Solana

1. Let the user decide what name they want
2.

You'll want to provide some sort of input box that allows a user to see if the name they want is available.

1. Check if the name is valid
2.

There are some restrictions for registering a name. In short, the name must fit in 32 bytes, must not have a space, or a /.
```

Copy import{ isNameValid }from"@clustersxyz/sdk"

constvalidName="clusters"; console.log(isNameValid(validName));// true

constinvalidName="c/c/c/c"; console.log(isNameValid(invalidName));// false
```

1. Once you know the name is valid, make sure the name is not taken
2.

```

Copy import{ isNameValid }from"@clustersxyz/sdk"

constname="clusters"; constnameAvailability=awaitgetNameAvailability(name); console.log(nameAvailability.isAvailable)// false
```

1. Get the raw transaction data for the on-chain transaction
2.

This call will get you the on-chain transaction data to register the names you want at the minimum registration fee of 0.01 ETH (converted to lamports).

If you want to impose a specific registration amount, you can do so in the names array. Input these amounts in Wei and they'll be converted to SOL. Check out the documentation on[getting the transaction data](#) to learn more. ```

Copy constnames=[{ name:'mynewname'}]; constsignerAddress='A59..iHqw'; constchainId='solana'; constdata=awaitgetRegistrationTransaction(names,signerAddress,chainId);
```

1. You can now display the registration fee
2.

```

Copy import{ LAMPORTS_PER_SOL }from'@solana/web3.js';

constregistrationFee=Number(data.registrationFee)/LAMPORTS_PER_SOL;

// Total Registration fee: 0.2248 SOL console.log(Total Registration fee:{registrationFee}{data.gasToken.symbol});
```

1. Submit the transaction
2.

```

Copy // Example of a solana provider // https://docs.phantom.app/solana/detecting-the-provider constsolanaProvider=window.phantom?.solana;

for(lettransactionofdata.transactionData) { const{signature}=awaitsolanaProvider.signAndSendTransaction(transaction); }
```

1. Track the status of the transaction
2.

Once the transaction is submitted, you can check when the registration has been processed. An alternative method is pinging thegetNameAvailability() function to see when the name becomes taken.

```
Copy constgetStatus=awaitgetTransactionStatus(signature); console.log(getStatus.status)// finalized
```