Using grant allowances to execute transactions

As one of the primary use case of the Fee Grant module is to help with onboarding new users, this section of the guide will show how to allow users to execute transactions using the allowance granted to them.

Query a grant allowance

Now that an allowance has been assigned to an account address, you should be able to verify the details of that grant. The returned information will include the granter's address, the allowance amount, and any other conditions set by the granter, such as the expiration date of the allowance.

The following command usingsecretcli will give the details of a grant allowance:

```

Copy secretcliqueryfeegrantgrant"granter_address""grantee_address"

```

Results :

```

Copy allowance: '@type':/cosmos.feegrant.v1beta1.BasicAllowance expiration:null spend_limit: -amount:"1000000" denom:uscrt grantee:secret1q0rth4fu4svxnw63vjd7w74nadzsdp0f23e0uy granter:secret1tq6y8waegggp4fv2fcxk3zmpsmlfadyc7lsd69

```

This can also be achieved via the following API endpoint whereapi_endpoint is an API endpoint connected to one of the Archway networks (Testnet or Mainnet).granter is the address that granted the allowance andgrantee is the address that received the allowance:

```

Copy {api_endpoint}/cosmos/feegrant/v1beta1/allowance/{granter}/{grantee}

```

Results :

```

Copy { "allowance":{ "granter":"secret1tq6y8waegggp4fv2fcxk3zmpsmlfadyc7lsd69", "grantee":"secret1q0rth4fu4svxnw63vjd7w74nadzsdp0f23e0uy", "allowance":{ "@type":"/cosmos.feegrant.v1beta1.BasicAllowance", "spend_limit":[ { "denom":"uscrt", "amount":"1000000" } ], "expiration":null } } }

```

Basic Keplr example

The following example will allow a user to unwrap sSCRT into SCRT using a feegrant on mainnet.

Prerequisites

Before moving forward, ensure that you have completed the following prerequisites:

- Install theKeplr
- extension on your browser
- Install thesecret.js
- library within your project
-

Execute transaction

1. Set the fee granter and contract addresses.
2.

```
```

Copy constsmartContractAddress="secret1k0jntykt7e4g3y88ltc60czgjuqdy4c9e8fzek";
constfeeGranterAddress="secret1tq6y8waegggp4fv2fcxk3zmpsmlfadyc7lsd69";

```

1. Define the transaction details which includes the message to the contract (MsgExecuteContract)
2. and sign and broadcast the transaction.
3.

```

Copy consttx=awaitsecretjs.tx .broadcast([newMsgExecuteContract({ sender:secretjs.address, contract_address:smartContractAddress, sent_funds:[], msg:{ redeem:{ amount, denom:"SCRT" } } }asany)], { gasLimit:150_000, gasPriceInFeeDenom:0.25, feeDenom:'uscrt', feeGranter:feeGranterAddress })

```

Last updated2 months ago On this page *Query a grant allowance * Basic Keplr example * Prerequisites * Execute transaction

Was this helpful? Edit on GitHub Export as PDF
```