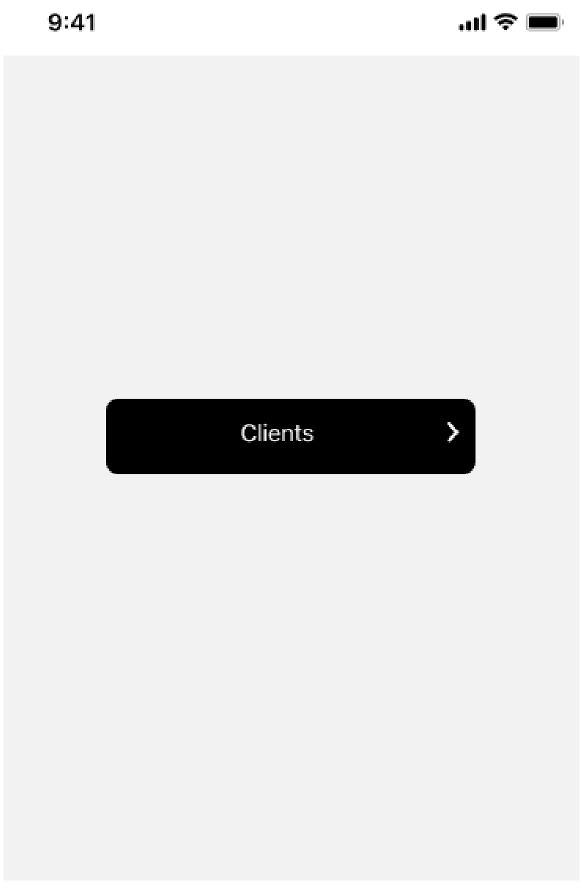
# React-native test

This Test of react native consist on create Three screens:

- Home Page
- Customers
- Customer details

## Home page

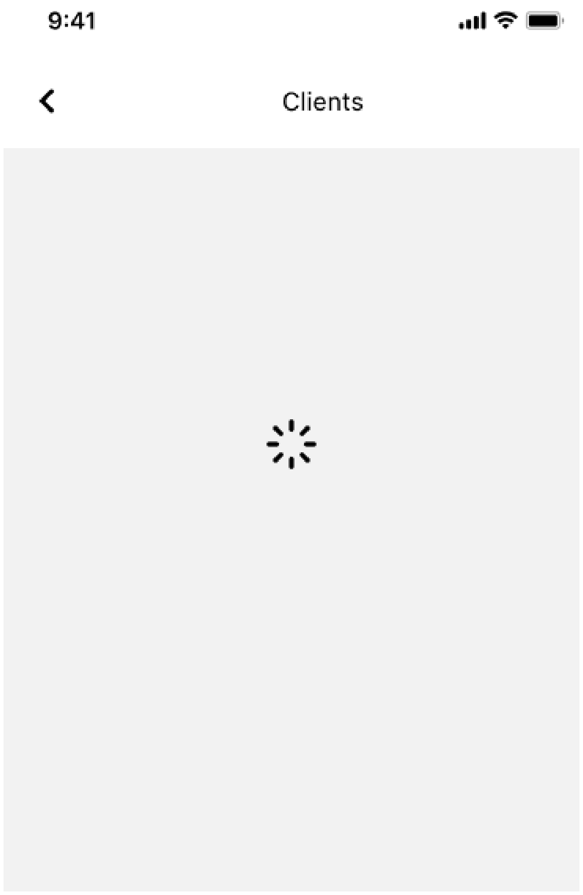


CTA to navigate to screen `Customers`

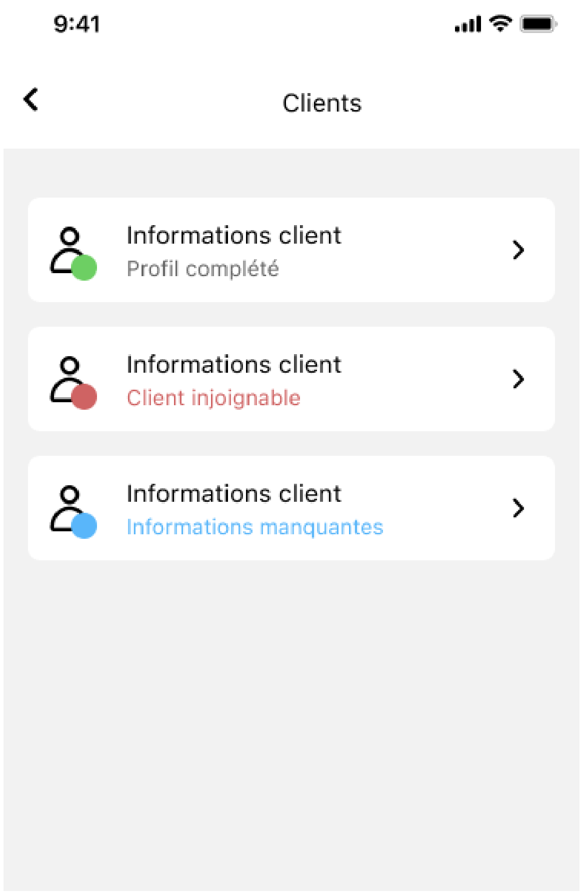
## Customers Page

Listing of retrieved customers via `/customers` service ( Mock service) see `postman` on root:

On arriving to this page a loader appears (you can use default loader ) when loading customers:



Once that loaded you should use adapter to parse data to `Customer` data :



Informations client
Profil complété

Informations client
Client injoignable

Informations client
Informations manquantes

## Component `CustomerShortInfo` :

If the customer missing data :

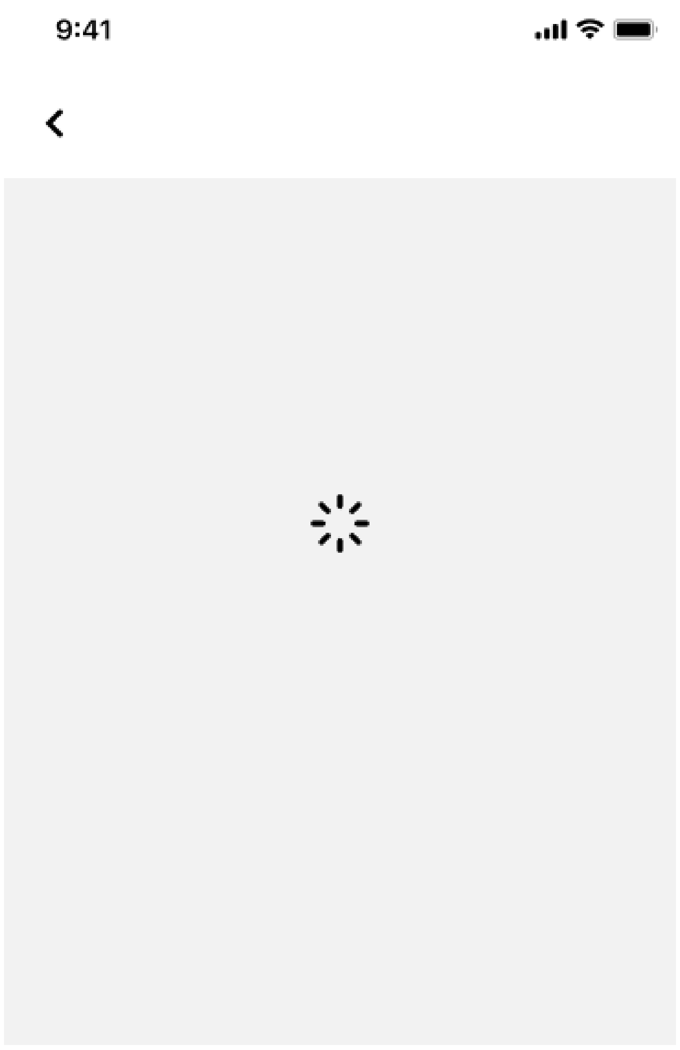
- `email` or `phone` The color should be **red** (see `$colors` under `src/layout/colors.ts` ) and text should be **“Client injoignable”**
- `address` or `birthday` the color should be blue  (see `$colors` under `src/layout/colors.ts` ) and text should be **“Informations manquantes”**
- All thoses fields are filled the color should be green and text **Profil complété**

onPress the user redirect to **customer details page**

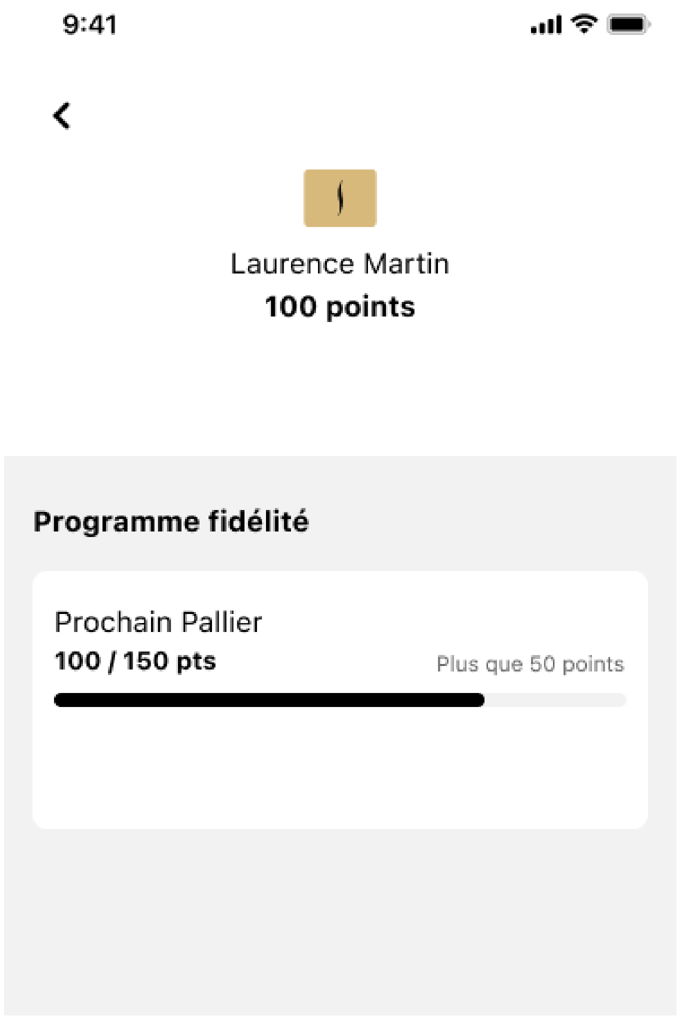
## Customer Page details

Once we screen is loaded we fetch the customer details by Id ( the customer details contains more details) the service is `/customer/{clientID}` see postman collection on root project

When loading details :



Once the customer details loaded :


9:41

Laurence Martin

**100 points**

**Programme fidélité**

Prochain Pallier

**100 / 150 pts** Plus que 50 points

## Profile infos

1 - Customer card :

Based on the **loyalty status**:

- For `Black` use black card (see the default setup on the current homepage )
- For `White` use red card
- For `Gold` use gold card

Example on the default setup `./src/App.tsx` :

```
<CardIcon color={$color.red} />
<CardIcon color={$color.black} />
```

```
<CardIcon color={$color.gold} />
```

2 - Full name of the customer

3 - Points (based on loyalty )

4 - Progress bar that shows their current `points` and how many more points they need to reach the next `150-point` threshold.

## Bonus

Components:

- To ensure that our application is accessible and usable for users who read from right to left, it's crucial that the component fully supports right-to-left ( `RTL` ) languages and text direction.
- Manage Services errors ( you can show red message if service fails)

Variantes / Schemas:

- Create on iOS and android variante DEV / E2E / PROD

## Requirement :

- T**o promote code reusability and ensure reliability, it's important for all components to be thoroughly tested and designed with reusability in mind.**
- **The code should be clean and have the capability to add features.**
- **Remember to avoid using large files, and make sure that each function is designed to do only one thing.**

## Git

- Create your repo with branch `feature/customers`

## ***Good Luck*** 😊