecruiter.com/partner/v0/job>

curl<https://api.ziprecruiter.com/partner/v0/job> \   
 -H "Authorization: Basic meowmeowmeow" \   
 -X POST \   
 -H "Content-Type: application/json" \   
 -d @job.json

**Parameters**

The job is an json-encoded object in the content. The job model is [described below](https://www.ziprecruiter.com/partner/documentation/?python#the-job-model).

**Update a job**

The job is an json-encoded object in the content. The job model is [described below](https://www.ziprecruiter.com/partner/documentation/?python#the-job-model). The job (identified its job\_id in the body) must already exist.

Note: updating a job always makes it active again, if it was previously closed.

**HTTP Request**

Either:

* PUT<https://api.ziprecruiter.com/partner/v0/job>
* PUT [https://api.ziprecruiter.com/partner/v0/job/<JOB\_ID](https://api.ziprecruiter.com/partner/v0/job/%3CJOB_ID)>

If the latter url is used, the JOB\_ID in the url must match the one in the body.

curl<https://api.ziprecruiter.com/partner/v0/job> \   
 -H "Authorization: Basic meowmeowmeow" \   
 -X PUT \   
 -H "Content-Type: application/json" \   
 -d @job.json

**Get a specific job**

This endpoint retrieves a specific job by id.   
At present, this only serves to demonstrate that the job exists in our system.

curl "<https://api.ziprecruiter.com/partner/v0/job/emlwcmVjcnVpdGVy-3215>" \   
 -H "Authorization: Basic meowmeowmeow"

**HTTP Request**

GET [https://api.ziprecruiter.com/partner/v0/job/<JOB\_ID](https://api.ziprecruiter.com/partner/v0/job/%3CJOB_ID)>

**URL Parameters**

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| JOB\_ID | The ID of the job to retrieve. |

If your `job\_id` has UTF-8, it must be properly encoded for the URI. See the note above.

**Close a Job**

curl "<https://api.ziprecruiter.com/partner/v0/job/emlwcmVjcnVpdGVy-3215>" \   
 -H "Authorization: Basic meowmeowmeow" \   
 -X DELETE

This request closes a specific job.

**HTTP Request**

DELETE [https://api.ziprecruiter.com/partner/v0/job/<JOB\_ID](https://api.ziprecruiter.com/partner/v0/job/%3CJOB_ID)>

**URL Parameters**

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| JOB\_ID | The ID of the job to close |

If your `job\_id` has UTF-8, it must be properly encoded for the URI. See the note above.

**Re-open a Job**

To re-open a job, simply [update it again](https://www.ziprecruiter.com/partner/documentation/?python#update-a-job).   
(It is fine to PUT with exactly the same fields.)

**The Job Model**

The job is represented as a hash (mapping, dictionary) of fields, encoded as JSON in the request or response body.   
Fields are optional unless otherwise specified.

Do not include fields that have no value, when you create a new job (POST). You may give a JSON null value for a field you wish to erase (such as apply\_url), in an update (PUT).

The job\_id and employer\_id fields are required, and must be unique in the partner's system. The employer\_name should be given the *first* time a job is created by that employer, it's optional after that.

A job location is required. You must always give country, and then either:

* postal\_code, or
* city and state, or
* latitude and longitude.

If you have more than one of the above, provide all that you have. (For remote jobs, you can give the location of the hiring company's headquarters.)

The job title is required.

A job\_type is required; it can be one of specific set of values (an enum), see below.

A minimal sample job, in json:

{   
 "employer\_id": "emlwcGFydG5lcg-042",   
 "employer\_name": "Stenographica, Ltd",   
 "job\_id": "emlwcmVjcnVpdGVy-3215",   
 "title": "Longhand",   
 "job\_type": "contract",   
 "state": "NY",   
 "city": "New York",   
 "country": "US",   
 "description": "<div>We need someone to write things by hand, on paper, legibly, in cursive.</div>"   
}

The basics are

* job\_id - REQUIRED. *unique* job id in the partner's system.
* employer\_id - REQUIRED. *unique* employer id in the partner's system.
* employer\_name - text
* title - REQUIRED. job title. text.
* job\_type - REQUIRED. enum, can be one of: full\_time, part\_time, contractor, contract, temporary, other
* country - REQUIRED. enum, can be one of: US, CA, AU
* city
* state - two- (or three-) letter [ISO-3166-2](https://en.wikipedia.org/wiki/ISO_3166-2) abbreviation (including [Canada](https://en.wikipedia.org/wiki/ISO_3166-2:CA) and [Australia](https://en.wikipedia.org/wiki/ISO_3166-2:AU))
* postal\_code
* description - text, minimum 25 chars. May include these HTML tags: p em i strong b br ol ul li hr div

Other fields:

* compensation\_min, compensation\_max - positive fixed point numeric (integer acceptable)
* compensation\_period - enum, can be one of: annual, monthly, weekly, daily, hourly
* compensation\_currency - enum, can be one of: USD, CAD, AUD
* show\_compensation - boolean, whether salary range should be shown with job listing
* apply\_url - a url, used for external apply, or for automated delivery with some ATS
* latitude, longitude - fixed point numbers, e.g. -106.81923
* accept\_remote - boolean, whether the job can be performed remotely. Acceptable values are true, false, 1, 0
* requisition\_id - string, unique requisition id of the job, which will be used during the candidate applications delivery
* cpc - numeric input, optionally with decimal fractions, that represents cost per click e.g. 12.49
* partner\_attributes - json object, with key-value pairs where key must be a string, containing any partner specific attributes e.g. {"foo": "bar", "foo1": 1}

For Australian provinces, the API supports both [two- and three-letter](https://en.wikipedia.org/wiki/ISO\_3166-2:AU#Changes) codes. However, the Ziprecruiter website currently only displays the two-letter codes.

**Job API Response**

The response to a successful request is a json object, as the example shown at right.

{   
 "job": {   
 "job\_id": "emlwcmVjcnVpdGVy-3215",   
 "title": "Longhand",   
 "job\_type": "full\_time",   
 "latitude": "40.71427000",   
 "longitude": "-74.00597000",   
 "city": "New York",   
 "state": "NY",   
 "country": "US",   
 "employer\_id": "emlwcGFydG5lcg-042",   
 "employer\_name": "Stenographica, Ltd",   
 "description": "<div>We need someone to write things by hand, on paper, legibly, in cursive.</div>"   
 },   
 "success": true   
}

The GET operation will also include the additional field job\_status (in addition to the job) with one of the values listed below:

* posted - the job is posted and live
* pending - the job is yet to be posted
* deleted - the job is closed and removed from our search engine

**Job API Change Log**

* 210928 enumerated permitted HTML tags in job description

Questions API

This section provides API specification for the optional Questions to be attached to an existing job listing, sometimes known as screening or pre-screening questions — a short series of questions presented to your applicants during the application process.

**Questions API Operations**

The endpoint, [https://api.ziprecruiter.com/partner/v0/job/<JOB\_ID>/questions](https://api.ziprecruiter.com/partner/v0/job/%3CJOB_ID%3E/questions) can be used for basic CRUD operations.

**URL Parameter**

JOB\_ID - The ID of the job to which the Questions are attached.

**Post Questions**

The body of the request must contain a json representation of the questions, [described below](https://www.ziprecruiter.com/partner/documentation/?python#the-question-model). The questions for the job (identified by JOB\_ID) cannot already exist.

curl<https://api.ziprecruiter.com/partner/v0/job/emlwcmVjcnVpdGVy-3215/questions> \   
 -H "Authorization: Basic meowmeowmeow" \   
 -X POST \   
 -H "Content-Type: application/json" \   
 -d @questions.json

**HTTP Request**

POST [https://api.ziprecruiter.com/partner/v0/job/<JOB\_ID>/questions](https://api.ziprecruiter.com/partner/v0/job/%3CJOB_ID%3E/questions)

**Parameters**

The questions are json-encoded object in the content. The question model is [described below](https://www.ziprecruiter.com/partner/documentation/?python#the-question-model).

**Update Questions**

The body of the request must contain a json representation of the questions, [described below](https://www.ziprecruiter.com/partner/documentation/?python#the-question-model). The questions for the job (identified by JOB\_ID) may already exist, which will be overwritten.

curl<https://api.ziprecruiter.com/partner/v0/job/emlwcmVjcnVpdGVy-3215/questions> \   
 -H "Authorization: Basic meowmeowmeow" \   
 -X PUT \   
 -H "Content-Type: application/json" \   
 -d @questions.json

**HTTP Request**

PUT [https://api.ziprecruiter.com/partner/v0/job/<JOB\_ID>/questions](https://api.ziprecruiter.com/partner/v0/job/%3CJOB_ID%3E/questions)

**Get Questions**

This endpoint retrieves the questions attached to the job specified by JOB\_ID.

curl "<https://api.ziprecruiter.com/partner/v0/job/emlwcmVjcnVpdGVy-3215/questions>" \   
 -H "Authorization: Basic meowmeowmeow"

**HTTP Request**

GET [https://api.ziprecruiter.com/partner/v0/job/<JOB\_ID>/questions](https://api.ziprecruiter.com/partner/v0/job/%3CJOB_ID%3E/questions)

**Delete Questions**

This request deletes all the questions attached the job specified by JOB\_ID.

curl "<https://api.ziprecruiter.com/partner/v0/job/emlwcmVjcnVpdGVy-3215/questions>" \   
 -H "Authorization: Basic meowmeowmeow" \   
 -X DELETE

**HTTP Request**

DELETE [https://api.ziprecruiter.com/partner/v0/job/<JOB\_ID>/questions](https://api.ziprecruiter.com/partner/v0/job/%3CJOB_ID%3E/questions)

**The Question Model**

A sample list of questions, in json:

[   
 {   
 "question\_id": "pet\_name",   
 "type" : "text",   
 "question" : "What is your pet's name?",   
 "required" : true   
 },   
 {   
 "question\_id": "pet\_age",   
 "type" : "text",   
 "format" : "integer",   
 "question" : "How old is your pet, in months?",   
 "min" : 1,   
 "max" : 250   
 },   
 {   
 "question\_id": "pet\_weight",   
 "type" : "text",   
 "format" : "decimal",   
 "question" : "What is your pet's weight, in kg?",   
 "min" : 0.01,   
 "max" : 250   
 },   
 {   
 "question\_id": "pet\_adopt\_date",   
 "type" : "date",   
 "question" : "When was your pet adopted, or born into your care?",   
 "min" : "1970-10-20",   
 "format" : "YYYY-MM-DD"   
 },   
 {   
 "question\_id": "pet\_sleep\_pref",   
 "type" : "select",   
 "question" : "Where does your pet like to sleep?",   
 "options" : [   
 {"value": "bed", "label": "My Bed"},   
 {"value": "floor", "label": "The Floor"},   
 {"value": "alone", "label": "Its Bed"},   
 {"value": "na", "label": "None of the Above"}   
 ]   
 },   
 {   
 "question\_id": "pet\_nose\_color",   
 "type" : "multiselect",   
 "question" : "What colors are present on your pet's nose?",   
 "options" : [   
 {"value": "0", "label": "Pink"},   
 {"value": "1", "label": "Brown"},   
 {"value": "2", "label": "Liver"},   
 {"value": "3", "label": "Yellow"},   
 {"value": "4", "label": "Green"},   
 {"value": "5", "label": "None of the Above"}   
 ]   
 },   
 {   
 "question\_id": "pet\_photo",   
 "type" : "upload",   
 "question" : "Please submit your cutest photo of your pet.",   
 "required" : true   
 }   
]

The questions are expressed as an array of objects, each object representing a question, encoded as json in the request or response body.

The following fields are available in all questions, regardless of type. All are required except for the required field. All fields should be text values unless specified otherwise.

* question\_id

A unique identifier for this question. This can just be the ordinal number of the question (i.e. 1, 2, 3...) if you don’t already have an identifier for your questions.

* type

The question type, which determines the appearance of the question to your applicants, as well what other fields are required and optional for the question here. The following types are valid:

* text
* date
* select
* multiselect
* upload

Each type may have further required and allowed fields; see below for more information.

* question

The question text shown to applicants. Simple HTML formatting, like paragraphs and emphasis, is permitted.

* required

Boolean, Optional. If present and true, this question cannot be skipped by applicants, and is required in order for the application to be considered complete and delivered to you.

**Supported Question Types**

* **text**

A free-form text entry with optional minimum and maximum text length requirements.

* min
* max

The minimum and maximum number of characters required to accept an answer.

* **integer and decimal**

An entry restricted to numeric input, optionally with decimal fractions, as well as optional minimum and maximum value requirements. This is expressed as an extension of the text type.

* type - Must be text
* format - Must be either integer or decimal
* min
* max

The minimum and maximum value required to accept an answer.

* **date**

A date entry with optional earliest and latest date requirements.

* min
* max

The optional earliest and latest date accepted, in YYYY-MM-DD format. Applicants will be shown a datepicker that restricts them to this range.

* format

Required, must be YYYY-MM-DD. This is required in order to be clear that only one date format supported.

* **select and multiselect**

A list of options where either only one may be selected, or any number of options may be selected.

* options

An array of objects, each with both of the following fields:

* label - The option text shown to applicants.
* value - The corresponding value, or option identifier that is returned to your ATS in answers, if applicable.
* **upload**

A file upload entry. Files from applicants for upload questions are limited to PDF and well-known image and text document formats, and may be no larger than 10MB. No additional fields are required.

**Questions API Response**

The response to a successful request is a json object, as the example shown at right.

{   
 "success" : true,   
 "job\_id" : "emlwcmVjcnVpdGVy-3215",   
 "questions": [   
 {   
 "question\_id": "pet\_name",   
 "type" : "text",   
 "question" : "What is your pet's name?",   
 "required" : true   
 },   
 {   
 "question\_id": "pet\_age",   
 "type" : "text",   
 "format" : "integer",   
 "question" : "How old is your pet, in months?",   
 "min" : 1,   
 "max" : 250,   
 "required" : false   
 },   
 {   
 "question\_id": "pet\_weight",   
 "type" : "text",   
 "format" : "decimal",   
 "question" : "What is your pet's weight, in kg?",   
 "min" : 0.01,   
 "max" : 250,   
 "required" : false   
 },   
 {   
 "question\_id": "pet\_adopt\_date",   
 "type" : "date",   
 "question" : "When was your pet adopted, or born into your care?",   
 "format" : "YYYY-MM-DD",   
 "min" : "1970-10-20",   
 "required" : false   
 },   
 {   
 "question\_id": "pet\_sleep\_pref",   
 "type" : "select",   
 "question" : "Where does your pet like to sleep?",   
 "options": [   
 { "value": "bed", "label": "My Bed" },   
 { "value": "floor", "label": "The Floor" },   
 { "value": "alone", "label": "Its Bed" },   
 { "value": "na", "label": "None of the Above" }   
 ],   
 "required" : false   
 },   
 {   
 "question\_id": "pet\_nose\_color",   
 "type" : "multiselect",   
 "question" : "What colors are present on your pet's nose?",   
 "options": [   
 { "value": "0", "label": "Pink" },   
 { "value": "1", "label": "Brown" },   
 { "value": "2", "label": "Liver" },   
 { "value": "3", "label": "Yellow" },   
 { "value": "4", "label": "Green" },   
 { "value": "5", "label": "None of the Above" }   
 ],   
 "required" : false   
 },   
 {   
 "question\_id": "pet\_photo",   
 "type" : "upload",   
 "question" : "Please submit your cutest photo of your pet.",   
 "required" : true   
 }   
 ]   
}

Features API

## Hospital Jobs Online (US)

<https://www.hospitaljobsonline.com/postjobs>

## Indeed (US)

When contacted via mail they said they are not allowing any job boards to show up in their feed.

<https://www.indeed.com/employers/sponsored-jobs>

## Resume Library (US)

They don't allow any job boards to post data but they allow you to access their job boards and find out the candidates based on our requirement.

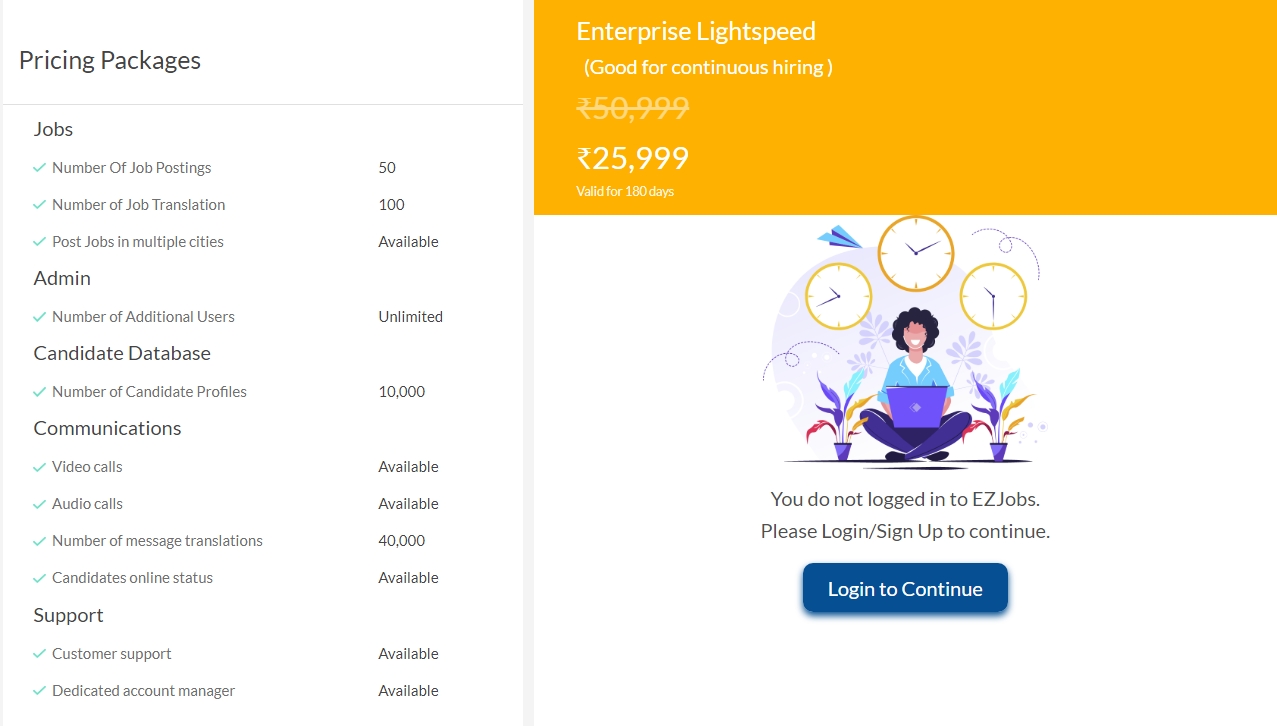
<https://www.resume-library.com/hiring/post-jobs>  
  
Resume Library Requested to call 1-800-672-6706 to get this activated.

## ClearanceJobs (US)

<https://support.clearancejobs.com/employers/posting-a-new-job>

## EZ Jobs (India)

<https://employer.ezjobs.io/#/>



## JS Firm (US)

<https://www.jsfirm.com/newadvertise>

## Apna (india)

Pricing is like 2 – 2.5 lakh for integration

Takes 7 days for integration.

They provided documentation.

enterprise account Package OFFER PRICE

Amount (Exclusive of GST) 100000

Job credit : 300

Validity (Days) : 360days

⚡Validity for : 360 days

⚡Get Unlimited Responses

⚡free viewing profile in database

⚡Send bulk SMS or WhatsApp to jobseekers

⚡Download this data in excel.

⚡ key account manager

Start your hiring with just

Offer price - 100000

## Naukri (india)

[Jobs - Recruitment - Job Search - Employment - Job Vacancies - Naukri.com](https://www.naukri.com/)

They requested for this info:

Company Name:

Company Location:

Email Address:

Phone number:

## Shine (india)

[Jobs - Job Search 2025 - Latest Job Vacancies - Recruitment - Shine.com](https://www.shine.com/)

They requested for this data:

**Company name**

**Number of candidates required**

**Number of recruiters**

**Location**

**IT/Non-IT Profiles from Pan India or any Specific Region**

**Is recruitment an ongoing process? (Yes/No)**

**Your current Designation**

**Company budget for recruitment**

**Name, email and contact number of decision maker**

## Timesjobs.com (india) (Paid)

[Jobs Search | Recruitment | Employment | Job Vacancies | TimesJobs](https://www.timesjobs.com/candidate/contact.html)

**Email Reply from TImesJobs.com:**

Thank you for giving us an opportunity to present **Integrated Recruitment Solutions from TimesJobs.com– Times of India, India’s leading Job Board** to your esteemed organization.  
  
Following your requirement with times jobs proposal , regarding **Api Process flow of job posting** please find enclosed a comprehensive proposal from **TimesJobs.com** on the same.

**Our Best Value Proposal 1 :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Product** | **Duration (Days/Month/Years)** | **Login-Ids** | **Best Price including**  **taxes** |
| **1000 JOB POSTING + PROCESS FLOW OF JOB APPLICATION** | **12 Month** | **1 Login id** | **1,41,000** |

Note :- This integration will help you to get the candidate flow to your own website . The access is strictly only for India we don't provide the access

apart from India . Payment should be done through an Indian account only . We don't accept any other currency mode apart from Indian rupees .

Here I am mentioning how the process flow works to get the application on your website .

1) The candidate is already logged in and he clicks on 'apply'

What happens:

a) The candidate is asked to choose the resume that he wishes to apply with (standard process for any application at TimesJobs)

b) Once he does that, Timesjobs window shows him a message to the effect of:

Dear <Name of the candidate>,

<Message stating that a new “Client” window has opened and candidate needs to complete the application process for “Client” from there>

In case he has bulk applied, application confirmation for the rest of the jobs is shown on this page in the Timesjobs window.

c) and Client's recruitment platform (Client's website) is opened in a new window that pops up in front of the candidate.

2) The candidate is registered but not logged in and he clicks on 'apply'

What happens:

a) The candidate is asked to log in first

The rest of the steps are as mentioned (a,b and c) for Scenario 1

3) The candidate is not registered with TimesJobs and he clicks on 'apply'

What happens:

a) The candidate is asked to register on TimesJobs to successfully apply to the job

b) On completing the registration process, the rest of the steps are as mentioned (a,b and c) for Scenario 1

Flow:

1. Clients will share/post jobs on TimesJobs and in the posting will mention the “own website page link”, on which they want the candidates of a particular job should land.

2. When on TimesJobs, a job will be seen by a candidate and when interested candidate will click on apply, he does that, Timesjobs window shows him a message to the effect of:

Dear <Name of the candidate>,

<Message stating that a new “Client” window has opened and candidate needs to complete the application process for “Client” from there>

3.Client website particular page will open in a new TAB along with the TimesJobs application confirmation page .

## Internshala (india)

[Internships | Jobs | Trainings & Placement Guarantee Courses | Post a Job](https://internshala.com/?form=MG0AV3)

## Freshersworld (india)

[Jobs: Search Jobs In India, Freshers Jobs Online, Govt Jobs, Recruitment | Freshersworld.com](https://www.freshersworld.com/)

## cutshort.io (india)

[Cutshort: Making Top Professionals More Successful.](https://cutshort.io/)

## herkey (india)

[Recruitment Solutions | Online Jobs Listing Service | JobsForHer](https://business.herkey.com/employer?form=MG0AV3)

## Quicker jobs (india)

[Jobs in India, Blue Collar Jobs, Vacancy 2025-26 - QuikrJobs](https://www.quikr.com/jobs?form=MG0AV3)

# Free Job Boards:

## Career Jet (UK)

<https://www.careerjet.co.in/recruiter>

**Email Reply from CareerJet:**

In order to help us to ensure that all of your jobs are listed as soon as possible, would it be possible to provide us with an XML (or JSON) feed to your job listings?

If you do not already have a feed, you can refer to our guidelines for creating one here: [https://www.careerjet.co.in/docs/feeds/xml](https://www.careerjet.co.in/docs/feeds/xml?ltag=2dcgCVO86SdWCiL2dXoEo6)

## Post Job Free (US)

<https://www.postjobfree.com/>

**Documentation on how post xml Job feed:**

**What is an XML job feed?**

XML job feed is a web page on your site that lists your jobs in a computer-readable XML format.

**Why do I need a job feed?**

XML job feed allows you to post jobs to postjobfree.com and other job boards automatically.

**How do I include my job feed to PostJobFree?**

|  |  |
| --- | --- |
| **Email:**  **\*** |  |
| **Company:** |  |
| **Contact person:** |  |
| **Link to job feed:** |  |
| **Notes:** |  |

**Example of XML job feed:**

<?xml version="1.0" encoding="utf-8"?>   
<source>   
<job>

<title>JobTitle</title>   
<date>2024-12-24 23:12:39</date>   
<referencenumber>UniqueData131</referencenumber>   
<url>https://www.site.com/job?jobid=UniqueData131</url>   
<company>CompanyName</company>   
<location>Louisville,KY,40202</location>   
<description>We have an immediate opening for a Diesel Technician, where you will be performing the majority of required maintenance and service on a wide range of equipment.   
...   
Responsibilities: Performs vehicle maintenance and repair duties   
...   
</description>   
<salary>$50K per year</salary>   
<email>hr@companyname.com</email>   
<cpc>$0.30</cpc>

</job>   
...   
</source>

Note:

Colored fields are required.

**Q: How will I receive job applications?**

You have 2 options to receive job seekers' applications from PostJobFree:

1) By email.

This is the simplest option to implement.

PostJobFree will send all job seekers' applications to your email.

If you want to send job applications to different emails, then please specify email for every job in your job feed.

You may specify an email address anywhere in the job description, or you may put the email address into a separate <email> element.

2) Web API.

That option is more involved and requires you to have a web API on your website.

The simplest API implementation is to add a webpage that will receive HTTP POST requests with the following parameters:

- id (referencenumber of the job)   
- resume   
- coverLetter   
- email (email address of job seeker who apply)   
- ip (IP address of job seeker who apply)

Using web API allows you to automatically receive and process job seekers' applications.

**Q: Can you redirect job seekers to my jobs to apply?**

We can, but usually with "redirect" integration jobs receive ~10x less applications.

We also prefer if you pay (PPC) for redirected traffic (e.g. $0.30/click CPC).

Free "redirect" integration is also available if your jobs are appealing to job seekers and are hard to find elsewhere.

If possible, we prefer sending applications over email to you -- for free (instead of redirecting job seekers to the external website on PPC basis).

**How much does jobs integration cost?**

We have both free and paid version of XML feed inclusion.

If we email candidates applications to you - usually it is free.

If you need us to redirect candidates to your website - we usually charge from $0.10 to $0.50 per candidate click(redirect).

## ZipRecruiter (US)

<https://www.ziprecruiter.in/>

**Documentation on JobBoard Integration:**

[Job API – ZipRecruiter Partner Platform Documentation](https://www.ziprecruiter.com/partner/documentation/#job-api)

how to authenticate?

To authenticate your requests to the ZipRecruiter API, you will need to use **Basic Authentication**. Here’s how you can do it:

1. **API Key**: First, you need to get your API key from ZipRecruiter.
2. **Authorization Header**: You will include this API key in the Authorization header of your HTTP requests.

Here are examples of how to set this up in different tools and languages:

Using cURL

In cURL, you can set the Authorization header like this:

## Jobrapido (Italy)

<https://corporate.jobrapido.com/post-a-job/>

1 advert @ 99.99 euro +l vat

2 adverts @ 179.99 euro + vat

Custom campaigns (Contact Job rapido)

## Jooble (Ukraine)

<https://se.jooble.org/employer/jobposting>

Link on how to post jobs on jooble using xml feed

[XML](https://jooble.org/files/xml_feed_specifications.pdf)

Email Reply from Jooble posting jobs on their job board:

First of all, we are sorry for the delay in replying! Thank you for choosing Jooble to post your job offer! Jooble is a job aggregator that collects job postings from various sources and displays them in one place, making it easier for job seekers to find opportunities. As part of this process, we check websites to identify and index available vacancies.

To display your job postings on Jooble, you have following options:

XML Feed. This is a file format that allows us to automatically receive job postings from your system. If you already have an XML feed, please send it in response to this email. For guidance on creating an XML feed, please refer to this detailed specification. <https://jooble.org/files/xml_feed_specifications.pdf>

Direct Link. If creating an XML feed isn’t possible, you can simply provide us with a direct link to the page on your website where vacancies are listed.

At this time, Jooble does not provide the ability for recruiters to post jobs directly to our job board by clicking the “Share on other job boards” button or similar feature. Instead, Jooble operates as a job aggregator, which means that we collect jobs from a variety of external sources, including employer websites, job boards, and ATS platforms.

If you are interested in having your jobs posted on Jooble, we offer an XML Feed Integration: To add your jobs to our site, please send us an XML feed in response to this email or provide us with your website address.

Information on how to create an XML feed can be found here:

<https://jooble.org/files/XML-feed/EN/jooble-xml-4job.pdf>

## Ladders (US)

<https://recruit.theladders.com/post>

## Jobomas (Mexico)

<https://hrm.jobomas.com/in>

Pricing for posting jobs on Jobomas:



## Glassdoor (US)

Glassdoor is Indeed's partner. When contacted via mail they requested the ATS name and ATS link.

<https://www.glassdoor.com/post-job/>

## Neuvoo (Canada)

Cant access the page

## Jobcredo(Credo) (US)

[Credo | We Connect](https://credosemi.com/)

## Google (US)

Documentation on how to post jobs on Google Jobs Board:

[Intro to How Structured Data Markup Works | Google Search Central | Documentation | Google for Developers](https://developers.google.com/search/docs/appearance/structured-data/intro-structured-data)

## Snaprecruit (US)

<https://snaprecruit.com/>

# Free with Options:

## LinkedIn (US)

[https://www.linkedin.com/](https://www.linkedin.com/feed/)

* Here’s a **step-by-step guide** to post jobs on LinkedIn via the Persimmon app using LinkedIn's Job Posting API:
* **1. Register Your Application on LinkedIn**
* Go to the [LinkedIn Developer Portal](https://developer.linkedin.com/).
* Click **Create App**.
* Fill in the required details:
* App name
* Company name (must match the LinkedIn company for job posting)
* App logo
* Once the app is created, note down your **Client ID** and **Client Secret**.
* **2. Request API Access for Job Posting (Need to Contact LinkedIn)**
* LinkedIn's Job Posting API requires special approval:
* Go to the **Products** section in your app dashboard.
* Add the **Talent Solutions API** or **Job Posting API** to your app.
* Submit your app for LinkedIn review.
* Provide a detailed description of how your app will use the Job Posting API.
* Include sample job posting data or screenshots if necessary.
* **3. Set Up OAuth 2.0 for Authentication**
* LinkedIn uses OAuth 2.0 for authentication. You'll need an access token to interact with the LinkedIn API.
* ***a. Define the Scopes***
* You’ll need the following scopes:
* w\_organization\_social (to manage job posts)
* r\_organization\_social (to read organization details)
* ***b. Implement OAuth Flow in FastAPI***
* Here’s an example of implementing LinkedIn OAuth in your Persimmon app:
* from fastapi import FastAPI, Request   
  from authlib.integrations.starlette\_client import OAuth   
  from starlette.responses import RedirectResponse   
     
  app = FastAPI()   
     
  # OAuth setup   
  oauth = OAuth ()   
  oauth.register(   
   name='linkedin',   
   client\_id='YOUR\_CLIENT\_ID',   
   client\_secret='YOUR\_CLIENT\_SECRET',   
   access\_token\_url='<https://www.linkedin.com/oauth/v2/accessToken>',
* authorize\_url='https://www.linkedin.com/oauth/v2/authorization',   
   client\_kwargs={'scope': 'w\_organization\_social r\_organization\_social'}   
  )   
     
  # Step 1: Redirect to LinkedIn for login   
  @app.get('/login')   
  async def login (request: Request):   
   redirect\_uri = "<http://localhost:8000/auth/callback>"   
   return await oauth.linkedin.authorize\_redirect(request, redirect\_uri)   
     
  # Step 2: Handle LinkedIn callback   
  @app.get('/auth/callback')   
  async def auth\_callback(request: Request):   
   token = await oauth.linkedin.authorize\_access\_token(request)   
   return {"access\_token": token['access\_token']}
* **4. Post a Job via LinkedIn API**
* ***a. Create a Job Post***
* Once you have the access token, use it to create a job post. Here's an example of making a POST request to the /jobs endpoint:
* **Endpoint: POST** [**https://api.linkedin.com/v2/jobs**](https://api.linkedin.com/v2/jobs)
* Example request in Python:
* import requests   
     
  ACCESS\_TOKEN = "your\_access\_token"   
     
  headers = {   
   "Authorization": f"Bearer {ACCESS\_TOKEN}",   
   "Content-Type": "application/json"   
  }   
     
  job\_data = {   
   "company": "urn:li: organization:123456", # Replace with your LinkedIn company URN   
   "title": "Software Engineer",   
   "description": "We are looking for a Software Engineer to join our team.",   
   "location": {   
   "city": "San Francisco",   
   "country": "US"   
   },   
   "employmentType": "FULL\_TIME",   
   "jobPostingInstructions": {   
   "applyMethod": {   
   "type": "COMPANY\_WEBSITE",   
   "url": "<https://example.com/jobs/apply>"   
   }   
   }   
  }   
     
  response = requests.post(   
   "<https://api.linkedin.com/v2/jobs>",   
   headers=headers,   
   json=job\_data   
  )   
     
  print(response.json())
* ***b. Response***
* A successful response will include the job post ID:
* {   
   "id": "1234567890",   
   "status": "ACTIVE"   
  }
* ***c. Check Job Post***
* To verify the posted job, you can:
* Use the **GET /jobs/{jobID}** endpoint to retrieve job details.
* Visit your LinkedIn company's job page.
* **5. Handle Errors**
* LinkedIn's API will return errors if:
* You exceed rate limits.
* Required fields are missing.
* Permissions are not properly set.
* Example error response:
* {   
   "serviceErrorCode": 401,   
   "message": "The token used in the request has expired",   
   "status": 401   
  }
* Always check for and handle these errors gracefully in your application.
* **6. Automate Job Posting Workflow in Persimmon**
* You can create a backend service in Persimmon to manage job postings:
* **Database Storage**: Store job details in your database.
* **Scheduled Jobs**: Use a scheduler (e.g., Celery or APScheduler) to post jobs to LinkedIn periodically.
* **Monitoring**: Log API requests and responses for debugging.
* **7. Testing Job Postings**
* In Development Mode:
* Add test users and companies in the LinkedIn Developer portal.
* Use the API to post jobs under the test company.
* Verify jobs using API responses or the LinkedIn job dashboard.
* **8. Move to Production**
* Submit your app for **Production Approval** in the LinkedIn Developer Portal.
* Once approved, you can start posting jobs for real LinkedIn companies.

## Corp2corp (US)

<https://corp2corpinc.com/>

corp2corp does not have a public API available for posting jobs. I contacted the corp2corp team but did not receive a response.

## Monster(Foundit) (US)

<https://www.foundit.in/>

Foundit.in does not have a public API available for posting jobs. I contacted the Monster/Foundit team but did not receive a response.

**Chat gpt explanation on how to post on monster:**

To post jobs automatically on **Foundit (formerly Monster)**, you can integrate their **Job Posting API** into your system. This allows for automated job posting directly from your platform. Below is a step-by-step guide on how to achieve this:

1. Request API Access from Foundit

* Contact Foundit's **sales** or **support team** to gain access to their API.
* They will provide:
  + API documentation
  + API Key (or authentication credentials)
  + Endpoint URLs for job posting

2. Set Up Authentication

Foundit APIs often use:

* **OAuth 2.0**
* **API key-based authentication**

Ensure your application can handle these types of authentication to securely access the API.

3. Prepare Job Posting Data

Structure the job posting details as required by the API. Most APIs will expect a payload in **JSON** format. For example:

{

"job\_title": "Software Developer",

"job\_description": "We are looking for a Software Developer with 3+ years of experience in Python.",

"skills": ["Python", "Django", "REST API"],

"experience\_min": 3,

"experience\_max": 5,

"job\_location": "Bangalore",

"job\_type": "Full-time",

"industry": "IT Services",

"education": "B.Tech/BE",

"application\_email": "jobs@yourcompany.com",

"expiry\_date": "2025-02-15"

}

4. Integrate the API in Your Platform

Write a script to make HTTP POST requests to Foundit's job posting API endpoint.

Here's an example using Python and the **requests** library:

import requests

# API endpoint and key

API\_URL = "https://api.foundit.com/v1/jobs"

API\_KEY = "your\_api\_key\_here"

# Job data payload

job\_data = {

"job\_title": "Software Developer",

"job\_description": "We are looking for a Software Developer with 3+ years of experience in Python.",

"skills": ["Python", "Django", "REST API"],

"experience\_min": 3,

"experience\_max": 5,

"job\_location": "Bangalore",

"job\_type": "Full-time",

"industry": "IT Services",

"education": "B.Tech/BE",

"application\_email": "jobs@yourcompany.com",

"expiry\_date": "2025-02-15"

}

# Headers for authentication

headers = {

"Authorization": f"Bearer {API\_KEY}",

"Content-Type": "application/json"

}

# Make POST request

response = requests.post(API\_URL, json=job\_data, headers=headers)

if response.status\_code == 201:

print("Job posted successfully:", response.json())

else:

print("Failed to post job:", response.status\_code, response.text)

5. Test the Integration

* Post a few test jobs to ensure the API integration works as expected.
* Verify the posted jobs on your Foundit recruiter dashboard.

6. Schedule Job Posting (Optional)

Use a task scheduler like **Cron** (Linux) or **Celery** (Python) to automate job postings at regular intervals.

7. Monitor and Log

* Log API responses to track success or errors.
* Set up notifications for failed job postings.

8. Use Webhooks (Optional)

If Foundit provides **webhooks**, you can automate job status updates or receive candidate applications automatically when jobs are posted.

Benefits of Automation:

* Saves time and effort in manual posting.
* Ensures consistency in job details.
* Enables bulk posting of jobs.

If Foundit does not provide an API, you can explore **web scraping** with caution or contact their team for alternative automation options.