Go test task “notifier” (go01)

refurbed

The Library

Write a library that implements an HTTP notification client. A client is configured with a URL to which notifications are sent. It implements a function that takes messages and notifies about them by sending HTTP `POST` requests to the configured URL with the message content in the request body. This operation should be non-blocking for the caller.

A great number of messages might arrive at once, so make sure to handle spikes in notification activity and don’t overload the event-handling service or exhaust your file descriptors. But be efficient and don’t just send requests serially.

Allow the caller to handle notification failures in case any requests should fail.

The Executable

Write a small program that uses the library above. It should read `stdin` and send new messages every *interval* (should be configurable). Each line should be interpreted as a new message that needs to be notified about.

The program should implement graceful shutdown on *SIGINT*.

Example usage information for clarification purposes (the solution doesn’t have to reproduce this output):

```
usage: notify --url=URL [<flags>]

Flags:
      --help             Show context-sensitive help (also
                         try --help-long and --help-man).
  -i, --interval=5s      Notification interval
```

Example call:

```
$ notify --url http://localhost:8080/notify < messages.txt
```