Introduction

This is a take-home assignment to be completed before the final Advidi technical interview. During the interview, we will discuss the contents of the assignment, amongst other things. The assignment should take about a day to complete. In the assignment, we aim to test both your frontend and backend knowledge.

If things become unclear during the assignment, please feel free to email to team@advidi.it for support.

Allowed Stack

For the backend, we want you to use *Kotlin* as a language in combination with Spring. You are free to choose any other techniques/libraries, and even cloud services are allowed, as long as you make them accessible to the public.

For the frontend please use *Angular*, specifically a version of 6 or higher.

Problem description

Advidi is an affiliate network, meaning we, in general, serve two parties, affiliates (publishers) and advertisers (sellers). Advertisers sell products, but don't possess the capability to advertise these products to the public. Affiliates are experts in online advertising and are very capable of selling these products. As Advidi we bring these two separate parties together in a single network.

Our advertisers generally create an '**offer'** for every product they sell. An offer contains information about:

- The *name* of the product the advertiser is selling.
- An *URL* to the 'offer page', the page where people purchase items.

Every sale in an offer is called a **'conversion'.** Conversions are simply transactions made on websites, such as the offer page, that are tied to some payment. In general conversions contain:

- A *timestamp* when the conversion happened.
- The corresponding *affiliate id* and *offer id.*
- A *payout*, the price an affiliate receives for a sale.
- A *received* value, the amount of money Advidi receives for a sale.

We would like you to make a simple dashboard for Advidi employees, where they can:

- Login with a username + password.
- View the offers in Advidi with some general information (id, name, URL).
- View the total received and payout values of an offer for all affiliates.

So, in essence, the exercise consists of an Angular SPA (single page application) with two views, the login view, and the offer view. As there are quite a lot of offers in Advidi, it would be needed to also include paging and (search) filters for both time ranges and the offer name.

The time ranges would apply to the included conversions in the offer overview. In this offer overview, you would be able to view a table that looks something similar to:

Offer id	Offer name	Offer URL	Total Paid	Total Received
1	TheName	https://www.offer.com	$1.234,12	$1,553.45

In reality, Advidi has many millions of conversions and thousands of offers. Please set up your application so that it could scale to such a workload.

In Advidi we often make use of multiple databases to increase performance. An example would be that the offers are located in a MySQL database and the conversions in a data warehouse, like Redshift.

You can mimic this common issue by introducing two databases for this assignment, where the offers and conversions are placed in two separate tables in two different databases. You can use any database, as long as they are the same type! For example, feel free to use MySql or MongoDB, but don’t mix both. For this assignment two of the same type of databases are more than sufficient.

You can find the data (in CSV) for the exercise here:

Offers: https://advidi-assignment.s3-eu-west-1.amazonaws.com/offers.csv
Conversions: https://advidi-assignment.s3-eu-west-1.amazonaws.com/conversions.csv

Submission

Although you might be using a new language or framework we do expect you to adhere to general coding best practices. You will be evaluated based on the quality of your code, your design, and the proper resolution of the problem description. The application should be production-ready. Please include a README on how to run your code. It should be **easy** to run your application and the setup on our side should not take longer than a few minutes.

Please assume that we don’t have any of your dependencies/databases already installed on our machines.

When finished, please zip your files and send them (or a link to your files) to team@advidi.it. Please hand in your assignment code **48 hours** before your interview. If you use github, please make the repository private and add “JordenvBreemen” as a viewer.