# Blockchain Smart Contract Exercise Boch Lukas & Cetinkaya Yusuf

## Currency Exchange

### Exchange Contract:

The contract is deployed on the Sepolia test network and is accessible under the following address: 0xFf370967E6E5B25b99ADd4258c2b2e93A55eEb09Ein Bild, das Text, Screenshot, drinnen enthält.

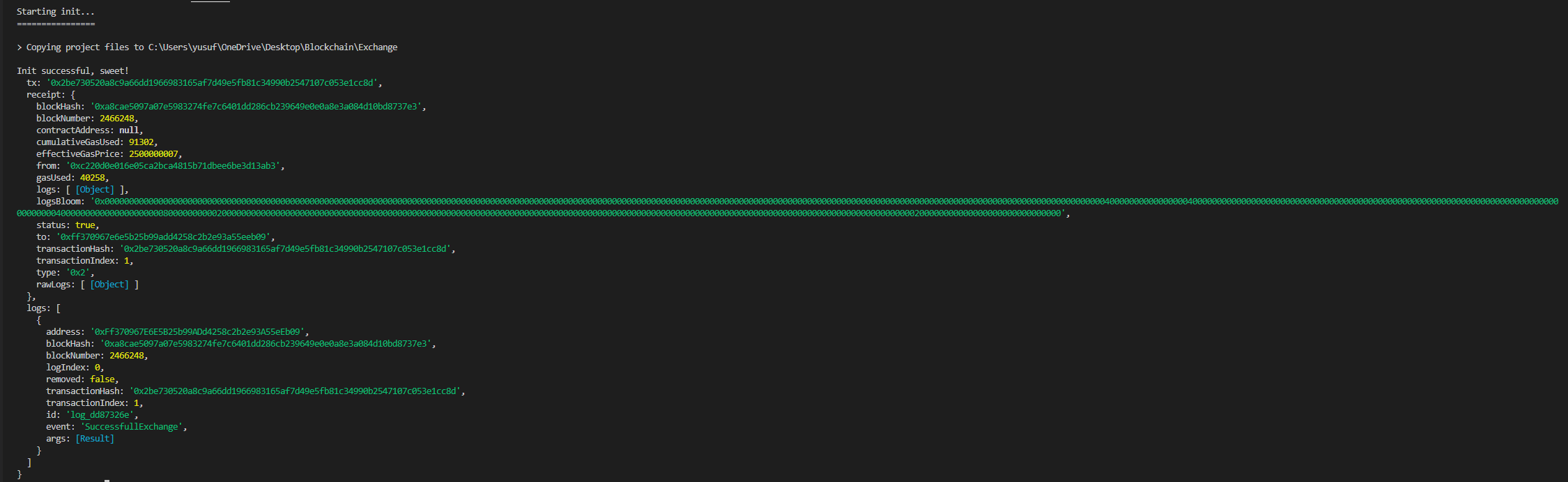
Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, drinnen enthält.

Automatisch generierte Beschreibung

### Access Contract Exchange

* Create a new project, Create a folder Exchange
* RUN: truffle init
* Copy build folder and the truffle config file
* Replace the dummy in “sepolia section” with your mnemonic and the infura key
* Run: truffle console --network sepolia
* Run: let instance = await Exchange.deployed()



### Explanation Contract Exchange

This small implementation of a smart contract is about being able to exchange the currencies Euro, Dollar and Pound. For this, 2% is deducted per transaction as a processing fee and the exchange rate is that of 12.12.2022. First, a struct "Wallet" was created. This contains the data: address owner, uint centEU, uint centUS, uint pennyUK. Then a wallet testWallet is created with the addresse: 0xc0ffee254729296a45a3885639AC7E10F9d54979 and 10000 of each currency. Since solidity does not support decimal numbers, the solution was to calculate with cent/penny and powers of ten in order to be able to perform precise percentage calculations. The address has in this simple example no purpose, but could be used by extension to distinguish between several wallets.

#### walletInfo()

The first function walletInfo only returns the address of the wallet and the respective amount of currencies. In the following pictures you can also see that the transactions have changed the value of the amounts of the currencies of the wallet.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

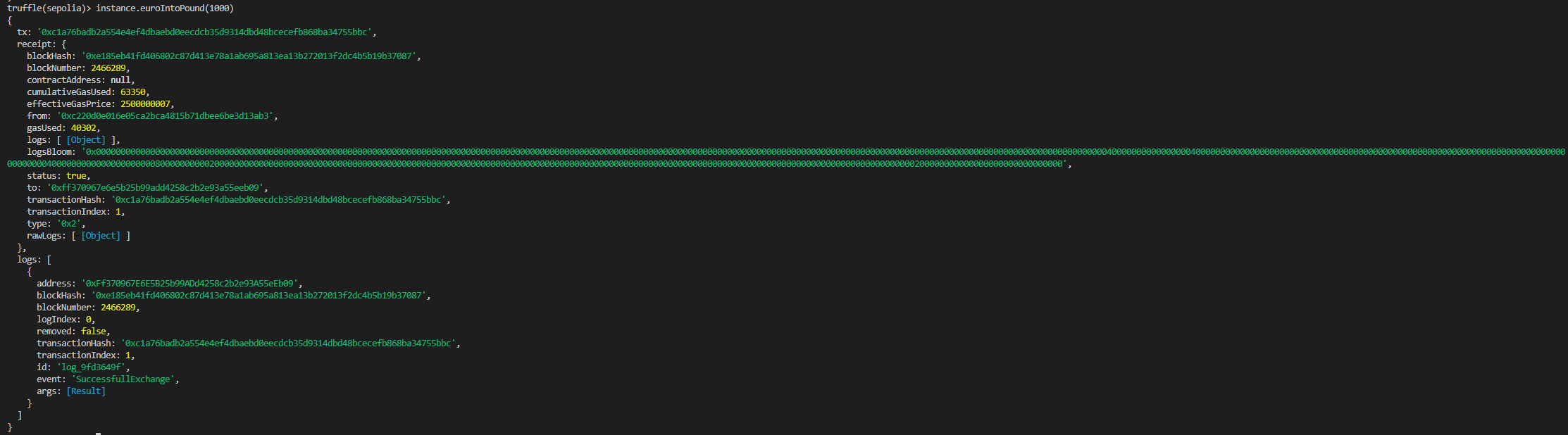
Automatisch generierte Beschreibung

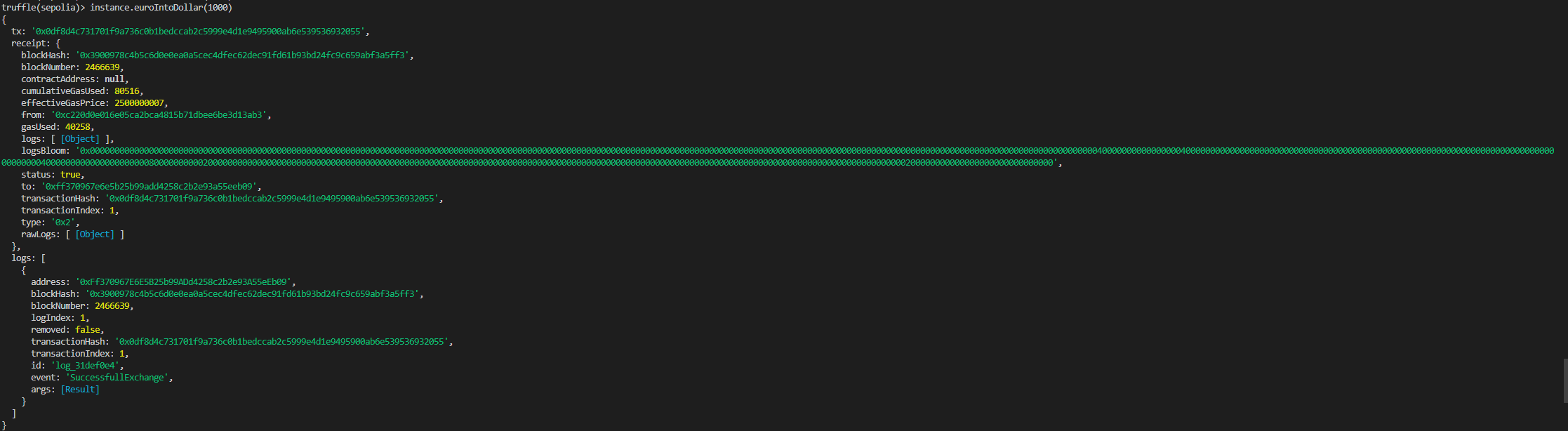
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

#### euroIntoDollar(uint \_centEU) & euroIntoPound(uint \_centEU)

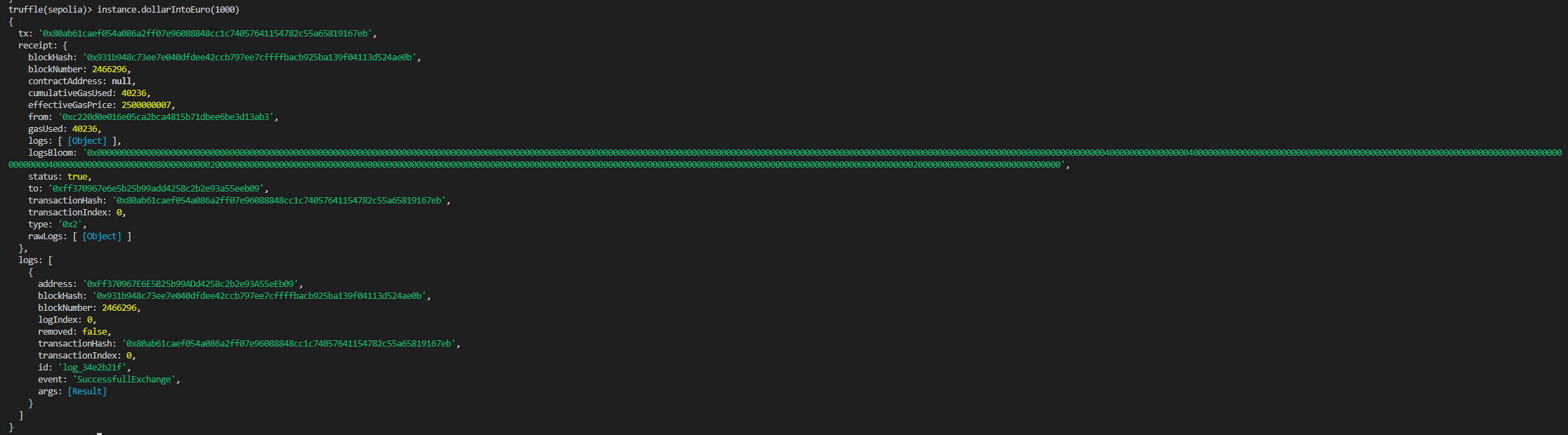
With these functions, you can have euros exchanged for either dollars or pounds. The input parameter is the amount to be exchanged as uint in cent (EU). You don't get a return value except a success message and some log data or an error message if there are not enough of the currency on the wallet. With the function walletInfo() you can see the change afterwards. In addition, an event is fired after each executed transaction. In this implementation, this is still irrelevant, but could be of great use in an extension.

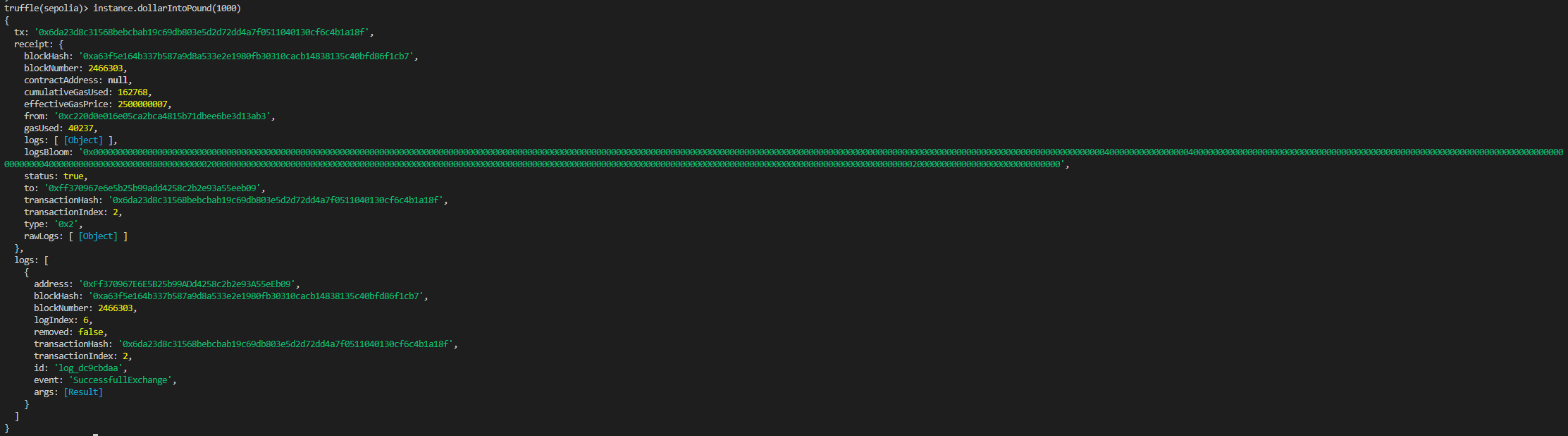




#### dollarIntoEuro(uint \_centUS) & dollarIntoPound(uint \_centUS)

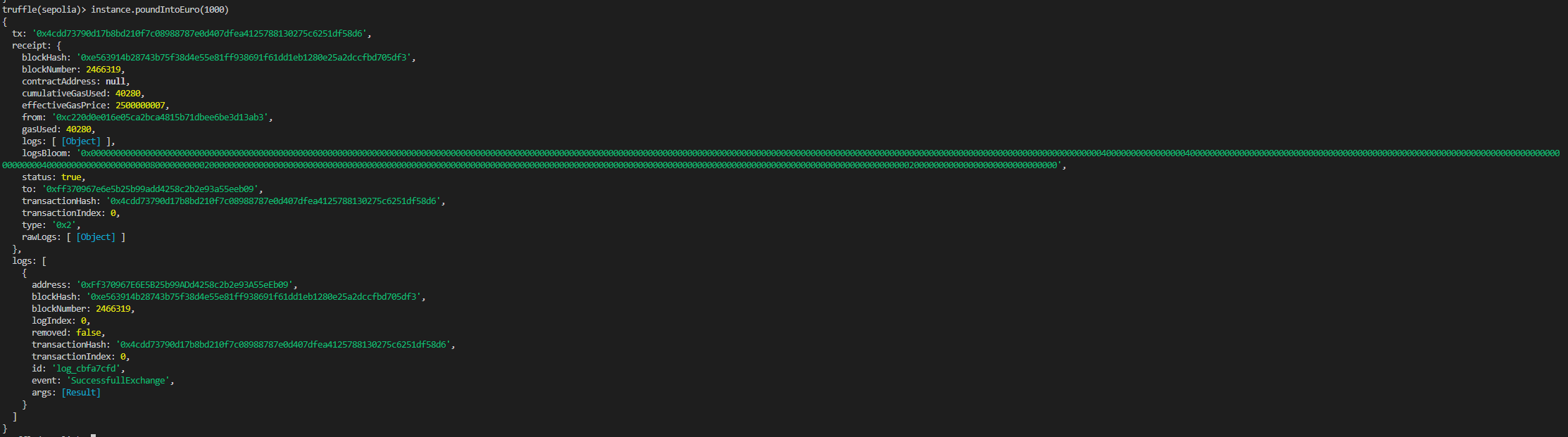
With these functions, you can have dollars exchanged for either euros or pounds. The input parameter is the amount to be exchanged as uint in cent (US). You don't get a return value except a success message and some log data or an error message if there are not enough of the currency on the wallet. With the function walletInfo() you can see the change afterwards. In addition, an event is fired after each executed transaction. In this implementation, this is still irrelevant, but could be of great use in an extension.

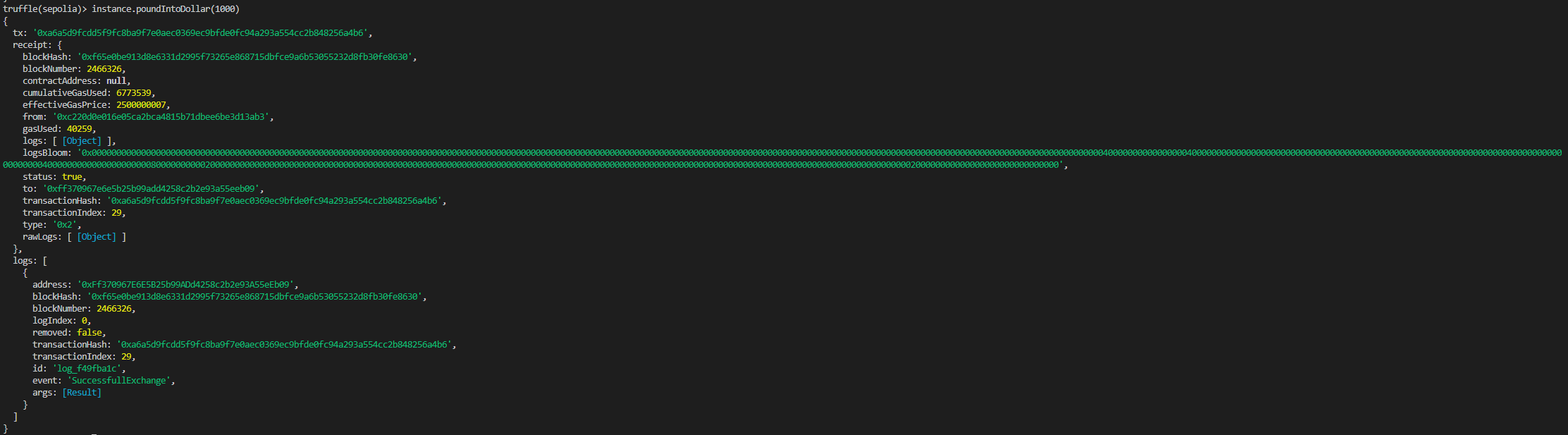




#### poundIntoDollar(uint \_pennyUK) & poundIntoEuro(uint \_pennyUK)

With these functions, you can have pounds exchanged for either dollars or euros. The input parameter is the amount to be exchanged as uint in penny (UK). You don't get a return value except a success message and some log data or an error message if there are not enough of the currency on the wallet. With the function walletInfo() you can see the change afterwards. In addition, an event is fired after each executed transaction. In this implementation, this is still irrelevant, but could be of great use in an extension.





#### Transactions After UsageEin Bild, das Text, drinnen, Screenshot, Computer enthält. Automatisch generierte Beschreibung

### truffle-config.js

For security reasons, we deleted our 12-word phrase and sepolia address from the file after it was done and replaced it with the placeholder "dummy", since the project will be on Github and you can never be too careful with something like that; and for good practice reasons ;)

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung