

```
D4D4D4;--ch-t-background: #1E1E1E;--ch-t-lighter-
inlineBackground: #1e1e1ee6;--ch-t-editor-background:
#1E1E1E;--ch-t-editor-foreground: #D4D4D4;--ch-t-editor-
rangeHighlightBackground: #ffffff0b;--ch-t-editor-
infoForeground: #3794FF;--ch-t-editor-
selectionBackground: #264F78;--ch-t-focusBorder:
#007FD4;--ch-t-tab-activeBackground: #1E1E1E;--ch-t-
tab-activeForeground: #ffffff;--ch-t-tab-
inactiveBackground: #2D2D2D;--ch-t-tab-
inactiveForeground: #ffffff80;--ch-t-tab-border: #252526;--
ch-t-tab-activeBorder: #1E1E1E;--ch-t-editorGroup-
border: #444444;--ch-t-editorGroupHeader-
tabsBackground: #252526;--ch-t-editorLineNumber-
foreground: #858585;--ch-t-input-background: #3C3C3C;-
-ch-t-input-foreground: #D4D4D4;--ch-t-icon-foreground:
#C5C5C5;--ch-t-sideBar-background: #252526;--ch-t-
sideBar-foreground: #D4D4D4;--ch-t-sideBar-border:
#252526;--ch-t-list-activeSelectionBackground: #094771;--
ch-t-list-activeSelectionForeground: #ffffffe;--ch-t-list-
hoverBackground: #2A2D2E; }
```

## Human approval for AI agent actions

This page describes a setup, where an autonomous agent proposes transactions, and one or more human signers approve and execute the transaction.

This setup benefits from increased security, as the Smart Account can not be compromised by convincing the agent to execute a malicious transaction.

On the other hand, it can take minutes or hours to collect the necessary approvals, which reduces the agility of this setup.

### Setup the Smart Account

For this setup, we recommend a 2-out-of-3, 3-out-of-5, or 5-out-of-7 threshold. The important considerations are:

- The AI agent should be one signer
- The threshold should be two more, so at least one human approval is required
- The amount of signers should be higher than the threshold to make sure the Safe Smart Account is functional when one key is lost

Here is an example setup:

#### Deploy Safe

You can add the AI agent as a signer in the [Safe Wallet \(opens in a new tab\)](#).

You can also setup the Safe Smart Account programmatically like this using the Safe [Protocol Kit](#) :

```
_16 import Safe from '@safe-global/protocol-kit' _16 _16 const AGENT_ADDRESS = // ... _16 const
AGENT_PRIVATE_KEY = // ... _16 const HUMAN_SIGNER_1_ADDRESS = // ... _16 const
HUMAN_SIGNER_2_ADDRESS = // ... _16 const RPC_URL = 'https://rpc.ankr.com/eth_sepolia' _16 _16 const newSafe =
await Safe.init({ _16 provider: RPC_URL, _16 signer: AGENT_PRIVATE_KEY, _16 safeOptions: { _16 owners:
[AGENT_ADDRESS, HUMAN_SIGNER_1_ADDRESS, HUMAN_SIGNER_2_ADDRESS], _16 threshold: 2 _16 } _16 })
Here, the AI agent creates the Safe Smart Account and adds two human signers for a 2-out-of-3 setup. The Smart Account
will be deployed when the first transaction is executed.
```

## Assemble and propose a transaction

The AI agent can now propose transactions. We recommend sending the transactions to the [Safe Transaction Service](#). By this, you make sure that the transactions show up in the Safe Wallet interface and can easily be checked, approved and executed by the human signers.

You can use the [API Kit](#) to propose transactions to the Safe Transaction Service.

A simple example transaction to the zero address can be proposed like this:

```
_29 import SafeApiKit from '@safe-global/api-kit' _29 _29 const apiKit = new SafeApiKit({ _29 chainId: 11155111n _29 })
_29 _29 const tx = await newSafe.createTransaction({ _29 transactions: [ _29 { _29 to:
'0x0000000000000000000000000000000000000000000000000000000000000000', _29 data: '0x', _29 value: '0' _29 } _29 ] _29 }) _29 _29 // Every
transaction has a Safe (Smart Account) Transaction Hash different than the final transaction hash _29 const safeTxHash =
await newSafe.getTransactionHash(tx) _29 // The AI agent signs this Safe (Smart Account) Transaction Hash _29 const
signature = await newSafe.signHash(safeTxHash) _29 _29 // Now the transaction with the signature is sent to the
Transaction Service with the Api Kit: _29 await apiKit.proposeTransaction({ _29 safeAddress: safeAddress, _29
safeTransactionData: tx.data, _29 safeTxHash, _29 senderSignature: signature.data, _29 senderAddress:
AGENT_ADDRESS _29 })
```

## Approve and execute the transactions

The transactions will now show up in the transaction interface of the [Safe Wallet\(opens in a new tab\)](#). The human signers now have to connect their Metamask, and approve and/or execute the transactions with a click. They can also use the [Mobile App\(opens in a new tab\)](#) to sign the transactions.

In the Safe Wallet, the human signers will see the transaction in the queued transaction view:

And can either add a signature or execute the transaction when enough signatures were collected:

## Next steps

Now your AI agent is equipped with a Safe Smart Account and you are in full control of the transactions. We are excited to see what you will build.

If you have a technical question, feel free to reach out on [Stack Exchange\(opens in a new tab\)](#) with the safe-core tag.

[Setup your Agent with a Safe account](#) [Multiple Agent setup](#) Was this page helpful?

[Report issue](#)