Sui Exchange Integration Guide

This topic describes how to integrate SUI, the token native to the Sui network, into a cryptocurrency exchange. The specific requirements and processes to implement an integration vary between exchanges. Rather than provide a step-by-step guide, this topic provides information about the primary tasks necessary to complete an integration. After the guidance about how to configure an integration, you can also find information and code samples related to staking on the Sui network.

The requirements to configure a SUI integration include:

For best results, run Sui Full nodes on Linux. Sui supports the Ubuntu and Debian distributions. You can also fun a Full node on macOS.

You can set up and configure a Sui Full node using Docker or directly from source code in the Sui GitHub repository.

Sui addresses do not require on-chain initialization, you can spend from an address if it corresponds to your private key. You can derive a 32-byte Sui address by hashing the signature scheme flag byte concatenated with public key bytes flag || pubkey using the BLAKE2b (256 bits output) hashing function.

Currently, Sui address supports these signature schemes: pure Ed25519, Secp256k1, Secp256r1 and multisig. The corresponding flag bytes are 0x00, 0x01, 0x02, 0x03 respectively.

The following code sample demonstrates how to derive a Sui address in Rust:

Sui supports both addresses with and without a 0x prefix. Sui recommends that you always include the 0x prefix in API calls and when you display user addresses.

You can track balance changes by calling sui_getBalance at predefined intervals. This call returns the total balance for an address. The total includes any coin or token type, but this document focuses on SUI. You can track changes in the total balance for an address between subsequent sui_getBalance requests.

The following bash example demonstrates how to use sui_getBalance for address 0x849d63687330447431a2e76fecca4f3c10f6884ebaa9909674123c6c662612a3 . If you use a network other than Devnet, replace the value for rpc with the URL to the appropriate Full node.

The response is a JSON object that includes the totalBalance for the address:

The following example demonstrates using sui_getBalance in Rust:

You can also track the balance for an address by subscribing to all of the events emitted from it. Use a filter to include only the events related to SUI coins, such as when the address acquires a coin or pays for a gas fee. The following example demonstrates how to filter events for an address using bash and cURL:

The response can include a large number of events. Add pagination to the response using the nextCursor key in the request. You can determine the corresponding txDigest and eventSeq from the id field of a transaction.

You can add the txDigest value instead of the first null within the params . The second null is an integer that defines how many results (up to 1000) to return and the true means ascending order. You can use the nextCursor so the response starts from a desired point.

The id field of any transaction looks like:

With this data, create a nextCursor as follows:

Sui is a DAG-based blockchain and uses checkpoints for node synchronization and global transaction ordering. Checkpoints differ from blocks in the following ways:

Sui Checkpoint API operations include:

To transfer a specific amount of SUI between addresses, you need a SUI token object with that specific value. In Sui, everything is an object, including SUI tokens. The amount of SUI in each SUI token object varies. For example, an address could own 3 SUI tokens with different values: one of 0.1 SUI, a second of 1.0 SUI, and a third with 0.005 SUI. The total balance for the address equals the sum of the values of the individual SUI token objects, in this case, 1.105 SUI.

You can merge and split SUI token objects to create token objects with specific values. To create a SUI token worth .6 SUI, split the token worth 1 SUI into two token objects worth .6 SUI and .4 SUI.

To transfer a specific amount of SUI, you need a SUI token worth that specific amount. To get a SUI token with that specific value, you might need to split or merge existing SUI tokens. Sui supports several methods to accomplish this, including some that do not require you to manually split or merge coins.

Sui supports the following API operations related to transferring SUI between addresses:

sui_transferObject Because SUI tokens are objects, you can transfer SUI tokens just like any other object. This method requires a gas token, and is useful in niche cases only.

sui_payAllSui This method accepts an array of SUI token IDs. It merges all existing tokens into one, deducts the gas fee, then sends the merged token to the recipient address.

The method is especially useful if you want to transfer all SUI from an address. To merge together all coins for an address, set the recipient as the same address. This is a native Sui method so is not considered a transaction in Sui.

sui_paySui This operation accepts an array of SUI token IDs, an array of amounts, and an array of recipient addresses.

The amounts and recipients array map one to one. Even if you use only one recipient address, you must include it for each amount in the amount array.

The operation merges all of the tokens provided into one token object and settles the gas fees. It then splits the token according to the amounts in the amounts array and sends the first token to the first recipient, the second token to the second recipient, and so on. Any remaining SUI on the token stays in the source address.

The benefits of this method include: no gas fees for merging or splitting tokens, and the abstracted token merge and split. The sui_paySui operation is a native function, so the merge and split operations are not considered Sui transactions. The gas fees for them match typical transactions on Sui.You can use this operation to split coins in your own address by setting the recipient as your own address. Note that the total value of the input coins must be greater than the total value of the amounts to send.

sui_pay This method is similar to sui_paySui, but it accepts any kind of coin or token instead of only SUI. You must include a gas token, and all of the coins or tokens must be the same type.

sui_transferSui This method accepts only one SUI token object and an amount to send to the recipient. It uses the same token for gas fees, so the amount to transfer must be strictly less than the value of the SUI token used.

Refer to Sui Signatures for more details on signature validity requirements.

The Sui blockchain uses a Delegated Proof-of-Stake mechanism (DPoS). This allows SUI token holders to stake their SUI tokens to any validator of their choice. When someone stakes their SUI tokens, it means those tokens are locked for the entire epoch. Users can withdraw their stake at any time, but new staking requests become active only at the start of the next epoch.

SUI holders who stake their tokens to validators earn rewards for helping secure the Sui network. Sui determines rewards for staking based on stake rewards on the network, and distributes them at the end of each epoch.

The total voting power in the Sui Network is always 10,000. The voting power of each individual validator is similar to basis points. For example, a voting power of 101 = 1.01%. Sui's quorum threshold (number of votes needed to confirm a transaction) is 6,667 (which is greater than 2/3). The voting power for a single validator is capped at 1,000 (10%) regardless of how much stake the validator has.

Sui supports the following API operations related to staking. You can find the source code in the sui_system module.

Requirements to configure a SUI integration

The requirements to configure a SUI integration include:

For best results, run Sui Full nodes on Linux. Sui supports the Ubuntu and Debian distributions. You can also fun a Full node on macOS.

You can set up and configure a Sui Full node using Docker or directly from source code in the Sui GitHub repository.

Sui addresses do not require on-chain initialization, you can spend from an address if it corresponds to your private key. You can derive a 32-byte Sui address by hashing the signature scheme flag byte concatenated with public key bytes flag || pubkey using the BLAKE2b (256 bits output) hashing function.

Currently, Sui address supports these signature schemes: pure Ed25519, Secp256k1, Secp256r1 and multisig. The corresponding flag bytes are 0x00, 0x01, 0x02, 0x03 respectively.

The following code sample demonstrates how to derive a Sui address in Rust:

Sui supports both addresses with and without a 0x prefix. Sui recommends that you always include the 0x prefix in API calls and when you display user addresses.

You can track balance changes by calling sui_getBalance at predefined intervals. This call returns the total balance for an address. The total includes any coin or token type, but this document focuses on SUI. You can track changes in the total balance for an address between subsequent sui_getBalance requests.

The following bash example demonstrates how to use sui_getBalance for address 0x849d63687330447431a2e76fecca4f3c10f6884ebaa9909674123c6c662612a3 . If you use a network other than Devnet, replace the value for rpc with the URL to the appropriate Full node.

The response is a JSON object that includes the totalBalance for the address:

The following example demonstrates using sui_getBalance in Rust:

You can also track the balance for an address by subscribing to all of the events emitted from it. Use a filter to include only the events related to SUI coins, such as when the address acquires a coin or pays for a gas fee. The following example demonstrates how to filter events for an address using bash and cURL:

The response can include a large number of events. Add pagination to the response using the nextCursor key in the request. You can determine the corresponding txDigest and eventSeq from the id field of a transaction.

You can add the txDigest value instead of the first null within the params . The second null is an integer that defines how many results (up to 1000) to return and the true means ascending order. You can use the nextCursor so the response starts from a desired point.

The id field of any transaction looks like:

With this data, create a nextCursor as follows:

Sui is a DAG-based blockchain and uses checkpoints for node synchronization and global transaction ordering. Checkpoints differ from blocks in the following ways:

Sui Checkpoint API operations include:

To transfer a specific amount of SUI between addresses, you need a SUI token object with that specific value. In Sui, everything is an object, including SUI tokens. The amount of SUI in each SUI token object varies. For example, an address could own 3 SUI tokens with different values: one of 0.1 SUI, a second of 1.0 SUI, and a third with 0.005 SUI. The total balance for the address equals the sum of the values of the individual SUI token objects, in this case, 1.105 SUI.

You can merge and split SUI token objects to create token objects with specific values. To create a SUI token worth .6 SUI, split the token worth 1 SUI into two token objects worth .6 SUI and .4 SUI.

To transfer a specific amount of SUI, you need a SUI token worth that specific amount. To get a SUI token with that specific value, you might need to split or merge existing SUI tokens. Sui supports several methods to accomplish this, including some that do not require you to manually split or merge coins.

Sui supports the following API operations related to transferring SUI between addresses:

sui_transferObject Because SUI tokens are objects, you can transfer SUI tokens just like any other object. This method requires a gas token, and is useful in niche cases only.

sui_payAllSui This method accepts an array of SUI token IDs. It merges all existing tokens into one, deducts the gas fee, then sends the merged token to the recipient address.

The method is especially useful if you want to transfer all SUI from an address. To merge together all coins for an address, set the recipient as the same address. This is a native Sui method so is not considered a transaction in Sui.

sui_paySui This operation accepts an array of SUI token IDs, an array of amounts, and an array of recipient addresses.

The amounts and recipients array map one to one. Even if you use only one recipient address, you must include it for each amount in

the amount array.

The operation merges all of the tokens provided into one token object and settles the gas fees. It then splits the token according to the amounts in the amounts array and sends the first token to the first recipient, the second token to the second recipient, and so on. Any remaining SUI on the token stays in the source address.

The benefits of this method include: no gas fees for merging or splitting tokens, and the abstracted token merge and split. The sui_paySui operation is a native function, so the merge and split operations are not considered Sui transactions. The gas fees for them match typical transactions on Sui.You can use this operation to split coins in your own address by setting the recipient as your own address. Note that the total value of the input coins must be greater than the total value of the amounts to send.

sui_pay This method is similar to sui_paySui, but it accepts any kind of coin or token instead of only SUI. You must include a gas token, and all of the coins or tokens must be the same type.

sui_transferSui This method accepts only one SUI token object and an amount to send to the recipient. It uses the same token for gas fees, so the amount to transfer must be strictly less than the value of the SUI token used.

Refer to Sui Signatures for more details on signature validity requirements.

The Sui blockchain uses a Delegated Proof-of-Stake mechanism (DPoS). This allows SUI token holders to stake their SUI tokens to any validator of their choice. When someone stakes their SUI tokens, it means those tokens are locked for the entire epoch. Users can withdraw their stake at any time, but new staking requests become active only at the start of the next epoch.

SUI holders who stake their tokens to validators earn rewards for helping secure the Sui network. Sui determines rewards for staking based on stake rewards on the network, and distributes them at the end of each epoch.

The total voting power in the Sui Network is always 10,000. The voting power of each individual validator is similar to basis points. For example, a voting power of 101 = 1.01%. Sui's quorum threshold (number of votes needed to confirm a transaction) is 6,667 (which is greater than 2/3). The voting power for a single validator is capped at 1,000 (10%) regardless of how much stake the validator has.

Sui supports the following API operations related to staking. You can find the source code in the sui_system module.

Configure a Sui Full node

You can set up and configure a Sui Full node using Docker or directly from source code in the Sui GitHub repository.

Sui addresses do not require on-chain initialization, you can spend from an address if it corresponds to your private key. You can derive a 32-byte Sui address by hashing the signature scheme flag byte concatenated with public key bytes flag || pubkey using the BLAKE2b (256 bits output) hashing function.

Currently, Sui address supports these signature schemes: pure Ed25519, Secp256k1, Secp256r1 and multisig. The corresponding flag bytes are 0x00, 0x01, 0x02, 0x03 respectively.

The following code sample demonstrates how to derive a Sui address in Rust:

Sui supports both addresses with and without a 0x prefix. Sui recommends that you always include the 0x prefix in API calls and when you display user addresses.

You can track balance changes by calling sui_getBalance at predefined intervals. This call returns the total balance for an address. The total includes any coin or token type, but this document focuses on SUI. You can track changes in the total balance for an address between subsequent sui_getBalance requests.

The following bash example demonstrates how to use sui_getBalance for address 0x849d63687330447431a2e76fecca4f3c10f6884ebaa9909674123c6c662612a3 . If you use a network other than Devnet, replace the value for rpc with the URL to the appropriate Full node.

The response is a JSON object that includes the totalBalance for the address:

The following example demonstrates using sui_getBalance in Rust:

You can also track the balance for an address by subscribing to all of the events emitted from it. Use a filter to include only the events related to SUI coins, such as when the address acquires a coin or pays for a gas fee. The following example demonstrates how to filter events for an address using bash and cURL:

The response can include a large number of events. Add pagination to the response using the nextCursor key in the request. You can determine the corresponding txDigest and eventSeq from the id field of a transaction.

You can add the txDigest value instead of the first null within the params . The second null is an integer that defines how many results (up to 1000) to return and the true means ascending order. You can use the nextCursor so the response starts from a desired point.

The id field of any transaction looks like:

With this data, create a nextCursor as follows:

Sui is a DAG-based blockchain and uses checkpoints for node synchronization and global transaction ordering. Checkpoints differ from blocks in the following ways:

Sui Checkpoint API operations include:

To transfer a specific amount of SUI between addresses, you need a SUI token object with that specific value. In Sui, everything is an object, including SUI tokens. The amount of SUI in each SUI token object varies. For example, an address could own 3 SUI tokens with different values: one of 0.1 SUI, a second of 1.0 SUI, and a third with 0.005 SUI. The total balance for the address equals the sum of the values of the individual SUI token objects, in this case, 1.105 SUI.

You can merge and split SUI token objects to create token objects with specific values. To create a SUI token worth .6 SUI, split the token worth 1 SUI into two token objects worth .6 SUI and .4 SUI.

To transfer a specific amount of SUI, you need a SUI token worth that specific amount. To get a SUI token with that specific value, you might need to split or merge existing SUI tokens. Sui supports several methods to accomplish this, including some that do not require you to manually split or merge coins.

Sui supports the following API operations related to transferring SUI between addresses:

sui_transferObject Because SUI tokens are objects, you can transfer SUI tokens just like any other object. This method requires a gas token, and is useful in niche cases only.

sui_payAllSui This method accepts an array of SUI token IDs. It merges all existing tokens into one, deducts the gas fee, then sends the merged token to the recipient address.

The method is especially useful if you want to transfer all SUI from an address. To merge together all coins for an address, set the recipient as the same address. This is a native Sui method so is not considered a transaction in Sui.

sui_paySui This operation accepts an array of SUI token IDs, an array of amounts, and an array of recipient addresses.

The amounts and recipients array map one to one. Even if you use only one recipient address, you must include it for each amount in the amount array.

The operation merges all of the tokens provided into one token object and settles the gas fees. It then splits the token according to the amounts in the amounts array and sends the first token to the first recipient, the second token to the second recipient, and so on. Any remaining SUI on the token stays in the source address.

The benefits of this method include: no gas fees for merging or splitting tokens, and the abstracted token merge and split. The sui_paySui operation is a native function, so the merge and split operations are not considered Sui transactions. The gas fees for them match typical transactions on Sui.You can use this operation to split coins in your own address by setting the recipient as your own address. Note that the total value of the input coins must be greater than the total value of the amounts to send.

sui_pay This method is similar to sui_paySui, but it accepts any kind of coin or token instead of only SUI. You must include a gas token, and all of the coins or tokens must be the same type.

sui_transferSui This method accepts only one SUI token object and an amount to send to the recipient. It uses the same token for gas fees, so the amount to transfer must be strictly less than the value of the SUI token used.

Refer to Sui Signatures for more details on signature validity requirements.

The Sui blockchain uses a Delegated Proof-of-Stake mechanism (DPoS). This allows SUI token holders to stake their SUI tokens to any validator of their choice. When someone stakes their SUI tokens, it means those tokens are locked for the entire epoch. Users can withdraw their stake at any time, but new staking requests become active only at the start of the next epoch.

SUI holders who stake their tokens to validators earn rewards for helping secure the Sui network. Sui determines rewards for

staking based on stake rewards on the network, and distributes them at the end of each epoch.

The total voting power in the Sui Network is always 10,000. The voting power of each individual validator is similar to basis points. For example, a voting power of 101 = 1.01%. Sui's quorum threshold (number of votes needed to confirm a transaction) is 6,667 (which is greater than 2/3). The voting power for a single validator is capped at 1,000 (10%) regardless of how much stake the validator has.

Sui supports the following API operations related to staking. You can find the source code in the sui_system module.

Set up Sui addresses

Sui addresses do not require on-chain initialization, you can spend from an address if it corresponds to your private key. You can derive a 32-byte Sui address by hashing the signature scheme flag byte concatenated with public key bytes flag || pubkey using the BLAKE2b (256 bits output) hashing function.

Currently, Sui address supports these signature schemes: pure Ed25519, Secp256k1, Secp256r1 and multisig. The corresponding flag bytes are 0x00, 0x01, 0x02, 0x03 respectively.

The following code sample demonstrates how to derive a Sui address in Rust:

Sui supports both addresses with and without a 0x prefix. Sui recommends that you always include the 0x prefix in API calls and when you display user addresses.

You can track balance changes by calling sui_getBalance at predefined intervals. This call returns the total balance for an address. The total includes any coin or token type, but this document focuses on SUI. You can track changes in the total balance for an address between subsequent sui_getBalance requests.

The following bash example demonstrates how to use sui_getBalance for address 0x849d63687330447431a2e76fecca4f3c10f6884ebaa9909674123c6c662612a3 . If you use a network other than Devnet, replace the value for rpc with the URL to the appropriate Full node.

The response is a JSON object that includes the totalBalance for the address:

The following example demonstrates using sui_getBalance in Rust:

You can also track the balance for an address by subscribing to all of the events emitted from it. Use a filter to include only the events related to SUI coins, such as when the address acquires a coin or pays for a gas fee. The following example demonstrates how to filter events for an address using bash and cURL:

The response can include a large number of events. Add pagination to the response using the nextCursor key in the request. You can determine the corresponding txDigest and eventSeq from the id field of a transaction.

You can add the txDigest value instead of the first null within the params . The second null is an integer that defines how many results (up to 1000) to return and the true means ascending order. You can use the nextCursor so the response starts from a desired point.

The id field of any transaction looks like:

With this data, create a nextCursor as follows:

Sui is a DAG-based blockchain and uses checkpoints for node synchronization and global transaction ordering. Checkpoints differ from blocks in the following ways:

Sui Checkpoint API operations include:

To transfer a specific amount of SUI between addresses, you need a SUI token object with that specific value. In Sui, everything is an object, including SUI tokens. The amount of SUI in each SUI token object varies. For example, an address could own 3 SUI tokens with different values: one of 0.1 SUI, a second of 1.0 SUI, and a third with 0.005 SUI. The total balance for the address equals the sum of the values of the individual SUI token objects, in this case, 1.105 SUI.

You can merge and split SUI token objects to create token objects with specific values. To create a SUI token worth .6 SUI, split the token worth 1 SUI into two token objects worth .6 SUI and .4 SUI.

To transfer a specific amount of SUI, you need a SUI token worth that specific amount. To get a SUI token with that specific value, you might need to split or merge existing SUI tokens. Sui supports several methods to accomplish this, including some that do not

require you to manually split or merge coins.

Sui supports the following API operations related to transferring SUI between addresses:

sui_transferObject Because SUI tokens are objects, you can transfer SUI tokens just like any other object. This method requires a gas token, and is useful in niche cases only.

sui_payAllSui This method accepts an array of SUI token IDs. It merges all existing tokens into one, deducts the gas fee, then sends the merged token to the recipient address.

The method is especially useful if you want to transfer all SUI from an address. To merge together all coins for an address, set the recipient as the same address. This is a native Sui method so is not considered a transaction in Sui.

sui_paySui This operation accepts an array of SUI token IDs, an array of amounts, and an array of recipient addresses.

The amounts and recipients array map one to one. Even if you use only one recipient address, you must include it for each amount in the amount array.

The operation merges all of the tokens provided into one token object and settles the gas fees. It then splits the token according to the amounts in the amounts array and sends the first token to the first recipient, the second token to the second recipient, and so on. Any remaining SUI on the token stays in the source address.

The benefits of this method include: no gas fees for merging or splitting tokens, and the abstracted token merge and split. The sui_paySui operation is a native function, so the merge and split operations are not considered Sui transactions. The gas fees for them match typical transactions on Sui.You can use this operation to split coins in your own address by setting the recipient as your own address. Note that the total value of the input coins must be greater than the total value of the amounts to send.

sui_pay This method is similar to sui_paySui, but it accepts any kind of coin or token instead of only SUI. You must include a gas token, and all of the coins or tokens must be the same type.

sui_transferSui This method accepts only one SUI token object and an amount to send to the recipient. It uses the same token for gas fees, so the amount to transfer must be strictly less than the value of the SUI token used.

Refer to Sui Signatures for more details on signature validity requirements.

The Sui blockchain uses a Delegated Proof-of-Stake mechanism (DPoS). This allows SUI token holders to stake their SUI tokens to any validator of their choice. When someone stakes their SUI tokens, it means those tokens are locked for the entire epoch. Users can withdraw their stake at any time, but new staking requests become active only at the start of the next epoch.

SUI holders who stake their tokens to validators earn rewards for helping secure the Sui network. Sui determines rewards for staking based on stake rewards on the network, and distributes them at the end of each epoch.

The total voting power in the Sui Network is always 10,000. The voting power of each individual validator is similar to basis points. For example, a voting power of 101 = 1.01%. Sui's quorum threshold (number of votes needed to confirm a transaction) is 6,667 (which is greater than 2/3). The voting power for a single validator is capped at 1,000 (10%) regardless of how much stake the validator has.

Sui supports the following API operations related to staking. You can find the source code in the sui_system module.

Displaying addresses

Sui supports both addresses with and without a 0x prefix. Sui recommends that you always include the 0x prefix in API calls and when you display user addresses.

You can track balance changes by calling sui_getBalance at predefined intervals. This call returns the total balance for an address. The total includes any coin or token type, but this document focuses on SUI. You can track changes in the total balance for an address between subsequent sui_getBalance requests.

The following bash example demonstrates how to use sui_getBalance for address 0x849d63687330447431a2e76fecca4f3c10f6884ebaa9909674123c6c662612a3 . If you use a network other than Devnet, replace the value for rpc with the URL to the appropriate Full node.

The response is a JSON object that includes the totalBalance for the address:

The following example demonstrates using sui_getBalance in Rust:

You can also track the balance for an address by subscribing to all of the events emitted from it. Use a filter to include only the events related to SUI coins, such as when the address acquires a coin or pays for a gas fee. The following example demonstrates how to filter events for an address using bash and cURL:

The response can include a large number of events. Add pagination to the response using the nextCursor key in the request. You can determine the corresponding txDigest and eventSeq from the id field of a transaction.

You can add the txDigest value instead of the first null within the params . The second null is an integer that defines how many results (up to 1000) to return and the true means ascending order. You can use the nextCursor so the response starts from a desired point.

The id field of any transaction looks like:

With this data, create a nextCursor as follows:

Sui is a DAG-based blockchain and uses checkpoints for node synchronization and global transaction ordering. Checkpoints differ from blocks in the following ways:

Sui Checkpoint API operations include:

To transfer a specific amount of SUI between addresses, you need a SUI token object with that specific value. In Sui, everything is an object, including SUI tokens. The amount of SUI in each SUI token object varies. For example, an address could own 3 SUI tokens with different values: one of 0.1 SUI, a second of 1.0 SUI, and a third with 0.005 SUI. The total balance for the address equals the sum of the values of the individual SUI token objects, in this case, 1.105 SUI.

You can merge and split SUI token objects to create token objects with specific values. To create a SUI token worth .6 SUI, split the token worth 1 SUI into two token objects worth .6 SUI and .4 SUI.

To transfer a specific amount of SUI, you need a SUI token worth that specific amount. To get a SUI token with that specific value, you might need to split or merge existing SUI tokens. Sui supports several methods to accomplish this, including some that do not require you to manually split or merge coins.

Sui supports the following API operations related to transferring SUI between addresses:

sui_transferObject Because SUI tokens are objects, you can transfer SUI tokens just like any other object. This method requires a gas token, and is useful in niche cases only.

sui_payAllSui This method accepts an array of SUI token IDs. It merges all existing tokens into one, deducts the gas fee, then sends the merged token to the recipient address.

The method is especially useful if you want to transfer all SUI from an address. To merge together all coins for an address, set the recipient as the same address. This is a native Sui method so is not considered a transaction in Sui.

sui_paySui This operation accepts an array of SUI token IDs, an array of amounts, and an array of recipient addresses.

The amounts and recipients array map one to one. Even if you use only one recipient address, you must include it for each amount in the amount array.

The operation merges all of the tokens provided into one token object and settles the gas fees. It then splits the token according to the amounts in the amounts array and sends the first token to the first recipient, the second token to the second recipient, and so on. Any remaining SUI on the token stays in the source address.

The benefits of this method include: no gas fees for merging or splitting tokens, and the abstracted token merge and split. The sui_paySui operation is a native function, so the merge and split operations are not considered Sui transactions. The gas fees for them match typical transactions on Sui.You can use this operation to split coins in your own address by setting the recipient as your own address. Note that the total value of the input coins must be greater than the total value of the amounts to send.

sui_pay This method is similar to sui_paySui, but it accepts any kind of coin or token instead of only SUI. You must include a gas token, and all of the coins or tokens must be the same type.

sui_transferSui This method accepts only one SUI token object and an amount to send to the recipient. It uses the same token for gas fees, so the amount to transfer must be strictly less than the value of the SUI token used.

Refer to Sui Signatures for more details on signature validity requirements.

The Sui blockchain uses a Delegated Proof-of-Stake mechanism (DPoS). This allows SUI token holders to stake their SUI tokens to any validator of their choice. When someone stakes their SUI tokens, it means those tokens are locked for the entire epoch. Users can withdraw their stake at any time, but new staking requests become active only at the start of the next epoch.

SUI holders who stake their tokens to validators earn rewards for helping secure the Sui network. Sui determines rewards for staking based on stake rewards on the network, and distributes them at the end of each epoch.

The total voting power in the Sui Network is always 10,000. The voting power of each individual validator is similar to basis points. For example, a voting power of 101 = 1.01%. Sui's quorum threshold (number of votes needed to confirm a transaction) is 6,667 (which is greater than 2/3). The voting power for a single validator is capped at 1,000 (10%) regardless of how much stake the validator has.

Sui supports the following API operations related to staking. You can find the source code in the sui_system module.

Track balance changes for an address

You can track balance changes by calling sui_getBalance at predefined intervals. This call returns the total balance for an address. The total includes any coin or token type, but this document focuses on SUI. You can track changes in the total balance for an address between subsequent sui_getBalance requests.

The following bash example demonstrates how to use sui_getBalance for address 0x849d63687330447431a2e76fecca4f3c10f6884ebaa9909674123c6c662612a3 . If you use a network other than Devnet, replace the value for rpc with the URL to the appropriate Full node.

The response is a JSON object that includes the totalBalance for the address:

The following example demonstrates using sui_getBalance in Rust:

You can also track the balance for an address by subscribing to all of the events emitted from it. Use a filter to include only the events related to SUI coins, such as when the address acquires a coin or pays for a gas fee. The following example demonstrates how to filter events for an address using bash and cURL:

The response can include a large number of events. Add pagination to the response using the nextCursor key in the request. You can determine the corresponding txDigest and eventSeq from the id field of a transaction.

You can add the txDigest value instead of the first null within the params . The second null is an integer that defines how many results (up to 1000) to return and the true means ascending order. You can use the nextCursor so the response starts from a desired point.

The id field of any transaction looks like:

With this data, create a nextCursor as follows:

Sui is a DAG-based blockchain and uses checkpoints for node synchronization and global transaction ordering. Checkpoints differ from blocks in the following ways:

Sui Checkpoint API operations include:

To transfer a specific amount of SUI between addresses, you need a SUI token object with that specific value. In Sui, everything is an object, including SUI tokens. The amount of SUI in each SUI token object varies. For example, an address could own 3 SUI tokens with different values: one of 0.1 SUI, a second of 1.0 SUI, and a third with 0.005 SUI. The total balance for the address equals the sum of the values of the individual SUI token objects, in this case, 1.105 SUI.

You can merge and split SUI token objects to create token objects with specific values. To create a SUI token worth .6 SUI, split the token worth 1 SUI into two token objects worth .6 SUI and .4 SUI.

To transfer a specific amount of SUI, you need a SUI token worth that specific amount. To get a SUI token with that specific value, you might need to split or merge existing SUI tokens. Sui supports several methods to accomplish this, including some that do not require you to manually split or merge coins.

Sui supports the following API operations related to transferring SUI between addresses:

sui_transferObject Because SUI tokens are objects, you can transfer SUI tokens just like any other object. This method requires a gas token, and is useful in niche cases only.

sui_payAllSui This method accepts an array of SUI token IDs. It merges all existing tokens into one, deducts the gas fee, then sends the merged token to the recipient address.

The method is especially useful if you want to transfer all SUI from an address. To merge together all coins for an address, set the recipient as the same address. This is a native Sui method so is not considered a transaction in Sui.

sui_paySui This operation accepts an array of SUI token IDs, an array of amounts, and an array of recipient addresses.

The amounts and recipients array map one to one. Even if you use only one recipient address, you must include it for each amount in the amount array.

The operation merges all of the tokens provided into one token object and settles the gas fees. It then splits the token according to the amounts in the amounts array and sends the first token to the first recipient, the second token to the second recipient, and so on. Any remaining SUI on the token stays in the source address.

The benefits of this method include: no gas fees for merging or splitting tokens, and the abstracted token merge and split. The sui_paySui operation is a native function, so the merge and split operations are not considered Sui transactions. The gas fees for them match typical transactions on Sui.You can use this operation to split coins in your own address by setting the recipient as your own address. Note that the total value of the input coins must be greater than the total value of the amounts to send.

sui_pay This method is similar to sui_paySui, but it accepts any kind of coin or token instead of only SUI. You must include a gas token, and all of the coins or tokens must be the same type.

sui_transferSui This method accepts only one SUI token object and an amount to send to the recipient. It uses the same token for gas fees, so the amount to transfer must be strictly less than the value of the SUI token used.

Refer to Sui Signatures for more details on signature validity requirements.

The Sui blockchain uses a Delegated Proof-of-Stake mechanism (DPoS). This allows SUI token holders to stake their SUI tokens to any validator of their choice. When someone stakes their SUI tokens, it means those tokens are locked for the entire epoch. Users can withdraw their stake at any time, but new staking requests become active only at the start of the next epoch.

SUI holders who stake their tokens to validators earn rewards for helping secure the Sui network. Sui determines rewards for staking based on stake rewards on the network, and distributes them at the end of each epoch.

The total voting power in the Sui Network is always 10,000. The voting power of each individual validator is similar to basis points. For example, a voting power of 101 = 1.01%. Sui's quorum threshold (number of votes needed to confirm a transaction) is 6,667 (which is greater than 2/3). The voting power for a single validator is capped at 1,000 (10%) regardless of how much stake the validator has.

Sui supports the following API operations related to staking. You can find the source code in the sui_system module.

Use events to track balance changes for an address

You can also track the balance for an address by subscribing to all of the events emitted from it. Use a filter to include only the events related to SUI coins, such as when the address acquires a coin or pays for a gas fee. The following example demonstrates how to filter events for an address using bash and cURL:

The response can include a large number of events. Add pagination to the response using the nextCursor key in the request. You can determine the corresponding txDigest and eventSeq from the id field of a transaction.

You can add the txDigest value instead of the first null within the params . The second null is an integer that defines how many results (up to 1000) to return and the true means ascending order. You can use the nextCursor so the response starts from a desired point.

The id field of any transaction looks like:

With this data, create a nextCursor as follows:

Sui is a DAG-based blockchain and uses checkpoints for node synchronization and global transaction ordering. Checkpoints differ from blocks in the following ways:

Sui Checkpoint API operations include:

To transfer a specific amount of SUI between addresses, you need a SUI token object with that specific value. In Sui, everything is

an object, including SUI tokens. The amount of SUI in each SUI token object varies. For example, an address could own 3 SUI tokens with different values: one of 0.1 SUI, a second of 1.0 SUI, and a third with 0.005 SUI. The total balance for the address equals the sum of the values of the individual SUI token objects, in this case, 1.105 SUI.

You can merge and split SUI token objects to create token objects with specific values. To create a SUI token worth .6 SUI, split the token worth 1 SUI into two token objects worth .6 SUI and .4 SUI.

To transfer a specific amount of SUI, you need a SUI token worth that specific amount. To get a SUI token with that specific value, you might need to split or merge existing SUI tokens. Sui supports several methods to accomplish this, including some that do not require you to manually split or merge coins.

Sui supports the following API operations related to transferring SUI between addresses:

sui_transferObject Because SUI tokens are objects, you can transfer SUI tokens just like any other object. This method requires a gas token, and is useful in niche cases only.

sui_payAllSui This method accepts an array of SUI token IDs. It merges all existing tokens into one, deducts the gas fee, then sends the merged token to the recipient address.

The method is especially useful if you want to transfer all SUI from an address. To merge together all coins for an address, set the recipient as the same address. This is a native Sui method so is not considered a transaction in Sui.

sui_paySui This operation accepts an array of SUI token IDs, an array of amounts, and an array of recipient addresses.

The amounts and recipients array map one to one. Even if you use only one recipient address, you must include it for each amount in the amount array.

The operation merges all of the tokens provided into one token object and settles the gas fees. It then splits the token according to the amounts in the amounts array and sends the first token to the first recipient, the second token to the second recipient, and so on. Any remaining SUI on the token stays in the source address.

The benefits of this method include: no gas fees for merging or splitting tokens, and the abstracted token merge and split. The sui_paySui operation is a native function, so the merge and split operations are not considered Sui transactions. The gas fees for them match typical transactions on Sui.You can use this operation to split coins in your own address by setting the recipient as your own address. Note that the total value of the input coins must be greater than the total value of the amounts to send.

sui_pay This method is similar to sui_paySui, but it accepts any kind of coin or token instead of only SUI. You must include a gas token, and all of the coins or tokens must be the same type.

sui_transferSui This method accepts only one SUI token object and an amount to send to the recipient. It uses the same token for gas fees, so the amount to transfer must be strictly less than the value of the SUI token used.

Refer to Sui Signatures for more details on signature validity requirements.

The Sui blockchain uses a Delegated Proof-of-Stake mechanism (DPoS). This allows SUI token holders to stake their SUI tokens to any validator of their choice. When someone stakes their SUI tokens, it means those tokens are locked for the entire epoch. Users can withdraw their stake at any time, but new staking requests become active only at the start of the next epoch.

SUI holders who stake their tokens to validators earn rewards for helping secure the Sui network. Sui determines rewards for staking based on stake rewards on the network, and distributes them at the end of each epoch.

The total voting power in the Sui Network is always 10,000. The voting power of each individual validator is similar to basis points. For example, a voting power of 101 = 1.01%. Sui's quorum threshold (number of votes needed to confirm a transaction) is 6,667 (which is greater than 2/3). The voting power for a single validator is capped at 1,000 (10%) regardless of how much stake the validator has.

Sui supports the following API operations related to staking. You can find the source code in the sui_system module.

Blocks vs checkpoints

Sui is a DAG-based blockchain and uses checkpoints for node synchronization and global transaction ordering. Checkpoints differ from blocks in the following ways:

Sui Checkpoint API operations include:

To transfer a specific amount of SUI between addresses, you need a SUI token object with that specific value. In Sui, everything is an object, including SUI tokens. The amount of SUI in each SUI token object varies. For example, an address could own 3 SUI tokens with different values: one of 0.1 SUI, a second of 1.0 SUI, and a third with 0.005 SUI. The total balance for the address equals the sum of the values of the individual SUI token objects, in this case, 1.105 SUI.

You can merge and split SUI token objects to create token objects with specific values. To create a SUI token worth .6 SUI, split the token worth 1 SUI into two token objects worth .6 SUI and .4 SUI.

To transfer a specific amount of SUI, you need a SUI token worth that specific amount. To get a SUI token with that specific value, you might need to split or merge existing SUI tokens. Sui supports several methods to accomplish this, including some that do not require you to manually split or merge coins.

Sui supports the following API operations related to transferring SUI between addresses:

sui_transferObject Because SUI tokens are objects, you can transfer SUI tokens just like any other object. This method requires a gas token, and is useful in niche cases only.

sui_payAllSui This method accepts an array of SUI token IDs. It merges all existing tokens into one, deducts the gas fee, then sends the merged token to the recipient address.

The method is especially useful if you want to transfer all SUI from an address. To merge together all coins for an address, set the recipient as the same address. This is a native Sui method so is not considered a transaction in Sui.

sui_paySui This operation accepts an array of SUI token IDs, an array of amounts, and an array of recipient addresses.

The amounts and recipients array map one to one. Even if you use only one recipient address, you must include it for each amount in the amount array.

The operation merges all of the tokens provided into one token object and settles the gas fees. It then splits the token according to the amounts in the amounts array and sends the first token to the first recipient, the second token to the second recipient, and so on. Any remaining SUI on the token stays in the source address.

The benefits of this method include: no gas fees for merging or splitting tokens, and the abstracted token merge and split. The sui_paySui operation is a native function, so the merge and split operations are not considered Sui transactions. The gas fees for them match typical transactions on Sui.You can use this operation to split coins in your own address by setting the recipient as your own address. Note that the total value of the input coins must be greater than the total value of the amounts to send.

sui_pay This method is similar to sui_paySui, but it accepts any kind of coin or token instead of only SUI. You must include a gas token, and all of the coins or tokens must be the same type.

sui_transferSui This method accepts only one SUI token object and an amount to send to the recipient. It uses the same token for gas fees, so the amount to transfer must be strictly less than the value of the SUI token used.

Refer to Sui Signatures for more details on signature validity requirements.

The Sui blockchain uses a Delegated Proof-of-Stake mechanism (DPoS). This allows SUI token holders to stake their SUI tokens to any validator of their choice. When someone stakes their SUI tokens, it means those tokens are locked for the entire epoch. Users can withdraw their stake at any time, but new staking requests become active only at the start of the next epoch.

SUI holders who stake their tokens to validators earn rewards for helping secure the Sui network. Sui determines rewards for staking based on stake rewards on the network, and distributes them at the end of each epoch.

The total voting power in the Sui Network is always 10,000. The voting power of each individual validator is similar to basis points. For example, a voting power of 101 = 1.01%. Sui's quorum threshold (number of votes needed to confirm a transaction) is 6,667 (which is greater than 2/3). The voting power for a single validator is capped at 1,000 (10%) regardless of how much stake the validator has.

Sui supports the following API operations related to staking. You can find the source code in the sui_system module.

SUI balance transfer

To transfer a specific amount of SUI between addresses, you need a SUI token object with that specific value. In Sui, everything is an object, including SUI tokens. The amount of SUI in each SUI token object varies. For example, an address could own 3 SUI tokens with different values: one of 0.1 SUI, a second of 1.0 SUI, and a third with 0.005 SUI. The total balance for the address

equals the sum of the values of the individual SUI token objects, in this case, 1.105 SUI.

You can merge and split SUI token objects to create token objects with specific values. To create a SUI token worth .6 SUI, split the token worth 1 SUI into two token objects worth .6 SUI and .4 SUI.

To transfer a specific amount of SUI, you need a SUI token worth that specific amount. To get a SUI token with that specific value, you might need to split or merge existing SUI tokens. Sui supports several methods to accomplish this, including some that do not require you to manually split or merge coins.

Sui supports the following API operations related to transferring SUI between addresses:

sui_transferObject Because SUI tokens are objects, you can transfer SUI tokens just like any other object. This method requires a gas token, and is useful in niche cases only.

sui_payAllSui This method accepts an array of SUI token IDs. It merges all existing tokens into one, deducts the gas fee, then sends the merged token to the recipient address.

The method is especially useful if you want to transfer all SUI from an address. To merge together all coins for an address, set the recipient as the same address. This is a native Sui method so is not considered a transaction in Sui.

sui_paySui This operation accepts an array of SUI token IDs, an array of amounts, and an array of recipient addresses.

The amounts and recipients array map one to one. Even if you use only one recipient address, you must include it for each amount in the amount array.

The operation merges all of the tokens provided into one token object and settles the gas fees. It then splits the token according to the amounts in the amounts array and sends the first token to the first recipient, the second token to the second recipient, and so on. Any remaining SUI on the token stays in the source address.

The benefits of this method include: no gas fees for merging or splitting tokens, and the abstracted token merge and split. The sui_paySui operation is a native function, so the merge and split operations are not considered Sui transactions. The gas fees for them match typical transactions on Sui.You can use this operation to split coins in your own address by setting the recipient as your own address. Note that the total value of the input coins must be greater than the total value of the amounts to send.

sui_pay This method is similar to sui_paySui, but it accepts any kind of coin or token instead of only SUI. You must include a gas token, and all of the coins or tokens must be the same type.

sui_transferSui This method accepts only one SUI token object and an amount to send to the recipient. It uses the same token for gas fees, so the amount to transfer must be strictly less than the value of the SUI token used.

Refer to Sui Signatures for more details on signature validity requirements.

The Sui blockchain uses a Delegated Proof-of-Stake mechanism (DPoS). This allows SUI token holders to stake their SUI tokens to any validator of their choice. When someone stakes their SUI tokens, it means those tokens are locked for the entire epoch. Users can withdraw their stake at any time, but new staking requests become active only at the start of the next epoch.

SUI holders who stake their tokens to validators earn rewards for helping secure the Sui network. Sui determines rewards for staking based on stake rewards on the network, and distributes them at the end of each epoch.

The total voting power in the Sui Network is always 10,000. The voting power of each individual validator is similar to basis points. For example, a voting power of 101 = 1.01%. Sui's quorum threshold (number of votes needed to confirm a transaction) is 6,667 (which is greater than 2/3). The voting power for a single validator is capped at 1,000 (10%) regardless of how much stake the validator has.

Sui supports the following API operations related to staking. You can find the source code in the sui_system module.

Sui API operations for transfers

Sui supports the following API operations related to transferring SUI between addresses:

sui_transferObject Because SUI tokens are objects, you can transfer SUI tokens just like any other object. This method requires a gas token, and is useful in niche cases only.

sui_payAllSui This method accepts an array of SUI token IDs. It merges all existing tokens into one, deducts the gas fee, then sends

the merged token to the recipient address.

The method is especially useful if you want to transfer all SUI from an address. To merge together all coins for an address, set the recipient as the same address. This is a native Sui method so is not considered a transaction in Sui.

sui_paySui This operation accepts an array of SUI token IDs, an array of amounts, and an array of recipient addresses.

The amounts and recipients array map one to one. Even if you use only one recipient address, you must include it for each amount in the amount array.

The operation merges all of the tokens provided into one token object and settles the gas fees. It then splits the token according to the amounts in the amounts array and sends the first token to the first recipient, the second token to the second recipient, and so on. Any remaining SUI on the token stays in the source address.

The benefits of this method include: no gas fees for merging or splitting tokens, and the abstracted token merge and split. The sui_paySui operation is a native function, so the merge and split operations are not considered Sui transactions. The gas fees for them match typical transactions on Sui.You can use this operation to split coins in your own address by setting the recipient as your own address. Note that the total value of the input coins must be greater than the total value of the amounts to send.

sui_pay This method is similar to sui_paySui, but it accepts any kind of coin or token instead of only SUI. You must include a gas token, and all of the coins or tokens must be the same type.

sui_transferSui This method accepts only one SUI token object and an amount to send to the recipient. It uses the same token for gas fees, so the amount to transfer must be strictly less than the value of the SUI token used.

Refer to Sui Signatures for more details on signature validity requirements.

The Sui blockchain uses a Delegated Proof-of-Stake mechanism (DPoS). This allows SUI token holders to stake their SUI tokens to any validator of their choice. When someone stakes their SUI tokens, it means those tokens are locked for the entire epoch. Users can withdraw their stake at any time, but new staking requests become active only at the start of the next epoch.

SUI holders who stake their tokens to validators earn rewards for helping secure the Sui network. Sui determines rewards for staking based on stake rewards on the network, and distributes them at the end of each epoch.

The total voting power in the Sui Network is always 10,000. The voting power of each individual validator is similar to basis points. For example, a voting power of 101 = 1.01%. Sui's quorum threshold (number of votes needed to confirm a transaction) is 6,667 (which is greater than 2/3). The voting power for a single validator is capped at 1,000 (10%) regardless of how much stake the validator has.

Sui supports the following API operations related to staking. You can find the source code in the sui_system module.

Signing transactions

Refer to Sui Signatures for more details on signature validity requirements.

The Sui blockchain uses a Delegated Proof-of-Stake mechanism (DPoS). This allows SUI token holders to stake their SUI tokens to any validator of their choice. When someone stakes their SUI tokens, it means those tokens are locked for the entire epoch. Users can withdraw their stake at any time, but new staking requests become active only at the start of the next epoch.

SUI holders who stake their tokens to validators earn rewards for helping secure the Sui network. Sui determines rewards for staking based on stake rewards on the network, and distributes them at the end of each epoch.

The total voting power in the Sui Network is always 10,000. The voting power of each individual validator is similar to basis points. For example, a voting power of 101 = 1.01%. Sui's quorum threshold (number of votes needed to confirm a transaction) is 6,667 (which is greater than 2/3). The voting power for a single validator is capped at 1,000 (10%) regardless of how much stake the validator has.

Sui supports the following API operations related to staking. You can find the source code in the sui_system module.

SUI Staking

The Sui blockchain uses a Delegated Proof-of-Stake mechanism (DPoS). This allows SUI token holders to stake their SUI tokens to any validator of their choice. When someone stakes their SUI tokens, it means those tokens are locked for the entire epoch.

Users can withdraw their stake at any time, but new staking requests become active only at the start of the next epoch.

SUI holders who stake their tokens to validators earn rewards for helping secure the Sui network. Sui determines rewards for staking based on stake rewards on the network, and distributes them at the end of each epoch.

The total voting power in the Sui Network is always 10,000. The voting power of each individual validator is similar to basis points. For example, a voting power of 101 = 1.01%. Sui's quorum threshold (number of votes needed to confirm a transaction) is 6,667 (which is greater than 2/3). The voting power for a single validator is capped at 1,000 (10%) regardless of how much stake the validator has.

Sui supports the following API operations related to staking. You can find the source code in the sui_system module.

Staking functions

Sui supports the following API operations related to staking. You can find the source code in the sui_system module.