Home Exercise - Android

First of all, welcome to this home exercise. We know your time is valuable and that's why we do not want you to spend more than about two hours on this exercise. We would like you to write a simple Android app so we can get a first impression of your work.

If you proceed to the next stage of the hiring process, your Android app will be used during the pair programming session where we extend it with more features. Therefor, keep in mind that when you write your app, it will grow in size during the hiring process.

Description

When an end-customer pays through Payoneer, our frontend software presents a list of payment methods like Visa, SEPA and PayPal. Our frontend software loads this list of payment methods using a token-based list URL.

We would like you to write a small app that loads this JSON list and presents the payment methods in a scrolling list in your app.

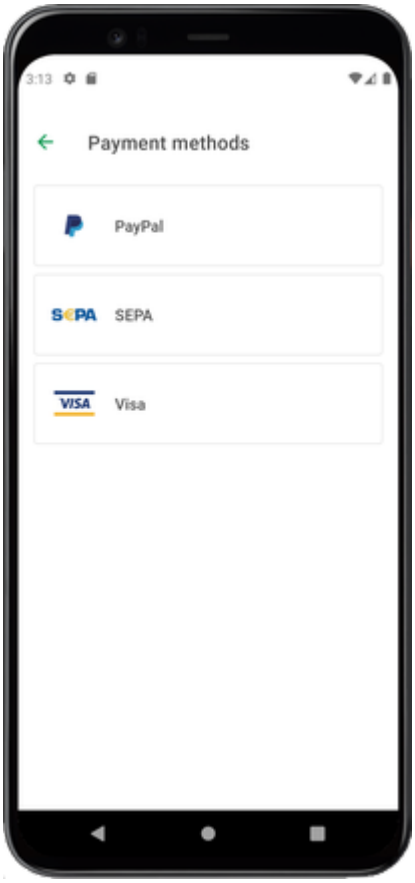
List URL

You can use the following URL to load the payment methods using a GET request in your app.

https://raw.githubusercontent.com/optile/checkout-android/develop/shared-test/lists/listresult.json

Look & Feel

We would like to see at least the logo image and label for each payment network in the list similar as in the image below. Of course the UI does not have to look the same as the example below, you can decide how you want to represent the data.



GitHub & Technologies

The sources of your app should be available to us through a public repository on GitHub.

The app must be written in Java and you can use any third-party libraries you like. You can also use the model classes from our Checkout Android SDK to represent the `ListResult` data (if you want).

https://github.com/optile/checkout-android/tree/develop/checkout/src/main/java/com/payoneer/checkout/model

Focus points

Here are a couple of points you could focus on:

- Network errors should be handled properly, e.g. `404`, `500` response codes and `IOExceptions`. You could show a popup dialog or log the error.
- Cool looking UI and UX but given the limited time we know this is not always possible.
- Unit tests, we can also write a couple during the pair programming session if there is no time left.
- Even for a small app with limited functionality we like it if it feels like a finished product.