Session keys with outside execution

Outside Execution (Meta-transactions in the Ethereum world) allows external contracts to execute transactions from outside an account contract, thereby opening up new use-cases such as sponsored transactions, deploying an account on behalf of a user, transaction scheduling (limit orders) etc.

The setup is exactly the same as the "basic" session keys. We simply have a few more steps to do:

1. Prepare the contract call

```

Copy // example for creating the calldata const erc20Contract = new Contract( Erc20Abi as Abi, ETHTokenAddress, sessionAccount )

const calldata = erc20Contract.populate("transfer", { recipient: address, amount: parseInputAmountToUint256(amount) })

```

1. Prepare Execution from outside

Here you have two possibilities. Depending on your setup, you can chose between a "higher level" function withcreateOutsideExecutionCall which returns a signed call and a "lower level" withcreateOutsideExecutionTypedData which returns the session acout signature and the typed data.

a. Get the raw Execute from outside call

```

Copy import { createOutsideExecutionCall } from "@argent/x-sessions"

const efoExecutionCall = await createOutsideExecutionCall({ session, // same object as before sessionKey, // same object as before calls: [calldata], argentSessionServiceUrl: ARGENT_SESSION_SERVICE_BASE_URL, network: CHAIN_ID === constants.NetworkName.SN_SEPOLIA ? "sepolia" : "mainnet", })

```

2.b. Get the signed Execute from outside call

```

Copy import { createOutsideExecutionTypedData } from "@argent/x-sessions"

const { signature, outsideExecutionTypedData } = await createOutsideExecutionTypedData({ session, sessionKey, calls: [calldata], argentSessionServiceUrl: ARGENT_SESSION_SERVICE_BASE_URL network // values "mainnet" | "sepolia", default to "mainnet" })

```

Previous Implement session keys Next Demo App