Engineering Task - Rate Limiter

Implement web service that acts as third party rate limiter service.

Specifications

The service should:

1. Accept following arguments during the startup (command line args):
   - `threshold` - Max number of requests per URL within a time period (`ttl`).
   - `ttl` - The time period in which URL visits will be counted.
2. Expose an endpoint to report URL visits in the following format:

   ```
   POST /report
   Content-Type: application/json

   {
       "url": "http://www.sample.com"
   }
   ```

3. The response to each request should be "block" true/false - depends on if the number of times the URL was reported reached the `threshold`:

   ```
   {
       "block": true
   }
   ```

4. Track the number of times each URL was reported within the `ttl` period.
5. Each URL should be hashed in order to reduce memory usage.

Notes

- Do not use external services (like redis, etc...) - it should be implemented in-memory.
- Implement the above in Golang/Java/Scala/C#.
- Assume this is a production service - make sure the code is well organized and readable and has no major performance issues.
- Make sure the service has no resource leaks.
- Logging will be appreciated.
- Clean code will be appreciated.

Example

TTL: 60,000 (1 minute), threshold: 10

```
00:00:00 URL /abc is reported, count=1, not blocked
00:00:05 URL /foo is reported, count=1, not blocked
00:00:12 URL /abc is reported, count=2, not blocked
00:00:20 URL /abc is reported, count=3, not blocked
00:00:22 URL /abc is reported, count=4, not blocked
00:00:30 URL /abc is reported, count=5, not blocked
00:00:35 URL /abc is reported, count=6, not blocked
00:00:39 URL /abc is reported, count=7, not blocked
00:00:42 URL /abc is reported, count=8, not blocked
00:00:43 URL /abc is reported, count=9, not blocked
00:00:48 URL /abc is reported, count=10, blocked
00:00:51 URL /foo is reported, count=2, not blocked
00:00:55 URL /abc is reported, count=11, blocked
00:01:00 URL /abc is reported, count=1, not blocked
00:01:05 URL /abc is reported, count=2, not blocked
```