The Front-end should mandatorily send “Login\_Type” as “JP” for Job-Provider and “JS” for Job-Seeker actions to the backend as given in the input formats.

When a new job is created by the Job Provider, the Job\_Status has to be “AVAILABLE”

Job Seeker should apply/request for a job from the ‘Open jobs’ listings page/tab. While sending to backend, the Job\_Status has to be ‘REQUESTED’.

Job applied by Job seeker can be accepted/rejected by the Job provider from the ‘Requested Jobs’ listings page/tab. While sending to backend, the Job\_Status has to be “ACCEPTED” or “DECLINED” respectively

Job Seeker can cancel a job that was already in “REQUESTED” status. While sending to backend, the Job\_Status has to be “REQUEST\_CANCELLED”

Job Seeker / Job Provider can cancel a job that was already in ‘ACCEPTED’ status from the respective tabs. While sending to backend, the Job\_Status has to be “ACCEPT\_CANCELLED”.

The Job Provider can delete a job from the “Open Jobs” listings page/tab. While sending to backend, the Job\_Status has to be “DELETED”.

When all the open jobs of a job id are accepted, the job has to be set to “READY” status and should not be displayed to Job Seeker for applying.

All capital letters to be used for Job\_Status, and Job\_Type data

Job Progress Update fields from Front-end to Back-end

{

"Job\_Id": event["Job\_Id"],

"Login\_Id\_JS": event["Login\_Id\_JS"],

"Login\_Id\_JP": event["Login\_Id\_JP"],

"Mobile\_Number\_JS": event["Mobile\_Number\_JS"],

"Mobile\_Number\_JP": event["Mobile\_Number\_JP"],

"DateFrom": event["DateFrom"],

"DateTo": event["DateTo"],

"Job\_Address\_Pincode": event["Job\_Address\_Pincode"],

"Job\_Type": event["Job\_Type"],

"Job\_Status": event["Job\_Status"].upper(),

"Name\_JP": event["Name\_JP"],

"Name\_JS": event["Name\_JS"],

"Payment\_Frequency": event["Payment\_Frequency"],

"Requested Wage": event["Requested Wage"],

"Wage\_Maximum": event["Wage\_Maximum"],

"Email\_JP": event["Email\_JP"],

"Email\_JS": event["Email\_JS"],

"Lattitude": event["Lattitude],

"Longtitude": event["Longtitude"]

}

Verify Login – Interface format:

API URL: <https://dk7vxhqz9b.execute-api.ap-south-1.amazonaws.com/jobConnect/Login>

Input:

{

"Login\_Id": "Ranjan102",

"Password": "1234"

}

API response example:

{

    "statusCode": 200,

    "body": "{\"Login\_Id\": \"RanjanRajesh09\", \"Status\": \"Email not yet verified\", \"User\_Name\": \"Rajesh Ranjan\", \"Mobile\_Number\": \"9840690981\", \"Email\": \"testmail123@gmail.com\"}"

}

“Email not yet verified” status indicates that the Login\_Id and Password matches but the Email of the user is yet to be verified. If we make Email verification as optional then we can consider this as a successful verification of Login\_Id and Password.

“Success” status indicates that the Email verification is already done and the Login\_Id and Password also matches.

“Failure” status indicates that the Login\_Id and Password did not match.

Verify Email - interface format:

API URL: https://ck67zhqfe7.execute-api.ap-south-1.amazonaws.com/JobConnect/verifyEmail

To send Access\_Code via E-Mail. Access\_Code will be generated in backend and send via E-Mail corresponding to the Login\_Id, if exists.

{

"Login\_Id": "Ranjan100",

"Action": "SendAccessCode"

}

To verify the Access\_Code. The Access\_Code sent in API request will be verified again the Access\_Code for the Login\_Id. Front-end application has to send the Access\_Code entered by Customer which was received via his/her E-Mail.

{

"Login\_Id": "Ranjan100",

"Action": "VerifyEmail",

"Access\_Code": "694630"

}

The Status field will have "Email Sent" when Access\_Code has been successfully sent via email for the Login\_Id given in input request.

The status field will have "Email Verified" when the Access\_Code has been successfully verified.

New User Registration:

API URL: <https://1d51g1pe1g.execute-api.ap-south-1.amazonaws.com/jobConnnect/registerUser>

Input:

{

"Login\_Id": "Ranjan100",

"Mobile\_Number": "9840690984",

"Login\_Type": "JS",

"Password": "1234",

"User\_Name": "Ranjan",

"Gender": "M",

"Age": "42",

"Profession": "Painter",

"Experience\_Years": "3",

"Experience\_Months": "6",

"Average\_Rating": "4.5",

"Street": "MG Road",

"City": "Yerpedu",

"State": "Andhra Pradesh",

"Pincode": "517619",

"Email": "ranjan.murali82@gmail.com",

"Email\_Status": "UNVERIFIED"

}

API Response, Example:

{

    "statusCode": 202,

    "body": "{\"Login\_Id\": \"RanjanRajesh09\", \"Status\": \"User Already Exist\"}"

}

Add New Job:

API URL: <https://hsfq9bsuv0.execute-api.ap-south-1.amazonaws.com/jobConnect/newJob>

{

  "Login\_Id": "Ranjan100",

  "Job\_Id": "240805182309",

  "Job\_Type": "Home Nurse",

  "Date\_From": "20-Sep-2024 8:00",

  "Date\_To": "24-Oct-2024 18:00",

  "Job\_Description": "Home nurse required for a lady senior citizen who is coming from Australia for a week. Female nurse preferred.",

  "Job\_Openings": "1",

  "Job\_Status": "Available",

  "Job\_City": "Yerpedu",

  "Job\_Pincode": "517619",

  "Job\_State": "Andhra Pradesh",

  "Job\_Street": "Kamma Street",

  "Lattitude": "",

  "Longtitude": "",

  "Payment\_Frequency": "Daily",

  "Wage\_Minimum": "Rs. 650",

  "Wage\_Maximum": "Rs. 750",

  "Name\_JP": "Ranjan100",

  "Mobile\_Number\_JP": "9840690980",

  "Email\_JP": " ranjan.murali82@gmail.com "

}

Get all the Job\_Types:

API URL: <https://dpezos8c2l.execute-api.ap-south-1.amazonaws.com/JobConnect/jobTypes>

getJobDetails:

“City\_Filter” option available only for Job Seekers to filter the jobs available in a particular city.

Job Provider can view the jobs with status in Job\_Status\_List = ["AVAILABLE", "READY", "DELETED"] that are relevant to him.

Both Job Provider and Job Seekers should be able to filter and view the jobs with the below status that are relevant to them.

Job\_Progress\_Status\_List = ["REQUESTED", "ACCEPTED", "REQUEST\_CANCELLED", "ACCEPT\_CANCELLED", "DECLINED"]

While creating a new Job, the Job\_Openings should be >= 1 and should not be zero or negative value. So we can show a default value of 1 for Job\_Openings while creating a new job and allow edit but validate for non-zero or negative value when the Job Poster submits the job.

For Job Seeker, the input format is: (Please note the City\_Filter is a mandatory field)

{

"Login\_Id": "Hiji1490",

"Login\_Type": "JS",

"Job\_Status": "AVAILABLE",

"Jobs\_Filter": [“PAINTER”],

"City\_Filter":[]

}

For Job Provider, the input format is:

{

"Login\_Id": "Ranjan100",

"Login\_Type": "JP",

"Job\_Status": "AVAILABLE",

"Jobs\_Filter": [

"PAINTER",

"PAINTER",

"PLUMBER",

"CARPENTER",

"HOME NURSE"

]

}