



# Backend Engineer Assignment

Congratulations! Your application piqued our interest. We would now like to see a bit how you tick!

Imagine you have a user account at [doodle.com](https://doodle.com). You can list the *doodles* you have created or participated in at <https://doodle.com/dashboard>. In our daily business, we call those *doodles* polls, which is a term normally used for elections. What we would want you to do is implement an alternative version of a user's list of polls.

We understand your time is precious and would not want you to spend more than 3 hours on this over the span of 1 week max. The outcome should be runnable locally using docker-compose or Kubernetes. Please provide all instructions (or a documented script) needed to run your assignment (i.e. how to build the images and run the containers).

It is ok if the assignment is not completed. Try to prioritize by what you think is more important. Tell us (in the form of a README file) what motivated your technologies choices, how you tackled the task, what you would do differently were you given more time, what you would differently a second time around, etc.

Once your assignment is completed please archive it (zip, tar, tar.gz will do) and send it via email to:

**[lfa@doodle.com](mailto:lfa@doodle.com)**

Please make sure you include the `.git` folder in your compressed file, and please do not publish your code online.

We would like you to develop a simple API endpoint that reads data from a database. You are free to use any of the following technologies:

- Code: preferably Java, possibly other JVM languages
- Database: MongoDB, Redis, MySQL, Postgres

Save the JSON file at the following URL:

**<https://boiling-tor-31289.herokuapp.com/users/me/polls>**

The file contains a list of Doodle polls in JSON format. We would like you to:

- Provide a way to import the JSON polls into one of the above containerized databases. We will use this to populate the database before running your project

- Design an API endpoint to list all polls created by a user
  - Data should be fetched from the database
  - The endpoint should return a list of polls in JSON format
  - The API should support text search
  - The API should support filtering on the creation date (e.g. return all polls created after 1 January 1970)

## Deployment (optional)

If you still have some time (congrats :)) why not deploying your project? Feel free to pick your favourite platform and share with us an address or a public IP!