Welcome to the software engineering technical test!

We hope that you find this exercise fun and interesting. There are no trick questions; we want to see your solution to a simple problem with well thought out and structured code. We realise that there are a lot of topics in the brief and that you may not have the experience or time to complete them all – that is fine, we do not expect production-ready code to be an output of this exercise so focus on what is most important first and document any tradeoffs you made in your README. If you have any questions about any part of this exercise, please do not hesitate to reach out to us.

There is no time limit on how much time to commit to the test; however, we suggest you spend no more than 1 day.

The Brief

Our product team would like you to develop a new call detail record (CDR) business intelligence platform. The idea behind it is that the call records can be queried to find information such as average call cost, longest calls, average number of calls in a given time period.

For this technical test we would like you to create a RESTful CDR API. This means that a front-end user interface is not required, neither is any consideration of any services which you might expect to be shared such as authentication.

A CDR will be provided as a comma separated value file; these files are delivered as often as daily and can be very large (gigabytes), so being able to simply upload new files is a must.

In addition to uploading new CDR files, we would like the API to expose the following functionality:

- Retrieve individual CDR by the CDR Reference
- Retrieve a count and total duration of all calls in a specified time period. Given the large amount of CDRs likely to be present, we would like to limit the maximum time period to 1 month and implement optional filtering on the call Type.
- Retrieve all CDRs for a specific Caller ID in a specified time period. Given the large amount of CDRs likely to be present, we would like to limit the maximum time period to 1 month and implement optional filtering on the call Type.
- Retrieve N most expensive calls, in GBP, for a specific Caller ID in a specified time period of maximum of 1 month, with optional filtering on the call Type.

The expectation from the product team is that we produce an MVP of the fully working system as soon as possible, then iteratively continue to add features. We hope that you will think about this expectation as you work.

You should design your solution so that it is maintainable, extendable and testable. We expect you to include an automated test suite in the solution.

We would also like you to think about how you will document your API so that it is easy for other teams to consume.

The Data

Column	Description	Notes
caller_id	Phone number of the caller	
recipient	Phone number of the recipient	
call_date	Date on which the call was made	
end_time	Time when the call ended	
duration	Duration of the call	Seconds
cost	The billable cost of the call	To 3 decimal places (decipence)
reference	Unique reference for the call	
currency	Currency for the cost	ISO alpha-3
type	Call Type	1 = Domestic, 2 = International

The Deliverable

Ensure you include the following in your README.md:

1. A covering note explaining the technology choices you have made
2. Any assumptions you've made and why
3. Instructions on how to run the application and the test suite locally
4. Are there any other considerations/future enhancements you would make given more time

We expect you to treat this project like you're working as part of a team, and your git etiquette should reflect this. Email as an attachment or include a link to the git bundled repository showing your commit history with all your commits on the master branch:

```
git bundle create tech-test.bundle --all —branches
```