

# Technical Test

## *Software Engineer, Backend*

### Contents of this Test

- Technical test instructions (this document).
- A number of sample “uploaded” CSV files for testing.

This test may be downloaded from

<https://rei-dev.ictnw.net/techtests/be/infinity-be-techtest.tgz>

### Task

This test may be completed in either **PHP or Go**, whichever you prefer, but please use only the **standard library** and no other frameworks or external packages (you are allowed to use the external MySQL driver package in Go).

You are tasked with writing a program to process files that have been uploaded by clients.

The program should run **once per minute using a system crontab**. When run, it should discover any CSV files in the “uploaded” directory, parse the rows in the file, insert their contents into a MySQL database, and then move each CSV file to the “processed” directory.

Clients are expected to upload files in the following format, and each file should be **validated** to ensure that format has been used:

- Comma separated format files with Unix line endings
- Containing the following columns **in any order**:
  - “eventDatetime” (timestamp, formatted as “yyyy-mm-dd hh:mm:ss”, required)
  - “eventAction” (string, 1-20 chars, required)
  - “callRef” (integer, required)
  - “eventValue” (decimal, optional)
  - “eventCurrencyCode” (3-letter ISO 4217 currency code, required only if “eventValue” is non-zero)

The program should ensure that a **table is created** to receive the data, if not already present in the schema. You will need to choose suitable data types to hold the data.

Also provided with this document is a directory containing sample “uploaded” files that you should use to test your program.

Although the sample files are small, some clients are expected to upload batches of files that take longer than a minute to process at peak times. You should ensure that your program **does not run** if another instance of the program is already running.

As this program is designed to run in the background via cron, any progress information should be logged to **syslog**, rather than output to the console.

**The leading call intelligence platform.**

## Submission

Create a public repository in GitHub, with a name of your choice. Create a branch in that repository and push your submission to it. Finally, create a pull request from that branch.

Your pull request should explain the approach you have taken, and any specific steps taken to handle errors.

Send a link to your pull request by email to [be.techtest@infinity.co](mailto:be.techtest@infinity.co) for review. Be sure to include your full name in this email so we can link it to your application.