Transaction insights

You can provide transaction insights in MetaMask's transaction confirmation window before a user signs a transaction. For example, you can show the user the percentage of gas fees they would pay for their transaction.

Steps

1. Request permission to display transaction insights

Request the [endowment:transaction-insight](#) permission by adding the following to your Snap's manifest file:

snap.manifest.json "initialPermissions" :

{ "endowment:transaction-insight" :

{ } } If you need to receive the origin of the transaction request, add allowTransactionOrigin to the permission object, and set it to true :

snap.manifest.json "initialPermissions" :

{ "endowment:signature-insight" :

{ "allowTransactionOrigin" :

true } }

2. Implement the onTransaction

entry point

Expose an [onTransaction](#) entry point, which receives a raw unsigned transaction payload, the chain ID, and the optional transaction origin. When a user submits a transaction using the MetaMask extension, MetaMask calls the onTransaction handler method.

The following is an example implementation of onTransaction :

index.ts import

type

{ OnTransactionHandler }

from

"@metamask/snaps-sdk" ; import

{ panel , heading , text }

from

"@metamask/snaps-sdk" ;

export

const onTransaction :

OnTransactionHandler

=

async

({ transaction , chainId , transactionOrigin , })

=>

{ const insights =

/ *Get insights* / ; return

{ content :

```
panel ( [ heading ( "My Transaction Insights" ) , text ( "Here are the insights:" ) , ... ( insights . map ( ( insight )
```

```
=>
```

```
text ( insight . value ) ) ) , ] ) , } ; } ;
```
The Snap tab in the transaction confirmation window displays the transaction insights:

Transaction severity level

Flask Only This feature is experimental and only available in [MetaMask Flask](#) , the canary distribution of MetaMask. A Snap providing transaction insights can return an optional severity level of "critical" . MetaMask shows a modal with the warning before the user can confirm the transaction.

index.ts import

```
type
```

```
{ OnTransactionHandler }
```

```
from
```

```
"@metamask/snaps-sdk" ; import
```

```
{ panel , heading , text }
```

```
from
```

```
"@metamask/snaps-sdk" ;
```

```
export
```

```
const onTransaction :
```

```
OnTransactionHandler
```

```
=
```

```
async
```

```
( { transaction , chainId , transactionOrigin , } )
```

```
=>
```

```
{ const insights =
```

/ *Get insights* / ; return

```
{ content :
```

```
panel ( [ heading ( "My Transaction Insights" ) , text ( "Here are the insights:" ) , ... ( insights . map ( ( insight )
```

```
=>
```

```
text ( insight . value ) ) ) , ] ) , severity :
```

```
"critical" , } ; } ;
```

Example

For a full end-to-end tutorial that walks you through creating a transaction insights Snap, see [Create a Snap to calculate gas fee percentages](#) . You can also see the [@metamask/transaction-insights-example-snap](#) package for a full example of implementing transaction insights.

[Edit this page](#)