

# How to use a Fireblocks signer with permissionless.js

[Fireblocks](#) is a user-friendly platform designed for building blockchain-based products and managing digital asset operations. It uses a direct custody approach, combining high performance with zero counterparty risk and multi-layered security. The platform includes secure MPC-based digital asset wallets, a policy engine for governance and transaction rules, and comprehensive treasury management. Fireblocks' security framework features multiple layers, including MPC-CMP technology, secure enclaves, and a robust policy engine, ensuring protection against cyberattacks, internal threats, and human errors. It's widely used for various operations like treasury, trading, and managing NFTs, smart contracts, and user wallets.

## Setup

To use Fireblocks with permissionless.js, first create an application that integrates with Fireblocks.

- Refer to the [Fireblocks documentation site](#)
- for instructions on setting up an application with the Fireblocks.
- For a quick start, Fireblocks provides a guide, available [here](#)
- .

## Integration

Integrating permissionless.js with Fireblocks is straightforward after setting up the project. Fireblocks provides an Externally Owned Account (EOA) wallet to use as a signer with permissionless.js accounts.

### Create the Fireblocks object

After following the Fireblocks documentation, you will have access to a `FireblocksWeb3Provider` object as shown below which you can use to create a `SmartAccountSigner` object:

```
```
```

```
import{ ChainId, FireblocksWeb3Provider, typeFireblocksProviderConfig, }from"@fireblocks/fireblocks-web3-provider"
import{ providerToSmartAccountSigner }from"permissionless"
```

```
// Config options here will be specific to your project. See the Fireblocks docs for more info.
```

```
constfireblocksProviderConfig:FireblocksProviderConfig={ apiKey: process.env.FIREBLOCKS_API_KEY, privateKey:
process.env.FIREBLOCKS_API_PRIVATE_KEY_PATH, chainId: ChainId.SEPOLIA, }
constfireblocksWeb3Provider=newFireblocksWeb3Provider(fireblocksProviderConfig)
```

```
// Convert the FireblocksWeb3Provider to a SmartAccountSigner
```

```
constsmartAccountSigner=providerToSmartAccountSigner(fireblocksWeb3Provider)
```

```
```
```

### Use with permissionless.js

```
SimpleAccount Safe Account Kernel Account Biconomy Account ```
```

```
SimpleAccount import{ signerToSimpleSmartAccount }from"permissionless/accounts" import{ createPublicClient, http
}from"viem" import{ generatePrivateKey, privateKeyToAccount }from"viem/accounts" import{ sepolia }from"viem/chains"
```

```
exportconstpublicClient=createPublicClient({ transport:http("https://rpc.ankr.com/eth_sepolia"), chain: sepolia, })
```

```
constsmartAccount=awaitsignerToSimpleSmartAccount(publicClient, { signer: smartAccountSigner,
factoryAddress:"0x9406Cc6185a346906296840746125a0E44976454", entryPoint:ENTRYPOINT_ADDRESS_V06, })
```

```
```
```