

# Ethereum Networks

We currently support the following networks for name registration

ID Name 1 Mainnet 10 Optimism 56 BNB Smart Chain 137 Polygon 8453 Base 42161 Arbitrum One 43114 Avalanche 11155111 Sepolia 17000 Holesky (testnet)

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. If you are on a chain, such as Polygon, that has a different gas token than ETH, you'll receive back the converted value in MATIC.

If you want to impose a specific registration amount, you can do so in the names array. Check out the documentation on [getting the transaction data](#) to learn more. ```

```
Copy constnames=[{ name:'mynewname'}]; constsignerAddress='0x..123'; constchainId='1'; constdata=awaitgetRegistrationTransaction(names,signerAddress,chainId);
```

```

1. You can now display the registration fee, bridge fee (if any), and what gas token the user will be paying with
- 2.

```

```
Copy import{ formatUnits }from'viem';
```

```
constregistrationFee=formatUnits(BigInt(data.registrationFee),data.gasToken.decimals); constbridgeFee=formatUnits(BigInt(data.bridgeFee),data.gasToken.decimals);
```

```
// Total Registration fee: 0.01 ETH console.log(Total Registration fee:{registrationFee}{data.gasToken.symbol});
```

```
// Bridge fee: 0 ETH console.log(Bridge fee:{bridgeFee}{data.gasToken.symbol});
```

```

1. Submit the transaction
- 2.

...

Copy // <https://viem.sh/docs/actions/wallet/sendTransaction>

```
const hash = await walletClient.sendTransaction({ data: data.transactionData.data, account: signerAddress, to: data.transactionData.to, value: BigInt(data.transactionData.value) })
```

...

1. Track the status of the transaction
- 2.

Once the transaction is submitted, you can check when the registration has been processed. This is especially useful when bridging. An alternative method is pinging the `getNameAvailability()` function to see when the name becomes taken.

...

Copy `const getStatus = await getTransactionStatus(hash); console.log(getStatus.status) // finalized`

...

[Previous](#) [Registering a name](#) [Next](#) [Solana](#) Last updated 2 days ago On this page