check-availability)

- Check requirements
- Registration
- Transaction Status

Communities

Currently only available on Sepolia and Holesky as we roll out the feature

Check availability

Check the availability of a single community cluster

GET /check/availability/:clusterName/:walletName

Example

```
Single check Copy curlhttps://api.clusters.xyz/v0.1/community/check/availability/mycommunity/foobar?testnet=true
```

Response

Single Check
```
Copy { "name":"mycommunity/foobar", "isAvailable":true }
```

Check requirements

Check if the community requirements for a specific wallet has been met for registration

GET /check/requirements/:clusterName?walletAddress=:walletName

Example

```
Copy curlhttps://api.clusters.xyz/v0.1/community/check/requirements/mycommunity?walletAddress=0x123?testnet=true
```

Response

```
Copy { "clusterName":"mycommunity", "walletAddress":"0x123", "isRequirementsMet":true }
```

Registration

POST /v0.1/community/register

Either registers a free community clusters OR returns transaction data for registrations that require payment.

Name Description HEADER(bearer) Authorization Authentication key from wallet clusterName The cluster name of the community being registered walletName The wallet name being registered under the community chainId (optional) If the price of a community cluster is USDC, a chainId should be passed so that the correct contracts are used when returning transaction data. It will default to mainnet. walletAddress (optional) If the registration sign up's are restricted to community owners, a wallet address needs to be included to know who to register the name to. Free EVM payment Solana Payment Error handling Example

```
Copy curl--requestPOST\ --url'https://api.clusters.xyz/v0.1/community/register?testnet=true'\ --header'Content-Type:
```

```
application/json'\ --header'Authorization: Bearer abc123'\ --data'{ "clusterName": "mycommunity", "walletName": "foobar" }'
```

Response

```
Copy { result:"success"}
```

Example

```
Copy curl--requestPOST\ --url'https://api.clusters.xyz/v0.1/community/register?testnet=true'\ --header'Content-Type: application/json'\ --header'X-AUTH-KEY: abc123'\ --data'{ "clusterName": "mycommunity", "walletName": "foobar", "chainId": "11155111" }'
```

Response

```
Copy { "clusterName": string; "walletName": string; "transactionInfo": { "type":"evm", "transaction":{ "to":string; "data"0x{string}; "value": string; } } }
```

Usage

```
Copy import{ createWalletClient,custom }from'viem' import{ sepolia }from'viem/chains'

consturl='https://api.clusters.xyz/v0.1/community/register?testnet=true'; constgetRegister=awaitfetch(url,{ method:'POST', headers:{ 'Content-type':'application/json', 'X-AUTH-KEY':'abc123' }, data:JSON.stringify({ clusterName,walletName }) }) constregister=awaitgetRegister.json();

constwalletClient=createWalletClient({ chain:sepolia, transport:custom(provider), });

const{to,data,value}=register.transactionInfo.transaction; consttx=awaitwalletClient.sendTransaction({ account, to, data, value });
```

Example

```
Copy curl--requestPOST\ --url'https://api.clusters.xyz/v0.1/community/register?testnet=true'\ --header'Content-Type: application/json'\ --header'X-AUTH-KEY: abc123'\ --data'{ "clusterName": "mycommunity", "walletName": "foobar", "chainId": "11155111" }'
```

Response

```
Copy { "clusterName": string; "walletName": string; "transactionInfo": { "type":"solana", "transaction":{ "data":string;// base64 string of serialized transactions } } }
```

Usage

```
Copy import*asweb3from'@solana/web3.js';

constmessage=web3.VersionedMessage.deserialize( Buffer.from(register.transactionInfo.transaction,'base64'), ); consttransaction=newweb3.VersionedTransaction(message); const{signature}=awaitprovider.signAndSendTransaction( transaction, );
```

``` Testnet only - We are in beta only phase of community clusters. You'll receive this error if testnet is not defined

```

Copy { error:"testnet_only"}

```

Community not found - If the specific cluster name is not a community

```

Copy { error: "not_found" }

```

Not a verified user - If sign ups are restricted to community owners and the auth key is not from an owner

```

Copy { error: "not_verified" }

```

No wallet address found - If sign ups are restricted to community owners and no wallet address was included in the body of the request

```

Copy { error: "no_walletAddress_found"}

```

Registration has not started yet - If sign ups for this community have not started yet

```

Copy { error: "not_active" }

```

Name is not available - If the name is taken

```

Copy { error: "not_available" }

```

Requirements have not been met - If the user registering the name has not met the requirements

```

Copy { error: "requirements_not_met" }

```

Invalid chain id - If the chain submitted in the request is invalid

```

Copy { error: "invalid_chain_id" }

```

Transaction Status

Checks the status of a transaction

GET /v0.1/community/tx/:tx

Example

```

```
Copy curlhttps://api.clusters.xyz/v0.1/community/tx/0x123?testnet=true
```

Response

```
Copy typestatus='not_found'|'pending'|'bridging'|'invalid'|'lost_bid'|'finalized'
```

200

```
Copy { "tx":"0x123", "status":"not_found" }
```

Previous Managing Wallets Next Data Availability Last updated1 day ago