BBC Sounds

RMS Junior Engineer Challenge

The BBC Sounds Radio and Music Services (RMS) team have a microservice architecture (HTTP/RPC web services) that produces and consumes formatted metadata for millions of listeners every day. This coding challenge asks you to think about how you parse, store, and update a small set of music-related data.

For the coding challenge, we have provided a CSV file containing information relating to 10 albums. Your solution should read the CSV file into memory and implement the following methods/functions:

- Add(<please define parameter(s) yourself>)
- Get(id)
- Get(artist)
- Update(<please define the parameter(s) yourself>)
- Delete(id)

You should provide unit tests as part of the solution. Unit tests should cover:

- Adding a new album (choose the album yourself. You can generate a version 4 random UUID here: https://www.uuidtools.com/v4)
- Get the album with id f337fd51-7bf5-44bf-9553-5826162bc83a
- Get all albums by the Gorillaz
- Update year_released for Melodrama by Lorde to be 2017
- Delete an album (choose one yourself)

There should be no need for external libraries or dependency management. Use the in-memory data for your unit tests; avoid overwriting or manipulating the CSV file. The unit tests mentioned above are essential but not exhaustive; please add any other unit tests you identify as pertinent.

The challenge should be completed in Scala, Java, Kotlin, Python, Rust or Go.

Please consider your approach. We will be looking at data structure choice, code clarity, documentation, logging and error handling when reviewing the code.

When submitting, we are happy to accept GitHub links or zip files. Please include instructions on how to run unit tests in a README.

Most importantly, have fun!