Creating and canceling Solana Fusion order

Creating and submitting an order

The example below demonstrates how to create and submit a Solana order for swapping 1 SOL to 1 USDC.

```
import { Address, FusionSwapContract, Sdk } from "@1inch/solana-fusion-sdk";
import { AxiosHttpProvider } from "@1inch/solana-fusion-sdk/axios";
import { Connection, Keypair, PublicKey, Transaction } from "@solana/web3.js"; // v1

const authKey = "...";
const secretKey = Buffer.from("secret", "base64"); // replace with real secret key
const rpc = "https://api.mainnet-beta.solana.com";

const connection = new Connection(rpc);
const maker = Keypair.fromSecretKey(secretKey);

const sdk = new Sdk(new AxiosHttpProvider(), {
  baseUrl: "https://api.1inch.dev/fusion",
  authKey,
  version: "v1.0",
});

// create order
// 1 SOL -> USDC
const order = await sdk.createOrder(
  Address.NATIVE, // SOL
  new Address("EPjFWdd5AufqSSqeM2qN1xzybapC8G4wEGGkZwyTDt1v"), // USDC
  1_000_000_000n, // 1 SOL
  Address.fromPublicKey(maker.publicKey),
);

const contract = FusionSwapContract.default();

// create escrow instruction
const ix = contract.create(order, {
  maker: Address.fromPublicKey(maker.publicKey),
  srcTokenProgram: Address.TOKEN_PROGRAM_ID,
});

// generate tx
const createOrderTx = new Transaction().add({
```

```
    ...ix,
    programId: new PublicKey(ix.programId.toBuffer()),
    keys: ix.accounts.map((a) => ({
      ...a,
      pubkey: new PublicKey(a.pubkey.toBuffer()),
    })),
  });

  createOrderTx.recentBlockhash = (
    await connection.getRecentBlockhash()
  ).blockhash;
  createOrderTx.sign(maker);

  // broadcast tx
  await connection.sendRawTransaction(createOrderTx.serialize());

  while (true) {
    const status = await sdk.getOrderStatus(order.getOrderHashBase58());

    if (!status.isActive()) {
      console.log("order finished", status);
      break;
    }

    await sleep(1000);
  }

  export function sleep(ms: number): Promise<void> {
    return new Promise((resolve) => setTimeout(resolve, ms));
  }
```

Canceling an order

The example below demonstrates how to cancel a Solana order.

```
  import { Address, FusionSwapContract, Sdk } from '@1inch/solana-fusion-sdk'
  import { AxiosHttpProvider } from '@1inch/solana-fusion-sdk/axios'
  import { Connection, Keypair, PublicKey, Transaction } from '@solana/web3.js' // v1

  const authKey = '...'
  const secretKey = Buffer.from('secret', 'base64') // replace with real secret key
  const rpc = 'https://api.mainnet-beta.solana.com'

  const connection = new Connection(rpc)
  const maker = Keypair.fromSecretKey(secretKey)

  const sdk = new Sdk(new AxiosHttpProvider(), { baseUrl: 'https://api.1inch.dev/fusion', authKey, version: 'v1.0'
  })

  const order = ... // see 'Creating and submitting an order' section
  const contract = FusionSwapContract.default()

  // close escrow instruction
  const ix = contract.cancelOwnOrder(order, {
```

```
  maker: Address.fromPublicKey(maker.publicKey),
  srcTokenProgram: Address.TOKEN_PROGRAM_ID
})

// generate tx
const cancelOrderTx = new Transaction().add({
  ...ix,
  programId: new PublicKey(ix.programId.toBuffer()),
  keys: ix.accounts.map((a) => ({
    ...a,
    pubkey: new PublicKey(a.pubkey.toBuffer())
  }))
})

cancelOrderTx.recentBlockhash = (await connection.getRecentBlockhash()).blockhash
cancelOrderTx.sign(maker)

// broadcast tx
await connection.sendRawTransaction(createOrderTx.serialize())
```

Previous
‹ SDK overview

Next
Filling and canceling Solana Fusion order ›