



2019 Telstra Graduate Program

Case Study - Coding

Purpose

During this case study you will be assessed on your ability to write code to solve a technical task.

Background

A cache is a system used to speed up the retrieval of files from a source. When a user requests a file from a source, they first check the cache to see if it is stored there. If it is, they retrieve the file from cache. If the file is not found in the cache only then do they go directly to the source. Once a file is retrieved from the source, it is stored in the cache for a temporary amount of time to speed up retrieval for the next person that requests that same file. More information about caches in general can be found at [1]

Your Task

Overview

Implement a simple online cache to store documents of the form:

```
{
  "id": int,
  "message": string
}
```

- `id` is the unique identifier for the document.
- `message` is the body of the document and can be any string

An example of a well formed message:

```
{
  "id": 2019,
  "message": "Telstra 2019 Graduate Program"
}
```

Your cache must be able to store and retrieve documents through a RESTful API that it exposes, as outlined below. The internal implementation of the cache, including implementation language, as well as any design choices such as how documents are stored, are at your own discretion.

Storing a document

Documents should be able to be stored in the cache by sending a `POST` request to the `/messages` endpoint. The request body should contain a valid document, such as the one above. Any documents sent to be stored in the cache are assumed to be well formed, with no errors.

Documents stored in the cache should have a Time To Live (TTL) [2] of **30 seconds**. After this time has elapsed, the document should be deleted from the cache and any attempts to retrieve it fail, as explained below.

Retrieving a document

Sending a GET request to `/messages/<id>` where `<id>` is a valid document id should return the requested document to the user. If the user has requested a document that has expired or does not exist, a message `Resource not found` with the appropriate response code should be returned.

Clearing the cache

Implement functionality to clear the cache of all documents.

Extensions

The below tasks are not a requirement of this case study but are simply for you to consider if you wish to extend the functionality of your cache.

- Make the TTL adjustable
- Reject any malformed documents if users attempt to save them to the cache
- Distribute your cache over multiple servers, implementing failover capability such that if one instance goes down, your cache is still fully functional.

Submission

Please print off a copy of your code and bring it with you to the interview. If you do not have access to a printer, please email your code to Telstragradprogram@team.telstra.com at least 4 hours prior to your interview so this can be printed or shared with your interviewing managers prior.

You should also host your code on an online repository such as GitHub [3] or Bitbucket [4], a link to which can then be shared with your interviewing manager. It is not a requirement that your solution be running live anywhere but if you would like to demonstrate your solution working, you may wish to deploy your application to somewhere like Heroku [5].

Presenting your case study

During the interview you will be asked to discuss your code with the interviewing managers. You will have a maximum of 5 minutes to present.

References

- [1] [https://en.wikipedia.org/wiki/Cache_\(computing\)](https://en.wikipedia.org/wiki/Cache_(computing))
- [2] <https://www.incapsula.com/cdn-guide/glossary/time-to-live-ttl.html>
- [3] <https://github.com/>
- [4] <https://bitbucket.org/>
- [5] <https://www.heroku.com/>