M. Kufleitner Winter term 2023

Practical Course: Logic App Development

Exercise sheet week 8

To be completed before 8th December 2023, 11am.

1. This week, you will continue with adding features to the dice app. The solution to this week's task should be in a folder `week8` in your repository; copy all necessary parts from last week to a new project. If your app does not yet satisfy all the mandatory properties of week 5, 6 and 7, you also need to include the missing functionality. This week's task is to add a timer and to query some JSON API over the internet.

   - Your app should have a timer on the dice tab which starts at 00:00.0 after every throw of the dice and shows the time since the last throw. Moreover, the statistics tab should now also provide timer related information: the minimal and maximal time between any two throws since the last reset and the time between the last two throws. The precision of the timer should be 100 milliseconds; times should be shown as MM:SS.C for minutes MM (with more than two digits if necessary), seconds SS, and C is the number of 100 milliseconds. The two main approaches for implementing a timer would be either `Stream.periodic` or `Timer.periodic`, see e.g.:

     `https://api.flutter.dev/flutter/dart-async/Stream/Stream.periodic.html`
     `https://api.flutter.dev/flutter/dart-async/Timer/Timer.periodic.html`

     Actually, internally `Stream.periodic` uses `Timer.periodic`.

   - You should add a new tab to your app. In this tab, you should query some API over the internet and show the user some information depending on the last throw of the dice. It is up to you which API to choose and which information to show but the information should depend on the last throw and come from the API. The API should answer your request using JSON.

     `https://dart.dev/guides/json`

     For instance, you could send a request to `https://www.themoviedb.org/` and, depending on the sum of the two dice, show the user a movie suggestion.

     `https://www.themoviedb.org/documentation/api`

     You need to upload your private API key to the repository (if the API requires one) so that I can test your solution. Also note that requesting an API key may take some time. A list of some APIs can, for instance, be found at:

     `https://github.com/public-apis/public-apis`

     You should use some "real" API and not a fake REST API like `jsonplaceholder`. Moreover, you should implement the networking functionality yourself, see e.g.,

     `https://docs.flutter.dev/cookbook/networking/fetch-data`

     and **not** use some third-party package for your particular API such as `https://pub.dev/packages/tmdb`.

   Both the timer and the API call rely on asynchronous programming. If you have not done it yet, you should make yourself familiar with the suggested resources on streams and asynchronous dart programming.