KINGSLAND
SCHOOL OF BLOCKCHAIN

Rest API For The Practical Project

Reference Guide

Rest API for the practical project

In this document, you will see the endpoints for your node along with their inputs and outputs.
Each endpoint is implemented in our project which you can find here:
https://stormy-everglades-34766.herokuapp.com

General information

Endpoint for receiving general information about the node.
URL : http://{host}:{port}/info
Method : GET
Output :

```
{
    "about"  :  "KingslandChain/0.9-csharp",
    "nodeId"  :  "1a22d3...9b2f",
    "chainId": "c6da93eb...c47f",
    "nodeUrl"  :  "http://chain-node-03.herokuapp.com",
    "peers"  :  2,
    "currentDifficulty"  :  5,
    "blocksCount"  :  25,
    "cumulativeDifficulty"  :  127,
    "confirmedTransactions"  : 208,
    "pendingTransactions"  :
7 }
```

- **about** – The name of the node

- **nodeId** – Identifier of the node (hash of Datetime + some random)
- **peers** – The number of peers connected to the node
- **chainId** – The identifier of the chain, if you have different chains (i.e. different genesis block)
- **currentDifficulty** – The current difficulty of the network
- **blocksCount** – The number of blocks in the node's chain
- **confirmedTransactions** – transactions that have been included in a block
- **pendingTransactions** – transactions sitting in the Mempool (transactions that are waiting to be included in a block)

Debug endpoint

This endpoint will print everything about the node. The blocks, peers, chain, pending transactions and much more.

URL : http://{host}:{port}/debug

Method : GET

Output :

```
{ "selfUrl": "http://localhost:5555", "peers": { ... },
  "chain": {
    "blocks": [{"index": 0, "transactions":[...], "difficulty": 0,
      "prevBlockHash" : "d9...9c", "minedBy" : "af...b2", "nonce": 0,
      "blockDataHash" : "af25...d9", "dateCreated" : "2018-01-..."},
      "blockHash" : "c962...a8"}, {...}, {...}],
      "pendingTransactions" : [{...}, ...], "currentDifficulty": 5,
      "miningJobs" : {"e3d8...5f": {...}, "25c1...a8": {...}, }
  }, "confirmedBalances" : {"2a7e...cf": 500020, ... } }
```

- **selfUrl** – the URL of the node
- **chain** – the chain with each block. Each block has ○ **index –** order of the block in the chain **o** **transactions –** array of the transactions ○ **Difficulty** – the difficulty of the block
    - ○ **minedBy** – address of the miner who submitted the block ○ **blockDataHash** – The hash of all the data in the block ○

        - **blockHash** – the hash of the blockDataHash plus the nonce
        - **nonce** – The proof for the block: (nonce + blockHash) need to hash to a value below the difficulty value
    - **dateCreated** – The timestamp of the block
- pendingTransactions – the array of pending transactions
- confirmedBalances – The balances of everyone

Reset the chain Endpoint

This endpoint will reset the chain and start it from the beginning; this is used only for debugging.

URL: http://{host}:{port}/reset-chain
Method : GET **Output**
:

```
{
    "message" : "The chain was reset to its genesis block"
}
```

All blocks Endpoint

The endpoint will print all the blocks in the node's chain.

URL: http://{host}:{port}/blocks
Method : GET

Output :

```
[
    {
        "index": 0,
        "transactions": [
            {
                "from": "0000000000000000000000000000000000000000",
                "to" : "f3a1e69b6176052fcc4a3248f1c5a91dea308ca9",
                "value" : 1000000000000,
                "fee": 0,
                "dateCreated": "2018-01-01T00:00:00.000Z"
,                 "data": "genesis Tx",
"senderPubKey":
"0000000000000000000000000000000000000000000000000000000000000000",
"transactionDataHash":
"8a684cb8491ee419e7d46a0fd2438cad82d1278c340b5d01974e7beb6b72ecc2",
                "senderSignature" : [
```

```
"0000000000000000000000000000000000000000000000000000000000000000",
"0000000000000000000000000000000000000000000000000000000000000000"
                ],
                "minedInBlockIndex": 0,
                "transferSuccessful": true
            }
        ],
        "difficulty": 0,
        "minedBy": "0000000000000000000000000000000000000000",
"blockDataHash":
"15cc5052fb3c307dd2bfc6bcaa057632250ee05104e4fb7cc75e59db1a92cefc",
"nonce": 0,
        "dateCreated": "2018-01-01T00:00:00.000Z",
"blockHash":
"c6da93eb4249cb5ff4f9da36e2a7f8d0d61999221ed6910180948153e71cc47f"
    }
]
```


The output will be an array of blocks. Each block has

- **index –** the order of the block in the chain
- **Transactions –** array of transactions that are included in the block ○ **From –** Address of the sender ○ **To –** Address of the receiver ○ **Value –** the amount of money ○ **Fee –** the fee for the transaction
    - **dateCreated –** the timestamp of the transaction
    - **data –** Some additional data, if you want to add some message to the transaction
    - **senderPubKey –** The public key of the sender ○ **senderSignature –** The signature of the sender
    - **minedInBlockIndex –** the block index in which the transaction is mined
    - **transferSuccessful –** true if the transaction is mined, false if it has not been mined
- **Difficulty** – the difficulty of the block

- **minedBy** – the miner's address
- **blockDataHash** – The hash of the data in the block
- **blockHash** – the hash of the blockDataHash plus the nonce
- **nonce** – The proof for the block, (nonce + blockHash) need to hash to a value below the difficulty value
- **dateCreated** – The timestamp of the block

Block by Index Endpoint

The endpoint will print the block with the index that you specify

URL : http://{host}:{port}/block/**{index}**
Method : GET
Input : index
Output :

```
{
    "index" : 1,
    "transactions" : [
        {
            "from" : "0000000000000000000000000000000000000000",
            "to" : "84ede81c58f5c490fc6e1a3035789eef897b5b35",
            "value" : 5000020,
            "fee" : 0,
            "dateCreated" : "2018-09-04T13:01:04.969Z",
"data" : "coinbase tx",              "senderPubKey" :
"0000000000000000000000000000000000000000000000000000000000000000",
"transactionDataHash" :
"b68df93232251cc0773bf384b3f90fafaeab0097e7e060f31f5fa413939e4dfa",
            "senderSignature" : [

"0000000000000000000000000000000000000000000000000000000000000000",
"0000000000000000000000000000000000000000000000000000000000000000"
            ],
            "minedInBlockIndex" : 1,
            "transferSuccessful" : true
        }
    ],
```

```
    "difficulty" : 1,
"prevBlockHash" :
"c6da93eb4249cb5ff4f9da36e2a7f8d0d61999221ed6910180948153e71cc47f",
"minedBy" : "84ede81c58f5c490fc6e1a3035789eef897b5b35",     "blockDataHash"
:
"c895c10776cf25499f045b1e3147806b969a3f7521a9ea6b1013447f3873495c",
"nonce" : 14,
    "dateCreated" : "2018-09-04T13:01:04.972Z",
"blockHash" :
"08065b32d9c776700b86a1efc6d778ccbda17932a54ed7b829aa3565588e2817"
}
```


- **Index –** The index of the block
- **Transactions –** The transactions that are included in the block ○ **From –** Address of the sender ○ **To –** Address of the receiver ○ **Value –** the amount of money ○ **Fee –** the fee for the transaction
    - **dateCreated –** the timestamp of the transaction
    - **data –** Some additional data, if you want to add some message to the transaction
    - **senderPubKey –** The public key of the sender ○ **senderSignature –** The signature of the sender
    - **minedInBlockIndex –** the block index in which the transaction is mined
    - **transferSuccessful –** true if the transaction is mined, false if it has not mined
- **Difficulty** – the difficulty of the block
- **minedBy** – the miner's address
- **blockDataHash** – The hash of the data in the block
- **blockHash** – the hash of the blockDataHash plus the nonce
- **nonce** – The proof for the block, (nonce + blockHash) need to hash to a value below the difficulty value
- **dateCreated** – The timestamp of the block
- **prevBlockHash** – The previous block hash

Get Pending Transactions Endpoint

This endpoint will print the list with transactions that have not been mined.

URL : http://{host}:{port}/transactions/pending
Method : GET
Output :


```
[
    {
        "from" : "f3a1e69b6176052fcc4a3248f1c5a91dea308ca9",
        "to" : "a1de0763f26176c6d68cc77e0a1c2c42045f2314",
        "value" : 400000,
        "fee" : 10,
        "dateCreated" : "2018-09-04T12:54:24.839Z",
"data" : "Faucet -> Alice (again)",         "senderPubKey"
:
"8c4431db61e9095d5794ff53a3ae4171c766cadef015f2e11bec22b98a80f74a0",
"transactionDataHash" :
"356c5628e7ab659b1d25765e332cfe6eec318008b96d0eba4dfd677032cc670b",
        "senderSignature" : [
            "7787cc91d311b6e5d04acda388f1ce01990b636d8a8026e0fe86704e12c5c1ed",
            "6293c000ae4af69510f939d3f459e6d7e1da464ec91c7a0a08dd00dc0b3a6cdc"
        ]
    }
]
```


- **From –** Address of the sender
- **To –** Address of the receiver
- **Value –** the amount of money
- **Fee –** the fee of the transaction

- **dateCreated –** the timestamp of the transaction
- **data –** Some additional data, if you want to add some message to the transaction
- **senderPubKey –** The public key of the sender
- **senderSignature –** The signature of the sender
- **minedInBlockIndex –** the block index in which the transaction is mined

- **transferSuccessful –** true if the transaction is mined, false if it has not been mined

Get Confirmed Transactions

This endpoint will print the list of the transactions that are included in blocks.

URL: http://{host}:{port}/transactions/confirmed
Method : GET
Output:


```
[
    {
        "from" : "0000000000000000000000000000000000000000",
        "to" : "f3a1e69b6176052fcc4a3248f1c5a91dea308ca9",
        "value" : 1000000000000,
        "fee" : 0,
        "dateCreated" : "2018-01-01T00:00:00.000Z",
"data" : "genesis tx",          "senderPubKey" :
"0000000000000000000000000000000000000000000000000000000000000000",
"transactionDataHash" :
"8a684cb8491ee419e7d46a0fd2438cad82d1278c340b5d01974e7beb6b72ecc2",
        "senderSignature" : [
            "00000000000000000000000000000000000000000000000000000000000000000000000000000000",
            "00000000000000000000000000000000000000000000000000000000000000000000000000000000"
],
        "minedInBlockIndex" : 0,
        "transferSuccessful" : true
    }
```

-

```
]
```

- **From –** Address of the sender
- **To –** Address of the receiver
- **Value –** the amount of money
- **Fee –** the fee for the transaction
- **dateCreated –** the timestamp of the transaction
  data – Some additional data, if you want to add some message to the transaction
- **senderPubKey –** The public key of the sender
- **senderSignature –** The signature of the sender
- **minedInBlockIndex –** the block index in which the transaction is mined
- **transferSuccessful –** true if the transaction is mined, false if it has not been mined

Get Transaction by Hash Endpoint

This endpoint will return a transaction identified by hash

-

URL : http://{host}:{port}/transactions/{hash}
Method : GET
Output:


```
{
    "from" : "0000000000000000000000000000000000000000",
    "to" : "f3a1e69b6176052fcc4a3248f1c5a91dea308ca9",
    "value" : 1000000000000,
    "fee" : 0,
    "dateCreated" : "2018-01-01T00:00:00.000Z",
"data" : "genesis tx",     "senderPubKey" :
"0000000000000000000000000000000000000000000000000000000000000000",
"transactionDataHash" :
"8a684cb8491ee419e7d46a0fd2438cad82d1278c340b5d01974e7beb6b72ecc2",
    "senderSignature" : [
        "0000000000000000000000000000000000000000000000000000000000000000",
        "0000000000000000000000000000000000000000000000000000000000000000"
],
    "minedInBlockIndex" : 0,
    "transferSuccessful"    :
true }
```


- **From –** Address of the sender
- **To –** Address of the receiver
- **Value –** the amount of money
- **Fee –** the fee for the transaction
- **dateCreated –** the timestamp of the transaction
  data – Some additional data, if you want to add some message to the transaction
- **senderPubKey –** The public key of the sender

-

- **senderSignature –** The signature of the sender
- **minedInBlockIndex –** the block index in which the transaction is mined
- **transferSuccessful –** true if the transaction is mined, false if it has not been mined

List All Account Balance

This endpoint will return all the balances in the network.

URL : http://{host}:{port}/balances
Method : GET
Output:

```
{
    "0000000000000000000000000000000000000000": -1000010000060,
    "f3a1e69b6176052fcc4a3248f1c5a91dea308ca9" : 999998799980,
    "84ede81c58f5c490fc6e1a3035789eef897b5b35" : 10000060,
    "a1de0763f26176c6d68cc77e0a1c2c42045f2314" : 99960,
    "b3d72ad831b3e9cdbdaeda5ff4ae8e9cf182e548" : 1100000,
    "22e2864c613e4f778bb25ddb2b0022d1fbb11c8c" : 0
}
```

List Transactions for Address

This endpoint will print all transactions for address.

URL : http://{host}:{port}/address/{address}/transactions
Method : GET
Input : address
Output:

```
[
    {
```

-

```
        "from" : "0000000000000000000000000000000000000000",
        "to" : "f3a1e69b6176052fcc4a3248f1c5a91dea308ca9",
        "value" : 1000000000000,
```

```
        "fee" : 0,
        "dateCreated" : "2018-01-01T00:00:00.000Z",
"data" : "genesis tx",          "senderPubKey" :
"0000000000000000000000000000000000000000000000000000000000000000",
"transactionDataHash" :
"8a684cb8491ee419e7d46a0fd2438cad82d1278c340b5d01974e7beb6b72ecc2",
        "senderSignature" : [
            "0000000000000000000000000000000000000000000000000000000000000000",
            "0000000000000000000000000000000000000000000000000000000000000000"
],
        "minedInBlockIndex" : 0,
        "transferSuccessful" : true
    }
]
```


- **From –** Address of the sender
- **To –** Address of the receiver
- **Value –** the amount of money
- **Fee –** the fee of the transaction
- **dateCreated –** the timestamp of the transaction
- **data –** Some additional data, if you want to add some message to the transaction
- **senderPubKey –** The public key of the sender
- **senderSignature –** The signature of the sender
- **minedInBlockIndex –** the block index in which the transaction is mined
- **transferSuccessful –** true if the transaction is mined, false if it has not been mined

Get Balance for Address Endpoint

This endpoint will return the balance of a specified address in the network.

URL : http://{host}:{port}/address/{address}/balance **Method** : GET
Input: address
Output:

```
{
    "safeBalance" : 999998799980,
    "confirmedBalance" : 999998799980,
    "pendingBalance" : 999998399970
}
```

- **safeBalance** – 6 confirmations or more
- **confirmedBalance** – 1 or more confirmations
- **pendingBalance** – 0 confirmations

Balances Invalid for Address

If the address is valid but it is not used, return zero for the balance; if it is an invalid address, return an error message.

URL: http://{host}:{port}/address/{invalidAddress}/balance
Method: POST
Input: invalidAddress
Output:

```
{
  "safeBalance" : 0,
  "confirmedBalance" : 0,
  "pendingBalance" : 0
}
```

Send Transaction

With this endpoint, you can broadcast a transaction to the network.

URL: http://{host}:{port}/transactions/send
Method : POST

Input :

```
{
  "from" : "c3293572dbe6ebc60de4a20ed0e21446cae66b17",
  "to" : "f51362b7351ef62253a227a77751ad9b2302f911",
  "value" : 25000, "fee" : 10, "dateCreated" : "2018-02-10T17:53:48.972Z",
"data" : "first payment (50%)", "senderPubKey":
  "c74a8458cd7a7e48f4b7ae6f4ae9f56c5c88c0f03e7c59cb4132b9d9d1600bba1",
  "senderSignature" : ["1aaf55dcb1...68b0", "87250a2841...7960"]
}
```

Output:

```
{ "transactionDataHash" : "cd8d9a345bb208c6f9b8acd6b8eef...20c8a" }
```

Get Mining Job Endpoint

This endpoint will prepare a block candidate and the miner will calculate the nonce for it.

URL: http://{host}:{port}/mining/get-mining-job/{minerAddress}
Method: GET
Input: minerAddress
Output:

```
{
  "index" : 50,
```

```
  "transactionsIncluded" : 17, "difficulty" : 5,
  "expectedReward" : 5000350, "rewardAddress" : "9a9f08...fe917",
  "blockDataHash" : "d2c6ee29ff14b499af985824ea12afccc8...e4cd",
}
```

- **Index** – the order of the block in the chain
- **transactionsIncluded** – the count of the transaction that will be included
- **difficulty** – the difficulty for the block
- **expectedReward** – the block reward, the miner will make a coinbase transaction with this amount
- **rewardAddress** – the miner's address to receive the reward
- **blockDataHash** – the hash of the block without the nonce, the miner will get this hash and increment the nonce to find the correct hash

Submit Block Endpoint

With this endpoint you will submit a mined block.
URL: http://{host}:{port}/mining/submit-mined-block
Method : POST
Input:

```
{
  "blockDataHash" : "df8f114897188bcc68b97ebe2b673d3c92d...742b",
"dateCreated" : "2018-02-11T20:38:56.692Z", "nonce": 1177127,
  "blockHash" : "00000641e21ffceea0fce17c6b2f21668cc52886...745b"
}
```

- **blockDataHash –** The hash of the data in the block
- **blockHash –** the blockDataHash + correct nonce
- **dateCreated –** timestamp for the block

Output

```
{
  "message" : "Block accepted, reward paid: 5000350 microcoins"
}
```

Debug: Mine a Block Endpoint

With this endpoint you can mine with the difficulty that you want. Use it only for debugging purposes.

URL : http://{host}:{port}/debug/mine/{minerAddress}/{difficulty}
Method : GET
Input : minerAddress, difficulty
Output:

```
{
  "index" : 44,
  "transactions" : [{"from" : "0x1...", "to" : "0x2...","value" : 5}, {...}],
"difficulty"  : 3,"minedBy" : "0x3", "dateCreated" : "2018-05-10",... }
```

List All Peers Endpoint

This endpoint will return all the peers of the node.

URL : http://{host}:{port}/peers
Method : GET
Output:

```
{
  "162269f6993d2b5440dddcd6" : "http://localhost:5556",
  "162266dff5753a87a3e72403" : "http://af6c7a.ngrok.org:5555",
}
```

Connect a Peer Endpoint

With this endpoint, you can manually connect to other nodes.

URL : http://{host}:{port}/peers/connect

Method : POST
Input :

```
{
   "peerUrl" : "http://212.50.11.109:5556"
}
```

Output:

```
{
    "message" : "Connected to peer http://212.50.11.109:5556"
}
```

Notify Peers about New Block Endpoint

This endpoint will notify the peers about a new block.

URL: http://{host}:{port}/peers/notify-new-block
Method : POST
Input:

```
{
    "blocksCount" : 51,
    "cumulativeDifficulty" : 283,
    "nodeUrl" : "http://chain-node-03.herokuapp.com:5555"
}
```

Output:

```
{ "message": "Thank you for the notification." }
```