<repository>

mina-ns

overview

repository: https://github.com/a6b8/mina-ns/
userName: a6b8
repository: mina-ns
branch: main
date: 2023-08-20T03:44:05+02:00 (1692495845)

<file>

path: /README.md

url: https://github.com/a6b8/mina-ns/blob/main/README.md

Mina Name System (Experiment)

This repository serves as an experiment, and all included implementations will likely be revised at a later point. It serves as a playground to advance the idea of an Inscription-Based Name System for Mina.

Table of Contents

Explorer Name Service

"To ensure that only the address holder may update the information, a transaction must be broadcast on the Mina blockchain utilizing the memo field to the specified address. To identify this is a name request, the memo must begin with Name: You should send an amount of 0 and only pay the transaction fee. "

Memos are limited to 32 characters that include the prefix.

> https://docs.minaexplorer.com/minaexplorer/explorer-name-service

GraphQL Archive Nodes

- Archive Node: https://docs.minaexplorer.com/minaexplorer/berkeley-testnet
- zkIgnite Funding: https://zkignite.minaprotocol.com/zkignite/dev4dev-track-1/funded/suggestion/381

Full Featurelist

Operations

Operations are the core of a command; they define how the command will be handled.

NR	NAME	STRUCT
1	create	single
2	update	single
3	delete	single
4	batch	batch

Single Struct

NR	KEY	REQUIRED	TYPE	OPERATIONS
A	project	true	string	
B	name	true	string	create, delete
C	sources	false	array object	create, update, delete
D	roles	false	array object	create, update, delete

Example

```
{
  "project": "mns::0.01",
  "operation": "create",
  "name": "meow.test",
  "sources": [
    {
      "provider": "ipfs",
      "source": "ipfs://..."
    }
  ],
  "roles": [
    {
      "role": "Owner",
      "address": "B62..."
    }
  ]
}
```

Project

This key determines which parser is responsible for the command. The provided string includes the following components.

NR	NAME	REGEX	DESCRIPTION
1	project tag	mns	A designated abbreviation for Mina Contract System
2	splitter	::	A splitter separates the project tag from the version number
3	version	\d+\.\d{2}	The version number of the protocol for proper evaluation

Example:

```
{
  "project": "mns::0.01"
}
```

Name

The name cannot be updated; it can only be deleted after creation. This field defines the name of the entry.

NR	KEY	REQUIRED	REGEX	ID
1	name	true	\w+\.test	X

Sources

This field can be used to link references to Smart Contracts. Additional fields can be considered, and validation dependencies based on the provider might also be taken into account.

NR	KEY	REQUIRED	REGEX	ID
1	provider	true	`ipfs\	ord`
2	source	true	[.\\d]+	X

Example:

```
{
   ...
   "sources": [
      {
         "provider": "ipfs",
         "source": "ipfs://..."
      }
      ...
   ]
}
```

> The key "sources" is optional, but if used, 1 and 2 must be passed as a JSON structure.

Roles

Roles can be defined here, allowing for the distribution of data management.

NR	KEY	REQUIRED	REGEX	ID
1	role	true	`owner\	contributor`
2	address	true	^B62[a-km-zA-HJ-NP-Z1-9]{52}$	X

Example

```
{
   ...
   "roles": [
      {
         "role": "Owner",
         "address": "B62..."
      }
   ]
}
```

> The key "roles" is optional, but if used, 1 and 2 must be passed as a JSON structure.

Batch Update

Here, various operations can be mixed together, saving space while still allowing for efficient reading.

Example:

```
{
  "project": "mns::0.01",
  "operation": "batch",
  "name": "meow.test",
  "batch": [
    {
      "operation": "create",
      "key": "sources",
      "provider": "ipfs",
      "source": "ipfs://123445556"
    },
    {
      "operation": "update",
      "key": "sources",
      "id": "ipfs://123445556",
      "provider": "ord",
      "source": "ord/123456789"
    },
    {
      "operation": "delete",
      "key": "sources",
      "id": "ipfs://123445556"
    },
    {
      "operation": "create",
      "key": "roles",
      "role": "owner",
      "address": "B62...abc"
    },
    {
      "operation": "update",
      "key": "roles",
      "id": "B62...abc",
      "role": "contributor",
      "address": "B62...xyz"
    },
    {
      "operation": "delete",
      "key": "roles",
      "id": "B62...abc"
    }
  ]
}
```


Minimal Featurelist

- "Name": [create, delete]

Memo Overview

Only 32 bytes are possible

https://mothereff.in/byte-counter

Memo Introduction:

https://garethtdavies.medium.com/prototyping-a-coda-blockchain-explorer-dbe5c12b4ae2

Encoding with bs58:

https://github.com/MinaProtocol/mina/pull/7079#issuecomment-746868482

Example
- E4YfGWVZK4c946WaUWKU1TVBHkjij17A5NK71qnkfgzDEoPfnHzME > aaaa.test

- Module: [base58check](https://pypi.org/project/base58check/)

Test

https://berkeley.minaexplorer.com/transaction/5JtugaKzy5ms55HpVhzBgpDxCAfeFmu2ju11ffWqJA

Contract

https://berkeley.minaexplorer.com/wallet/B62qkvW2gDNdzwnUtM7Zx8dLA8TRNf9MeuycEs5bL46H

User

https://berkeley.minaexplorer.com/wallet/B62qkJ3BSoHtxd7ndHuETioVPEfG4VcNUA7p4x2Y1PfK3

Fetch Memo

Get Address by Name

Known: Name

- NumericHash 24144640
- Memo: E4ZNHV411wMefbDqatmQPx8ZMZYg3cGHm5nCQ9kqy2mBBngMJ2BzX

```
query MyQuery {
  events(sortBy: BLOCKHEIGHT_DESC, query: {event_in: "24144640", zkAppCommandHash:
{zkappCommand: {accountUpdates: {body: {publicKey:
"B62qkvW2gDNdzwnUtM7Zx8dLA8TRNf9MeuycEs5bL46HVCjnZwNMWh5"}}, memo:
"E4ZNHV411wMefbDqatmQPx8ZMZYg3cGHm5nCQ9kqy2mBBngMJ2BzX"}}}) {
    zkAppCommandHash {
      zkappCommand {
        accountUpdates {
          body {
            events
            publicKey
          }
        }
        memo
        feePayer {
          body {
            publicKey
          }
        }
      }
      blockHeight
      dateTime
    }
  }
}
```

Get Name by Address

Known: Address

```
query MyQuery {
  zkapps(sortBy: BLOCKHEIGHT_DESC, query: {zkappCommand: {feePayer: {body: {publicKey:
"B62qkJ3BSoHtxd7ndHuETioVPEfG4VcNUA7p4x2Y1PfK3dPrgG2qyEa"}}, accountUpdates: {body:
{publicKey: "B62qkvW2gDNdzwnUtM7Zx8dLA8TRNf9MeuycEs5bL46HVCjnZwNMWh5"}}}}) {
    zkappCommand {
      accountUpdates {
        body {
          events
          publicKey
        }
      }
      memo
    }
  }
}
```</file>

<file>
path: /convertToFieldNumber.js
url: https://github.com/a6b8/mina-ns/blob/main/convertToFieldNumber.js

~~~
const md5 = require( 'md5' )

const inputString = '{"p":"mns","name":"meow.test"}'
const hash = md5( inputString )

const numericHash = parseInt( hash, 16 ) % 100000000

console.log( numericHash )
~~~
</file>

<file>
path: /createEnvironment.mjs
url: https://github.com/a6b8/mina-ns/blob/main/createEnvironment.mjs

~~~
import { EasyMina } from 'easymina'
import fs from 'fs'

const easyMina = new EasyMina()
await easyMina.setEnvironment( {} )
await easyMina.deployContract( {} )


~~~
</file>

<file>
path: /createMemo.mjs
url: https://github.com/a6b8/mina-ns/blob/main/createMemo.mjs

~~~

import { PrivateKey, PublicKey, Mina, Field } from 'snarkyjs'
import fs from 'fs'
import { Inscription } from './src/Inscription.mjs'

function addPath({ path }) {
    const result = Object
        .entries(path)
        .reduce((acc, a, index) => {
            const [key, value] = a
            const p = &nbsp;.mina/${key}/${value['path']}&nbsp;
            acc[key] = {
                'path': value['path'],
                'content': JSON.parse(fs.readFileSync(p, 'utf-8'))
            }
            return acc
~~~

```
        }, {} )
    return result
}


function addState( { config } ) {
    const state = {
        'accounts': {}
    }
    state['accounts'] = [
        [ 'destination', 'contracts' ],
        [ 'deployer', 'deployers' ]
    ]
        .reduce( ( acc, a, index ) => {
            const [ newKey, oldKey ] = a
            acc[ newKey ] = PrivateKey.fromBase58(
                config['path'][ oldKey ]['content']['data']['address']['private']
            )

            return acc
        }, {} )

    return state
}


const config = {
    'path': {
        'contracts': {
            'path': 'default--1691966956.json',
            'content': null
        },
        'deployers': {
            'path': 'default--1691962167.json',
            'content': null
        }
    }
}


console.log( 'CREATE MEMO' )
console.log( '  Add Path' )
config['path'] = addPath( { 'path': config['path'] } )
const state = addState( { config } )

console.log( '  Set Network' )
const node = 'https://proxy.berkeley.minaexplorer.com/graphql'
const Berkeley = Mina.BerkeleyQANet( node )
Mina.setActiveInstance( Berkeley )

console.log( '  Payload')
```

```
const memo = 'aaaa.test' //&amp;nbsp;{"p":"mns","name":"aaaa.test"}&amp;nbsp;
const inscription = new Inscription( memo )

console.log( &amp;nbsp;    ${memo}&amp;nbsp; )
console.log( &amp;nbsp;    ${inscription.getNumericHash()}&amp;nbsp; )
console.log( &amp;nbsp;    ${inscription.getBase58()}&amp;nbsp; )

console.log( '  Import' )
const { Main } = await import( './workdir/build/default.mjs' )

console.log( '  Compile' )
await Main.compile()

console.log( '  App Instance' )
const zkAppInstance = new Main( state['accounts']['destination'].toPublicKey() )

console.log( '  Transaction' )

const n = inscription.getNumericHash()
const txn1 = await Mina.transaction(
    {
        'feePayerKey': state['accounts']['deployer'],
        'fee': 100_000_000,
        'memo': memo //JSON.stringify( test['payload'] )
    },
    () => { zkAppInstance.mns( Field( inscription.getNumericHash( n ) ) ) }
)

console.log( '  Prove' )
await txn1.prove()
const result = await txn1
    .sign( [ state['accounts']['deployer'] ] )
    .send()


console.log( '  Hash' )
console.log( '    ', result.hash() )

console.log( '  Waiting...' )
await result.wait()




/*
const deployTxn = await this.#snarkyjs.Mina.transaction(
    {
        'feePayerKey': struct['deployer']['encodedPrivate'],
        'fee': struct['transaction']['fee']
    },
    () => {
        this.#snarkyjs.AccountUpdate
```

```
            .fundNewAccount( struct['deployer']['encodedPrivate'] )

        zkApp.deploy( {
            'zkappKey': struct['destination']['encodedPrivate'],
            'verificationKey': this.#contract['verificationKey'],
            'zkAppUri': 'hello-world'
        } )

        zkApp.init(
            struct['destination']['encodedPrivate']
        )
    }
)

const response =  await deployTxn
    .sign( [
        struct['deployer']['encodedPrivate'],
        struct['destination']['encodedPrivate']
    ] )
    .send()

console.log( Main )
*/

/*
const zkApp = new this.#contract['class']( 
    struct['destination']['encodedPrivate'].toPublicKey()
)

const deployTxn = await this.#snarkyjs.Mina.transaction(
    {
        'feePayerKey': struct['deployer']['encodedPrivate'],
        'fee': struct['transaction']['fee']
    },
    () => {
        this.#snarkyjs.AccountUpdate
            .fundNewAccount( struct['deployer']['encodedPrivate'] )

        zkApp.deploy( {
            'zkappKey': struct['destination']['encodedPrivate'],
            'verificationKey': this.#contract['verificationKey'],
            'zkAppUri': 'hello-world'
        } )

        zkApp.init(
            struct['destination']['encodedPrivate']
        )
    }
)

const response =  await deployTxn
    .sign( [
```

```
      struct['deployer']['encodedPrivate'],
      struct['destination']['encodedPrivate']
    ] )
    .send()


*/

~~~
</file>


<file>
## path: /encodeMemo.mjs
url: https://github.com/a6b8/mina-ns/blob/main/encodeMemo.mjs

~~~

import { Inscription } from './src/Inscription.mjs'


const string = &amp;nbsp;aaaa.test                    &amp;nbsp;;
const inscription = new Inscription( string )

console.log( 'ENCODE' )
console.log( '  inscription       ', string )
console.log( '  Base58            ', inscription.getBase58( string ) )
console.log( '  NumericHash       ', inscription.getNumericHash( string ) )

console.log( 'DECODE' )
console.log( '  inscription       ', string )
console.log( '  Base58            ', inscription.decodeBase58( inscription.getBase58( string ) ) )
// console.log( '  NumericHash       ', inscription.decodeNumericHash( inscription.getNumericHash( string ) )
)


// https://garethtdavies.medium.com/prototyping-a-coda-blockchain-explorer-dbe5c12b4ae2

// E4YfGWVZK4c946WaUWKU1TVBHkjij17A5NK71qnkfgzDEoPfnHzME
// aaaa.test

const test = inscription.decodeBase58(
'E4YfGWVZK4c946WaUWKU1TVBHkjij17A5NK71qnkfgzDEoPfnHzME' )
console.log( 'decoded', test )
const test2 = inscription.encodeBase58( test )
console.log( 'encoded', test2 )


// https://github.com/MinaProtocol/mina/pull/7079#issuecomment-746868482
~~~
</file>
```

<file>
path: /package.json
url: https://github.com/a6b8/mina-ns/blob/main/package.json

~~~
{
  "name": "minans",
  "version": "1.0.0",
  "description": "Dieses Repo ist ein Experiment und alle beinhalteten Ausführung werden wahrscheinlich zu einem späteren Zeipunkt überarbeitet. Es dient playground um die Idee eines Inscribtion Based Name System für Mina voranzutreiben.",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "bs58": "^5.0.0",
    "easymina": "^0.1.1",
    "md5": "^2.3.0"
  }
}

~~~
</file>

<file>
path: /src/BatchUpdate.mjs
url: https://github.com/a6b8/mina-ns/blob/main/src/BatchUpdate.mjs

~~~
import fs from 'fs'


export class BatchUpdate {
    #config
    #tests
    #state


    constructor() {
        this.#config = {
            'mns': {
                'name': 'mns',
                'splitter': '::',
                'version': '0.01',
                'receiver': '{{sender}}',
            },
~~~

```
'operations': {
    'single': [ 'create', 'update', 'delete' ],
    'batch': [ 'batch' ]
},
'main': {
    //'required': true,
    'operations': [ 'create', 'delete' ],
    'type': 'single',
    'identifier': 'name',
    'validations': [
        {
            //'required': true,
            'key': 'name',
            'value': 'regex__domainName'
        }
    ]
},
'additionals': {
    'sources': {
        //'required': false,
        'operations': [ 'create', 'update', 'delete' ],
        'type': 'array',
        'identifier': 'source',
        'validations': [
            {
                //'required': true,
                'key': 'provider',
                'value': 'regex__sourceProvider'
            },
            {
                //'required': true,
                'key': 'id',
                'value': 'regex__sourceURL'
            },
            {
                //'required': true,
                'key': 'source',
                'value': 'regex__sourceURL'
            }
        ]
    },
    'roles': {
        //'required': false,
        'operations': [ 'create', 'update', 'delete' ],
        'type': 'array',
        'identifier': 'owner',
        'validations': [
            {
                //'required': true,
                'key': 'role',
                'value': 'regex__role'
```

```
                },
                {
                    //'required': true,
                    'key': 'id',
                    'value': 'regex__minaAddress'
                },
                {
                    //'required': true,
                    'key': 'address',
                    'value': 'regex__minaAddress'
                }
            ]
        }
    },
    'regex': {
        'domainName': "\w+\.test",
        'minaAddress': "^B62[a-km-zA-HJ-NP-Z1-9]{52}{{content}}quot;,
        'sourceType': "SmartContractIMemo",
        'sourceProvider': "ipfslord",
        'sourceURL': "[.\\d]+",
        'role': "OwnerIContributor"
    }
}

return true
}


init() {
    this.#state = {
        'accounts': {}
    }
    return this
}


singleToBatchUpdate( { payload } ) {
    const commands = this.#commandToBatchCommands( { payload } )
    const name = payload['name']
    const batch = this.#getBatchCommand( { commands, name } )
    return batch
}


#getBatchCommand( { commands, name } ) {
    const struct = {
        'project': 'mns::0.01',
        'operation': 'batch',
        'name': name,
        'batch': commands
    }
    return struct
```

```
}


#commandToBatchCommands( { payload } ) {
    if( !Object.hasOwn( payload, 'operation' ) ) {
        console.log(  Payload has not key 'operation'.  )
        process.exit( 1 )
    }

    if( !this.#config['operations']['single'].includes( payload['operation'] ) ) {
        console.log(  Payload uses a wrong 'operation' key (${payload['operation']}).  )
        process.exit( 1 )
    }

    let messages = {}
    const commands = [
     //   [ 'name', this.#config['main'], [ payload ] ],
        [ 'sources', this.#config['additionals']['sources'], payload['sources'] ],
        [ 'roles', this.#config['additionals']['roles'], payload['roles'] ]
    ]
        .reduce( ( acc, a, index ) => {
            const [ key, validation, groups ] = a

            if( groups !== undefined ) {
                messages = groups
                    .reduce( ( aaa, group, pindex ) => {
                        const [ valid, messages ] = this.#validateKey( {
                            'validation': validation,
                            'object': group
                        } )

                        !Object.hasOwn( aaa, key ) ? aaa[ key ] = [] : ''
                        aaa[ key ].push( messages )
                        aaa[ key ] = aaa[ key ].flat( 1 )

                        if( valid ) {
                            const struct = {
                                'key': key,
                                'operation': payload['operation']
                            }

                            const cmd = validation['validations']
                                .reduce( ( abb, b, rindex ) => {
                                    abb[ b['key'] ] = group[ b['key'] ]
                                    return abb
                                }, struct )

                            acc.push( cmd )
                        }

                        return aaa
                    }, {} )
```

```
            } else {
                // console.log( 'not found')
            }

            return acc
        }, [] )

    Object
        .entries( messages )
        .forEach( ( a, index ) => {
            const [ key, values ] = a

            values
                .forEach( ( value, rindex ) => {
                    index === 0 ? console.log(    ${key}  ) : ''
                    console.log(    - ${value}  )
                } )
        } )

    return commands
}


#validateKey( { validation, object } ) {
    let messages = []

    const result = validation['validations']
        .map( a => {
            const { required, key, value } = a
            const checks = {
                'key': false,
                'value': false,
                'overall': false
            }

            if( Object.hasOwn( object, key ) ) {
                checks['key'] = true
            } else {
                const msg =  Key: ${key} not found. 
                messages.push( msg )
            }

            if( checks['key'] ) {
                const regex = this.#keyPathToValue( {
                    'data': this.#config,
                    'keyPath': value
                } )

                if( object[ key ].match( regex ) !== null ) {
                    checks['value'] = true
                } else {
                    const msg2 =  Value ${key}, ${object[ key ]} is not valid (${regex}). 
```

```
                    messages.push( msg2 )
                }
            }

            if( checks['key'] && checks['value'] ) {
                checks['overall'] = true
            } else {
                checks['overall'] = false
            }

            return checks
        } )

        return [ result, messages ]
    }


    #keyPathToValue( { data, keyPath, separator='__' } ) {
        if( typeof keyPath !== 'string' ) {
            return undefined
        }

        const result = keyPath
            .split( separator )
            .reduce( ( acc, key, index ) => {
                if( !acc ) return undefined
                if( !acc.hasOwnProperty( key ) ) return undefined
                acc = acc[ key ]
                return acc
            }, data )

        return result
    }
}




~~~
</file>


<file>
## path: /src/Inscription.mjs
url: https://github.com/a6b8/mina-ns/blob/main/src/Inscription.mjs

~~~
import bs58 from 'bs58'
import md5 from 'md5';


export class Inscription {
```

```
#state


constructor( string ) {
    this.#state = {
        'jsonString': string,
        'base58': null,
        'numericHash': null
    }

    this.#state['base58'] = this.encodeBase58( string )
    this.#state['numericHash'] = this.encodeNumericHash( string )

    return true
}


getBase58() {
    return this.#state['base58']
}


getNumericHash() {
    return this.#state['numericHash']
}


encodeBase58( string ) {
    const bytes = Buffer.from( string, 'utf-8' )
    const encodedString = bs58.encode( bytes )
    return encodedString
}


encodeBase582( inputString ) {
    const bytes = Buffer.from(inputString, 'utf-8');
    const encodedString = bs58.encode(bytes);

    // Pad the encoded string to ensure it's 32 bytes long
    const padding = '0'.repeat(32 - encodedString.length);
    const encoded32Bytes = padding + encodedString;

    return encoded32Bytes;
}


encodeNumericHash( string ) {
    const hash = md5( string )
    const numericHash = parseInt( hash, 16 ) % 100000000
    return numericHash
}
```

```
    decodeBase58( encodedString ) {
        const decodedBuffer = bs58.decode( encodedString )
        const decodedString = decodedBuffer.toString( 'utf8' )

        const characterString = decodedString
            .split( ',' )
            .map( code => String.fromCharCode( code ) ).join( '' )

        return characterString
    }

/*
    decodeNumericHash( numericHash ) {
        const hexHash = numericHash.toString( 16 )
        const originalValue = md5( hexHash )

        return originalValue
    }
*/
}
~~~
</file>


<file>
## path: /test.py
url: https://github.com/a6b8/mina-ns/blob/main/test.py

~~~
import base58

base58_string = "aaaa.test"

# Decode the Base58 string to bytes
decoded_bytes = base58.b58decode(base58_string)

# Check the length of the decoded bytes
current_length = len(decoded_bytes)

# Pad the bytes to 32 bytes if needed
if current_length < 32:
    padding_length = 32 - current_length
    padded_bytes = b'\x00' * padding_length + decoded_bytes
else:
    padded_bytes = decoded_bytes

# Now you have a 32-byte padded result
print(padded_bytes)


~~~
```

</file>

<file>
path: /testBatchUpdate.mjs
url: https://github.com/a6b8/mina-ns/blob/main/testBatchUpdate.mjs

~~~
import { BatchUpdate } from './src/BatchUpdate.mjs'

const test = {
    'sender': '{{alice}}',
    'payload': {
        'operation': 'update',
        'name': 'meow.test',
        'sources': [
            {
                'provider': 'ipfs',
                'id': 'ipfs://...',
                'source': 'ipfs://...2'
            }
        ],
        'roles': [
            {
                'provider': 'ord',
                'id': 'B62qkJ3BSoHtxd7ndHuETioVPEfG4VcNUA7p4x2Y1PfK3dPrgG2qyEa',
                'role': 'Contributor',
                'address': 'B62qkJ3BSoHtxd7ndHuETioVPEfG4VcNUA7p4x2Y1PfK3dPrgG2qyEb'
            }
        ]
    }
}


const batchupdate = new BatchUpdate()
const batch = batchupdate
    .init()
    .singleToBatchUpdate( { 'payload': test['payload'] } )
~~~
</file>

<file>
path: /tsconfig.json
url: https://github.com/a6b8/mina-ns/blob/main/tsconfig.json

~~~
{
    "compilerOptions": {
        "target": "ES2019",
        "module": "es2022",
        "lib": [
~~~

```
            "dom",
            "esnext"
        ],
        "outDir": "workdir/build/",
        "rootDir": "workdir/typescript/",
        "strict": true,
        "strictPropertyInitialization": false,
        "skipLibCheck": true,
        "forceConsistentCasingInFileNames": true,
        "esModuleInterop": true,
        "resolveJsonModule": true,
        "moduleResolution": "node",
        "experimentalDecorators": true,
        "emitDecoratorMetadata": true,
        "allowJs": true,
        "declaration": false,
        "sourceMap": false,
        "noFallthroughCasesInSwitch": true,
        "allowSyntheticDefaultImports": true,
        "isolatedModules": true
    },
    "include": [
        "workdir/typescript/"
    ],
    "exclude": []
}
~~~
</file>


<file>
## path: /workdir/build/default.js
url: https://github.com/a6b8/mina-ns/blob/main/workdir/build/default.js

~~~
var __decorate = (this && this.__decorate) || function (decorators, target, key, desc) {
    var c = arguments.length, r = c < 3 ? target : desc === null ? desc =
Object.getOwnPropertyDescriptor(target, key) : desc, d;
    if (typeof Reflect === "object" && typeof Reflect.decorate === "function") r = Reflect.decorate(decorators,
target, key, desc);
    else for (var i = decorators.length - 1; i >= 0; i--) if (d = decorators[i]) r = (c < 3 ? d(r) : c > 3 ? d(target, key,
r) : d(target, key)) || r;
    return c > 3 && r && Object.defineProperty(target, key, r), r;
};
var __metadata = (this && this.__metadata) || function (k, v) {
    if (typeof Reflect === "object" && typeof Reflect.metadata === "function") return Reflect.metadata(k, v);
};
import { Field, SmartContract, state, State, method, } from 'snarkyjs';
export class Main extends SmartContract {
    constructor() {
        super(...arguments);
        this.events = { 'easyMina': Field, 'mns': Field };
```

```
        this.num = State();
    }
    init() {
        super.init();
        this.num.set(Field(3));
        this.emitEvent('easyMina', Field(123456789));
    }
    update(square) {
        const currentState = this.num.get();
        this.num.assertEquals(currentState);
        square.assertEquals(currentState.mul(currentState));
        this.num.set(square);
    }
    mns(square) {
        this.emitEvent('mns', Field(square));
    }
}
__decorate([
    state(Field),
    __metadata("design:type", Object)
], Main.prototype, "num", void 0);
__decorate([
    method,
    __metadata("design:type", Function),
    __metadata("design:paramtypes", [Field]),
    __metadata("design:returntype", void 0)
], Main.prototype, "update", null);
__decorate([
    method,
    __metadata("design:type", Function),
    __metadata("design:paramtypes", [Field]),
    __metadata("design:returntype", void 0)
], Main.prototype, "mns", null);

~~~
</file>


<file>
## path: /workdir/build/default.mjs
url: https://github.com/a6b8/mina-ns/blob/main/workdir/build/default.mjs

~~~
var __decorate = (this && this.__decorate) || function (decorators, target, key, desc) {
    var c = arguments.length, r = c < 3 ? target : desc === null ? desc =
Object.getOwnPropertyDescriptor(target, key) : desc, d;
    if (typeof Reflect === "object" && typeof Reflect.decorate === "function") r = Reflect.decorate(decorators,
target, key, desc);
    else for (var i = decorators.length - 1; i >= 0; i--) if (d = decorators[i]) r = (c < 3 ? d(r) : c > 3 ? d(target, key,
r) : d(target, key)) || r;
    return c > 3 && r && Object.defineProperty(target, key, r), r;
};
```

```
var __metadata = (this && this.__metadata) || function (k, v) {
    if (typeof Reflect === "object" && typeof Reflect.metadata === "function") return Reflect.metadata(k, v);
};
import { Field, SmartContract, state, State, method, } from 'snarkyjs';
export class Main extends SmartContract {
    constructor() {
        super(...arguments);
        this.events = { 'easyMina': Field, 'mns': Field };
        this.num = State();
    }
    init() {
        super.init();
        this.num.set(Field(3));
        this.emitEvent('easyMina', Field(123456789));
    }
    update(square) {
        const currentState = this.num.get();
        this.num.assertEquals(currentState);
        square.assertEquals(currentState.mul(currentState));
        this.num.set(square);
    }
    mns(square) {
        this.emitEvent('mns', Field(square));
    }
}
__decorate([
    state(Field),
    __metadata("design:type", Object)
], Main.prototype, "num", void 0);
__decorate([
    method,
    __metadata("design:type", Function),
    __metadata("design:paramtypes", [Field]),
    __metadata("design:returntype", void 0)
], Main.prototype, "update", null);
__decorate([
    method,
    __metadata("design:type", Function),
    __metadata("design:paramtypes", [Field]),
    __metadata("design:returntype", void 0)
], Main.prototype, "mns", null);

~~~
</file>


<file>
## path: /workdir/typescript/default.ts
url: https://github.com/a6b8/mina-ns/blob/main/workdir/typescript/default.ts

~~~
import {
```

```
  Field,
  SmartContract,
  state,
  State,
  method,
} from 'snarkyjs';
export class Main extends SmartContract {
    events = { 'easyMina': Field, 'mns': Field };
    @state(Field) num = State<Field>();


    init() {
        super.init();
        this.num.set( Field( 3 ) );
        this.emitEvent( 'easyMina', Field( 123456789 ) );
    }

    @method update( square: Field ) {
        const currentState = this.num.get();
        this.num.assertEquals( currentState );
        square.assertEquals( currentState.mul( currentState ) );
        this.num.set( square );
    }

    @method mns( square: Field ) {
        this.emitEvent( 'mns', Field( square ) );
    }
}
~~~
</file>

</repository>
```