Write a reverse proxy that accepts requests and forwards the original request to an origin server. For each request, keep an in-memory cache of the response body with a TTL of 30 seconds. Use the cache data if the request is made multiple times.

For example, if I query https://blockstream.info/api/blocks/0 through the reverse proxy, I should get the following response directly from the origin:

```
[{"id":"000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f
1b60a8ce26f","height":0,"version":1,"timestamp":1231006505,"t
x_count":1,"size":285,"weight":816,"merkle_root":"4a5e1e4baab
89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b","previ
ousblockhash":null,"mediantime":1231006505,"nonce":2083236893
,"bits":486604799,"difficulty":1}]
```

If I query the same origin within 30 seconds, I should get the same response without forwarding the request to the origin.

The solution must be written in Rust. The cache layer must not rely on a framework or dependency.