Technical assignment

Golang software engineer

Description

The goal of this assignment is the implementation of a JSON/HTTP service, in golang, that returns the matching timestamps of a periodic task.

A periodic task is described by the following properties:

- Period (every hour, every day, ...)
- Invocation point (where inside the period should be invoked)
- Timezone (days/months/years are timezone-depended)

The service should return all matching timestamps of a periodic task (ptlist) between 2 points in time (t1, t2). t1, t2 and the entries of ptlist are in UTC with seconds accuracy, in the following form: 20060102T150405Z

The supported periods should be: *1h*, *1d*, *1mo*, *1y*. The invocation timestamp should be at the start of the period (e.g. for *1h* period a matching timestamp is considered the 20210729T010000Z). The service should accept as command-line argument the listen addr/port.

On success, HTTP status 200 OK and a *JSON array* with all matching timestamps, in UTC, for the requested period should be returned. On failure, HTTP status 400 and a *JSON object* with appropriate fields should be returned.

Examples

Here are some examples of successful requests/responses:

```
GET /ptlist?period=1h&tz=Europe/Athens&t1=20210714T204603Z&t2=20210715T123456Z
  [
        "20210714T210000Z",
        "20210714T220000Z",
        "20210714T230000Z",
        "20210715T000000Z",
        "20210715T010000Z",
        "20210715T020000Z",
        "20210715T030000Z",
        "20210715T040000Z",
        "20210715T050000Z",
        "20210715T060000Z",
        "20210715T070000Z",
        "20210715T080000Z",
        "20210715T090000Z",
        "20210715T100000Z",
        "20210715T110000Z",
        "20210715T120000Z"
  ]


GET /ptlist?period=1d&tz=Europe/Athens&t1=20211010T204603Z&t2=20211115T123456Z
  [
        "20211010T210000Z",
        "20211011T210000Z",
```

```
        "20211012T210000Z",
        "20211013T210000Z",
        "20211014T210000Z",
        "20211015T210000Z",
        "20211016T210000Z",
        "20211017T210000Z",
        "20211018T210000Z",
        "20211019T210000Z",
        "20211020T210000Z",
        "20211021T210000Z",
        "20211022T210000Z",
        "20211023T210000Z",
        "20211024T210000Z",
        "20211025T210000Z",
        "20211026T210000Z",
        "20211027T210000Z",
        "20211028T210000Z",
        "20211029T210000Z",
        "20211030T210000Z",
        "20211031T220000Z",
        "20211101T220000Z",
        "20211102T220000Z",
        "20211103T220000Z",
        "20211104T220000Z",
        "20211105T220000Z",
        "20211106T220000Z",
        "20211107T220000Z",
        "20211108T220000Z",
        "20211109T220000Z",
        "20211110T220000Z",
        "20211111T220000Z",
        "20211112T220000Z",
        "20211113T220000Z",
        "20211114T220000Z"
  ]


GET /ptlist?period=1mo&tz=Europe/Athens&t1=20210214T204603Z&t2=20211115T123456Z
  [
        "20210228T220000Z",
        "20210331T210000Z",
        "20210430T210000Z",
        "20210531T210000Z",
        "20210630T210000Z",
        "20210731T210000Z",
        "20210831T210000Z",
        "20210930T210000Z",
        "20211031T220000Z"
  ]


GET /ptlist?period=1y&tz=Europe/Athens&t1=20180214T204603Z&t2=20211115T123456Z
  [
        "20181231T220000Z",
        "20191231T220000Z",
        "20201231T220000Z"
  ]
```

An example of unsuccessful request/response:

```
GET /ptlist?period=1w&tz=Europe/Athens&t1=20180214T204603Z&t2=20211115T123456Z
    {
      "status": "error",
      "desc": "Unsupported period"
    }
```

Deliverable

Mandatory: Code with minimal instructions to build/run should be committed in a public repository (github, bitbucket, ...).

Optional (nice to have):

- Extensible (easy to support new periods)
- Run in docker
- Unit-test