

# Task

<https://pastebin.com/DBnaYSDT>

You are given an API interface of some service for streaming data, which is the price of a “Ticker” (eg. BTC price from an exchange). The price of BTC can be different on different exchanges, so our target is to build a “fair” price for BTC, combining the price from different sources. Let’s say there are up to 100 possible exchanges from where the price can be streamed.

You need to develop an algorithm which uses these data streams as input and as output providing an online “index price” in the form of minute bars, where the bar is a pair (timestamp, price). Output can be provided in any form (file, console, etc.). An example output if the service is working for ~2 minutes would look like:

**Timestamp, IndexPrice**

1577836800, 100.1

1577836860, 102

## Requirements:

Data from the streams can come with delays, but strictly in increasing time order for each stream. Stream can return an error, in that case the channel is closed. Bars timestamps should be solid minute as shown in example and provided in on-line manner, price should be the most relevant to the bar timestamp. How to combine different prices into the index is up to you.

Code should be written using Go language, apart from that you are free to choose how and what to do. You can also write some mock streams in order to test your code.

The interface is artificial, so if you need to change something or to have additional assumptions - you are free to do this, but don’t forget to mention that. Your code will be reviewed but won’t be executed on our side.

We expect source code to be published on GitHub and shared with us followed by the readme file with the description of the solution written in English. There might be a technical call after the task completion where we can discuss the solution in detail, ask some questions etc.