

# Sui RPC

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

SuiJSON is a JSON-based format with restrictions that allow Sui to align JSON inputs more closely with Move call arguments.

This table shows the restrictions placed on JSON types to make them SuiJSON compatible:

Due to the loosely typed nature of JSON/SuiJSON and the strongly typed nature of Move types, you sometimes need to overload SuiJSON types to represent multiple Move types.

For example SuiJSON::Number can represent both u8 and u32 . This means you have to coerce and sometimes convert types.

Which type you coerce depends on the expected Move type. For example, if the Move function expects a u8, you must have received a SuiJSON::Number with a value less than 256. More importantly, you have no way to easily express Move addresses in JSON, so you encode them as hex strings prefixed by 0x .

Additionally, Move supports u128 and u256 but JSON doesn't. As a result Sui allows encoding numbers as strings.

## Type coercion reasoning

Due to the loosely typed nature of JSON/SuiJSON and the strongly typed nature of Move types, you sometimes need to overload SuiJSON types to represent multiple Move types.

For example SuiJSON::Number can represent both u8 and u32 . This means you have to coerce and sometimes convert types.

Which type you coerce depends on the expected Move type. For example, if the Move function expects a u8, you must have received a SuiJSON::Number with a value less than 256. More importantly, you have no way to easily express Move addresses in JSON, so you encode them as hex strings prefixed by 0x .

Additionally, Move supports u128 and u256 but JSON doesn't. As a result Sui allows encoding numbers as strings.

## Type coercion rules