

# Custom name resolution

Flask Only This feature is experimental and only available in [MetaMask Flask](#) , the canary distribution of MetaMask. You can implement custom domain resolution and reverse resolution using the following steps.

## Steps

### 1. Request permission to provide name resolution

Request the [endowment:name-lookup](#) permission.

For example, to resolve Ethereum Mainnet domains, add the following to your Snap's manifest file:

snap.manifest.json "initialPermissions" :

```
{ "endowment:name-lookup" :
```

```
{ "chains" :
```

```
[ "eip155:1" ] } }
```

### 2. Implement custom name resolution

Expose an [onNameLookup](#) entry point, which receives achainId and either adomain or anaddress . The following example implements a very basic resolution from Unstoppable Domains domain names to Ethereum addresses inonNameLookup :

```
index.ts import
```

```
type
```

```
{ OnNameLookupHandler }
```

```
from
```

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

```
const
```

```
UNSTOPPABLE_API_KEY
```

```
=
```

```
"xxx" ;
```

```
export
```

```
const onNameLookup :
```

```
OnNameLookupHandler
```

```
=
```

```
async
```

```
( request )
```

```
=>
```

```
{ const
```

```
{ chainId , domain }
```

```
= request ;
```

```
if
```

```
( domain && chainId ===
```

```
"eip155:1" )
```

```
{ const response =
```

await

```
fetch ( https://api.unstoppabledomains.com/resolve/domains/ { domain } ,
```

```
{ headers :
```

```
{ accept :
```

```
"application/json" , authorization :
```

```
Bearer { UNSTOPPABLE_API_KEY } , } , } ) ; const data =
```

```
await response . json ( ) ; const resolvedAddress = data . records [ "crypto.ETH.address" ] ; if
```

```
( address )
```

```
{ return
```

```
{ resolvedAddresses :
```

```
[ { resolvedAddress , protocol :
```

```
"Unstoppable Domains"
```

```
}] , } ; } }
```

```
return
```

```
null ; } ;
```

## Example

See the [@metamask/name-lookup-example-snap](#) package for a full example of implementing custom name resolution.

[Edit this page](#)