Cron jobs

You can schedule actions to run periodically at fixed times or intervals, also known as "cron jobs." For example, you can display a dialog or notification in MetaMask at a specific time each day.

Steps

1. Configure a cron job

To configure a cron job, request theendowment:cronjob permission, specifying one or more cron jobs in thejobs array. Define each job with acron expression and a request object, which MetaMask sends to the Snap's cron job handler when the job is executed.

For example, to configure a job that executes every minute, add the following to your Snap's manifest file:

```
snap.manifest.json "initialPermissions" :

{ "endowment:cronjob" :

{ "jobs" :

[ { "expression" :

"  * * *" , "request" :

{ "method" :

"execute" } } ] } }
```

2. Implement a cron job handler

Expose anonCronjob entry point, which is triggered at the specified schedule with the requests defined in theendowment:cronjob permission. The following example handles theexecute method specified in the previous example:

```
index.ts import

type

{ OnCronjobHandler }

from

"@metamask/snaps-sdk" ;

export

const onCronjob :

OnCronjobHandler

=

async

( { request } )

=>

{ switch

( request . method )

{ case

"execute" : // Cron jobs can execute any method that is available to the Snap. return snap . request ( { method :

"snap_dialog" , params :

{ type :

"alert" , content :
```

panel ([heading ("Cron job") , text ("This dialog was triggered by a cron job.") ,]) , } , }) ;

default : throw

new

Error ("Method not found.") ; } } ; Access data from cron jobs When accessing encrypted data from cron jobs usingsnap_manageState , MetaMask requires the user to enter their password if the wallet is locked. This interaction can be confusing to the user, since the Snap accesses the data in the background without the user being aware.

If the cron job requires access to encrypted state, usesnap_getClientStatus to ensure that MetaMask is unlocked before accessing state. This will prevent an unexpected password request, improving the user's experience.

If the cron job does not require access to sensitive data, store that data in unencrypted state by settingencrypted tofalse when usingsnap_manageState .

Example

See the@metamask/cronjob-example-snap package for a full example of implementing cron jobs.

Edit this page