

Migration Guide

0.1.0

This migration guide is assuming you migrating to ^0.1.0, but you are still looking to use EntryPoint version v0.6.

Replaced transport with bundlerTransport inside createSmartAccountClient

```
const smartAccountClient = createSmartAccountClient({ transport: http(bundlerUrl), bundlerTransport: http(bundlerUrl), // ... })
```

Replaced sponsorUserOperation middleware API

Instead of accepting just asponsorUserOperation middleware, createSmartAccountClient now accepts amiddleware function that can specify asponsorUserOperation function internally, as well as a gasPrice function.

```
const smartAccountClient = createSmartAccountClient({ sponsorUserOperation: paymasterClient.sponsorUserOperation, middleware: { gasPrice: async () =>
return(await bundlerClient.getUserOperationGasPrice()).fast }, sponsorUserOperation: paymasterClient.sponsorUserOperation, // ... })
```

In addition, a function can also be passed to middleware directly.

```
const smartAccountClient=createSmartAccountClient({ middleware:async({userOperation,entryPoint})=>{ // ... return modifiedUserOperation } // ... })
```

sponsorUserOperation

`inipimlicoPaymasterActions` and `stackupPaymasterActions` now returns just the `paymasterAndData` and estimated gas limits instead of the whole user operation

[illegible]

ModifiedproviderToSmartAccountSigner

API to use keyword arguments

```
import { providerToSmartAccountSigner } from "permissionless"

const provider = // ... const signerAddress = // ...

const smartAccountSigner = await providerToSmartAccountSigner(provider, signerAddress)
const smartAccountSigner = await providerToSmartAccountSigner(provider, { signerAddress: signerAddress })
```

All clients must now take in anentryPoint

```

address
***

constbundlerClient=createBundlerClient({ transport:http(bundlerUrl), entryPoint:ENTRYPOINT_ADDRESS_V06, })

constsmartAccountClient=createSmartAccountClient({ account, entryPoint:ENTRYPOINT_ADDRESS_V06, chain: sepolia, bundlerTransport:http(bundlerUrl), })

constbundlerClient=createPimlicoBundlerClient({ transport:http(bundlerUrl), entryPoint:ENTRYPOINT_ADDRESS_V06, })

exportconstpaymasterClient=createPimlicoPaymasterClient({ transport:http(paymasterUrl), entryPoint:ENTRYPOINT_ADDRESS_V06, })

exportconstpaymasterClient=createStackupPaymasterClient({ transport:http(paymasterUrl), entryPoint:ENTRYPOINT_ADDRESS_V06, })

constbundlerClient=createClient({ transport:http(bundlerUrl), chain: sepolia, }) .extend(bundlerActions) .extend(bundlerActions(ENTRYPOINT_ADDRESS_V06))

```

entryPoint

is no longer passed in to bundler, paymaster, and smart account actions

```
const userOperationHash = await bundlerClient.sendUserOperation({ userOperation: sponsoredUserOperation, entryPoint: ENTRYPOINT_ADDRESS_V07, });
```