

Jobs

Types

JobHeader

Name	Type	Notes
id	uuid string	Unique id of job
title	string	
company	string	
category	short string or integer	Unique id of category lookup
department	short string or integer	Unique id of department lookup
specialization	short string or integer	Unique id of specialization lookup
positionType	short string or integer	Unique id of position type lookup
experience	short string or integer	Unique id of experience lookup
education	short string or integer	Unique id of education lookup
salary	string	This could be TBD, a single figure, a range or something else, so we can't use a number
city	string	
province	string	e.g. ON
postDate	ISO date string	
closingDate	ISO date string	
logo	string	Company logo image url

saved	boolean	Indicates whether the user has saved the job
appliedDate	ISO date string (optional)	The date the user applied for this job. If the user hasn't applied for this job yet, this should be null or undefined.

JobLocation

Name	Type	Notes
address	string	Street address
city	string	
province	string	e.g. ON
postalCode	string	

JobContact

Name	Type	Notes
mailingAddress	string (optional)	
fax	string (optional)	
phone	string (optional)	
email	string (optional)	

JobDetail

Name	Type	Notes
<i>Include all JobHeader fields, plus:</i>		
description	Markdown string	The text of the job post,

		which will be converted to HTML for display
contact	JobContact	
location	JobLocation	

JobFilterLocation

Name	Type	Notes
city	string	
province	string	e.g. ON

JobFilter

Name	Type	Notes
keywords	string (optional)	Keywords to search for in the title and description fields of the job
category	short string or integer (optional)	Unique id of category lookup
department	short string or integer (optional)	Unique id of department lookup
specialization	short string or integer (optional)	Unique id of specialization lookup
positionType	short string or integer (optional)	Unique id of position type lookup
experience	short string or integer (optional)	Unique id of experience lookup
education	short string or integer	Unique id of education lookup
postDate	ISO date string (optional)	Include jobs posted on or after this date

location	JobFilterLocation (optional)	
----------	------------------------------	--

LocalName

Name	Type	Notes
en	string	English name
fr	string	French name

Lookup

Name	Type	Notes
id	short string or integer	Unique id of lookup record
name	LocalName	

JobApplication

Name	Type	Notes
id	unique id	Unique id of application
job	JobHeader	
appliedDate	ISO datetime string	

Methods

GET /jobs

Gets a filtered, paginated list of jobs matching a filter.

Request body:

Name	Type	Notes
startIndex	integer	Index of first record in result set to return
count	integer (max 50)	Number of jobs to return
filter	JobFilter	

Response body:

Name	Type	Notes
jobs	Array of JobHeader	Array of jobs from the result set, from start index up to count
totalCount	integer	Total number of jobs in result set, which match the filter

HTTP status codes:

200 (must have valid response body), 4xx/5xx (any errors)

GET /job/[jobId]

Gets the details of a single job.

URL arguments:

Name	Type	Notes
jobId	uuid string	Unique id of job

Response body:

JobDetail object

HTTP status codes:

200 (must have valid response body), 404 (job doesn't exist), 4xx/5xx (any other errors)

GET /categories

Gets all the available categories.

Response body:

Array of Lookup

HTTP status codes:

200 (must have valid response body), 4xx/5xx (any errors)

GET /departments

Gets all the available departments.

Response body:

Array of Lookup

HTTP status codes:

200 (must have valid response body), 4xx/5xx (any errors)

GET /specializations

Gets all the available specializations.

Response body:

Array of Lookup

HTTP status codes:

200 (must have valid response body), 4xx/5xx (any errors)

GET /position-types

Gets all the available position types.

Response body:

Array of Lookup

HTTP status codes:

200 (must have valid response body), 4xx/5xx (any errors)

GET /experiences

Gets all the available experience values.

Response body:

Array of Lookup

HTTP status codes:

200 (must have valid response body), 4xx/5xx (any errors)

GET /educations

Gets all the available education values.

Response body:

Array of Lookup

HTTP status codes:

200 (must have valid response body), 4xx/5xx (any errors)

POST /jobs/save

Save a job for the logged-in user

Request body:

Name	Type	Notes
id	uuid string	Unique id of job

Response body:

JobDetail

HTTP status codes:

200 (must have valid response), 404 (job not found), 4xx/5xx (any other errors)

POST /jobs/unsave

Unsave a job for the logged-in user

Request body:

Name	Type	Notes
id	uuid string	Unique id of job

Response body:

JobDetail

HTTP status codes:

200 (must have valid response), 404 (job not found), 4xx/5xx (any other errors)

GET /jobs/saved

Get all the saved jobs for the logged-in user

Response body:

Array of JobHeader

HTTP status codes:

200 (must have valid response), 4xx/5xx (any other errors)

POST /jobs/[jobId]/apply

Submit a job application for the logged-in user.

Request body:

Name	Type	Notes
cellPhone	string (optional)	
coverLetterText	string (optional)	Text body of cover letter
resumeld	unique id of resume	

Response body:

Name	Type	Notes
cellPhone	string (optional)	
coverLetterText	string (optional)	Text body of cover letter
resumeld	unique id of resume	
job	JobHeader	

HTTP status codes:

200 (must have valid response), 404 (job not found), 4xx/5xx (any other errors)

GET /jobs/applications

Return the logged-in user's job application history.

Response body:

Array of JobApplication

HTTP status codes:

200 (must have valid response), 4xx/5xx (any other errors)

Profile

Types

CoverLetterContent

Name	Type	Notes
name	string	
description	string (optional)	
text	string	Plain text body of cover letter
active	boolean	Flag for user to toggle whether or not this cover letter will appear in the select cover letter dropdown when applying for a job. Included just for the user's convenience.

CoverLetter

Name	Type	Notes
<i>Include all CoverLetterContent fields, plus:</i>		
id	uuid string/integer	Unique id of cover letter
modifiedDate	ISO datetime string	The last time any field on the record was modified. Should be updated any time the APIs modify the cover letter record.

ResumeContent

Name	Type	Notes
name	string	
description	string (optional)	
fileId	unique id of resume file	
active	boolean	Indicates whether to show in the dropdown for the user to select. Multiple resumes can be active.
searchable	boolean	Indicates whether employers can find it in a search

Resume

Name	Type	Notes
<i>Include all ResumeContent fields, plus:</i>		
id	unique id	Unique id of resume
text	string	Text content extracted from the PDF
processing	boolean	Indicates whether the PDF is still be processed to extract text.
modifiedDate	ISO datetime string	Indicates the last date/time any field was changed.

ResumeFile

Name	Type	Notes
------	------	-------

id	unique id	Unique id of resume file
url	string	Download url of file

ProfileAddress

Name	Type	Notes
street	string (optional)	
city	string (optional)	
postalCode	string (optional)	
province	string	e.g. ON

Profile

Name	Type	Notes
firstName	string	
lastName	string	
email	string	
password	string (optional)	Only included if updating the password
oldPassword	string (optional)	Only included if updating the password
phone	string (optional)	
phoneExt	string (optional)	
cellPhone	string (optional)	
address	ProfileAddress	
choice1	unique id of job specialization (optional)	

choice2	unique id of job specialization (optional)	
choice3	unique id of job specialization (optional)	
newGraduate	boolean	
coopStudent	boolean	

SavedSearch

Name	Type	Notes
id	unique id	
name	string	
filter	JobFilter	
createdDate	ISO date string	

Methods

GET /profile/cover-letters

Returns all the cover letters saved by the logged-in user.

Response body:

Array of CoverLetter

HTTP status codes:

200 (must have valid response body), 4xx/5xx (any errors)

POST /profile/cover-letter

Adds a new cover letter for the logged-in user.

Request body:

CoverLetterContent

Response body:

CoverLetter

HTTP status codes:

200 (must have valid response body), 4xx/5xx (any errors)

POST /profile/cover-letter/[coverLetterId]

Overwrites a cover letter the logged-in user has previously saved. All field values should be overwritten with the values provided, including nulls.

Request body:

CoverLetterContent

Response body:

CoverLetter

HTTP status codes:

200 (must have valid response body), 404 (cover letter doesn't exist), 4xx/5xx (any errors)

DELETE /profile/cover-letter/[coverLetterId]

Deletes a cover letter the logged-in user has previously saved. Returns the deleted record.

Response body:

CoverLetter

HTTP status codes:

200 (must have valid response body), 404 (cover letter doesn't exist), 4xx/5xx (any other errors)

POST /profile/cover-letter-active/[coverLetterId]

Sets the active flag on a cover letter the user has previously saved.

Request body:

Name	Type	Notes
active	boolean	Value for the cover letter active field.

Response body:

CoverLetter

HTTP status codes:

200 (must have valid response body), 404 (cover letter doesn't exist), 4xx/5xx (any other errors)

GET /profile/resumes

Gets all resumes the logged-in user has previously saved.

Response body:

Array of Resume

HTTP status codes:

200 (must have valid response body), 4xx/5xx (any other errors)

POST /profile/resume-file

Uploads a resume PDF file. Called before creating the resume record.

Request body:

Multipart file content

Response body:

ResumeFile

HTTP status codes:

200 (must have valid response body), 4xx/5xx (any other errors)

POST /profile/resume

Saves a resume for the logged-in user. Must have uploaded the resume file first.

Request body:

ResumeContent

Response body:

Resume

HTTP status codes:

200 (must have valid response body), 4xx/5xx (any other errors)

POST /profile/resume-active/[resumeld]

Sets the active flag on a resume the user has previously saved.

Request body:

Name	Type	Notes
active	boolean	Value for the resume active field.

Response body:

Resume

HTTP status codes:

200 (must have valid response body), 404 (resume doesn't exist), 4xx/5xx (any other errors)

POST /profile/resume-searchable/[resumeld]

Sets the searchable flag on a resume the user has previously saved.

Request body:

Name	Type	Notes
searchable	boolean	Value for the resume searchable field.

Response body:

Resume

HTTP status codes:

200 (must have valid response body), 404 (resume doesn't exist), 4xx/5xx (any other errors)

POST /profile/resume/[resumeld]

Overwrites a resume the logged-in user has previously saved. All field values should be overwritten with the values provided, including nulls.

Request body:

ResumeContent

Response body:

Resume

HTTP status codes:

200 (must have valid response body), 404 (resume doesn't exist), 4xx/5xx (any errors)

DELETE /profile/resume/[resumeld]

Deletes a resume the logged-in user has previously saved. Returns the deleted record.

Response body:

Resume

HTTP status codes:

200 (must have valid response body), 404 (resume doesn't exist), 4xx/5xx (any other errors)

POST /profile/register

Register a user.

Request body:

Name	Type	Notes
firstName	string	
lastName	string	
email	string	
phone	string (optional)	
province	string	e.g. ON
password	string	

Response body:

Profile

HTTP status codes:

200 (must have valid response body), 4xx/5xx (any other errors)

GET /profile/details

Get the user profile details.

Response body:

Profile

HTTP status codes:

200 (must have valid response body), 4xx/5xx (any other errors)

POST /profile/details

Update the user profile details.

Request body:

Profile

Response body:

Profile

HTTP status codes:

200 (must have valid response body), 4xx/5xx (any other errors)

POST /profile/search

Save a search for the logged-in user.

Request body:

Name	Type	Notes
name	string	
filter	JobFilter	

Response body:

SavedSearch

HTTP status codes:

200 (must have valid response), 4xx/5xx (any other errors)

GET /profile/searches

Gets all the previously saved searches for the logged-in user.

Response body:

Array of SavedSearch

HTTP status codes:

200 (must have valid response), 4xx/5xx (any other errors)

DELETE /profile/search/[searchId]

Deletes a previously saved search. Returns the deleted search.

Response body:

SavedSearch

HTTP status codes:

200 (must have valid response), 4xx/5xx (any other errors)

Localization

Methods

GET /localized/[groupName]

Returns a named group of localized strings.

URL arguments:

Name	Type	Notes
groupName	string	

Request headers:

Name	Type	Notes
Accept-Language	string	e.g. en, fr

Response body:

Object with each key in the group and the corresponding string value in the specified language.

For example, for 'Accept-Language: en':

```
{
  'key1': 'english text 1',
  'key2': 'english text 2'
}
```

HTTP status codes:

200 (must have valid response), 4xx/5xx (any errors)