CODING CHALLENGE

21st

OVERVIEW

Create a service which fetches data from an endpoint (provided by our challenge docker container) at x second intervals and caches the newest result in memory.

RUN THE CHALLENGE DOCKER CONTAINER

```
docker run -p 8080:80 21re/coding-challenge
```

OUR ENDPOINT

`GET /` on the docker container returns a list of items separated by a `\n` character.

```
» curl localhost:8080 | head
   N
   N
   N
   N
   A
   A
   A
   A
   A
   ...
```

SERVICE ENDPOINT

Your App should expose an HTTP endpoint that allows clients to access the cached data at a given index

- Use HTTP framework of your choice
- GET `/$index` return an item at a given index (start at 0 or 1)
- Please provide instructions in a README file explaining how to run your server locally

```
» curl localhost:1337/1
   N
» curl localhost:1337/7
   A
```

COMPRESSION

Items returned by endpoint A will contain repeated characters at high frequencies. Your cache should use *Run-length encoding (RLE)** compression for internal storage in memory.

You can use this trait as starting point, but you don't need to. It's just an idea.

```
trait Compressor {
 def compress[A]: ??? => Seq[Repeat[A]]
}

case class Repeat[A](count: Int, element: A)
```

CONSTRAINT

Please use a statically typed language adequate for the position you are applying to.

* https://en.wikipedia.org/wiki/Run-length_encoding