

Hello!

Thank you very much for taking the time to apply to Mentimeter as our next QA Lead/DevOps Engineer. In order to help us figure out if we are a good fit for one another, we would like you to complete a task that, should it be completed well enough, will give us a good base for discussion for the rest of the interview process.

While we would like to keep the *scope* of this case small, we would like to stress that we are most interested in *how* you complete the case. We are looking for a product that you think is “production ready” (or as close as is possible within the imaginary scope of this task). “Production ready” for us means - ready to use by millions of users and ready for other developers to work on in a sustainable way.

We would like you to particularly think about how you would safely deploy this code with many other developers working in the same code base, and how those other developers can run this code easily.

For example, you could dockerize the application and implement a strategy to automatically run its tests at important points in the development cycle.

Given that this is a (lot) simplified and slightly unrealistic scenario, it is okay to take shortcuts where you think it’s needed and appropriate, in order to not spend too much time on the case.

Submission

Please finish off by writing a paragraph or two about the tradeoffs you made in your implementation, which shortcuts you may have taken and why, and the main things missing for your code to “go to production”.

Please make a git(hub) repo containing your solution. If you do not want the repository to be public, please add github.com/ostenbom as a collaborator, or send us a zip of the entire repository including its .git directory. If you are submitting as a zip, please email the solution to ingrid.joelsson@mentimeter.com We will be paying attention to the structure, documentation and production quality of the task.

After

The Task

We would like you to create a much simplified, static mentimeter “results” page for a single multiple-choice question.

It should consist of:

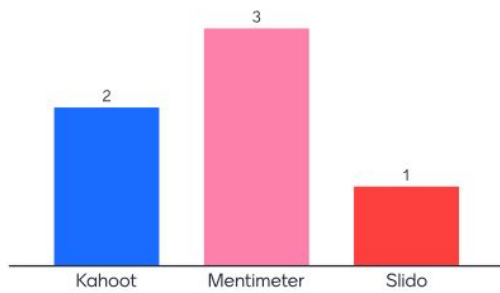
- A **backend service** that pulls information about the question from
 - the “questions” endpoint:
<https://api.mentimeter.com/questions/48d75c359ce4>
 - And the “results” endpoint:
<https://api.mentimeter.com/questions/48d75c359ce4/result>

The backend service should respond with a JSON payload containing the question name and results, for example your service should respond on <http://localhost:backendport/48d75c359ce4> with:

```
{
  "question": "Which is the best interactive presentation platform?",
  "results": [
    {
      "id": 1,
      "label": "Kahoot",
      "score": [
        2
      ]
    },
    {
      "id": 2,
      "label": "Mentimeter",
      "score": [
        3
      ]
    },
    {
      "id": 3,
      "label": "Slido",
      "score": [
        1
      ]
    }
  ],
}
```

- A **frontend service** that should pull JSON from your backend service and display the results. Looks don't matter. Do use a 3rd party library to help you make a bar-chart. For example, <http://localhost:frontendport/48d75c359ce4> should respond with something like:

Which is the best interactive presentation platform?



You are free to use any languages or technologies that you feel comfortable with. We predominantly use React, Node, Ruby and Go if you feel like dabbling with our tech stack.