

QNS JSON-RPC API Документация

Версия: 2.0.0

Дата обновления: 2026-02-05

Всего методов: 252+

WebSocket подписок: 28

Обзор

QNS предоставляет полный JSON-RPC 2.0 API, совместимый с Ethereum, с расширениями для:

- Post-quantum криптографии (Dilithium3, Kyber1024)
 - Multi-VM архитектуры (EVM, TVM, QVM)
 - Neural DAG с GHOSTDAG упорядочиванием (10 кластеров)
 - Cross-shard операций
 - DeFi примитивов (AMM, Lending, Flash Loans с комиссией 0.09%)
 - Neural AI компонентов (BlockchainBrain)
 - NFT и токенов (ERC-20, ERC-721, ERC-1155)
 - Security мониторинга и MEV защиты
-

Эндпоинты

Pyuniq Testnet (Основная)

```
HTTP:      https://node.qsnchain.com/testnet/  
WebSocket: wss://node.qsnchain.com/testnet/ws  
Chain ID:  99991  
Symbol:    tQNS
```

Mainnet (Скоро)

```
HTTP:      https://node.qsnchain.com/qvm/mainnet/  
WebSocket: wss://node.qsnchain.com/qvm/mainnet/ws  
Chain ID:  1  
Symbol:    QNS
```

Devnet

```
HTTP:      https://node.qsnchain.com/devnet/  
WebSocket: wss://node.qsnchain.com/devnet/ws  
Chain ID:  99992  
Symbol:    dQNS
```

Ethereum-совместимые методы (eth_*)

Блокчейн информация

Метод	Описание	Параметры
<code>eth_blockNumber</code>	Текущий номер блока	-
<code>eth_chainId</code>	Chain ID сети	-
<code>eth_gasPrice</code>	Текущая цена газа в wei	-
<code>eth_syncing</code>	Статус синхронизации	-

Блоки

Метод	Описание	Параметры
<code>eth_getBlockByNumber</code>	Блок по номеру	<code>blockNumber</code> , <code>fullTx</code>
<code>eth_getBlockByHash</code>	Блок по хешу	<code>blockHash</code> , <code>fullTx</code>
<code>eth_getBlockTransactionCountByNumber</code>	Кол-во транзакций в блоке	<code>blockNumber</code>
<code>eth_getBlockTransactionCountByHash</code>	Кол-во транзакций по хешу блока	<code>blockHash</code>

Аккаунты

Метод	Описание	Параметры
<code>eth_getBalance</code>	Баланс адреса	<code>address</code> , <code>blockTag</code>
<code>eth_getTransactionCount</code>	Nonce адреса	<code>address</code> , <code>blockTag</code>
<code>eth_getCode</code>	Код контракта	<code>address</code> , <code>blockTag</code>
<code>eth_getStorageAt</code>	Значение storage слота	<code>address</code> , <code>slot</code> , <code>blockTag</code>
<code>eth_getProof</code>	Merkle proof (EIP-1186)	<code>address</code> , <code>storageKeys</code> , <code>blockTag</code>
<code>eth_accounts</code>	Список аккаунтов	-

Транзакции

Метод	Описание	Параметры
<code>eth_sendRawTransaction</code>	Отправить подписанную транзакцию	<code>signedTxData</code>
<code>eth_sendTransaction</code>	Отправить транзакцию	<code>txObject</code>

Метод	Описание	Параметры
<code>eth_getTransactionByHash</code>	Транзакция по хешу	<code>txHash</code>
<code>eth_getTransactionReceipt</code>	Квитанция транзакции	<code>txHash</code>
<code>eth_getTransactionByBlockHashAndIndex</code>	Транзакция по блоку и индексу	<code>blockHash</code> , <code>index</code>
<code>eth_getTransactionByBlockNumberAndIndex</code>	Транзакция по номеру блока и индексу	<code>blockNumber</code> , <code>index</code>

Вызовы и оценка газа

Метод	Описание	Параметры
<code>eth_call</code>	Read-only вызов контракта	<code>callObject</code> , <code>blockTag</code>
<code>eth_estimateGas</code>	Оценка газа для транзакции	<code>txObject</code>

Логи и фильтры

Метод	Описание	Параметры
<code>eth_getLogs</code>	Логи по фильтру	<code>filterObject</code>
<code>eth_newFilter</code>	Создать фильтр для логов	<code>filterObject</code>
<code>eth_newBlockFilter</code>	Фильтр для новых блоков	-
<code>eth_newPendingTransactionFilter</code>	Фильтр для pending транзакций	-
<code>eth_getFilterChanges</code>	Изменения по фильтру	<code>filterId</code>
<code>eth_getFilterLogs</code>	Все логи по фильтру	<code>filterId</code>
<code>eth_uninstallFilter</code>	Удалить фильтр	<code>filterId</code>

QNS-специфичные методы (qsn_*)

Информация о сети и блокчейне

Метод	Описание
<code>qsn_chainInfo</code>	Полная информация о сети
<code>qsn_getNetworkStats</code>	Статистика сети
<code>qsn_getNetworkStatsExtended</code>	Расширенная статистика
<code>qsn_getTpsMetrics</code>	Метрики TPS (200,000+ target)
<code>qsn_getGasStats</code>	Статистика газа
<code>qsn_getSyncStatus</code>	Статус синхронизации

Метод	Описание
<code>qsn_health</code>	Проверка здоровья

Аккаунты и кошельки

Метод	Описание
<code>qsn_getBalance</code>	Баланс адреса
<code>qsn_getAccountInfo</code>	Полная информация об аккаунте
<code>qsn_createWallet</code>	Создать кошелек (возвращает 3 адреса: quant, evm, ton)
<code>qsn_getQuantAddress</code>	Конвертация в quant-адрес

qsn_createWallet

Создаёт новый кошелек с гибридной криптографией (Ed25519 + Dilithium3) и возвращает **все три формата адресов**:

Ответ:

```
{
  "success": true,
  "addresses": {
    "quant": "quant27N2UqA8inQRFGAD8yXXSKffh1vr2kx7ngYQoWNeYdyjvSB9xC",
    "evm": "0xd50c170a57621b05b6889ae39e824bd51eb88130",
    "ton": "EQBvW8Z5huBkJYdnfAEM5JqTNLuuP..."
  },
  "quantAddress": "quant...",
  "evmAddress": "0x...",
  "tonAddress": "EQ...",
  "privateKey": "0x...",
  "publicKey": "0x...",
  "quantumPublicKey": "0x...",
  "quantumSecretKey": "0x...",
  "algorithm": {
    "classical": "Ed25519",
    "quantum": "Dilithium3 (NIST Level 3)"
  }
}
```

Форматы адресов:

- `quant...` - QVM/QRС-20 (workchain 0)
- `0x...` - EVM/ERC-20 (workchain 1)
- `EQ.../UQ.../0:hex` - TVM/Jetton TEP-74 (workchain 2)

Токены (Multi-VM: QRC-20, ERC-20, Jetton)

Метод	Описание	VM/Workchain
<code>qsn_deployToken</code>	Деплой QRC-20 токена	QVM (workchain 0)

Метод	Описание	VM/Workchain
<code>qsn_deployErc20</code>	Деплой ERC-20 токена	EVM (workchain 1)
<code>qsn_deployJetton</code>	Деплой Jetton токена (TEP-74)	TVM (workchain 2)
<code>qsn_tokenTransfer</code>	Трансфер токенов	All
<code>qsn_tokenBalanceOf</code>	Баланс токенов	All
<code>qsn_getTokenInfo</code>	Информация о токене	All

qsn_deployErc20

Деплой ERC-20 токена на EVM workchain с адресом формата `0x...`

Параметры:

```
{
  "from": "0x... или quant...",
  "name": "My Token",
  "symbol": "MTK",
  "initialSupply": "1000000",
  "decimals": 18,
  "signature": "0x...",
  "publicKey": "0x...",
  "quantumSignature": "0x...",
  "quantumPublicKey": "0x..."
}
```

Ответ:

```
{
  "success": true,
  "txHash": "0x...",
  "contractAddress": "0x...",
  "vmType": "EVM",
  "workchainId": 1
}
```

qsn_deployJetton

Деплой Jetton токена (TEP-74) на TVM workchain с адресом формата `EQ.../UQ...`

Параметры:

```
{
  "from": "EQ... или quant...",
  "name": "My Jetton",
  "symbol": "MJT",
  "initialSupply": "1000000",
  "decimals": 9,
  "signature": "0x...",
  "publicKey": "0x..."
}
```

```
"quantumSignature": "0x...",  
"quantumPublicKey": "0x..."  
}
```

Ответ:

```
{  
  "success": true,  
  "txHash": "0x...",  
  "contractAddress": "EQ...",  
  "vmType": "TVM",  
  "workchainId": 2,  
  "standard": "TEP-74"  
}
```

Стейкинг и валидаторы

Метод	Описание
<code>qsn_stake</code>	Застейкать токены
<code>qsn_unstake</code>	Снять стейк
<code>qsn_delegate</code>	Делегировать токены
<code>qsn_claimRewards</code>	Забрать награды
<code>qsn_getStakingInfo</code>	Информация о стейкинге
<code>qsn_getValidators</code>	Список валидаторов (171 всего)
<code>qsn_getActiveValidators</code>	Активные валидаторы
<code>qsn_getValidatorInfo</code>	Информация о валидаторе
<code>qsn_getNeuralValidators</code>	Нейронные валидаторы (50)
<code>qsn_getValidatorsByTier</code>	Валидаторы по тиру

Трёхуровневая система валидаторов

Уровень	Минимум	Награда	Макс. валидаторов
Execution	10,000 QNS	1.0x	100
Neural	50,000 QNS	1.5x	50
Verification	100,000 QNS	2.0x	21

Всего: 171 валидатор

Neural/AI методы (BlockchainBrain)

Метод	Описание
<code>qsn_getNeuralState</code>	Состояние нейросети

Метод	Описание
<code>qsn_getBrainMetrics</code>	Метрики BlockchainBrain
<code>qsn_getPredictions</code>	Предсказания AI
<code>qsn_getReasoningTrace</code>	Трассировка reasoning
<code>qsn_analyzeTransaction</code>	AI-анализ транзакции
<code>qsn_predictGas</code>	AI-предсказание газа
<code>qsn_getAIRecommendations</code>	AI-рекомендации
<code>qsn_detectMEV</code>	Обнаружение MEV
<code>qsn_getAddressRisk</code>	Риск-скоринг адреса

DAG и кластеры (GHOSTDAG)

Метод	Описание
<code>qsn_getDAGState</code>	Состояние DAG
<code>qsn_getDAGBlock</code>	Блок DAG
<code>qsn_getDAGChildren</code>	Дети блока DAG
<code>qsn_getDAGParents</code>	Родители блока DAG
<code>qsn_getClusters</code>	Список 10 кластеров
<code>qsn_getClusterInfo</code>	Информация о кластере
<code>qsn_getLinearization</code>	Линеаризация DAG

Безопасность и MEV защита

Метод	Описание
<code>qsn_getMEVStats</code>	Статистика MEV
<code>qsn_getDetectedAttacks</code>	Обнаруженные атаки
<code>qsn_getSecurityScore</code>	Security score
<code>qsn_getActiveThreats</code>	Активные угрозы
<code>qsn_getDefenseActions</code>	Защитные действия
<code>qsn_getSecurityDashboard</code>	Dashboard безопасности

QVM методы (qvm_*)

Базовые операции

Метод	Описание
<code>qvm_call</code>	Вызов контракта
<code>qvm_estimateGas</code>	Оценка газа
<code>qvm_sendRawTransaction</code>	Отправить транзакцию
<code>qvm_getBalance</code>	Баланс адреса
<code>qvm_chainId</code>	Chain ID

DeFi - АММ пулы

Метод	Описание
<code>qvm_defi_createPool</code>	Создать пул
<code>qvm_defi_getPoolInfo</code>	Информация о пуле
<code>qvm_defi_swap</code>	Обмен токенов
<code>qvm_defi_addLiquidity</code>	Добавить ликвидность
<code>qvm_defi_removeLiquidity</code>	Убрать ликвидность

DeFi - Lending

Метод	Описание
<code>qvm_defi_createLendingMarket</code>	Создать рынок кредитования
<code>qvm_defi_supply</code>	Поставить ликвидность
<code>qvm_defi_borrow</code>	Взять кредит
<code>qvm_defi_repay</code>	Погасить кредит
<code>qvm_defi_flashLoanFee</code>	Комиссия flash loan (0.09%)

Post-Quantum криптография

Метод	Описание
<code>qvm_dilithiumSign</code>	Подпись Dilithium3
<code>qvm_dilithiumVerify</code>	Верификация Dilithium3
<code>qvm_kyberEncapsulate</code>	Kyber KEM encapsulation

Bridge методы (bridge_*)

Метод	Описание
<code>bridge_status</code>	Статус моста
<code>bridge_deposits</code>	Список депозитов
<code>bridge_withdrawals</code>	Список выводов
<code>bridge_initiateDeposit</code>	Инициировать депозит
<code>bridge_initiateWithdrawal</code>	Инициировать вывод
<code>bridge_getSupportedChains</code>	Поддерживаемые сети (10+)
<code>bridge_estimateFee</code>	Оценка комиссии

Поддерживаемые сети

Сеть	Chain ID	Статус
Ethereum	1	Active
Polygon	137	Active
BSC	56	Active
TON	-	Active
Bitcoin	-	Planned

WebSocket подписки (28)

Подключение

```
wss://node.qsnchain.com/testnet/ws # Testnet
wss://node.qsnchain.com/qvm/mainnet/ws # Mainnet
```

Типы подписок

Базовые события

- `newHeads` — новые блоки
- `newPendingTransactions` — pending транзакции
- `logs` — логи контрактов
- `syncing` — статус синхронизации

Neural/Brain мониторинг

- `brainState` — состояние мозга
- `brainLearning` — события обучения

- `brainPrediction` — предсказания
- `neuralActivity` — neural активации

Безопасность и MEV

- `mevEvents` — обнаружение MEV атак
- `threatEvents` — обнаружение угроз
- `defenseEvents` — защитные действия
- `securityScore` — изменения security score

Rate Limiting

Параметр	Значение
<code>max_requests_per_second</code>	100 req/sec per IP
<code>max_tx_per_minute_per_address</code>	20 tx/min
<code>max_pending_tx</code>	10,000
<code>max_faucet_per_day</code>	1 req/day per address

Коды ошибок

Стандартные JSON-RPC

Код	Сообщение	Описание
-32700	Parse error	Некорректный JSON
-32600	Invalid Request	Невалидный запрос
-32601	Method not found	Метод не найден
-32602	Invalid params	Невалидные параметры
-32603	Internal error	Внутренняя ошибка

QNS-специфичные

Код	Сообщение	Описание
-32010	Rate limit exceeded	Превышен rate limit
-32011	Invalid signature	Невалидная подпись
-32012	Insufficient balance	Недостаточный баланс
-32016	Cross-shard error	Ошибка кросс-шард операции
-32017	Quantum signature required	Требуется PQ подпись

Примеры использования

cURL

```
# Получить номер блока
curl -X POST https://node.qsnchain.com/testnet/ \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","id":1,"method":"eth_blockNumber","params":[]}'

# QNS chain info
curl -X POST https://node.qsnchain.com/testnet/ \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","id":1,"method":"qsn_chainInfo","params":[]}'

# Brain metrics
curl -X POST https://node.qsnchain.com/testnet/ \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","id":1,"method":"qsn_getBrainMetrics","params":[]}'
```

JavaScript (ethers.js v6)

```
import { ethers } from 'ethers';

const provider = new ethers.JsonRpcProvider('https://node.qsnchain.com/testnet/');

// Получить номер блока
const blockNumber = await provider.getBlockNumber();

// QNS-специфичные методы
const chainInfo = await provider.send('qsn_chainInfo', []);
const brainMetrics = await provider.send('qsn_getBrainMetrics', []);
```

Python (web3.py)

```
from web3 import Web3

w3 = Web3(Web3.HTTPProvider('https://node.qsnchain.com/testnet/'))

# Получить номер блока
block_number = w3.eth.block_number

# QNS-специфичные методы
chain_info = w3.provider.make_request('qsn_chainInfo', [])
brain_metrics = w3.provider.make_request('qsn_getBrainMetrics', [])
```

Ключевые метрики

Параметр	Значение
TPS Target	200,000+
Block Time	1-2 сек
Finality	~3 сек
Max Validators	513 (300+150+63)
DAG Clusters	10
RPC Methods	252+
WebSocket Subscriptions	28
Flash Loan Fee	0.09% (9 BPS)
QVM Opcodes	252+
Post-Quantum	Dilithium3, Kyber1024

Документация обновлена: 2026-02-05