**CartonCloud - Android Developer Challenge**

**Tools / Frameworks:**

* Android Studio
* Java
* Kotlin
* MetaWeather API
* CookieBar2 library

**Tasks**:

1. Java:

Create a basic Android app project. Show a single view featuring a button triggering an API call to retrieve all of yesterday’s weather information for Brisbane (first retrieving Brisbane’s woeid). Potentially this information may include multiple entries, collected at different points in time during the day. The returned information is to be displayed in a table in a view shown from within the Android app.

2. Database:

Store the obtained weather data in a local database and use it to populate the table on startup until new data is obtained.

3. Kotlin:

To the above add an appropriate message to be displayed on success/error to the user on loading/refreshing the weather table using the CookieBar2 library (in Kotlin), using a custom layout to fit the style of the rest of your app and a button to close the message.

From the main activity, adding a full screen dialog fragment opened when clicking on an entry in the table view. This dialog fragment should show details of the chosen entry obtained from database (including weather state, wind direction, temperature, wind speed, air pressure, humidity, visibility, predictability and applicable date).

4. API:

Could you please explain how you would approach the following situation:

You have an existing (production) mobile application using an API endpoint that provides a variable amount of data, but always provides all the data at once, without paging the results. These results are displayed to the end user in a table.

The backend team want to apply the following changes to the API endpoint:

* Introduce paging to the existing endpoint.
* Split out some of the data into a separate endpoint, queried using an id provided by the original endpoint.

Which potential issues can you see with their solution and how would you suggest tackling these problems (in either the backend, mobile app or both).

**Resources:**

* MetaWeather API:  
  <https://www.metaweather.com/api/>
* CookieBar2 library:  
  <https://github.com/AviranAbady/CookieBar2>