

Error Handling

Similarly to how [error handling in viem works](#), every module in permissionless.js exports an accompanying error type which you can use to strongly type your catch statements.

These types come in the form of `ErrorType`. For example, the `estimateUserOperationGas` action exports an `EstimateUserOperationGasErrorType` type.

Unfortunately, [TypeScript doesn't have an abstraction for typed exceptions](#), so the most pragmatic & vanilla approach would be to explicitly cast error types in the catch statement.

example.ts client.ts ```

```
File example.ts import{ typeEstimateUserOperationGasErrorType, typeSenderAlreadyDeployedError, ENTRYPOINT_ADDRESS_V07 }from'permissionless' import{bundlerClient}from'./client'
```

```
try{ constestimateResult=awaitbundlerClient.estimateUserOperationGas({ userOperation, }) }catch(e) {
constestimationError=easEstimateUserOperationGasErrorType<"v0.7">
estimationError.name:"EstimateUserOperationGasError"}
```

```
consterror=estimationError.walk( (e)=>e instanceofSenderAlreadyDeployedError, )
if(error instanceofSenderAlreadyDeployedError) {
error.SenderAlreadyDeployedError.name:"SenderAlreadyDeployedError"}
```

```
}}
```

```
```
```