Hi! Welcome to the Golang coding challenge 😀.

Crypto Server

Please make use of the free API from https://api.hitbtc.com/ to complete this challenge.

Create a micro-service with the following endpoint,

GET /currency/{symbol}

Returns the real-time crypto prices of the given currency symbol.

Sample Response:

```
{
  "id": "ETH",
  "fullName": "Ethereum",
  "ask": "0.054464",
  "bid": "0.054463",
  "last": "0.054463",
  "open": "0.057133",
  "low": "0.053615",
  "high": "0.057559",
  "feeCurrency": "BTC"
}
```

GET /currency/all

Returns the real-time crypto prices of all the supported currencies.

Response:

```
{
  "currencies": [
    {
      "id": "ETH",
      "fullName": "Ethereum",
      "ask": "0.054464",
      "bid": "0.054463",
      "last": "0.054463",
      "open": "0.057133",
      "low": "0.053615",
      "high": "0.057559",
      "feeCurrency": "BTC"
    },
    {
      "id": "BTC",
      "fullName": "Bitcoin",
      "ask":"7906.72",
      "bid":"7906.28",
      "last":"7906.48",
      "open":"7952.3",
      "low":"7561.51",
      "high":"8107.96",
      "feeCurrency": "USD"
    }
  ]
}
```

Note:

At the moment the supported symbols are only `BTCUSD` and `ETHBTC` but must be configurable. `symbol` must be a valid symbol. Please sync the real-time currency info from Socket Market Data store it in-memory and serve the values from there.

Rules

- Golang must be used to complete the challenge.

- Avoid using libraries and if you use so, please justify their use.
- Dependencies need to be managed if you use any external libraries.
- Solution must be able to be readily run or deployed and of production quality.
- Ideal time to complete this challenge is 4 hours but must attempt to complete within a day.