Communicate errors

The Snaps SDK exposes a set of known errors that can be thrown from your Snap code without crashing the Snap. See theSnaps known errors reference for the full list of errors.

Import and throw errors

To throw these known errors, first import them from the@metamask/snaps-sdk package, then throw them where needed. For example:

```
index.ts import

type

{ OnRpcRequestHandler }

from

"@metamask/snaps-sdk" ; import

{ MethodNotFoundError }

from

"@metamask/snaps-sdk" ;

export

const onRpcRequest :

OnRpcRequestHandler

=

async

( { request } )

=>

{ switch

( request . method )

{ case

"hello" : return

"Hello World!" ; default : // Throw a known error to avoid crashing the Snap. throw

new

MethodNotFoundError ( ) ; } } ;
```

Pass data with the error

The error class constructors exported by@metamask/snaps-sdk have the following signature:

```
class

SnapJsonRpcError

extends

SnapError

{ new

( message ? :

string , data ? : Record < string , Json
```

) } Both parameters are optional. If you don't passmessage , then a pre-determined message is used. If you don't passdata , then an empty object is passed.

data can be any JSON-serializable object.

Detect known errors in dapps

Known errors are thrown back to the caller as JSON-RPC errors. They have a numericcode , amessage string, and adata object.

TheSnaps known errors reference lists all the known errors with their codes and intended usage.

Example

See the@metamask/error-example-snap package for a full example of communicating errors.

Edit this page