



FACULTY
OF INFORMATICS

Masaryk University

NEM block parser and validator

PA193 Project

Richard Kalinec, Michal Karm, Jakub Kremláček

Overview

1. NEM

1.1 Blockchain structure

1.2 Transactions

2. Implementation details

2.1 JSON parsing

2.2 Models

3. Testing

4. Contributions

Blockchain structure

- NEM is capable of multiple chains (public and private)
- Chain is made of blocks
- Blocks contain:
 - timestamp, signature, hash of previous block, type, version, signer and height
 - transactions

Transactions

- NEM currently offers eight transaction types, the main being:
 - transfer transaction version 1 (amount) and 2 (mosaics, namespaces)
 - importance transfer transaction (importance of node)
 - provision namespace transaction
 - mosaic definition creation transaction
 - mosaic supply change transaction
- Mosaics and namespaces, assets, private "currencies" in the public chain, domains, sheep, magic.beans

Overview

1. NEM

1.1 Blockchain structure

1.2 Transactions

2. Implementation details

2.1 JSON parsing

2.2 Models

3. Testing

4. Contributions

JSON parsing

- each block can be downloaded from NEM node network via its REST API
- retrieved JSON file is processed into classes `JsonObject ("{*}")` and `JsonAttribute ("\"NAME\":VAL")`
- Json-classes are then unmarshalled into NEM models
- parser does not support entire JSON syntax - only parts used by NEM json files are supported (f.e.: NULL attributes are not allowed)

Models

All models inherit from Validatable for easy validation via
`bool Validatable::isValid()`

- Block
 - attributes are validated by isValid methods, where possible
- Transaction
 - attributes are sanity checked in their setter methods, e.g. `fee > 0`
 - further validation takes place in isValid methods, e.g. fee calculation

Overview

1. NEM

1.1 Blockchain structure

1.2 Transactions

2. Implementation details

2.1 JSON parsing

2.2 Models

3. Testing

4. Contributions

Testing

- testing was performed using Microsoft Unit Testing Framework for C++
- each class is tested separately
- tests are runnable within Visual Studio or externally as a dll

Overview

1. NEM

1.1 Blockchain structure

1.2 Transactions

2. Implementation details

2.1 JSON parsing

2.2 Models

3. Testing

4. Contributions

Contributions

- Richard Kalinec
 - Flowchart design and preliminary validation design
 - Block attributes (except for transactions) validation
- Michal Karm
 - Transaction validation
 - Transfer version 1 and version 2 transaction testing
- Jakub Kremláček
 - repository + CI maintenance
 - model structure design
 - serialization "module" implementation
 - VS-related issues solving

Thank you for your attention.