

# Levels Beyond Coding Assessment

There are two separate tasks to complete for this exercise. One for backend work and another for frontend work. This is intended to be an example of what you would commit in a real-world scenario so keep in mind all the items that go into delivering software, if there are limitations that are beyond the scope of this exercise feel free to call those out for us and give us your thoughts.

## Backend

---

Your Backend task is to build a simple RESTful, JSON API to power a note-taking application.

### Guidelines

The API should be implemented in Java. Any frameworks/libraries you use are up to you. I should be able to download your code from github.com and run it on my machine, so you should include instructions for setup/configuration. The easier it is for me to setup and deploy, the better. We use Macbooks, please keep that in mind. The GET and POST calls are outlined below, please complete the other basic CRUD calls on your own.

### Notes

The notes API should live at the route `/api/notes`. So, if your API server is running on localhost, I would expect to access the 'notes' API at `http://localhost/api/notes`.

### The Note Model

```
{
  "id" : 1,
  "body" : "Ask Larry about the TPS reports."
}
```

### Create a New Note

When I POST note JSON to the notes route, a new note will be created.

```
POST /api/notes
```

```
BODY a note
```

```
Returns: a saved note...
```

Example

```
curl -i -H "Content-Type: application/json" -X POST -d '{"body" : "Pick up milk!"}' http://localhost/api/notes
```

Returns:

```
{
  "id" : 2,
  "body" : "Pick up milk!"
}
```

## Get an Existing Note

I can get a note using an API call:

```
GET /api/notes/{id}
```

```
Returns: the requested note..
```

Example:

```
curl -i -H "Content-Type: application/json" -X GET http://localhost/api/notes/1
```

Returns:

```
{
  "id" : 1,
  "body" : "Ask Larry about the TPS reports."
}
```

## Get All of my Notes

I can get all notes using an API call:

```
GET /api/notes
```

```
Returns: A list of my notes
```

Example:

```
curl -i -H "Content-Type: application/json" -X GET http://localhost/api/notes
```

Returns:

```
[
  {
    "id" : 2,
    "body" : "Pick up milk!"
  },
  {
    "id" : 1,
    "body" : "Ask Larry about the TPS reports."
  }
]
```

Also, I'd like to be able to pass in an optional query parameter that will allow me to search notes by their bodies.

Example:

```
curl -i -H "Content-Type: application/json" -X GET http://localhost/api/notes?query=milk
```

Returns a list of every note with the word 'milk' in it.

## Front End

---

Your task is to integrate with the Github REST API to search for issues in the [Angular Github](#) repo for the previous 7 days.

### Guidelines

Use the [Github Issues API](#). Working with AngularJS or ReactJS will be seen as a plus, we want to see how you work in the languages we use here at Levels Beyond. With the results from the API, display, in HTML, the returned values with their title, body, user login, and assignee login.

### Notes

The HTML display can be as plain or intricate as you like. We value UI/UX highly.

