

Request permissions

[Snaps must request permission](#) to access certain powerful JavaScript globals or API methods. [Dapps must also request permission](#) to access certain API methods to communicate with Snaps.

Snaps and dapps follow the [EIP-2255 wallet permissions specification](#).

Request permissions from a Snap

Snaps API methods

Request permission to call [Snaps API methods](#) in the `initialPermissions` field of the Snap [manifest file](#). For example, to request to call `snap_dialog`, add the following to the manifest file:

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

```
{ "snap_dialog" :
```

```
{ } } note All Snaps API methods except the following interactive UI methods require requesting permission in the manifest file:
```

- [snap_createInterface](#)
- [snap_getInterfaceState](#)
- [snap_updateInterface](#)

Endowments

Endowments are a type of permission. Request endowments in the `initialPermissions` field of the Snap [manifest file](#). See the [Snaps permissions reference](#) for the full list of endowments.

For example, to request the [endowment:network-access](#) permission, add the following to the manifest file:

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

```
{ "endowment:network-access" :
```

```
{ } }
```

Dynamic permissions

Dynamic permissions are not requested in the manifest file. Instead, your Snap can acquire dynamic permissions during its lifecycle.

For example, request permission to call the [eth_accounts](#) MetaMask JSON-RPC API method by calling [eth_requestAccounts](#). See the [eth_accounts dynamic permission](#) for more information.

Request permissions from a dapp

Dapps that communicate with Snaps using [wallet_snap](#) and [wallet_invokeSnap](#) must request permission to do so by calling the [wallet_requestSnaps](#) MetaMask JSON-RPC API method.

The following example calls `wallet_requestSnaps` to request permission to connect to the `hello-snap` Snap, then calls `wallet_invokeSnap` to invoke the `hello` JSON-RPC method exposed by the Snap:

```
index.js // If the Snap is not already installed, the user will be prompted to install it. await
```

```
window . ethereum . request ( { method :
```

```
"wallet_requestSnaps" , params :
```

```
{ // Assuming the Snap is published to npm using the package name "hello-snap". "npm:hello-snap" :
```

```
{ } , } , } ) ;
```

```
// Invoke the "hello" JSON-RPC method exposed by the Snap. const response =
```

```
await
```

```
window . ethereum . request ( { method :
```

```
"wallet_invokeSnap" , params :
```

```
{
```

```
  snapId :
```

```
  "npm:hello-snap" ,
```

```
  request :
```

```
{
```

```
  method :
```

```
  "hello"
```

```
}
```

```
}, } ) ;
```

```
console . log ( response ) ;
```

```
// "world!" note Learn more about implementingcustom JSON-RPC APIs in a Snap.
```

[Edit this page](#)