Lifecycle hooks

You can implement lifecycle hooks to automatically run an action, such as displaying a dialog or notification, when a user installs or updates your Snap.

Steps

1. Request permission to implement lifecycle hooks

Request theendowment:lifecycle-hooks permission. Add the following to your Snap's manifest file:

snap.manifest.json "initialPermissions" :

{ "endowment:lifecycle-hooks" :

{ } }

2. Run an action on installation

To run an action when a user installs your Snap, expose theonInstall entry point and implement the action. For example, you can useonInstall to perform any initialization that is required upon installation.

The following example displays analert dialog upon installation:

index.ts import

type

{ OnInstallHandler }

from

"@metamask/snaps-sdk" ; import

{ heading , panel , text }

from

"@metamask/snaps-sdk" ;

export

const onInstall :

OnInstallHandler

=

async

( )

=>

{ await snap . request ({ method :

"snap_dialog" , params :

{ type :

"alert" , content :

panel ([heading ("Installation successful") , text ("To use this Snap, visit the companion dapp atmetamask.io." ,) ,]) , } , }) ; } ;

3. Run an action on update

To run an action when a user updates your Snap, expose theonUpdate entry point and implement the action. For example, you can useonUpdate to perform any migrations that are required upon update.

The following example displays analert dialog upon update:

index.ts import

type

{ OnUpdateHandler }

from

"@metamask/snaps-sdk" ; import

{ heading , panel , text }

from

"@metamask/snaps-sdk" ;

export

const onUpdate :

OnUpdateHandler

=

async

( )

=>

{ await snap . request ({ method :

"snap_dialog" , params :

{ type :

"alert" , content :

panel ([heading ("Update successful") , text ("New features added in this version:" ,) , text ("Added a dialog that appears when updating.") ,]) , } , }) ; } ;

Example

See the@metamask/lifecycle-hooks-example-snap package for a full example of implementing lifecycle hooks.

Edit this page