Custom EVM accounts

The Keyring API integrates custom EVM accounts inside MetaMask. You can use the Keyring API to display custom accounts, such as multi-party computation (MPC) accounts andERC-4337 accounts , alongside regular MetaMask accounts in the user interface:

To use the Keyring API, you firstimplement the API in an account management Snap(also known as an "account Snap"). You can thencall Keyring API methods from a companion dappto enable users to create and interact with the custom accounts.

see also * Create an account management Snap * Create an account management companion dapp * Account management Snap security guidelines * Keyring API reference

System context diagram

The following diagram shows the system context when interacting with accounts managed by an account management Snap:

mermaid-svg-9055620{font-family:arial,verdana,sans-serif;font-size:16px;fill:#333;}#mermaid-svg-9055620 .error-icon{fill:#552222;}#mermaid-svg-9055620 .error-text{fill:#552222;stroke:#552222;}#mermaid-svg-9055620 .edge-thickness-normal{stroke-width:2px;}#mermaid-svg-9055620 .edge-thickness-thick{stroke-width:3.5px;}#mermaid-svg-9055620 .edge-pattern-solid{stroke-dasharray:0;}#mermaid-svg-9055620 .edge-pattern-dashed{stroke-dasharray:3;}#mermaid-svg-9055620 .edge-pattern-dotted{stroke-dasharray:2;}#mermaid-svg-9055620 .marker{fill:#333333;stroke:#333333;}#mermaid-svg-9055620 .marker.cross{stroke:#333333;}#mermaid-svg-9055620 svg{font-family:arial,verdana,sans-serif;font-size:16px;}#mermaid-svg-9055620 .label{font-family:arial,verdana,sans-serif;color:#333;}#mermaid-svg-9055620 .cluster-label text{fill:#333;}#mermaid-svg-9055620 .cluster-label span{color:#333;}#mermaid-svg-9055620 .label text,#mermaid-svg-9055620 span{fill:#333;color:#333;}#mermaid-svg-9055620 .node rect,#mermaid-svg-9055620 .node circle,#mermaid-svg-9055620 .node ellipse,#mermaid-svg-9055620 .node polygon,#mermaid-svg-9055620 .node path{fill:#ECECFF;stroke:#9370DB;stroke-width:1px;}#mermaid-svg-9055620 .node .label{text-align:center;}#mermaid-svg-9055620 .node.clickable{cursor:pointer;}#mermaid-svg-9055620

```
.arrowheadPath{fill:#333333;}#mermaid-svg-9055620 .edgePath .path{stroke:#333333;stroke-width:2.0px;}#mermaid-svg-9055620 .flowchart-link{stroke:#333333;fill:none;}#mermaid-svg-9055620 .edgeLabel{background-color:#e8e8e8;text-align:center;}#mermaid-svg-9055620 .edgeLabel rect{opacity:0.5;background-color:#e8e8e8;fill:#e8e8e8;}#mermaid-svg-9055620 .cluster rect{fill:#ffffde;stroke:#aaaa33;stroke-width:1px;}#mermaid-svg-9055620 .cluster text{fill:#333;}#mermaid-svg-9055620 .cluster span{color:#333;}#mermaid-svg-9055620 div.mermaidTooltip{position:absolute;text-align:center;max-width:200px;padding:2px;font-family:arial,verdana,sans-serif;font-size:12px;background:hsl(80, 100%, 96.2745098039%);border:1px solid #aaaa33;border-radius:2px;pointer-events:none;z-index:100;}#mermaid-svg-9055620 .flowchartTitleText{text-anchor:middle;font-size:18px;fill:#333;}#mermaid-svg-9055620 :root{--mermaid-font-family:arial,verdana,sans-serif;}
```

Use to submit requests and manage accounts Start requests Use to manage requests and accounts Submit requests Manage requests and accounts Submit requests and manage accounts Notify about account and request events User MetaMask Dapp Snap companion dapp Snap

The diagram contains the following components:

- User
-
    - The user interacting with the dapp, the Snap companion dapp, and MetaMask.
- Dapp
-
    - The dapp requesting an action to be performed on an account.
- MetaMask
-
    - The wallet the dapp connects to.
- MetaMask routes requests to the account management Snap and lets the user perform some level of
- account management.
- Snap
-
    - The account management Snap that implements the Keyring API to manage the user's
- accounts and handle requests that use these accounts.
- Snap companion dapp
-
    - The Snap's user interface component that allows the user to interact with
- the Snap to manage accounts and requests.

Account management Snap installation flow

The first process a user encounters when using an account management Snap is the Snap installation flow. This process can be initiated through MetaMask's or the Snap companion dapp. The flow looks like the following:

mermaid-svg-4500947{font-family:arial,verdana,sans-serif;font-size:16px;fill:#333;}#mermaid-svg-4500947 .error-icon{fill:#552222;}#mermaid-svg-4500947 .error-text{fill:#552222;stroke:#552222;}#mermaid-svg-4500947 .edge-thickness-normal{stroke-width:2px;}#mermaid-svg-4500947 .edge-thickness-thick{stroke-width:3.5px;}#mermaid-svg-4500947 .edge-pattern-solid{stroke-dasharray:0;}#mermaid-svg-4500947 .edge-pattern-dashed{stroke-dasharray:3;}#mermaid-svg-4500947 .edge-pattern-dotted{stroke-dasharray:2;}#mermaid-svg-4500947 .marker{fill:#333333;stroke:#333333;}#mermaid-svg-4500947 .marker.cross{stroke:#333333;}#mermaid-svg-4500947 svg{font-family:arial,verdana,sans-serif;font-size:16px;}#mermaid-svg-4500947 .actor{stroke:hsl(259.6261682243, 59.7765363128%, 87.9019607843%);fill:#ECECFF;}#mermaid-svg-4500947 text.actor>tspan{fill:black;stroke:none;}#mermaid-svg-4500947 .actor-line{stroke:grey;}#mermaid-svg-4500947 .messageLine0{stroke-width:1.5;stroke-dasharray:none;stroke:#333;}#mermaid-svg-4500947 .messageLine1{stroke-width:1.5;stroke-dasharray:2,2;stroke:#333;}#mermaid-svg-4500947 #arrowhead path{fill:#333;stroke:#333;}#mermaid-svg-4500947 .sequenceNumber{fill:white;}#mermaid-svg-4500947 #sequencenumber{fill:#333;}#mermaid-svg-4500947 #crosshead path{fill:#333;stroke:#333;}#mermaid-svg-4500947 .messageText{fill:#333;stroke:none;}#mermaid-svg-4500947 .labelBox{stroke:hsl(259.6261682243, 59.7765363128%, 87.9019607843%);fill:#ECECFF;}#mermaid-svg-4500947 .labelText,#mermaid-svg-4500947 .labelText>tspan{fill:black;stroke:none;}#mermaid-svg-4500947 .loopText,#mermaid-svg-4500947

.loopText>tspan{fill:black;stroke:none;}#mermaid-svg-4500947 .loopLine{stroke-width:2px;stroke-dasharray:2,2;stroke:hsl(259.6261682243, 59.7765363128%, 87.9019607843%);fill:hsl(259.6261682243, 59.7765363128%, 87.9019607843%);}#mermaid-svg-4500947 .note{stroke:#aaaa33;fill:#fff5ad;}#mermaid-svg-4500947 .noteText,#mermaid-svg-4500947 .noteText>tspan{fill:black;stroke:none;}#mermaid-svg-4500947 .activation0{fill:#f4f4f4;stroke:#666;}#mermaid-svg-4500947 .activation1{fill:#f4f4f4;stroke:#666;}#mermaid-svg-4500947 .activation2{fill:#f4f4f4;stroke:#666;}#mermaid-svg-4500947 .actorPopupMenu{position:absolute;}#mermaid-svg-4500947 .actorPopupMenuPanel{position:absolute;fill:#ECECFF;box-shadow:0px 8px 16px 0px rgba(0,0,0,0.2);filter:drop-shadow(3px 5px 2px rgb(0 0 0 / 0.4));}#mermaid-svg-4500947 .actor-man line{stroke:hsl(259.6261682243, 59.7765363128%, 87.9019607843%);fill:#ECECFF;}#mermaid-svg-4500947 .actor-man circle,#mermaid-svg-4500947 line{stroke:hsl(259.6261682243, 59.7765363128%, 87.9019607843%);fill:#ECECFF;stroke-width:2px;}#mermaid-svg-4500947 :root{--mermaid-font-family:arial,verdana,sans-serif;}

User MetaMask Snap Snap companion dapp

alt [Optional]

Add account Snap

1 Display suggested Snaps

2 Select Snap

3 Open in a new tab

4 Install Snap?

5 Display permissions dialog

6 Approve permissions

7 OK

8 User MetaMask Snap Snap companion dapp The MetaMask account selection modal has an option calledAdd account Snap .

This option shows a list of account management Snaps. Each Snap redirects the user to the companion dapp that contains the user interface to configure and manage the Snap.

Custom account creation flow

Once the account management Snap is installed, the user can use the Snap companion dapp to create or import custom accounts. The flow looks like the following:

mermaid-svg-5067649{font-family:arial,verdana,sans-serif;font-size:16px;fill:#333;}#mermaid-svg-5067649 .error-icon{fill:#552222;}#mermaid-svg-5067649 .error-text{fill:#552222;stroke:#552222;}#mermaid-svg-5067649 .edge-thickness-normal{stroke-width:2px;}#mermaid-svg-5067649 .edge-thickness-thick{stroke-width:3.5px;}#mermaid-svg-5067649 .edge-pattern-solid{stroke-dasharray:0;}#mermaid-svg-5067649 .edge-pattern-dashed{stroke-dasharray:3;}#mermaid-svg-5067649 .edge-pattern-dotted{stroke-dasharray:2;}#mermaid-svg-5067649 .marker{fill:#333333;stroke:#333333;}#mermaid-svg-5067649 .marker.cross{stroke:#333333;}#mermaid-svg-5067649 svg{font-family:arial,verdana,sans-serif;font-size:16px;}#mermaid-svg-5067649 .actor{stroke:hsl(259.6261682243, 59.7765363128%, 87.9019607843%);fill:#ECECFF;}#mermaid-svg-5067649 text.actor>tspan{fill:black;stroke:none;}#mermaid-svg-5067649 .actor-line{stroke:grey;}#mermaid-svg-5067649 .messageLine0{stroke-width:1.5;stroke-dasharray:none;stroke:#333;}#mermaid-svg-5067649 .messageLine1{stroke-width:1.5;stroke-dasharray:2,2;stroke:#333;}#mermaid-svg-5067649 #arrowhead path{fill:#333;stroke:#333;}#mermaid-svg-5067649 .sequenceNumber{fill:white;}#mermaid-svg-5067649 #sequencenumber{fill:#333;}#mermaid-svg-5067649 #crosshead path{fill:#333;stroke:#333;}#mermaid-svg-5067649 .messageText{fill:#333;stroke:none;}#mermaid-svg-5067649 .labelBox{stroke:hsl(259.6261682243, 59.7765363128%, 87.9019607843%);fill:#ECECFF;}#mermaid-svg-5067649

.labelText,#mermaid-svg-5067649 .labelText>tspan{fill:black;stroke:none;}#mermaid-svg-5067649 .loopText,#mermaid-svg-5067649 .loopText>tspan{fill:black;stroke:none;}#mermaid-svg-5067649 .loopLine{stroke-width:2px;stroke-dasharray:2,2;stroke:hsl(259.6261682243, 59.7765363128%, 87.9019607843%);fill:hsl(259.6261682243, 59.7765363128%, 87.9019607843%);}#mermaid-svg-5067649 .note{stroke:#aaaa33;fill:#fff5ad;}#mermaid-svg-5067649 .noteText,#mermaid-svg-5067649 .noteText>tspan{fill:black;stroke:none;}#mermaid-svg-5067649 .activation0{fill:#f4f4f4;stroke:#666;}#mermaid-svg-5067649 .activation1{fill:#f4f4f4;stroke:#666;}#mermaid-svg-5067649 .activation2{fill:#f4f4f4;stroke:#666;}#mermaid-svg-5067649 .actorPopupMenu{position:absolute;}#mermaid-svg-5067649 .actorPopupMenuPanel{position:absolute;fill:#ECECFF;box-shadow:0px 8px 16px 0px rgba(0,0,0,0.2);filter:drop-shadow(3px 5px 2px rgb(0 0 0 / 0.4));}#mermaid-svg-5067649 .actor-man line{stroke:hsl(259.6261682243, 59.7765363128%, 87.9019607843%);fill:#ECECFF;}#mermaid-svg-5067649 .actor-man circle,#mermaid-svg-5067649 line{stroke:hsl(259.6261682243, 59.7765363128%, 87.9019607843%);fill:#ECECFF;stroke-width:2px;}#mermaid-svg-5067649 :root{--mermaid-font-family:arial,verdana,sans-serif;}

User MetaMask Snap Snap companion dapp

Create new account

1 Custom logic to create account

2 keyring_createAccount(options)

3 Custom logic to create account

4 snap_manageAccounts("notify:accountCreated", account)

5 Approve account creation

6 OK

7 OK

8 Done

9 User MetaMask Snap Snap companion dapp The companion dapp presents a user interface allowing the user to configure their custom account. The dapp creates an account usingkeyring_createAccount .

The Snap keeps track of the accounts that it creates usingsnap_manageState . Once the Snap has created an account, it notifies MetaMask usingsnap_manageAccounts .

Once the Snap has created an account, that account can be used to sign messages and transactions.

Transaction flows

The Keyring API supports two flows for handling requestssynchronous andasynchronous .

In general, you should use the asynchronous flow when the request requires user interaction (for example, using a hardware key or a threshold signature scheme) or when the request takes a long time to complete. You should use the synchronous flow for any other use case.

Synchronous transaction flow

The synchronous flow looks like the following:

mermaid-svg-1300813{font-family:arial,verdana,sans-serif;font-size:16px;fill:#333;}#mermaid-svg-1300813 .error-icon{fill:#552222;}#mermaid-svg-1300813 .error-text{fill:#552222;stroke:#552222;}#mermaid-svg-1300813 .edge-thickness-normal{stroke-width:2px;}#mermaid-svg-1300813 .edge-thickness-thick{stroke-width:3.5px;}#mermaid-svg-1300813 .edge-pattern-solid{stroke-dasharray:0;}#mermaid-svg-1300813 .edge-pattern-dashed{stroke-dasharray:3;}#mermaid-svg-1300813 .edge-pattern-dotted{stroke-dasharray:2;}#mermaid-svg-1300813 .marker{fill:#333333;stroke:#333333;}#mermaid-svg-1300813 .marker.cross{stroke:#333333;}#mermaid-svg-1300813 svg{font-family:arial,verdana,sans-serif;font-size:16px;}#mermaid-svg-1300813 .actor{stroke:hsl(259.6261682243, 59.7765363128%, 87.9019607843%);fill:#ECECFF;}#mermaid-svg-1300813 text.actor>tspan{fill:black;stroke:none;}#mermaid-svg-1300813 .actor-line{stroke:grey;}#mermaid-svg-1300813 .messageLine0{stroke-width:1.5;stroke-dasharray:none;stroke:#333;}#mermaid-svg-1300813 .messageLine1{stroke-width:1.5;stroke-dasharray:2,2;stroke:#333;}#mermaid-svg-1300813 #arrowhead path{fill:#333;stroke:#333;}#mermaid-svg-

1300813 .sequenceNumber{fill:white;}#mermaid-svg-
1300813 #sequencenumber{fill:#333;}#mermaid-svg-
1300813 #crosshead path{fill:#333;stroke:#333;}#mermaid-
svg-1300813
.messageText{fill:#333;stroke:none;}#mermaid-svg-
1300813 .labelBox{stroke:hsl(259.6261682243,
59.7765363128%,
87.9019607843%);fill:#ECECFF;}#mermaid-svg-1300813
.labelText,#mermaid-svg-1300813
.labelText>tspan{fill:black;stroke:none;}#mermaid-svg-
1300813 .loopText,#mermaid-svg-1300813
.loopText>tspan{fill:black;stroke:none;}#mermaid-svg-
1300813 .loopLine{stroke-width:2px;stroke-
dasharray:2,2;stroke:hsl(259.6261682243, 59.7765363128%,
87.9019607843%);fill:hsl(259.6261682243, 59.7765363128%,
87.9019607843%);}#mermaid-svg-1300813
.note{stroke:#aaaa33;fill:#fff5ad;}#mermaid-svg-1300813
.noteText,#mermaid-svg-1300813
.noteText>tspan{fill:black;stroke:none;}#mermaid-svg-
1300813 .activation0{fill:#f4f4f4;stroke:#666;}#mermaid-
svg-1300813
.activation1{fill:#f4f4f4;stroke:#666;}#mermaid-svg-1300813
.activation2{fill:#f4f4f4;stroke:#666;}#mermaid-svg-1300813
.actorPopupMenu{position:absolute;}#mermaid-svg-
1300813
.actorPopupMenuPanel{position:absolute;fill:#ECECFF;box-
shadow:0px 8px 16px 0px rgba(0,0,0,0.2);filter:drop-
shadow(3px 5px 2px rgb(0 0 0 / 0.4));}#mermaid-svg-
1300813 .actor-man line{stroke:hsl(259.6261682243,
59.7765363128%,
87.9019607843%);fill:#ECECFF;}#mermaid-svg-1300813
.actor-man circle,#mermaid-svg-1300813
line{stroke:hsl(259.6261682243, 59.7765363128%,
87.9019607843%);fill:#ECECFF;stroke-
width:2px;}#mermaid-svg-1300813 :root{--mermaid-font-

family:arial,verdana,sans-serif;}

User Dapp MetaMask Snap

Create new sign request

1 ethereum.request(request)

2 Display request to user

3 Approve request

4 keyring_submitRequest(request)

5 Custom logic to handle request

6 { pending: false, result }

7 result

8 Done

9 User Dapp MetaMask Snap The flow starts when a user or dapp initiates a sign request. At that point, MetaMask detects that this interaction is requested for an account controlled by the account management Snap.

After the user approves the transaction in MetaMask, MetaMask calls keyring_submitRequest , which receives the original RPC request and returns a response withpending set tofalse , andresult set to the requested signature.

Asynchronous transaction flow

The asynchronous flow looks like the following:

mermaid-svg-5764969{font-family:arial,verdana,sans-serif;font-size:16px;fill:#333;}#mermaid-svg-5764969 .error-icon{fill:#552222;}#mermaid-svg-5764969 .error-text{fill:#552222;stroke:#552222;}#mermaid-svg-5764969 .edge-thickness-normal{stroke-width:2px;}#mermaid-svg-5764969 .edge-thickness-thick{stroke-width:3.5px;}#mermaid-svg-5764969 .edge-pattern-solid{stroke-dasharray:0;}#mermaid-svg-5764969 .edge-pattern-dashed{stroke-dasharray:3;}#mermaid-svg-5764969 .edge-pattern-dotted{stroke-dasharray:2;}#mermaid-svg-5764969 .marker{fill:#333333;stroke:#333333;}#mermaid-svg-5764969 .marker.cross{stroke:#333333;}#mermaid-svg-5764969 svg{font-family:arial,verdana,sans-serif;font-size:16px;}#mermaid-svg-5764969 .actor{stroke:hsl(259.6261682243, 59.7765363128%, 87.9019607843%);fill:#ECECFF;}#mermaid-svg-5764969 text.actor>tspan{fill:black;stroke:none;}#mermaid-svg-5764969 .actor-line{stroke:grey;}#mermaid-svg-5764969 .messageLine0{stroke-width:1.5;stroke-

```
dasharray:none;stroke:#333;}#mermaid-svg-5764969
.messageLine1{stroke-width:1.5;stroke-
dasharray:2,2;stroke:#333;}#mermaid-svg-5764969
#arrowhead path{fill:#333;stroke:#333;}#mermaid-svg-
5764969 .sequenceNumber{fill:white;}#mermaid-svg-
5764969 #sequencenumber{fill:#333;}#mermaid-svg-
5764969 #crosshead path{fill:#333;stroke:#333;}#mermaid-
svg-5764969
.messageText{fill:#333;stroke:none;}#mermaid-svg-
5764969 .labelBox{stroke:hsl(259.6261682243,
59.7765363128%,
87.9019607843%);fill:#ECECFF;}#mermaid-svg-5764969
.labelText,#mermaid-svg-5764969
.labelText>tspan{fill:black;stroke:none;}#mermaid-svg-
5764969 .loopText,#mermaid-svg-5764969
.loopText>tspan{fill:black;stroke:none;}#mermaid-svg-
5764969 .loopLine{stroke-width:2px;stroke-
dasharray:2,2;stroke:hsl(259.6261682243, 59.7765363128%,
87.9019607843%);fill:hsl(259.6261682243, 59.7765363128%,
87.9019607843%);}#mermaid-svg-5764969
.note{stroke:#aaaa33;fill:#fff5ad;}#mermaid-svg-5764969
.noteText,#mermaid-svg-5764969
.noteText>tspan{fill:black;stroke:none;}#mermaid-svg-
5764969 .activation0{fill:#f4f4f4;stroke:#666;}#mermaid-
svg-5764969
.activation1{fill:#f4f4f4;stroke:#666;}#mermaid-svg-5764969
.activation2{fill:#f4f4f4;stroke:#666;}#mermaid-svg-5764969
.actorPopupMenu{position:absolute;}#mermaid-svg-
5764969
.actorPopupMenuPanel{position:absolute;fill:#ECECFF;box-
shadow:0px 8px 16px 0px rgba(0,0,0,0.2);filter:drop-
shadow(3px 5px 2px rgb(0 0 0 / 0.4));}#mermaid-svg-
5764969 .actor-man line{stroke:hsl(259.6261682243,
59.7765363128%,
87.9019607843%);fill:#ECECFF;}#mermaid-svg-5764969
```

.actor-man circle,#mermaid-svg-5764969 line{stroke:hsl(259.6261682243, 59.7765363128%, 87.9019607843%);fill:#ECECFF;stroke-width:2px;}#mermaid-svg-5764969 :root{--mermaid-font-family:arial,verdana,sans-serif;}

User Dapp MetaMask Snap Snap companion dapp

alt [There is a redirect URL]

Create new sign request

1 ethereum.request(request)

2 Display request to user

3 Approve request

4 keyring_submitRequest(request)

5 Save request to Snap's state

6 { pending: true, redirect? }

7 Acknowledge redirection

8 Open redirect URL in a new tab

9 keyring_getRequest(id)

10 request

11 Custom logic to handle request

12 keyring_approveRequest(id, data?)

13 Custom logic to handle request

14 snap_manageAccounts("notify:requestApproved", { id, result })

15 result

16 OK

17 OK

18 Done

19 User Dapp MetaMask Snap Snap companion dapp The flow starts the same way as the synchronous flow : a user or dapp initiates a sign request. After approval, MetaMask calls keyring_submitRequest .

Since the Snap doesn't answer the request directly, it stores the pending request in its internal state using snap_manageState . The Snap sends a{ pending: true, redirect? } response to indicate that the request will be handled asynchronously. This response can optionally contain a redirect URL that MetaMask will open in a new tab to allow the user to interact with the Snap companion dapp.

The companion dapp gets the Snap's pending request using keyring_getRequest . It resolves the request using keyring_approveRequest , and the Snap resolves the request using snap_manageAccounts , notifying MetaMask of the result.

EOA methods

An account management Snap can implement the following methods to support dapp requests from externally owned accounts (EOAs):

- personal_sign
- eth_signTypedData_v4
- eth_signTransaction
- Deprecated signing methods

Account abstraction (ERC-4337)

Flask Only This feature is experimental and only available inMetaMask Flask , the canary distribution of MetaMask. Account abstraction, specified byEIP-4337 , introducesuser operations and enables users to manage smart contract accounts containing arbitrary verification logic. Users can use these ERC-4337 accounts instead of externally owned accounts as primary accounts.

An account management Snap can implement the following methods to support dapp requests from ERC-4337 accounts:

- eth_prepareUserOperation
- eth_patchUserOperation
- eth_signUserOperation

The user operation signing flow in an ERC-4337 compatible account Snap looks like the following:

mermaid-svg-5749205{font-family:arial,verdana,sans-serif;font-size:16px;fill:#333;}#mermaid-svg-5749205 .error-icon{fill:#552222;}#mermaid-svg-5749205 .error-text{fill:#552222;stroke:#552222;}#mermaid-svg-5749205 .edge-thickness-normal{stroke-width:2px;}#mermaid-svg-5749205 .edge-thickness-thick{stroke-width:3.5px;}#mermaid-svg-5749205 .edge-pattern-solid{stroke-dasharray:0;}#mermaid-svg-5749205 .edge-pattern-dashed{stroke-dasharray:3;}#mermaid-svg-5749205 .edge-pattern-dotted{stroke-dasharray:2;}#mermaid-svg-5749205 .marker{fill:#333333;stroke:#333333;}#mermaid-svg-5749205 .marker.cross{stroke:#333333;}#mermaid-svg-5749205 svg{font-family:arial,verdana,sans-serif;font-size:16px;}#mermaid-svg-5749205 .actor{stroke:hsl(259.6261682243, 59.7765363128%, 87.9019607843%);fill:#ECECFF;}#mermaid-svg-5749205 text.actor>tspan{fill:black;stroke:none;}#mermaid-svg-5749205 .actor-line{stroke:grey;}#mermaid-svg-5749205 .messageLine0{stroke-width:1.5;stroke-dasharray:none;stroke:#333;}#mermaid-svg-5749205 .messageLine1{stroke-width:1.5;stroke-dasharray:2,2;stroke:#333;}#mermaid-svg-5749205 #arrowhead path{fill:#333;stroke:#333;}#mermaid-svg-5749205 .sequenceNumber{fill:white;}#mermaid-svg-5749205 #sequencenumber{fill:#333;}#mermaid-svg-5749205 #crosshead path{fill:#333;stroke:#333;}#mermaid-svg-5749205 .messageText{fill:#333;stroke:none;}#mermaid-svg-

5749205 .labelBox{stroke:hsl(259.6261682243, 59.7765363128%, 87.9019607843%);fill:#ECECFF;}#mermaid-svg-5749205 .labelText,#mermaid-svg-5749205 .labelText>tspan{fill:black;stroke:none;}#mermaid-svg-5749205 .loopText,#mermaid-svg-5749205 .loopText>tspan{fill:black;stroke:none;}#mermaid-svg-5749205 .loopLine{stroke-width:2px;stroke-dasharray:2,2;stroke:hsl(259.6261682243, 59.7765363128%, 87.9019607843%);fill:hsl(259.6261682243, 59.7765363128%, 87.9019607843%);}#mermaid-svg-5749205 .note{stroke:#aaaa33;fill:#fff5ad;}#mermaid-svg-5749205 .noteText,#mermaid-svg-5749205 .noteText>tspan{fill:black;stroke:none;}#mermaid-svg-5749205 .activation0{fill:#f4f4f4;stroke:#666;}#mermaid-svg-5749205 .activation1{fill:#f4f4f4;stroke:#666;}#mermaid-svg-5749205 .activation2{fill:#f4f4f4;stroke:#666;}#mermaid-svg-5749205 .actorPopupMenu{position:absolute;}#mermaid-svg-5749205 .actorPopupMenuPanel{position:absolute;fill:#ECECFF;box-shadow:0px 8px 16px 0px rgba(0,0,0,0.2);filter:drop-shadow(3px 5px 2px rgb(0 0 0 / 0.4));}#mermaid-svg-5749205 .actor-man line{stroke:hsl(259.6261682243, 59.7765363128%, 87.9019607843%);fill:#ECECFF;}#mermaid-svg-5749205 .actor-man circle,#mermaid-svg-5749205 line{stroke:hsl(259.6261682243, 59.7765363128%, 87.9019607843%);fill:#ECECFF;stroke-width:2px;}#mermaid-svg-5749205 :root{--mermaid-font-family:arial,verdana,sans-serif;}

Dapp MetaMask Snap

Currently, only one transaction per userOp is supported

alt [The account is already deployed] [The account is not deployed and the initCode is not present] alt [The gas isn't set]

Transaction intents

1 eth_prepareUserOperation(transaction intents)

2 userOp details

3 Check if account is already deployed

4 Remove the initCode if set

5 Throw an error (without the exact reason)

6 Estimate and set gas values

7 Estimate and set gas fees

8 eth_patchUserOperation(userOp object)

9 Partial userOp object

10 Update paymasterAndData and remove dummy signature

11 eth_signUserOperation(userOp object, entry point)

12 Signature

13 Update userOp's signature

14 Submit userOp to bundler and wait for transaction hash

15 Transaction hash

16 Dapp MetaMask Snap See theERC-4337 methods for more information about their parameters and response details.

Examples

See the following example account management Snap implementations:

- Simple Account Snap
- Simple Account Abstraction Snap
- (ERC-4337)
- Biconomy Smart Account Snap
- (ERC-4337)
- Silent Shard Snap
- Safeheron MPC Snap
- Capsule Keyring Snap

Edit this page