Frontend Interacting with Multiple Contracts

This example showcases how to interact with multiple contracts from a single frontend.

Particularly, this example shows how to:

1. Query data from multiple contracts.
2. Call methods in multiple contracts simultaneously.

Query Data from Multiple Contracts

To query multiple contracts simply perform multipleview calls:

frontend/index.js loading ... [See full example on GitHub](#)

Dispatching Multiple Transactions

Thewallet object enables to dispatch multiple transactions simultaneously. However, please notice that the transactions execute independently.

Dispatching multiple transactions at once is just a nice way to improve UX, because the user interacts with the wallet only once.

frontend/index.js loading ... [See full example on GitHub](#) In this example, the user signs two independent transactions:

1. A transaction to callset_greeting
2. in our[Hello NEAR example](#)
3. A transaction to calladd_message
4. in our[GuestBook example](#)

caution Even when the user accepts signing the transactions at the same time, the transactions remainindependent . This is, if one fails, the other isNOT rolled back.

Batch Actions

You can aggregate multiple[actions](#) directed towards a same contract into a single transaction. Batched actions executesequentially , with the added benefit that, ifone fails then theyall get reverted.

// Register a user and transfer them FT on a single take const

REGISTER_DEPOSIT

=

"1250000000000000000000" ;

const ftTx =

{ receiverId :

FT_ADDRESS , actions :

[{ type :

'FunctionCall' , params :

{ methodName :

'storage_deposit' , args :

{

account_id :

""

} , gas :

THIRTY_TGAS ,

```
deposit :

REGISTER_DEPOSIT } } , { type :

'FunctionCall' , params :

{ methodName :

'ft_transfer' , args :

{

receiver_id :

"" ,

amount : amount_in_yocto } , gas :

THIRTY_TGAS ,

deposit :

1

} } ] }

// Ask the wallet to sign and send the transaction await wallet . signAndSendTransactions ( {

transactions :

[ ftTx ]

} )
```

Edit this page Last updatedonFeb 9, 2024 bygagdiez Was this page helpful? Yes No