**Welcome to Software Engineering I Project B Group 9: Bounty Board API**

This is a REST API for posting job and user profile information to our backend database.

**Database Objects**

**Object: Job**

**Attributes:**

1. **id:** Required, Unique, String. Automatically assigned by the API for each job added.
2. **posterName:** Required, String.Name of the user who create the posting.
3. **jobTitle:** Required, String. A title for the job posting.
4. **jobDescription:** Required, String. A description of the posted job.
5. **reputationScore:** Float. An average rating for the posting user. This will eventually be tied to the User object.
6. **payRate:** Float. Rate of pay for a job. Can be hourly or lump sum.
7. **isHourly:** Boolean. Should be set to true for hourly payrates, false for lump sum.
8. **timeEstimate:** String. Estimated time required to complete the job.
9. **city:** String. City in which the job is to be performed.
10. **state:** String. State in which the job is to be performed.
11. **streetAddress:** String. Street address where the job is to be performed.
12. **isInProgress:** Boolean. Defaults to false. Should be set to true when job is assigned.
13. **datePosted:** String. Date on which the job posting was added to the database. Assigned automatically when the job is created.

**HTTP Requests**

**Base URL:** <https://postskynetbountyboard.appspot.com>

**Post a new job** – Add a new job posting

POST /job

Code: 201 Created = Success

Body: JSON

Body Format Example:

{"poster\_name":"Andrew Ryan","job\_title":"Would you kindly play golf with me?","job\_description":"Plastic surgery assistance needed","reputation\_score":4.2,"pay\_rate":10.50,"is\_hourly":false,"time\_estimate":"A minute or so...","city":"Rapture","state":"Ociania","street\_address":"Fort Frollic"}

**Get all jobs** – Get a list of last 100 jobs (this can be expanded) returned as a JSON string.

GET /job

Code: 200 = Success

Code: 404 Not Found = No jobs exist in the database

Body: None

**Get a job by Id** – Get a single job by Id

GET /job/{job\_id}

Code: 200 = Success

Code: 404 Not Found = Target Id not found

Body: None

**Get all jobs by city (filtered)** – Get a list of all jobs in a specific city (case insensitive)

POST /job/city

Code: 200 = Success

Code: 404 Not Found = No jobs exist in the database that meet the requirements

Body: {“city”:city }

Example: {“city”:“chicago” }

**Get all jobs by state (filtered)** – Get a list of all jobs in a specific state (case insensitive)

POST /job/state

Code: 200 = Success

Code: 404 Not Found = No jobs exist in the database that meet the requirements

Body: {“state”:state }

Example: {“state”:“washington” }

Note: Case insensitive, but you might want to control if states are entered as a full name or a 2- letter code.

**Get all jobs by city *and* state (filtered)** – Get a list of all jobs in a specific city and state (case insensitive)

POST /job/citystate

Code: 200 = Success

Code: 404 Not Found = No jobs exist in the database that meet the requirements

Body: {“city”:city, “state”:state }

Example: {“city”:“seattle”, “state”:“Washington” }

**Get all jobs with a minimum rating (filtered)** – Get a list of all jobs with a rating >= a target rating (case insensitive)

POST /job/rating

Code: 200 = Success

Code: 404 Not Found = No jobs exist in the database that meet the requirements

Body: {“reputation\_score”:score }

Example: {“reputation\_score”:4.0}

**Get all jobs by city, state and rating (filtered)** – Get a list of all jobs in a specific city and state that have a minimum rating (case insensitive)

POST /job/location/rating

Code: 200 = Success

Code: 404 Not Found = No jobs exist in the database that meet the requirements

Body: {“city”:city, “state”:state, “reputation\_score”:score }

Example: {“city”:“seattle”, “state”:“Washington”, “reputation\_score”:4.0 }

**Get all jobs by keyword/phrase (filtered)** – Get a list of all jobs with titles that contain a keyword or phrase (case insensitive)

POST /job/keyword

Code: 200 = Success

Code: 404 Not Found = No jobs exist in the database that meet the requirements

Body: {“keyword”:keyword}

Example: {“keyword”:“trash”}

**Get all jobs newer than a target date (filtered)** – Get a list of all jobs with posting dates equal to or newer than a target date. Date format must be mm/dd/yyyy.

POST /job/date

Code: 200 = Success

Code: 404 Not Found = No jobs exist in the database that meet the requirements

Body: {“date”:date}

Example: {“date”:“05/06/2017”}

**Delete a job** – Delete a single job by Id

DELETE /job/{job\_id}

Success Code: 200

Code: 404 Not Found = Target Id not found

Body: None

**Modify an existing job** - Change the value of any combination of the following attributes of an existing job: job\_title, poster\_name, job\_description, pay\_rate, is\_hourly, time\_estimate, city, state, and street\_address.

PATCH /job/{job\_id}

Success Code: 201 Create

Code: 404 Not Found = Job with target Id was not found

Body: A JSON string containing key:value pairs for any combination of the modifiable attributes

listed above.

**Assign a job** – Set a job to isInProgress = true to indicate that the job has been taken

PUT /job/assign/{job\_id}

Success Code: 201 Create

Code: 403 Forbidden = Job with target Id is already assigned (isInProgress = true)

Body: None

**Unassign a job** – Set a job to isInProgress = false to indicate that the job is open

PUT /job/unassign/{job\_id}

Success Code: 201 Create

Code: 403 Forbidden = Job with target Id is not assigned (isInProgress = false)

Body: None