Custom Networks

Summary

The SDK supports network switching forEVM supported and custom networks. Configuring additional networks is done by providing theevmNetworks parameter inside theDynamicContextProvider settings

Setup

evmNetworks

Once you have determined the list of networks you want to support, you will need to pass an array ofEvmNetwork to theDynamicContextProvider settings. The reference for an EvmNetwork can be foundhere . The following example sets the Ethereum mainnet and Polygon as supported networks for the application. A comprehensive list of networks can be found atchainlist.org

```
// Setting up list of evmNetworks const evmNetworks= [ { blockExplorerUrls: [ 'https://etherscan.io/' ] , chainId: 1 ,
chainName: 'Ethereum Mainnet' , iconUrls: [ 'https://app.dynamic.xyz/assets/networks/eth.svg' ] , name: 'Ethereum' ,
nativeCurrency: { decimals: 18 , name: 'Ether' , symbol : 'ETH' , } , networkId: 1 , rpcUrls: [ 'https://mainnet.infura.io/v3/' ] ,
vanityName: 'ETH Mainnet' , } , { blockExplorerUrls: [ 'https://etherscan.io/' ] , chainId: 5 , chainName: 'Ethereum Goerli' ,
iconUrls: [ 'https://app.dynamic.xyz/assets/networks/eth.svg' ] , name: 'Ethereum' nativeCurrency: { decimals: 18 , name:
'Ether' , symbol : 'ETH' , } , networkId: 5 , rpcUrls: [ 'https://goerli.infura.io/v3/9aa3d95b3bc440fa88ea12eaa4456161' ] ,
vanityName: 'Goerli' , } , { blockExplorerUrls: [ 'https://polygonscan.com/' ] , chainId: 137 , chainName: 'Matic Mainnet' ,
iconUrls: [ "https://app.dynamic.xyz/assets/networks/polygon.svg" ] name: 'Polygon' , nativeCurrency: { decimals: 18 , name:
'MATIC' , symbol : 'MATIC' , } , networkId: 137 , rpcUrls: [ 'https://polygon-rpc.com' ] , vanityName: 'Polygon' , } , ] ; const
App = ( ) => ( < DynamicContextProvider settings= { { appLogoUrl:
'https://upload.wikimedia.org/wikipedia/commons/3/34/Examplelogo.svg' , appName: 'Example App' , environmentId:
'REPLACE_WITH_YOUR_ENV_ID' , evmNetworks, } }

    < Home/

    < / DynamicContextProvider> ) ; export default App;
```

Custom EVM Network

See theEvmNetwork reference for all props available to define a custom network.

You are now ready to switch networks!

Usage

Using theWalletConnector instance provided byuseDynamicContext , you have two useful methods for network switching

Method Description supportsNetworkSwitching(): boolean whether the connector supports network switching. switchNetwork({ networkChainId, networkName }: { networkChainId?: number; networkName?: string; }): Promise switch to another network by provider either the network name or chain id specified in the list of EvmNetwork When callingswitchNetwork with a connector supporting network switching, the SDK will either request the user to confirm the network switch or add the network if it was not previously set.

```
const { walletConnector} = useDynamicContext ( ) ; if ( walletConnector. supportsNetworkSwitching ( ) ) { await
walletConnector. switchNetwork ( { networkChainId: 137 } ) ; console . log ( "Success! Network switched" ) ; }
```

Was this page helpful?

Yes No Configure in Dashboard RPC Urls twitter linkedin slack Powered by Mintlify