



Introduction

Hello fellow Code Enthusiast!

As part of our interview process we would like you to complete a programming test. The purpose of this assignment is to develop a small app. Please read the description, then create an app to solve the problem at hand.

Technical requirements

Please use the following technologies:

JS, Git, SCSS any third party library and framework can be used

Description

As part of the test assignment we would like to ask you to build a web application which tackles the everyday problems we face at Wunderfleet. Please create a git repository for this task, before you start working.

The web application should be a basic user registration, like typically known from similar projects. The registration process contains 4 separated steps. Only one step is shown at a time to the customer.

View 1: Insert personal information

- Firstname, lastname, telephone

View 2: Insert address information

- Address including street, house number, zip code, city

View 3: Insert payment information

- Account owner
- IBAN (doesn't need to be validated)

View 4: Success page

- Success message, when data are successful saved
- Show the “paymentDataId”, returned by demo endpoint

Endpoint to use

When clicking the “next” button on View 3, the inserted data should be saved locally in the browser storage and the payment data should be posted to our external “demo-payment” api with the following configuration:

POST save-payment-data

`https://37f32cl571.execute-api.eu-central-1.amazonaws.com/default/wunderfleet-recruiting-backend-dev-save-payment-data`

the endpoint returns 200 OK on success with a response object containing the `paymentDataId` property

BODY raw

```
{
  "customerId": 1,
  "iban": "DE8234",
  "owner": "Max Mustermann"
}
```

Endpoint:

`https://37f32cl571.execute-api.eu-central-1.amazonaws.com/default/wunderfleet-recruiting-backend-dev-save-payment-data`

Save the “paymentDataId” to the customer dataset in your database.

On errors, we’re returning an error and 4xx Status Code. In the raw Body of the POST request you need to replace the example data with the inserted data from the user.

Additional information

1. The user is able to leave the registration on every step/view until he finished the whole registration successfully.
2. Accordingly to this he should be redirected to the last opened step when he’s joining the process again(re-navigating to the webpage). So the state of already inserted data needs to be saved somehow.

3. There is no need to validate all form inputs, just use the basic html validators if present.
4. Blank input is not valid and won't be tested.
5. You don't need to implement access control features.
6. Please add your custom SCSS style to your project, according to the <https://www.wundermobility.com/> corporate identity. Please think about a good SCSS structure. Feel free to extend existing libraries.
7. When you are ready and done, please create a remote repository in Bitbucket, push your project into this repository and share it with us.
8. Please add a README.txt in your project and answer the following questions in this file:
 - a. Describe possible optimizations for your code.
 - b. Which things could be done better, than you've done it?
9. Keep the UI minimal but straightforward! If you feel the need to add some cool features to our small app, feel free to do so. It will be a big plus if you structure your code using
 - a. Angular/React .
 - b. Typescript
 - c. Reactive Programming
 - d. State Management minimal structure for the JS application (using redux/ngrx) and have abstraction layers in place (effects/reducers/actions)
 - e. Reactive forms

Resources

Feel free to use any resources such as :

- bootstrap or angular material
- boilerplate for an Angular/React project with pre-setup structure
- forms library

Feel free to contact us with any questions regarding the assignment or the submission process. We would be happy to help!

Good luck! :)