Sui RPC

> **INFO**
>
> The Sui RPC is upgrading from JSON-RPC to GraphQL. See GraphQL for Sui RPC for more information.

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:

JSON	SuiJSON Restrictions	Move Type Mapping
Number	Must be unsigned integer	u8, u6, u32, u64 (encoded as String), u128 (encoded as String), u256 (encoded as String)
String	No restrictions	Vector`<u8>`, Address, ObjectID, TypeTag, Identifier, Unsigned integer (256 bit max)
Boolean	No restrictions	Bool
Array	Must be homogeneous JSON and of SuiJSON type	Vector
Null	Not allowed	N/A
Object	Not allowed	N/A

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

Move Type	SuiJSON Representations	Valid Examples
Bool	Bool	true, false
u8	Supports 3 formats: Unsigned number < 256. Decimal string with value < 256. One byte hex string prefixed with 0x.	7 "70" "0x43"
u16	Three formats are supported Unsigned number < 65536. Decimal string with value < 65536. Two byte hex string prefixed with 0x.	712 "570" "0x423"
u32	Three formats are supported Unsigned number < 4294967296. Decimal string with value < 4294967296. One byte hex string prefixed with 0x.	9823247 "987120" "0x4BADE93"
u64	Supports two formats Decimal string with value < U64::MAX. Up to 8 byte hex	"747944370" "0x2B1A39A15E"

Move Type	SuiJSON Representations	Valid Examples
	string prefixed with 0x.	
u128	Two formats are supported Decimal string with value < U128::MAX. Up to 16 byte hex string prefixed with 0x.	"74794734937420002470" "0x2B1A39A1514E1D
u256	Two formats are supported Decimal string with value < U256::MAX. Up to 32 byte hex string prefixed with 0x.	"747947349374200024707479473493742000 "0x2B1762FECADA39753FCAB2A1514E1D8A
Address	32 byte hex string prefixed with 0x	"0xbc33e6e4818f9f2ef77d020b35c24be738213e64d9e5883
ObjectID	32 byte hex string prefixed with 0x	"0x1b879f00b03357c95a908b7fb568712f5be862c5cb0a589
Identifier	Typically used for module and function names. Encoded as one of the following: A String whose first character is a letter and the remaining characters are letters, digits or underscore. A String whose first character is an underscore, and there is at	"function", "_function", "some_name", "____some_nam

Move Type	SuiJSON Representations	Valid Examples
	least one further letter, digit or underscore	
Vector<Move Type> / Option<Move Type>	Homogeneous vector of aforementioned types including nested vectors of primitive types (only "flat" vectors of ObjectIDs are allowed)	[1,2,3,4]: simple U8 vector [[3,600],[],[0,7,4]]: nestec ["0x2B1A39A1514E1D8A7CE45919CFEB4FE "0x2B1A39A1514E1D8A7CE45919CFEB4FEF"]: Obje
Vector<u8>	For convenience, Sui allows: U8 vectors represented as UTF-8 (and ASCII) strings.	"√®ˆbo72 √∂†Δ˚–œ∑π2ie": UTF-8 "abcdE738-2 _=

Edit this page