

# signerToKernelSmartAccount

Creates a KernelAccount instance controlled by aSmartAccountSigner . Check out[this guide](#) for a complete tutorial.

## Usage

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

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

```
constkernelAccount=awaitsignerToEcdsaKernelSmartAccount(publicClient, { signer,
entryPoint:"0x5FF137D4b0FDCD49DcA30c7CF57E578a026d2789",// v0.6 entrypoint index:0n,// optional address:"0x..."//
optional, only if you are using an already created account });
```

```
```
```

## Returns

- Type:
- KernelSmartAccount

The Kernel smart account instance.

## Parameters

### signer

- Type:
- SmartAccountSigner

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

### entryPoint

- Type:
- Address

The address of the EntryPoint contract.

### index (optional)

- Type:
- bigint

The index (which is basically a salt) 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.