

Querying Sui RPC with GraphQL (Alpha)

Refer to [Access Sui Data](#) for an overview of options to access Sui network data.

Based on valuable feedback from the community, the GraphQL RPC release stage has been updated from beta to alpha. Refer to the high-level timeline for beta and GA releases in the previously linked document.

The quickest way to access the GraphQL service for Sui RPC is through the online IDE that provides a complete toolbox for fetching data and executing transactions on the network. The online IDE provides features such as auto-completion (use Ctrl+Space or just start typing), built-in documentation (Book icon, top-left), multi-tabs, and more.

The online IDE is available for [Mainnet](#) and [Testnet](#). This guide contains various queries that you can try directly in the IDE.

For more details about some concepts used in the examples below, please see the [GraphQL concepts](#) page, and consult the [reference](#) for full documentation on the supported schema.

GraphQL introspection exposes the schema supported by the RPC service. The IDE's "Docs" pane (Book icon, top-left) and Search dialog (Cmd + K on macOS or Ctrl + K on Windows and Linux) offer a way to browse introspection output interactively.

The [official documentation](#) provides an overview on introspection, and how to interact with it programmatically.

This example finds the total stake rewards, the reference gas price, the number of checkpoints and the total gas fees for epoch 100. Note that in the query, the id argument is optional, and defaults to the latest epoch.

This example gets a transaction block by its digest and shows information such as the gas sponsor's address, the gas price, the gas budget, and effects from executing that transaction block.

This example finds all the transactions that touched (modified/transferred/deleted) a given object. This is useful for when we want to trace the flow of a Coin/StakeSui/NFT.

This example uses GraphQL [variables](#) and [pagination](#). When using the online IDE, copy the variables JSON to the "Variables" window, below the main editor.

Variables :

This example finds the last ten transaction blocks that called the public_transfer function, (as a move call transaction command).

This example makes usage of the filter last, which indicates that the user only wants the last ten transaction blocks known to the service.

This example finds the balance changes of all the transactions where a given address called a staking-related function. This is useful when you want to get your staking or unstaking history.

Variables :

This example uses aliases and [fragments](#).

This query can be used to paginate over the dynamic fields of an object. This works even when the object in question is [wrapped](#), by using the [owner](#) query, so can be used for iterating over the elements of on-chain data structures, like tables and bags. See [The Move Book](#) to learn more about dynamic collections available in Move.

This example uses [fragments](#) and [variables](#).

Sets up a paginated query, starting at the genesis checkpoint, reading five checkpoints at a time, in increasing order of sequence number. The value of pageInfo.hasNextPage determines whether there is another page to be read, and the value of pageInfo.endCursor is fed back in as the cursor to read \$after.

This example uses GraphQL [variables](#) and [pagination](#).

Sets up a paginated query, starting at the latest indexed checkpoint, reading five checkpoints at a time, in decreasing order of sequence number. The value of pageInfo.hasPreviousPage determines whether there is another page to be read, and the value of pageInfo.startCursor is fed back in as the cursor to read \$before.

This example uses GraphQL [variables](#) and [pagination](#).

Transaction execution takes in two arguments, txBytes and signatures . txBytes is the serialized unsigned transaction data, which can be generated when using the Sui CLI's client call [command](#) , to call a Move function by passing the --serialize-unsigned-transaction flag. The signatures can be generated using Sui CLI's [keytool](#) command sui keytool sign . More information on Sui CLI can be found [here](#) .

Variables :

You can find other examples in the [repository](#) , grouped into sub-directories. For example, there are directories for [transaction block effects](#) , [protocol configs](#) , [stake connection](#) , and more.

Examples in the repository are designed to work with the version of GraphQL built at the same revision. The links above point to examples intended for [GraphQL v2024.1](#) , the latest production version at the time of writing.

Discovering the schema

GraphQL introspection exposes the schema supported by the RPC service. The IDE's "Docs" pane (Book icon, top-left) and Search dialog (Cmd + K on macOS or Ctrl + K on Windows and Linux) offer a way to browse introspection output interactively.

The [official documentation](#) provides an overview on introspection, and how to interact with it programmatically.

This example finds the total stake rewards, the reference gas price, the number of checkpoints and the total gas fees for epoch 100. Note that in the query, the id argument is optional, and defaults to the latest epoch.

This example gets a transaction block by its digest and shows information such as the gas sponsor's address, the gas price, the gas budget, and effects from executing that transaction block.

This example finds all the transactions that touched (modified/transferred/deleted) a given object. This is useful for when we want to trace the flow of a Coin/StakeSui/NFT.

This example uses GraphQL [variables](#) and [pagination](#) . When using the online IDE, copy the variables JSON to the "Variables" window, below the main editor.

Variables :

This example finds the last ten transaction blocks that called the public_transfer function, (as a move call transaction command).

This example makes usage of the filter last , which indicates that the user only wants the last ten transaction blocks known to the service.

This example finds the balance changes of all the transactions where a given address called a staking-related function. This is useful when you want to get your staking or unstaking history.

Variables :

This example uses aliases and [fragments](#) .

This query can be used to paginate over the dynamic fields of an object. This works even when the object in question is [wrapped](#) , by using the [owner](#) query, so can be used for iterating over the elements of on-chain data structures, like tables and bags. See [The Move Book](#) to learn more about dynamic collections available in Move.

This example uses [fragments](#) and [variables](#) .

Sets up a paginated query, starting at the genesis checkpoint, reading five checkpoints at a time, in increasing order of sequence number. The value of pageInfo.hasNextPage determines whether there is another page to be read, and the value of pageInfo.endCursor is fed back in as the cursor to read \$after .

This example uses GraphQL [variables](#) and [pagination](#) .

Sets up a paginated query, starting at the latest indexed checkpoint, reading five checkpoints at a time, in decreasing order of sequence number. The value of pageInfo.hasPreviousPage determines whether there is another page to be read, and the value of pageInfo.startCursor is fed back in as the cursor to read \$before .

This example uses GraphQL [variables](#) and [pagination](#) .

Transaction execution takes in two arguments, txBytes and signatures . txBytes is the serialized unsigned transaction data, which can

be generated when using the Sui CLI's client call [command](#) , to call a Move function by passing the --serialize-unsigned-transaction flag. The signatures can be generated using Sui CLI's [keytool](#) command sui keytool sign . More information on Sui CLI can be found [here](#) .

Variables :

You can find other examples in the [repository](#) , grouped into sub-directories. For example, there are directories for [transaction block effects](#) , [protocol configs](#) , [stake connection](#) , and more.

Examples in the repository are designed to work with the version of GraphQL built at the same revision. The links above point to examples intended for [GraphQL v2024.1](#) , the latest production version at the time of writing.

Finding the reference gas price for latest epoch

This example finds the total stake rewards, the reference gas price, the number of checkpoints and the total gas fees for epoch 100. Note that in the query, the id argument is optional, and defaults to the latest epoch.

This example gets a transaction block by its digest and shows information such as the gas sponsor's address, the gas price, the gas budget, and effects from executing that transaction block.

This example finds all the transactions that touched (modified/transferred/deleted) a given object. This is useful for when we want to trace the flow of a Coin/StakeSui/NFT.

This example uses GraphQL [variables](#) and [pagination](#) . When using the online IDE, copy the variables JSON to the "Variables" window, below the main editor.

Variables :

This example finds the last ten transaction blocks that called the public_transfer function, (as a move call transaction command).

This example makes usage of the filter last , which indicates that the user only wants the last ten transaction blocks known to the service.

This example finds the balance changes of all the transactions where a given address called a staking-related function. This is useful when you want to get your staking or unstaking history.

Variables :

This example uses aliases and [fragments](#) .

This query can be used to paginate over the dynamic fields of an object. This works even when the object in question is [wrapped](#) , by using the [owner](#) query, so can be used for iterating over the elements of on-chain data structures, like tables and bags. See [The Move Book](#) to learn more about dynamic collections available in Move.

This example uses [fragments](#) and [variables](#) .

Sets up a paginated query, starting at the genesis checkpoint, reading five checkpoints at a time, in increasing order of sequence number. The value of pageInfo.hasNextPage determines whether there is another page to be read, and the value of pageInfo.endCursor is fed back in as the cursor to read \$after .

This example uses GraphQL [variables](#) and [pagination](#) .

Sets up a paginated query, starting at the latest indexed checkpoint, reading five checkpoints at a time, in decreasing order of sequence number. The value of pageInfo.hasPreviousPage determines whether there is another page to be read, and the value of pageInfo.startCursor is fed back in as the cursor to read \$before .

This example uses GraphQL [variables](#) and [pagination](#) .

Transaction execution takes in two arguments, txBytes and signatures . txBytes is the serialized unsigned transaction data, which can be generated when using the Sui CLI's client call [command](#) , to call a Move function by passing the --serialize-unsigned-transaction flag. The signatures can be generated using Sui CLI's [keytool](#) command sui keytool sign . More information on Sui CLI can be found [here](#) .

Variables :

You can find other examples in the [repository](#) , grouped into sub-directories. For example, there are directories for [transaction block effects](#) , [protocol configs](#) , [stake connection](#) , and more.

Examples in the repository are designed to work with the version of GraphQL built at the same revision. The links above point to examples intended for [GraphQL v2024.1](#) , the latest production version at the time of writing.

Finding information about a specific historical epoch

This example finds the total stake rewards, the reference gas price, the number of checkpoints and the total gas fees for epoch 100. Note that in the query, the id argument is optional, and defaults to the latest epoch.

This example gets a transaction block by its digest and shows information such as the gas sponsor's address, the gas price, the gas budget, and effects from executing that transaction block.

This example finds all the transactions that touched (modified/transferred/deleted) a given object. This is useful for when we want to trace the flow of a Coin/StakeSui/NFT.

This example uses GraphQL [variables](#) and [pagination](#) . When using the online IDE, copy the variables JSON to the "Variables" window, below the main editor.

Variables :

This example finds the last ten transaction blocks that called the public_transfer function, (as a move call transaction command).

This example makes usage of the filter last , which indicates that the user only wants the last ten transaction blocks known to the service.

This example finds the balance changes of all the transactions where a given address called a staking-related function. This is useful when you want to get your staking or unstaking history.

Variables :

This example uses aliases and [fragments](#) .

This query can be used to paginate over the dynamic fields of an object. This works even when the object in question is [wrapped](#) , by using the [owner](#) query, so can be used for iterating over the elements of on-chain data structures, like tables and bags. See [The Move Book](#) to learn more about dynamic collections available in Move.

This example uses [fragments](#) and [variables](#) .

Sets up a paginated query, starting at the genesis checkpoint, reading five checkpoints at a time, in increasing order of sequence number. The value of pageInfo.hasNextPage determines whether there is another page to be read, and the value of pageInfo.endCursor is fed back in as the cursor to read \$after .

This example uses GraphQL [variables](#) and [pagination](#) .

Sets up a paginated query, starting at the latest indexed checkpoint, reading five checkpoints at a time, in decreasing order of sequence number. The value of pageInfo.hasPreviousPage determines whether there is another page to be read, and the value of pageInfo.startCursor is fed back in as the cursor to read \$before .

This example uses GraphQL [variables](#) and [pagination](#) .

Transaction execution takes in two arguments, txBytes and signatures . txBytes is the serialized unsigned transaction data, which can be generated when using the Sui CLI's client call [command](#) , to call a Move function by passing the --serialize-unsigned-transaction flag. The signatures can be generated using Sui CLI's [keytool](#) command sui keytool sign . More information on Sui CLI can be found [here](#) .

Variables :

You can find other examples in the [repository](#) , grouped into sub-directories. For example, there are directories for [transaction block effects](#) , [protocol configs](#) , [stake connection](#) , and more.

Examples in the repository are designed to work with the version of GraphQL built at the same revision. The links above point to examples intended for [GraphQL v2024.1](#) , the latest production version at the time of writing.

Finding a transaction block by its digest

This example gets a transaction block by its digest and shows information such as the gas sponsor's address, the gas price, the gas budget, and effects from executing that transaction block.

This example finds all the transactions that touched (modified/transferred/deleted) a given object. This is useful for when we want to trace the flow of a Coin/StakeSui/NFT.

This example uses GraphQL [variables](#) and [pagination](#) . When using the online IDE, copy the variables JSON to the "Variables" window, below the main editor.

Variables :

This example finds the last ten transaction blocks that called the `public_transfer` function, (as a move call transaction command).

This example makes usage of the filter `last` , which indicates that the user only wants the last ten transaction blocks known to the service.

This example finds the balance changes of all the transactions where a given address called a staking-related function. This is useful when you want to get your staking or unstaking history.

Variables :

This example uses aliases and [fragments](#) .

This query can be used to paginate over the dynamic fields of an object. This works even when the object in question is [wrapped](#) , by using the [owner](#) query, so can be used for iterating over the elements of on-chain data structures, like tables and bags. See [The Move Book](#) to learn more about dynamic collections available in Move.

This example uses [fragments](#) and [variables](#) .

Sets up a paginated query, starting at the genesis checkpoint, reading five checkpoints at a time, in increasing order of sequence number. The value of `pageInfo.hasNextPage` determines whether there is another page to be read, and the value of `pageInfo.endCursor` is fed back in as the cursor to read \$after .

This example uses GraphQL [variables](#) and [pagination](#) .

Sets up a paginated query, starting at the latest indexed checkpoint, reading five checkpoints at a time, in decreasing order of sequence number. The value of `pageInfo.hasPreviousPage` determines whether there is another page to be read, and the value of `pageInfo.startCursor` is fed back in as the cursor to read \$before .

This example uses GraphQL [variables](#) and [pagination](#) .

Transaction execution takes in two arguments, `txBytes` and `signatures` . `txBytes` is the serialized unsigned transaction data, which can be generated when using the Sui CLI's client call [command](#) , to call a Move function by passing the `--serialize-unsigned-transaction` flag. The signatures can be generated using Sui CLI's [keytool](#) command `sui keytool sign` . More information on Sui CLI can be found [here](#) .

Variables :

You can find other examples in the [repository](#) , grouped into sub-directories. For example, there are directories for [transaction block effects](#) , [protocol configs](#) , [stake connection](#) , and more.

Examples in the repository are designed to work with the version of GraphQL built at the same revision. The links above point to examples intended for [GraphQL v2024.1](#) , the latest production version at the time of writing.

Finding the last ten transactions that are not a system transaction

This example finds all the transactions that touched (modified/transferred/deleted) a given object. This is useful for when we want to trace the flow of a Coin/StakeSui/NFT.

This example uses GraphQL [variables](#) and [pagination](#) . When using the online IDE, copy the variables JSON to the "Variables" window, below the main editor.

Variables :

This example finds the last ten transaction blocks that called the `public_transfer` function, (as a move call transaction command).

This example makes usage of the filter `last`, which indicates that the user only wants the last ten transaction blocks known to the service.

This example finds the balance changes of all the transactions where a given address called a staking-related function. This is useful when you want to get your staking or unstaking history.

Variables :

This example uses aliases and [fragments](#).

This query can be used to paginate over the dynamic fields of an object. This works even when the object in question is [wrapped](#), by using the [owner](#) query, so can be used for iterating over the elements of on-chain data structures, like tables and bags. See [The Move Book](#) to learn more about dynamic collections available in Move.

This example uses [fragments](#) and [variables](#).

Sets up a paginated query, starting at the genesis checkpoint, reading five checkpoints at a time, in increasing order of sequence number. The value of `pageInfo.hasNextPage` determines whether there is another page to be read, and the value of `pageInfo.endCursor` is fed back in as the cursor to read \$after.

This example uses GraphQL [variables](#) and [pagination](#).

Sets up a paginated query, starting at the latest indexed checkpoint, reading five checkpoints at a time, in decreasing order of sequence number. The value of `pageInfo.hasPreviousPage` determines whether there is another page to be read, and the value of `pageInfo.startCursor` is fed back in as the cursor to read \$before.

This example uses GraphQL [variables](#) and [pagination](#).

Transaction execution takes in two arguments, `txBytes` and `signatures`. `txBytes` is the serialized unsigned transaction data, which can be generated when using the Sui CLI's client call [command](#), to call a Move function by passing the `--serialize-unsigned-transaction` flag. The `signatures` can be generated using Sui CLI's [keytool](#) command `sui keytool sign`. More information on Sui CLI can be found [here](#).

Variables :

You can find other examples in the [repository](#), grouped into sub-directories. For example, there are directories for [transaction block effects](#), [protocol configs](#), [stake connection](#), and more.

Examples in the repository are designed to work with the version of GraphQL built at the same revision. The links above point to examples intended for [GraphQL v2024.1](#), the latest production version at the time of writing.

Finding all transactions that touched a given object

This example finds all the transactions that touched (modified/transferred/deleted) a given object. This is useful for when we want to trace the flow of a Coin/StakeSui/NFT.

This example uses GraphQL [variables](#) and [pagination](#). When using the online IDE, copy the variables JSON to the "Variables" window, below the main editor.

Variables :

This example finds the last ten transaction blocks that called the `public_transfer` function, (as a move call transaction command).

This example makes usage of the filter `last`, which indicates that the user only wants the last ten transaction blocks known to the service.

This example finds the balance changes of all the transactions where a given address called a staking-related function. This is useful when you want to get your staking or unstaking history.

Variables :

This example uses aliases and [fragments](#) .

This query can be used to paginate over the dynamic fields of an object. This works even when the object in question is [wrapped](#) , by using the [owner](#) query, so can be used for iterating over the elements of on-chain data structures, like tables and bags. See [The Move Book](#) to learn more about dynamic collections available in Move.

This example uses [fragments](#) and [variables](#) .

Sets up a paginated query, starting at the genesis checkpoint, reading five checkpoints at a time, in increasing order of sequence number. The value of pageInfo.hasNextPage determines whether there is another page to be read, and the value of pageInfo.endCursor is fed back in as the cursor to read \$after .

This example uses GraphQL [variables](#) and [pagination](#) .

Sets up a paginated query, starting at the latest indexed checkpoint, reading five checkpoints at a time, in decreasing order of sequence number. The value of pageInfo.hasPreviousPage determines whether there is another page to be read, and the value of pageInfo.startCursor is fed back in as the cursor to read \$before .

This example uses GraphQL [variables](#) and [pagination](#) .

Transaction execution takes in two arguments, txBytes and signatures . txBytes is the serialized unsigned transaction data, which can be generated when using the Sui CLI's client call [command](#) , to call a Move function by passing the --serialize-unsigned-transaction flag. The signatures can be generated using Sui CLI's [keytool](#) command sui keytool sign . More information on Sui CLI can be found [here](#) .

Variables :

You can find other examples in the [repository](#) , grouped into sub-directories. For example, there are directories for [transaction block effects](#) , [protocol configs](#) , [stake connection](#) , and more.

Examples in the repository are designed to work with the version of GraphQL built at the same revision. The links above point to examples intended for [GraphQL v2024.1](#) , the latest production version at the time of writing.

Filtering transaction blocks by a function

This example finds the last ten transaction blocks that called the public_transfer function, (as a move call transaction command).

This example makes usage of the filter last , which indicates that the user only wants the last ten transaction blocks known to the service.

This example finds the balance changes of all the transactions where a given address called a staking-related function. This is useful when you want to get your staking or unstaking history.

Variables :

This example uses aliases and [fragments](#) .

This query can be used to paginate over the dynamic fields of an object. This works even when the object in question is [wrapped](#) , by using the [owner](#) query, so can be used for iterating over the elements of on-chain data structures, like tables and bags. See [The Move Book](#) to learn more about dynamic collections available in Move.

This example uses [fragments](#) and [variables](#) .

Sets up a paginated query, starting at the genesis checkpoint, reading five checkpoints at a time, in increasing order of sequence number. The value of pageInfo.hasNextPage determines whether there is another page to be read, and the value of pageInfo.endCursor is fed back in as the cursor to read \$after .

This example uses GraphQL [variables](#) and [pagination](#) .

Sets up a paginated query, starting at the latest indexed checkpoint, reading five checkpoints at a time, in decreasing order of sequence number. The value of pageInfo.hasPreviousPage determines whether there is another page to be read, and the value of pageInfo.startCursor is fed back in as the cursor to read \$before .

This example uses GraphQL [variables](#) and [pagination](#) .

Transaction execution takes in two arguments, txBytes and signatures . txBytes is the serialized unsigned transaction data, which can be generated when using the Sui CLI's client call [command](#) , to call a Move function by passing the --serialize-unsigned-transaction flag. The signatures can be generated using Sui CLI's [keytool](#) command sui keytool sign . More information on Sui CLI can be found [here](#) .

Variables :

You can find other examples in the [repository](#) , grouped into sub-directories. For example, there are directories for [transaction block effects](#) , [protocol configs](#) , [stake connection](#) , and more.

Examples in the repository are designed to work with the version of GraphQL built at the same revision. The links above point to examples intended for [GraphQL v2024.1](#) , the latest production version at the time of writing.

Finding transaction balance changes

This example finds the balance changes of all the transactions where a given address called a staking-related function. This is useful when you want to get your staking or unstaking history.

Variables :

This example uses aliases and [fragments](#) .

This query can be used to paginate over the dynamic fields of an object. This works even when the object in question is [wrapped](#) , by using the [owner](#) query, so can be used for iterating over the elements of on-chain data structures, like tables and bags. See [The Move Book](#) to learn more about dynamic collections available in Move.

This example uses [fragments](#) and [variables](#) .

Sets up a paginated query, starting at the genesis checkpoint, reading five checkpoints at a time, in increasing order of sequence number. The value of pageInfo.hasNextPage determines whether there is another page to be read, and the value of pageInfo.endCursor is fed back in as the cursor to read \$after .

This example uses GraphQL [variables](#) and [pagination](#) .

Sets up a paginated query, starting at the latest indexed checkpoint, reading five checkpoints at a time, in decreasing order of sequence number. The value of pageInfo.hasPreviousPage determines whether there is another page to be read, and the value of pageInfo.startCursor is fed back in as the cursor to read \$before .

This example uses GraphQL [variables](#) and [pagination](#) .

Transaction execution takes in two arguments, txBytes and signatures . txBytes is the serialized unsigned transaction data, which can be generated when using the Sui CLI's client call [command](#) , to call a Move function by passing the --serialize-unsigned-transaction flag. The signatures can be generated using Sui CLI's [keytool](#) command sui keytool sign . More information on Sui CLI can be found [here](#) .

Variables :

You can find other examples in the [repository](#) , grouped into sub-directories. For example, there are directories for [transaction block effects](#) , [protocol configs](#) , [stake connection](#) , and more.

Examples in the repository are designed to work with the version of GraphQL built at the same revision. The links above point to examples intended for [GraphQL v2024.1](#) , the latest production version at the time of writing.

Fetching a dynamic field on an object

This example uses aliases and [fragments](#) .

This query can be used to paginate over the dynamic fields of an object. This works even when the object in question is [wrapped](#) , by using the [owner](#) query, so can be used for iterating over the elements of on-chain data structures, like tables and bags. See [The Move Book](#) to learn more about dynamic collections available in Move.

This example uses [fragments](#) and [variables](#) .

Sets up a paginated query, starting at the genesis checkpoint, reading five checkpoints at a time, in increasing order of sequence

number. The value of pageInfo.hasNextPage determines whether there is another page to be read, and the value of pageInfo.endCursor is fed back in as the cursor to read \$after .

This example uses GraphQL [variables](#) and [pagination](#) .

Sets up a paginated query, starting at the latest indexed checkpoint, reading five checkpoints at a time, in decreasing order of sequence number. The value of pageInfo.hasPreviousPage determines whether there is another page to be read, and the value of pageInfo.startCursor is fed back in as the cursor to read \$before .

This example uses GraphQL [variables](#) and [pagination](#) .

Transaction execution takes in two arguments, txBytes and signatures . txBytes is the serialized unsigned transaction data, which can be generated when using the Sui CLI's client call [command](#) , to call a Move function by passing the --serialize-unsigned-transaction flag. The signatures can be generated using Sui CLI's [keytool](#) command sui keytool sign . More information on Sui CLI can be found [here](#) .

Variables :

You can find other examples in the [repository](#) , grouped into sub-directories. For example, there are directories for [transaction block effects](#) , [protocol configs](#) , [stake connection](#) , and more.

Examples in the repository are designed to work with the version of GraphQL built at the same revision. The links above point to examples intended for [GraphQL v2024.1](#) , the latest production version at the time of writing.

Fetching all dynamic fields on an object

This query can be used to paginate over the dynamic fields of an object. This works even when the object in question is [wrapped](#) , by using the [owner](#) query, so can be used for iterating over the elements of on-chain data structures, like tables and bags. See [The Move Book](#) to learn more about dynamic collections available in Move.

This example uses [fragments](#) and [variables](#) .

Sets up a paginated query, starting at the genesis checkpoint, reading five checkpoints at a time, in increasing order of sequence number. The value of pageInfo.hasNextPage determines whether there is another page to be read, and the value of pageInfo.endCursor is fed back in as the cursor to read \$after .

This example uses GraphQL [variables](#) and [pagination](#) .

Sets up a paginated query, starting at the latest indexed checkpoint, reading five checkpoints at a time, in decreasing order of sequence number. The value of pageInfo.hasPreviousPage determines whether there is another page to be read, and the value of pageInfo.startCursor is fed back in as the cursor to read \$before .

This example uses GraphQL [variables](#) and [pagination](#) .

Transaction execution takes in two arguments, txBytes and signatures . txBytes is the serialized unsigned transaction data, which can be generated when using the Sui CLI's client call [command](#) , to call a Move function by passing the --serialize-unsigned-transaction flag. The signatures can be generated using Sui CLI's [keytool](#) command sui keytool sign . More information on Sui CLI can be found [here](#) .

Variables :

You can find other examples in the [repository](#) , grouped into sub-directories. For example, there are directories for [transaction block effects](#) , [protocol configs](#) , [stake connection](#) , and more.

Examples in the repository are designed to work with the version of GraphQL built at the same revision. The links above point to examples intended for [GraphQL v2024.1](#) , the latest production version at the time of writing.

Paginating checkpoints forward, five at a time

Sets up a paginated query, starting at the genesis checkpoint, reading five checkpoints at a time, in increasing order of sequence number. The value of pageInfo.hasNextPage determines whether there is another page to be read, and the value of pageInfo.endCursor is fed back in as the cursor to read \$after .

This example uses GraphQL [variables](#) and [pagination](#) .

Sets up a paginated query, starting at the latest indexed checkpoint, reading five checkpoints at a time, in decreasing order of sequence number. The value of pageInfo.hasPreviousPage determines whether there is another page to be read, and the value of pageInfo.startCursor is fed back in as the cursor to read \$before .

This example uses GraphQL [variables](#) and [pagination](#) .

Transaction execution takes in two arguments, txBytes and signatures . txBytes is the serialized unsigned transaction data, which can be generated when using the Sui CLI's client call [command](#) , to call a Move function by passing the --serialize-unsigned-transaction flag. The signatures can be generated using Sui CLI's [keytool](#) command sui keytool sign . More information on Sui CLI can be found [here](#) .

Variables :

You can find other examples in the [repository](#) , grouped into sub-directories. For example, there are directories for [transaction block effects](#) , [protocol configs](#) , [stake connection](#) , and more.

Examples in the repository are designed to work with the version of GraphQL built at the same revision. The links above point to examples intended for [GraphQL v2024.1](#) , the latest production version at the time of writing.

Paginating checkpoints backwards, five at a time

Sets up a paginated query, starting at the latest indexed checkpoint, reading five checkpoints at a time, in decreasing order of sequence number. The value of pageInfo.hasPreviousPage determines whether there is another page to be read, and the value of pageInfo.startCursor is fed back in as the cursor to read \$before .

This example uses GraphQL [variables](#) and [pagination](#) .

Transaction execution takes in two arguments, txBytes and signatures . txBytes is the serialized unsigned transaction data, which can be generated when using the Sui CLI's client call [command](#) , to call a Move function by passing the --serialize-unsigned-transaction flag. The signatures can be generated using Sui CLI's [keytool](#) command sui keytool sign . More information on Sui CLI can be found [here](#) .

Variables :

You can find other examples in the [repository](#) , grouped into sub-directories. For example, there are directories for [transaction block effects](#) , [protocol configs](#) , [stake connection](#) , and more.

Examples in the repository are designed to work with the version of GraphQL built at the same revision. The links above point to examples intended for [GraphQL v2024.1](#) , the latest production version at the time of writing.

Executing a transaction

Transaction execution takes in two arguments, txBytes and signatures . txBytes is the serialized unsigned transaction data, which can be generated when using the Sui CLI's client call [command](#) , to call a Move function by passing the --serialize-unsigned-transaction flag. The signatures can be generated using Sui CLI's [keytool](#) command sui keytool sign . More information on Sui CLI can be found [here](#) .

Variables :

You can find other examples in the [repository](#) , grouped into sub-directories. For example, there are directories for [transaction block effects](#) , [protocol configs](#) , [stake connection](#) , and more.

Examples in the repository are designed to work with the version of GraphQL built at the same revision. The links above point to examples intended for [GraphQL v2024.1](#) , the latest production version at the time of writing.

Other examples

You can find other examples in the [repository](#) , grouped into sub-directories. For example, there are directories for [transaction block effects](#) , [protocol configs](#) , [stake connection](#) , and more.

Examples in the repository are designed to work with the version of GraphQL built at the same revision. The links above point to examples intended for [GraphQL v2024.1](#) , the latest production version at the time of writing.

Related links