Considerations

- The solution must be based on microservice architecture
- The solution must expose the capability to transfer funds through a Rest API
- The transfer API must not be public , include a security layer to authenticate the users who call the API
- You must validate the input request
- You must take care handling the posible exceptions thrown by the application and the format in case there is one exception
- For now the API only support USD transfers but in the future could be added more currencies
- Is required use a database and persist the information

Rules

- Given an user request a transfer when the origin account does not have enough funds then notify the error *insufficient* funds
- Given an user request a transfer when the user reach the max limit of transfer per day then notify the error the user exceed the limit number of transfers allowed by day (max limit 3 per day)
- Given an user request a transfer when there is no errores and the amount is bigger than 100 then calculate a tax of 5% of the amount, this tax will be charged to the origin account
- Given an user request a transfer when there is no errores and the amount is less than 100 then calculate a tax of 2% of the amount, this tax will be charged to the origin account
- Given an user request a transfer when there is no errores then the original account must decrease its balance (according transfer amount + tax value) and the recipient account must increase according the transfer amount
- Given an user request a transfer when there is no errores then call the API GET https://data.fixer.io/api/latest (See https://fixer.io/) to transform the amount of the transfer to CAD currency and return it in the response (is not necessary you get or by any api key but is required you prepare your code to include it)

Success Request Example