Getting Started with SecretJS

Learn how to install and use SecretJS.

Overview

In this tutorial you will learn how to use Secret.js to connect a web3 wallet, upload a contract, instantiate a contract, execute a contract, and query a contract, amongst other use cases! For the complete Secret.js docs, seehere .

All of the following examples use thepulsar-3 testnet with LCD endpointhttps://api.pulsar3.scrttestnet.com .

Public LCD endpoints can be foundhere for both mainnet and testnet.

Installation

Create a package.json file:

```

Copy npminit-y

```

Then install secretJS:

```

Copy npminstallsecretjs

```

Set up Secret Network Client

```

Copy import{ SecretNetworkClient,Wallet }from"secretjs";

constwallet=newWallet("Your mnemonic words go here");

constsecretjs=newSecretNetworkClient({ chainId:"pulsar-3", url:"https://api.pulsar3.scrttestnet.com", wallet:wallet, walletAddress:wallet.address, });

```

Upload a Contract

```

Copy import{ SecretNetworkClient,Wallet }from"secretjs"; import*asfsfrom"fs";

constwallet=newWallet("Your mnemonic words go here");

constcontract_wasm=fs.readFileSync("../contract/contract.wasm");

constsecretjs=newSecretNetworkClient({ chainId:"pulsar-3", url:"https://api.pulsar3.scrttestnet.com", wallet:wallet, walletAddress:wallet.address, });

letupload_contract=async()=>{ lettx=awaitsecretjs.tx.compute.storeCode( { sender:wallet.address, wasm_byte_code:contract_wasm, source:"", builder:"", }, { gasLimit:4_000_000, } );

constcodeId=Number( tx.arrayLog.find((log)=>log.type==="message"&&log.key==="code_id") .value );

console.log("codeId: ",codeId);

constcontractCodeHash=( awaitsecretjs.query.compute.codeHashByCodeId({ code_id:codeId }) ).code_hash; console.log(Contract hash:{contractCodeHash}); };

upload_contract();

```

Instantiate a Contract

```

Copy import{ SecretNetworkClient,Wallet }from"secretjs"; import*asfsfrom"fs";

constwallet=newWallet("Your mnemonic words go here");

constcontract_wasm=fs.readFileSync("../contract/contract.wasm");

constsecretjs=newSecretNetworkClient({ chainId:"pulsar-3", url:"https://api.pulsar3.scrttestnet.com", wallet:wallet, walletAddress:wallet.address, });

// Add your contract codeId here letcodeId=""

// Add your contractCodeHash here letcontractCodeHash=""

letinstantiate_contract=async()=>{ //instantiate message is empty in this example. If your contract needs to be instantiated with additional variables, be sure to include them.

constinitMsg={}; lettx=awaitsecretjs.tx.compute.instantiateContract( { code_id:codeId, sender:wallet.address, code_hash:contractCodeHash, init_msg:initMsg, label:"Demo"+Math.ceil(Math.random()*10000), }, { gasLimit:400_000, } );

//Find the contract_address in the logs constcontractAddress=tx.arrayLog.find( (log)=>log.type==="message"&&log.key==="contract_address" ).value;

console.log(contractAddress); };

instantiate_contract();

```

Last updated3 months ago On this page *Overview * Installation * Set up Secret Network Client * Upload a Contract * Instantiate a Contract

Was this helpful? Edit on GitHub Export as PDF