

# Transactional Reporting System

<https://github.com/marcov4lente/TransactionalReportingSystem>

Built with the Symfony Console package, using Composer as the package management solution.

Using a mock Currency Service, The Transactional Reporting System supports 3 currencies, namely GBP, EUR and USD. It is therefore able to generate and display reports in any of these three currencies.

Using Monolog, the system logs all INFO and ERROR events to the logs directory.

The Logging Management, Configuration Management and Validation Management exist as traits, as it is intended for them to be shared with future command classes, that may perform other functions such as emailing reports or data analysis.

The system includes a custom build ASCII table builder as well, to properly format the output report as text within the terminal.

## Requirements

- PHP 5.5 or above.
- PHP SQLITE PDO Driver
- Apache 2 / Ngin
- Please insure that the logs folder is writeable.

## Installation

- Clone the repository to the desired destination folder on the destination system using the git clone command.
- Change directory to the application's root folder.
- Run a composer update, so that all required vendor packages are downloaded and installed.

## Running reports

### command

Generate a detailed transaction report for a specific user, for a specified time interval and type.

```
php application.php reports:generate [-m|--merchant="..."] [-d|--date="..."] [-i|--interval="..."] [-c|--currency="..."]
```

### Arguments

--merchant: Specify the user ID to generate the analysis report against.

--date: Specify the day, month or year for which the system is to generate the report for, accepted format: daily = yyyy-mm-dd, monthly = yyyy-mm, yearly = yyyy.

--interval: Report interval type: d = daily, m = monthly, y = yearly.

--currency: Specify the currency code for which the system is to output the generated report in, format: GBP.

## Sample report 1

In Great Britain Pounds (GBP), for Merchant of ID 1, for the 2nd of May 2010.

```
$ php application.php reports:generate --merchant=1 --date=2010-05-02 --interval=d --currency=GBP
```

Sample output

INFO: Starting the report generator...

INFO: Compiling data, and converting all amounts to GBP...

```
-----  
| merchant | date      | amount |  
-----  
| 1        | 2010-05-02 | 0.99   |  
| 1        | 2010-05-02 | 0.06   |  
-----
```

INFO: Report generated succesfully, 2 records found. have a lovely day!

## Sample report 2

In Euros (EUR), for Merchant of ID 2, for the period of May 2010.

```
$ php application.php reports:generate --merchant=2 --date=2010-05 --interval=m --currency=EUR
```

Sample output

INFO: Starting the report generator...

INFO: Compiling data, and converting all amounts to EUR...

```
-----  
| merchant | date      | amount |  
-----  
| 2        | 2010-05   | 25.11   |  
-----
```

INFO: Report generated succesfully, 1 records found. have a lovely day!

## Sample report 3

In United States Dollars (USD), for Merchant of ID 1, for the year of 2010.

```
$ php application.php reports:generate --merchant=1 --date=2010 --interval=y --currency=USD
```

Sample output

INFO: Starting the report generator...

INFO: Compiling data, and converting all amounts to USD...

```
-----  
| merchant | date      | amount |  
-----  
| 1        | 2010      | 7.86   |  
-----
```

INFO: Report generated succesfully, 1 records found. have a lovely day!



## Unit testing

Unit testing can be carried out using PHPUnit.

### Test the generation of a yearly report

\$ phpunit YearlyReportTest.php

### Test the generation of a monthly report

\$ phpunit MonthlyReportTest.php

### Test the generation of a daily report

\$ phpunit DailyReportTest.php

## Possible future additions

The following features were considered but due to the time constraints they were not implemented:

- Email reports through to a specified email address using the Mandril.com API
- Data intelligence, calculating trends such as transaction increases or decreases for a specific merchant over a certain period of time.