# dryRunTransactionBlock

Simulate running a transaction to inspect its effects without committing to them on-chain.

`txBytes` either a `TransactionData` struct or a `TransactionKind` struct, BCS-encoded and then Base64-encoded. The expected type is controlled by the presence or absence of `txMeta`: If present, `txBytes` is assumed to be a `TransactionKind`, if absent, then `TransactionData`.

`txMeta` the data that is missing from a `TransactionKind` to make a `TransactionData` (sender address and gas information). All its fields are nullable.

`skipChecks` optional flag to disable the usual verification checks that prevent access to objects that are owned by addresses other than the sender, and calling non-public, non-entry functions, and some other checks. Defaults to false.

```
dryRunTransactionBlock(
    txBytes: String!
    txMeta: TransactionMetadata
    skipChecks: Boolean
): DryRunResult!
```

## Arguments

`dryRunTransactionBlock.txBytes`.`String!` **non-null** **scalar**

`dryRunTransactionBlock.txMeta`.`TransactionMetadata` **input**

`dryRunTransactionBlock.skipChecks`.`Boolean` **scalar**

## Type

`DryRunResult` **object**

✏️ Edit this page

# Sui