Contract

When you instantiate an instance ofContract you need to specify the names of the functions you have on your smart contract. Then the new instance ofContract will have methods with the same names as your smart contract functions. For example if you deployed a contract withmy_smart_contract_function function on it, then this will work:

const contract =

new

Contract (account ,

"example-contract.testnet" ,

{ changeMethods :

["my_smart_contract_function"] ,

// your smart-contract has a function my_smart_contract_function }) ; //contract object has my_smart_contract_function function on it: contract . my_smart_contract_function ( ) ;

Load Contract

- Standard
- Using Wallet

const

{

Contract

}

= nearAPI ;

const contract =

new

Contract (account ,

// the account object that is connecting "example-contract.testnet" , { // name of contract you're connecting to viewMethods :

["getMessages"] ,

// view methods do not change state but usually return a value changeMethods :

["addMessage"] ,

// change methods modify state }) ;ClassContract const

{

Contract

}

= nearAPI ;

const contract =

new

Contract (wallet . account ( ) ,

// the account object that is connecting "example-contract.testnet" , { // name of contract you're connecting to viewMethods :

["getMessages"] ,

// view methods do not change state but usually return a value changeMethods :

["addMessage"] ,

// change methods modify state }) ;ClassContract

Call Contract

- Change Method
- Change Method w/ callbackUrl and meta
- View Method
- View Method w/ args

const contract =

new

Contract (account ,

"example-contract.testnet" ,

{ changeMethods :

["method_name"] , }) ; await contract . method_name ({ arg_name :

"value" ,

// argument name and value - pass empty object if no args required } , "300000000000000" ,

// attached GAS (optional) "100000000000000000000000"

// attached deposit in yoctoNEAR (optional)) ; const contract =

new

Contract (account ,

"example-contract.testnet" ,

{ changeMethods :

["method_name"] , }) ; await contract . method_name ({ callbackUrl :

"https://example.com/callback" ,

// callbackUrl after the transaction approved (optional) meta :

"some info" ,

// meta information NEAR Wallet will send back to the application.meta will be attached to thecallbackUrl as a url param args :

{ arg_name :

"value" ,

// argument name and value - pass empty object if no args required } , gas :

300000000000000 ,

// attached GAS (optional) amount :

100000000000000000000000 ,

// attached deposit in yoctoNEAR (optional) }) ; const contract =

new

Contract (account ,

"example-contract.testnet" ,

{ viewMethods :

["view_method_name"] , }) ; const response =

await contract . view_method_name ( ) ; const contract =

new

Contract (account ,

"example-contract.testnet" ,

{ viewMethods :

["view_method_name"] , }) ; const response =

await contract . view_method_name ({

arg_name :

"arg_value"

}) ;ClassContract Edit this page Last updatedonJan 31, 2024 bygagdiez Was this page helpful? Yes No

Previous Account Next Utilities