

# Email Guesser

Let's imagine that Babbel is offering a new service to other companies.

The new offering to businesses aims to make us a lucrative option for companies to promote language learning for their employees. We often reach out to different representatives from businesses to promote the value proposition we offer.

We maintain a database of the contacts we already have in various companies. However, we do not always have the email addresses of the people we should pitch to in these companies. We do have their full names and company domain.

In this exercise, you have to create an end-to-end solution for the use case above. The solution has two parts:

1. A tiny SPA (Single Page Application) with the UI to accept the full name and company domain and display the derived email address. The SPA would ideally use React for implementation. You do not have to add authentication to the SPA. Design can be minimal and the visual elements can be simple. We aren't expecting anything fancy. Also, for the UI - feel free to use your creativity :)
2. An HTTP microservice with a single endpoint to derive the email address for a person, given their full name and company domain. We can assume that all email addresses of one company follow the same format. That is, if we know the full name and email address of one person in the company, we should be able to derive the email addresses of other employees given their full name. Feel free to use any preferred backend language. At Babbel, we use JavaScript/TypeScript (NodeJS), Ruby, Python and Go.

For this exercise, we can assume that the email addresses are always one of the following two formats (for an employee called "Jane Doe" working at a company with a domain "domain.com"):

- first\_name\_last\_name : [janedoe@domain.com](mailto:janedoe@domain.com)
- first\_name\_initial\_last\_name : [jdoe@domain.com](mailto:jdoe@domain.com)

## The challenge

Assuming a sample data set exists in a static JSON file such as the one below:

```
{
  "Jane Doe": "jdoe@babbel.com",
  "Jay Arun": "jayarun@linkedin.com",
  "David Stein": "davidstein@google.com",
  "Mat Lee": "matlee@google.com",
  "Marta Dahl": "mdahl@babbel.com",
  "Vanessa Boom": "vboom@babbel.com"
}
```

Derive the email addresses of the following contact persons after determining the email format from the sample set. The service should be able to handle a request where a derivation is not possible.

1. Nina Simons, "babbel.com"
2. Priya Kuber, "linkedin.com"
3. Matthew Hall, "google.com"
4. Robert Miller, "slideshare.net"

## Points to remember

- Do not spend more than 8 hours on this task.
- We don't expect everything to be finished. Try to focus on completing at least one portion of the challenge: frontend or backend.
- Please write tests for (at least) the most significant code flows.
- If you have any questions, feel free to reach out.
- Have fun!