**EVM-based Chains**

The source data is in \_data/chains. Each chain has its own file with the filename being the [CAIP-2](https://github.com/ChainAgnostic/CAIPs/blob/master/CAIPs/caip-2.md) representation as name and .json as extension.

**Example:**

{

"name": "Ethereum Mainnet",

"chain": "ETH",

"rpc": [

"https://mainnet.infura.io/v3/${INFURA\_API\_KEY}",

"https://api.mycryptoapi.com/eth"

],

"faucets": [],

"nativeCurrency": {

"name": "Ether",

"symbol": "ETH",

"decimals": 18

},

"features": [{ "name": "EIP155" }, { "name": "EIP1559" }],

"infoURL": "https://ethereum.org",

"shortName": "eth",

"chainId": 1,

"networkId": 1,

"icon": "ethereum",

"explorers": [{

"name": "etherscan",

"url": "https://etherscan.io",

"icon": "etherscan",

"standard": "EIP3091"

}]

}

When an icon is used in either the network or an explorer, there must be a JSON in \_data/icons with the name used. (e.g. in the above example there must be a ethereum.json and a etherscan.json in there) - The icon JSON files look like this:

[

{

"url": "ipfs://QmdwQDr6vmBtXmK2TmknkEuZNoaDqTasFdZdu3DRw8b2wt",

"width": 1000,

"height": 1628,

"format": "png"

}

]

where:

* The URL MUST be publicly resolvable through IPFS
* width and height MUST be positive integers
* format is either "png", "jpg" or "svg"
* size MUST be less than 250kb

If the chain is an L2 or a shard of another chain you can link it to the parent chain like this:

{

...

"parent": {

"type" : "L2",

"chain": "eip155-1",

"bridges": [ {"url":"https://bridge.arbitrum.io"} ]

}

}

where you need to specify the type and the reference to an existing parent. The field about bridges is optional.

You can add a status field e.g. to deprecate (via status deprecated) a chain (a chain should never be deleted as this would open the door to replay attacks) Other options for status are active (default) or incubating

**Aggregation**

There are also aggregated json files with all chains automatically assembled:

* <https://chainid.network/chains.json>
* <https://chainid.network/chains_mini.json> (miniaturized - fewer fields for smaller filesize)

**Constraints**

* the shortName and name MUST be unique - see e.g. EIP-3770 on why
* if referencing a parent chain - the chain MUST exist in the repo
* if using an IPFS CID for the icon - the CID MUST be retrievable via ipfs get - not only through some gateway (means please do not use pinata for now)
* for more constraints you can look into the CI

**Collision management**

We cannot allow more than one chain with the same chainID - this would open the door to replay attacks. The first pull request gets the chainID assigned. When creating a chain we can expect that you read EIP155 which states this repo. All pull requests trying to replace a chainID because they think their chain is better than the other will be closed. The only way to get a chain reassigned is when the old chain gets deprecated. This can e.g. be used for testnets that are short-lived. But then you will get the redFlag "reusedChainID" that should be displayed in clients to warn them about the dangers here.

**Getting your PR merged**

**before PR is submitted**

Before submitting a PR, please ensure all checks pass by running:

$ ./gradlew run

BUILD SUCCESSFUL in 7s

9 actionable tasks: 9 executed

Additionally, run Prettier to format your JSON according to the style [defined here](https://github.com/ethereum-lists/chains/blob/master/.prettierrc.json)e.g. run

npx prettier --write \_data/\*/\*.json

**Once PR is submitted**

* Make sure CI is green. There will likely be no review when the CI is red.
* When making changes that fix the CI problems - please re-request a review - otherwise it is too much work to track such changes with so many PRs daily

**Usages**

**Tools**

* [MESC](https://paradigmxyz.github.io/mesc)

**Explorers**

* [Otterscan](https://otterscan.io/)

**Wallets**

* [WallETH](https://walleth.org/)
* [TREZOR](https://trezor.io/)
* [Minerva Wallet](https://minerva.digital/)

**EIPs**

* EIP-155
* EIP-3014
* EIP-3770
* EIP-4527

**Listing sites**

* [chainid.network](https://chainid.network/) / [chainlist.wtf](https://chainlist.wtf/)
* [chainlist.org](https://chainlist.org/)
* [Chainlink docs](https://docs.chain.link/)
* [dRPC Chainlist - Load-balanced public nodes](https://drpc.org/chainlist)
* [eth-chains](https://github.com/taylorjdawson/eth-chains)
* [EVM-BOX](https://github.com/izayl/evm-box)
* [evmchain.info](https://evmchain.info/)
* [evmchainlist.org](https://evmchainlist.org/)
* [networks.vercel.app](https://networks.vercel.app/)
* [Wagmi compatible chain configurations](https://spenhouet.com/chains)
* [chainlist.simplr.sh - Info packaged single pager](https://chainlist.simplr.sh/)

**Other**

* [FaucETH](https://github.com/komputing/FaucETH)
* [Sourcify playground](https://playground.sourcify.dev/)
* [Smart Contract UI](https://xtools-at.github.io/smartcontract-ui)
* Your project - contact us to add it here!