Outside Execution

Enabling meta-transactions on Starknet

Outside Execution will work for everyone on Sepolia testnet but if you need to provide support for it in your dApp onmainnet , please reach out to us atecosystem@argent.xyz . Outside execution allows external contracts the ability to execute transactions from outside an account contract, thereby creating opportunities for meta-transactions, transaction scheduling (limit orders) etc.

While we wait for native paymasters on Starknet, there are certain use cases that requires giving external contracts access to execute transactions through an account contract. This is currently not possible due to the structure of theexecute entrypoint in existing account contracts.

In a bid to prevent re-entrancy attacks, theexecute entrypoint prevents calls from external contracts, which means in order to enable meta-transactions we need to introduce a new entrypoint that allows calls from external contracts,execute_from_outside .

```

Copy fnexecute_from_outside( refself:ContractState, outside_execution:OutsideExecution, signature:Array)->Array>

```

Studying the interface above, we notice theexecute_from_outside entrypoint requires two

paramenters:

- TheOutsideExecution
- struct
- A valid signature

In the next section, let's take a look at how you can execute "outside transactions" as an application builder.

Executing from Outside as a dApp developer

To get started executing outside transactions:

1. Build your
2. OutsideExecution
3. Struct

AnOutsideExecution struct contains the details of the transaction to be executed from outside.

```

Copy structOutsideExecution{ caller:ContractAddress, nonce:felt252, execute_after:u64, execute_before:u64, calls:Span}

```

For an outside execution transaction to be valid, it needs to contain acaller (the address allowed to call theexecute_from_outside), anonce (a unique value to prevent signature reuse), anexecute_after value (specifying the time after which the transaction can succeed), anexecute_before value (specifying the time before which the transaction should succeed), and finally an array of calls to be executed.

2.Sign it using EIP-712 typed data hashing

The signature signs over the EIP712 message encoding ofoutside_execution . Dapps are encouraged to request signatures following the EIP712 standard for a clearer UX.

Refer to the standardhere .

1. Call the
2. execute_from_outside
3. method on the account contract
4.

To do this we advise that you first verify that the account contract being interacted with has support for outside execution. To do this, simply call thesupports_interface method on the account contract:

```
```

Copy letacccount=IErc165Dispatcher{ contract_address:acount_address };
letis_supported=account.supports_interface(ERC165_OUTSIDE_EXECUTION_INTERFACE_ID);

```

The interface ID to be queried for is:0x68cfd18b92d1907b8ba3cc324900277f5a3622099431ea85dd8089255e4181 .

If it returns true, then you can go ahead to call theexecute_from_outside method:

```

Copy letacccount=IOutsideExecutionDispatcher{ contract_address:acount_address };

```

Previous Multicalls Next Deploy accounts on behalf of users

Last updated2 months ago
```