# Reference

The [API Kit(opens in a new tab)](#) facilitates the interaction with the [Safe Transaction Service API(opens in a new tab)](#) .

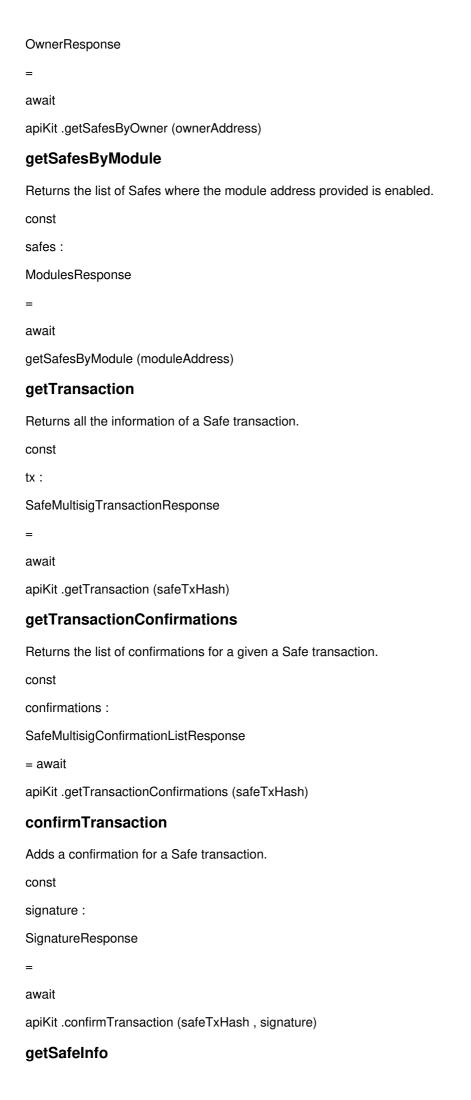## Install dependencies

To add the API Kit to your project, run:

yarn

add

@safe-global/api-kit

## Reference

### getServiceInfo

Returns the information and configuration of the service.

const

serviceInfo :

SafeServiceInfoResponse

=

await

apiKit .getServiceInfo ()

### getServiceSingletonsInfo

Returns the list of Safe singletons.

const

singletons :

SafeSingletonResponse

=

await

apiKit .getServiceSingletonsInfo ()

### decodeData

Decodes the specified Safe transaction data.

const

decodedData

=

await

apiKit .decodeData (data)

### getSafesByOwner

Returns the list of Safes where the address provided is an owner.

const

safes :

OwnerResponse

=

await

apiKit .getSafesByOwner (ownerAddress)

## getSafesByModule

Returns the list of Safes where the module address provided is enabled.

const

safes :

ModulesResponse

=

await

getSafesByModule (moduleAddress)

## getTransaction

Returns all the information of a Safe transaction.

const

tx :

SafeMultisigTransactionResponse

=

await

apiKit .getTransaction (safeTxHash)

## getTransactionConfirmations

Returns the list of confirmations for a given a Safe transaction.

const

confirmations :

SafeMultisigConfirmationListResponse

= await

apiKit .getTransactionConfirmations (safeTxHash)

## confirmTransaction

Adds a confirmation for a Safe transaction.

const

signature :

SignatureResponse

=

await

apiKit .confirmTransaction (safeTxHash , signature)

## getSafeInfo

Returns the information and configuration of the provided Safe address.

```
const
safeInfo :
SafeInfoResponse
=
await
apiKit .getSafeInfo (safeAddress)
```

## getSafeDelegates

Returns the list of delegates for a given Safe address.

```
const
delegateConfig :
GetSafeDelegateProps
= { safeAddress ,
// Optional delegateAddress ,
// Optional delegatorAddress ,
// Optional label ,
// Optional limit ,
// Optional offset // Optional } const
delegates :
SafeDelegateListResponse
=
await
apiKit .getSafeDelegates (delegateConfig)
```

## addSafeDelegate

Adds a new delegate for a given Safe address.

```
const
delegateConfig :
AddSafeDelegateProps
= { safeAddress ,
// Optional delegateAddress , delegatorAddress , label , signer } await
apiKit .addSafeDelegate (delegateConfig)
```

## removeSafeDelegate

Removes a delegate for a given Safe address.

```
const
delegateConfig :
DeleteSafeDelegateProps
= { delegateAddress , delegatorAddress , signer } await
```

```
apiKit .removeSafeDelegate (delegateConfig)
```

## getSafeCreationInfo

Returns the creation information of a Safe.

```
const

safeCreationInfo :

SafeCreationInfoResponse

=

await

apiKit .getSafeCreationInfo ( safeAddress )
```

## estimateSafeTransaction

Estimates the safeTxGas for a given Safe multi-signature transaction.

```
const

estimateTx :

SafeMultisigTransactionEstimateResponse

= await

apiKit .estimateSafeTransaction (safeAddress , safeTransaction)
```

## proposeTransaction

Creates a new multi-signature transaction and stores it in the Safe Transaction Service.

```
const

transactionConfig :

ProposeTransactionProps

= { safeAddress , safeTxHash , safeTransactionData , senderAddress , senderSignature , origin } await

apiKit .proposeTransaction (transactionConfig)
```

## getIncomingTransactions

Returns the history of incoming transactions of a Safe account.

```
const

incomingTxs :

TransferListResponse

=

await

apiKit .getIncomingTransactions (safeAddress)
```

## getModuleTransactions

Returns the history of module transactions of a Safe account.

```
const

moduleTxs :

SafeModuleTransactionListResponse
```

```
=
await
apiKit .getModuleTransactions ( safeAddress )
```

## getMultisigTransactions

Returns the history of multi-signature transactions of a Safe account.

```
const
multisigTxs :
SafeMultisigTransactionListResponse
=
await
apiKit .getMultisigTransactions ( safeAddress )
```

## getPendingTransactions

Returns the list of multi-signature transactions that are waiting for the confirmation of the Safe owners.

```
const
pendingTxs :
SafeMultisigTransactionListResponse
=
await
apiKit .getPendingTransactions ( safeAddress )
const
pendingTxs :
SafeMultisigTransactionListResponse
=
await
apiKit .getPendingTransactions ( safeAddress , currentNonce )
```

## getAllTransactions

Returns a list of transactions for a Safe. The list has different structures depending on the transaction type.

```
const
allTxs :
SafeMultisigTransactionListResponse
=
await
apiKit .getAllTransactions ( safeAddress )
const
allTxsOptions :
AllTransactionsOptions
= { executed , queued , trusted } const
```

```
allTxs :

SafeMultisigTransactionListResponse

=

await

apiKit .getAllTransactions ( safeAddress , allTxsOptions )
```

## getNextNonce

Returns the right nonce to propose a new transaction right after the last pending transaction.

```
const

nextNonce

=

await

apiKit .getNextNonce (safeAddress)
```

## getTokenList

Returns the list of all the ERC20 tokens handled by the Safe.

```
const

tokens :

TokenInfoListResponse

=

await

apiKit .getTokenList ()
```

## getToken

Returns the information of a given ERC20 token.

```
const

token :

TokenInfoResponse

=

await

apiKit .getToken (tokenAddress)
```

## getMessage

Returns a message from its hash.

```
const

message :

SafeMessage

=

await

apiKit .getMessage (safeMessageHash)
```

## getMessages

Returns the list of messages associated to a Safe account.

```
const messagesResponse: SafeMessageListResponse = await apiKit.getMessages(safeAddress, { ordering, limit, offset })
```

## addMessage

Creates a new message with an initial signature.

```
await apiKit.addMessage(safeAddress, { message, signature, safeAppId })
```

## addMessageSignature

Adds a signature to an existing message.

```
await apiKit.addMessageSignature(safeMessageHash, signature)
```

[API Kit](#) [Migrating to V1](#)

Was this page helpful?

[Report issue](#)