

Configuring New Chains in the Hyperlane Explorer

Hyperlane can be [deployed](#) to any chain by anyone, this open access is called Permissionless Interoperability (PI). However the messages on PI chains cannot be identified by the default Hyperlane Agents. To view details about messages from PI chains, first configure the explorer with metadata about that chain.

Adding Chain Metadata to Hyperlane Explorer

1. On the Explorer's homepage, click either the Origin or Destination button to open the chain selection. This will open the chain selection modal. From there, click the plus icon in the top-right.
2. On the form that appears, enter the chain's metadata using JSON or YAML conforming to the [ChainMetadata schema](#)
3. .
4. If a valid Etherscan-based block explorer config is provided, the Hyperlane Explorer will utilize it to find the desired messages. If not, it will use the RPC URL. Note, Explorers with API keys (even just free-tier keys), perform faster and more reliably.

Chain Metadata Examples

A minimal example in JSON:

```
{ "chainId": 11155111, "name": "sepolia", "protocol": "ethereum", "rpcUrls": [{ "http": "https://foobar.com" }] }
```

 A more detailed example in YAML:

Yaml metadata example

blockExplorers : -

apiUrl : https : //api - sepolia.etherscan.io/api family : etherscan name : Etherscan url : https : //sepolia.etherscan.io blocks : confirmations :

1 estimateBlockTime :

13 reorgPeriod :

2 chainId :

11155111 deployer : name : Abacus Works url : https : //www.hyperlane.xyz displayName : Sepolia domainId :

11155111 isTestnet :

true name : sepolia nativeToken : decimals :

18 name : Ether symbol : ETH protocol : ethereum rpcUrls : -

http : https : //ethereum - sepolia.publicnode.com -

http : https : //ethereum - sepolia.blockpi.network/v1/rpc/public info If the origin or destinationdomainId of chains in your messages doesn't match theirchainId then you must include thedomainId field in your chain config. [Edit this page](#) [Previous](#) [How to Use the Hyperlane Explorer](#) [Next GraphQL API](#)