Set up MetaMask

1. Connect MetaMask to other EVM chains
2. Get testnet tokens for other EVM chains to pay for gas
3. Import Axelar ERC20 tokens on other EVM chains
4. Enable hex data in transactions

Connect MetaMask to other EVM chains

In order to complete exercises for an EVM chain[chain] you need to connect your MetaMask to[chain] .

Open MetaMask. In the "Networks" dropdown list choose "Add Network". Enter the data for your desired[chain] below and click "Save". Repeat for any chains you like.

Mainnet

EVM chain Chain Name Chain ID Native Token RPC URL Explorer URL Add Chain Ethereum Ethereum 1 ETHURL URL astro-island,astro-slot,astro-static-slot{display:contents} (()=>{var e=async t=>{await(await t())()};(self.Astro||(self.Astro={})).only=e;window.dispatchEvent(new Event("astro:only"));})();;(()=>{var b=Object.defineProperty;var f=(a,s,i)=>s in a?b(a,s,{enumerable:!0,configurable:!0,writable:!0,value:i}):a[s]=i;var d=(a,s,i)=>(f(a,typeof s!="symbol"?s+"":s,i),i);var u;{let a={0:t=>m(t),1:t=>i(t),2:t=>new RegExp(t),3:t=>new Date(t),4:t=>new Map(i(t)),5:t=>new Set(i(t)),6:t=>BigInt(t),7:t=>new URL(t),8:t=>new Uint8Array(t),9:t=>new Uint16Array(t),10:t=>new Uint32Array(t)},s=t=>{let[e,r]=t;return e in a?ae:void 0},i=t=>t.map(s),m=t=>typeof t!="object"||t===null?t:Object.fromEntries(Object.entries(t).map(([e,r])=>[e,s(r)]));customElements.get("astro-island")||customElements.define("astro-island",(u=class extends HTMLElement{constructor(){super(...arguments);d(this,"Component");d(this,"hydrator");d(this,"hydrate",async()=>{var l;if(!this.hydrator||!this.isConnected)return;let e=(l=this.parentElement)==null?void 0:l.closest("astro-island[ssr]");if(e){e.addEventListener("astro:hydrate",this.hydrate,{once:!0});return}let r=this.querySelectorAll("astro-slot"),c={},h=this.querySelectorAll("template[data-astro-template]");for(let n of h){let o=n.closest(this.tagName);o!=null&&o.isSameNode(this)&&(c[n.getAttribute("data-astro-template")||"default"]=n.innerHTML,n.remove())}for(let n of r){let o=n.closest(this.tagName);o!=null&&o.isSameNode(this)&&(c[n.getAttribute("name")||"default"]=n.innerHTML)}let p;try{p=this.hasAttribute("props")?m(JSON.parse(this.getAttribute("props"))):{}}catch(n){let o=this.getAttribute("component-url")||"",y=this.getAttribute("component-export");throw y&&(o+=(export {y})),console.error([hydrate] Error parsing props for component {o},this.getAttribute("props"),n),n}await this.hydrator(this)(this.Component,p,c,{client:this.getAttribute("client")}),this.removeAttribute("ssr"),this.dispatchEvent(new CustomEvent("astro:hydrate"))});d(this,"unmount",()=>{this.isConnected||this.dispatchEvent(new CustomEvent("astro:unmount"))})}disconnectedCallback(){document.removeEventListener("astro:after-swap",this.unmount),document.addEventListener("astro:after-swap",this.unmount,{once:!0})}connectedCallback(){!this.hasAttribute("await-children")||this.firstChild?this.childrenConnectedCallback():new MutationObserver((e,r)=>{r.disconnect(),setTimeout(()=>this.childrenConnectedCallback(),0)}).observe(this,{childList:!0})}async childrenConnectedCallback(){let e=this.getAttribute("before-hydration-url");e&&await import(e),this.start()}start(){let e=JSON.parse(this.getAttribute("opts")),r=this.getAttribute("client");if(Astro[r]===void 0){window.addEventListener(astro:{r},()=>this.start(),{once:!0});return}Astror,[h,{default:p}]=await Promise.all([import(this.getAttribute("component-url")),c?import(c):()=>()=>{}]),l=this.getAttribute("component-export")||"default";if(!l.includes("."))this.Component=h[l];else{this.Component=h;for(let n of l.split("."))this.Component=this.Component[n]}return this.hydrator=p,this.hydrate},e,this)}attributeChangedCallback(){this.hydrate()}},d(u,"observedAttributes",["props"]),u))}})(); Avalanche Avalanche 43114 AVAX URL URL Fantom Fantom 250 FTM URL URL Polygon Polygon 137 MATIC URL URL Moonbeam Moonbeam 1284 GLMR URL URL

Testnet

EVM chain Chain Name Chain ID Native Token RPC URL Explorer URL Add Chain Ethereum Ethereum Goerli 5 ETHURL URL Avalanche Avalanche Fuji 43113 AVAX URL URL Fantom Fantom Testnet 4002 FTM URL URL Polygon Polygon Mumbai 80001 MATIC URL URL Moonbeam Moonbase Alpha 1287 DEV URL URL

Get testnet tokens for EVM chains

You need native tokens for each[chain] in order to pay transaction fees (gas) on[chain] .

You can get native tokens from a faucet. Search the internet for "[chain] testnet faucet" or use the links below.

- Ethereum
- Avalanche
- Fantom
- Moonbeam
- — No known web faucet; need to join theMoonbeam discord
- .

- Polygon

Import Axelar ERC20 tokens

Tokens transferred to an EVM chain using Axelar are not visible in MetaMask until you import them.

1. Use the “Networks” dropdown list, select your desired[chain]
2. .
3. View “Assets” and select “Import Tokens”.
4. Paste into “Token Contract Address” the ERC20 address for the token. (“Token symbol” and “token decimal” should be fetched automatically.)

Axelar token contract addresses for each[chain] can be found atTestnet resources .

Enable hex data in transactions

Some advanced exercises require you to send a transaction with hex data from MetaMask. The “hex data” field is invisible by default. Edit your settings to make it visible.

Accounts dropdown list -> Settings -> Advanced -> Show Hex Data, switch on.

Edit this page