

signerToSimpleSmartAccount

Creates a SimpleAccount instance controlled by aSmartAccountSigner .

Usage

```
example.ts publicClient.ts signer.ts ```
```

File example.ts import{ signerToSimpleSmartAccount }from"permissionless/accounts" import{ publicClient }from"./publicClient" import{ signer }from"./signer"

```
constsimpleAccount=awaitsignerToSimpleSmartAccount(publicClient, { signer,
factoryAddress:"0x9406Cc6185a346906296840746125a0E44976454",
entryPoint:"0x5FF137D4b0FDCD49DcA30c7CF57E578a026d2789", })
```

```
```
```

## Returns

- Type:
- SimpleSmartAccount

The smart account instance.

## Parameters

### signer

- Type:
- SmartAccountSigner

The signer that will be used to sign messages and user operations.

### factoryAddress

- Type:
- Address

The address of the simple account factory that will be used to deploy the smart account.

The canonical simple account factory for v0.6 of the EntryPoint is0x9406Cc6185a346906296840746125a0E44976454

### entryPoint

- Type:
- Address

The address of the EntryPoint contract.

### saltNonce (optional)

- Type:
- bigint

The salt nonce that will be used to deploy the smart account. If not provided,0 will be used.

### address (optional)

- Type:
- Address

If you provide an address, the smart account can not be deployed. This should be used if you want to interact with an existing smart account.

The address of the smart account. If not provided, the deterministic smart account address will be used.