# Gasless USDC to EURC Swap

This implementation demonstrates how to get quotes and execute swaps between USDC to EURC using the Soroswap Aggregator API.

As we suppose that wallets that will use the Gasless Swaps are wallets that already suport USDC, our first script will set up a wallet with 1 USDC with an sponsored trustline. While, the next ones will execute gasless swaps using the API, for vaious cases

## Setup

```
yarn install
cp .env.example .env
```

Setup your API KEY and SECRET KEYS

## Usage

### 1. Setup the user and referral account.

- Sponsor the creation of a referral acount with USDC and EURC trustline
- Sponsor the creation of a user account with USDC trustline and 1 USDC

```
yarn ts-node src/createSponsoredAccounts.ts
```

### 2. Swaps 0.5 USDC into EURC using Soroswap API and gaslessTrustline=true

PROF

```
yarn ts-node src/gaslessTrustlineExactInUSDC.ts # swap 0.5 USDC for EURC
with gaslessTrustline
```

```
const quote = await soroswapSdk.quote({... gaslessTrustlie: create,
...})
const buildResponse = await soroswapSdk.build({quote, sponsor, from, to,
referralId})
const swapTransaction = TransactionBuilder.fromXDR(buildResponse.xdr,
Networks.PUBLIC)

swapTransaction.sign(sponsorKeypair)
swapTransaction.sign(userKeypair)

const sendResponse = await soroswapSdk.send(swapTransaction.toXDR())
```

3. Swaps ExactIn 0.1 EURC to USDC using Soroswap API, forcing SDEX and sponsoring the gas.

```
yarn ts-node src/gaslessSwapSDEXExactInEURC.ts
```

```
const quote = await soroswapSdk.quote({... protocols:[sdex]})
const buildResponse = await soroswapSdk.build({quote, sponsor, from, to,
referralId})
const swapTransaction = TransactionBuilder.fromXDR(buildResponse.xdr,
Networks.PUBLIC)

swapTransaction.sign(sponsorKeypair)
swapTransaction.sign(userKeypair)

const sendResponse = await soroswapSdk.send(swapTransaction.toXDR())
```

4. Swaps EURC to ExactOut 0.1 USDC using Soroswap API, forcing SDEX and sponsoring the gas

```
yarn ts-node src/gaslessSwapSDEXExactOutUSDC.ts
```

```
const quote = await soroswapSdk.quote({... protocols:[sdex]})
const buildResponse = await soroswapSdk.build({quote, sponsor, from, to,
referralId})
const swapTransaction = TransactionBuilder.fromXDR(buildResponse.xdr,
Networks.PUBLIC)

swapTransaction.sign(sponsorKeypair)
swapTransaction.sign(userKeypair)

const sendResponse = await soroswapSdk.send(swapTransaction.toXDR())
```

PROF

5. Swap ExactIn 0.1 EURC to USDC using Soroswap API, and using only Soroban Protocols (soroswap, aqua, phoenix)

```
yarn ts-node src/gaslessSwapSorobanExactInEURC.ts
```

```
```

```
const quote = await soroswapSdk.quote({... protocols:[soroswap, phoenix,
aqua]})
const buildResponse = await soroswapSdk.build({quote, sponsor, from, to,
referralId})
const swapTransaction = TransactionBuilder.fromXDR(buildResponse.xdr,
Networks.PUBLIC)

swapTransaction.sign(userKeypair)
const feeBumpTransaction: FeeBumpTransaction =
      TransactionBuilder.buildFeeBumpTransaction(
        sponsorAddress, // The new tx will be paid by the sponsor
        swapTransaction.fee + 1, // New fee needs to be higher
        swapTransaction as Transaction, // Tx signed by the user
        Networks.PUBLIC
      );
feeBumpTransaction.sign(sponsorKeypair)

const sendResponse = await soroswapSdk.send(feeBumpTransaction.toXDR())
```

6. Swap USDC to ExactOut EURC using Soroswap API, and using only Soroban Protocols (soroswap, aqua, phoenix)

```
yarn ts-node src/gaslessSwapSorobanExactOutEURC.ts
```

## Now, one way to do it for all protocols (SDEX and Soroban)

```
yarn ts-node src/gaslessSwapSorobanExactOutEURC.ts
```

PROF

The best way to support all transactions:
If the tx is SDEX, the sponsoring is as usual. If the tx is soroban, it wraps it and create a feebump tx ("soroban sponsorship")

```
const quote = await soroswapSdk.quote({... protocols:[soroswap, phoenix,
aqua]})
const buildResponse = await soroswapSdk.build({quote, sponsor, from, to,
referralId})
const swapTransaction = TransactionBuilder.fromXDR(buildResponse.xdr,
Networks.PUBLIC)

let signedTransaction: Transaction | FeeBumpTransaction
if (quote.platform === SupportedPlatforms.SDEX) {
    swapTransaction.sign(userKeypair)
    swapTransaction.sign(sponsorKeypair)
    signedTransaction = swapTransaction as Transaction
```

```
} else {
    swapTransaction.sign(userKeypair)
    const feeBumpTransaction: FeeBumpTransaction =
    TransactionBuilder.buildFeeBumpTransaction(
        sponsorAddress, // The new tx will be paid by the sponsor
        swapTransaction.fee + 1, // New fee needs to be higher
        swapTransaction as Transaction, // Tx signed by the user
        Networks.PUBLIC
    );
    feeBumpTransaction.sign(sponsorKeypair)
    signedTransaction = feeBumpTransaction as FeeBumpTransaction
}
const sendResponse = await soroswapSdk.send(signedTransaction.toXDR())
```