

## AA23 reverted (or OOG)

OOG = Out Of Gas

The `validateUserOp` function of the smart account either reverted or ran out of gas.

## Possible solutions

- Verify that `theverificationGasLimit`
  - is high enough to cover `thevalidateUserOp`
  - function's gas costs.
- Verify that `thevalidateUserOp`
  - function is implemented with the correct logic, and that the user operation is supposed to be valid.
- If you aren't
  - using a paymaster, verify that `thesender`
    - address has enough native tokens to cover the required prefund. Consider leveraging functions like [getRequiredPrefund](#)
- .

...

```
const requiredPrefund = getRequiredPrefund({ userOperation })
```

```
const senderBalance = await publicClient.getBalance({ address: userOperation.sender });
```

```
if(senderBalance<requiredPrefund) { thrownewError(Sender address does not have enough native tokens ) }
```

...

- If you are
- looking to use a paymaster to cover the gas fees, verify that the `paymasterAndData`
- field is set.

...

[illegible]

- Make sure the signature verification logic of the smart account is implemented correctly. Namely, correct implementations should not
  - revert when the signature is invalid. If you are not looking to use neither time-based signatures nor a signature aggregator, you should simply return `uint(1)`
  - for invalid signatures in `thevalidateUserOp`
  - function.
  - If all else fails, investigate why `thevalidateUserOp`
  - function reverted using tools like [Tenderly](#)
  - .