

Summary

This document describes a set of custom scalars used in the GraphQL interface to Tezos node data.

Motivation

Using custom scalars tailored to the Tezos types gives a lot of benefits for validation. Projects implementing Tezos GraphQL schema using these scalars will be able to provide validation. It is especially useful for validating hashes and addresses, which are using `base58check` encoding.

Specification

Scalars

```
# Tezos address. Represented as public key hash (Base58Check-encoded) prefixed
with tz1, tz2, tz3 or KT1.
```

```
scalar Address
```

```
# Timestamp specified as a ISO-8601 UTC date string (2020-02-04T15:31:39Z)
```

```
scalar DateTime
```

```
# JSON represents any valid JSON object
```

```
scalar JSON
```

```
# Raw Michelson expression represented as JSON
```

```
scalar MichelsonExpression
```

```
# Arbitrary precision number represented as string in JSON.
```

```
scalar BigNumber
```

```
# Micro tez. Positive bignumber. 1 tez = 1,000,000 micro tez.
```

```
scalar Mutez
```

```
# Operation identifier (Base58Check-encoded) prefixed with o.
```

```
scalar OperationHash
```

```
# Block identifier (Base58Check-encoded) prefixed with B.
```

```
scalar BlockHash
```

```
# Protocol identifier (Base58Check-encoded) prefixed with P.
```

```
scalar ProtocolHash
```

```
# Context identifier (Base58Check-encoded) prefixed with Co.
```

```
scalar ContextHash
```

```
# Operations identifier (Base58Check-encoded) prefixed with LLo (List of a list of
operations).
```

```
scalar OperationsHash
```

```
# Chain identifier (Base58Check-encoded) prefixed with Net.
```

```
scalar ChainId
```

```
# Generic signature (Base58Check-encoded) prefixed with sig.  
scalar Signature  
  
# Public key (Base58Check-encoded) prefixed with edpk, sppk or p2pk.  
scalar PublicKey  
  
# Nonce hash (Base58Check-encoded).  
scalar NonceHash
```

Implementations

Tezos GraphQL schema

Reference Tezos GraphQL schema using the scalars is defined here: <https://gitlab.com/tezos-graphql/schema>

Scalar resolvers

Reference GraphQL scalar resolvers implementation for base58check types is available here: <https://gitlab.com/tezos-graphql/tezos-graphql-nodejs/-/blob/master/src/resolvers/scalars/base58-resolvers.ts>

Related work

TaaS-GraphQL

Reference implementation of the Tezos GraphQL schema and respective scalar resolvers is available here <https://gitlab.com/tezos-graphql/tezos-graphql-nodejs>.

TezosLive.io

Publicly available endpoint exposing Tezos node data over GraphQL using a reference implementation from [tezos-graphql-nodejs](https://gitlab.com/tezos-graphql/tezos-graphql-nodejs) project is hosted at [TezosLive.io](https://tezoslive.io).

EIP-1767: GraphQL interface to Ethereum node data

[EIP-1767: GraphQL interface to Ethereum node data](#) is a specification of the GraphQL interface for Ethereum node data. There are not many similarities between the scalars or their GraphQL schemes as their respective RPCs differ greatly.