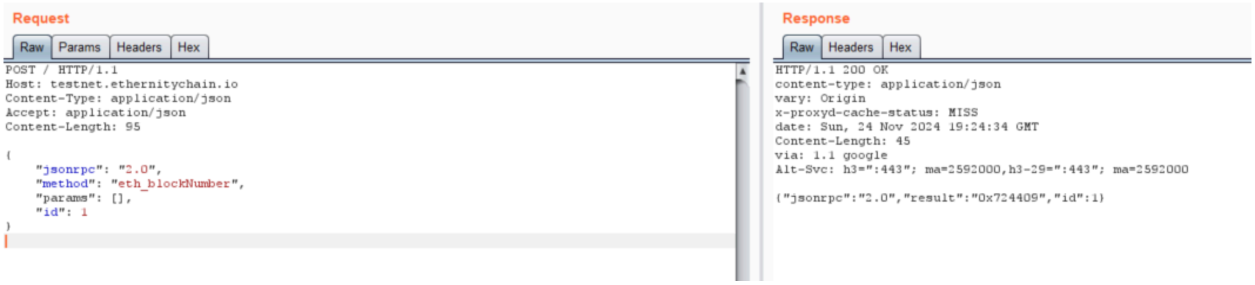
Functional Test Case Report for RPC Endpoints

Endpoint: https://testnet.ethernitychain.io

1. **API Method Validation: PASSED**
    - **Objective**: Ensure only allowed HTTP methods are functional.
    - **Steps**:
        1. Send requests with methods: GET, POST, PUT, DELETE, OPTIONS.
        2. Observe the response for each.
    - **Expected Result**: Only permitted methods return `200 OK`; others return `405 Method Not Allowed`.
2. **Input Validation: PASSED**
    - **Objective**: Verify if the endpoint handles malformed input.
    - **Steps**:
        1. Send a GET request with invalid query parameters (e.g., `?invalidParam=123`).
        2. Send a POST request with invalid payloads.
    - **Expected Result**: The endpoint responds with a `400 Bad Request` or appropriate error.
3. **Valid Payload Test: PASSED**
    - **Objective**: Ensure the endpoint processes a valid payload correctly.
    - **Steps**: Submit a properly formatted JSON payload (if applicable).

- **Expected Result**: The response contains the expected output without errors.

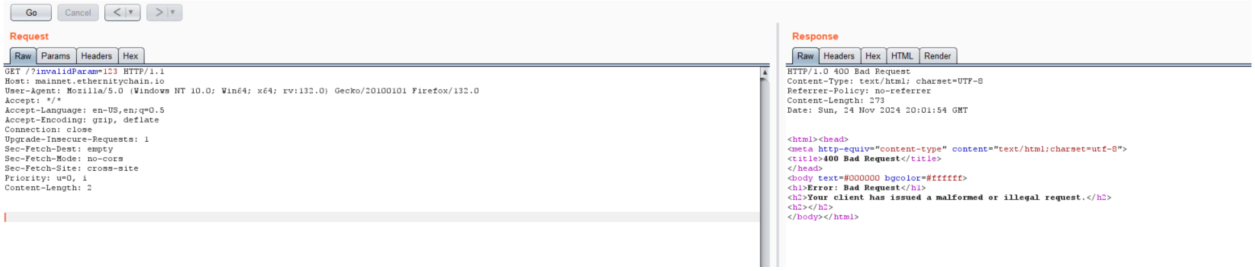


```
POST / HTTP/1.1
Host: testnet.ethernitychain.io
Content-Type: application/json
Accept: application/json
Content-Length: 95

{
    "jsonrpc": "2.0",
    "method": "eth_blockNumber",
    "params": [],
    "id": 1
}
```

```
HTTP/1.1 200 OK
content-type: application/json
vary: Origin
x-proxyd-cache-status: MISS
date: Sun, 24 Nov 2024 19:24:34 GMT
Content-Length: 45
via: 1.1 google
Alt-Svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000

{"jsonrpc":"2.0","result":"0x724409","id":1}
```

Endpoint: https://mainnet.ethernitychain.io

4. **API Method Validation: PASSED**
    - **Objective**: Ensure only allowed HTTP methods are functional.
    - **Steps**:
        1. Send requests with methods: GET, POST, PUT, DELETE, OPTIONS.
        2. Observe the response for each.
    - **Expected Result**: Only permitted methods return `200 OK`; others return `405 Method Not Allowed`.
5. **Input Validation: PASSED**
    - **Objective**: Verify if the endpoint handles malformed input.
    - **Steps**:
        1. Send a GET request with invalid query parameters (e.g., `?invalidParam=123`).
        2. Send a POST request with invalid payloads.
    - **Expected Result**: The endpoint responds with a `400 Bad Request` or appropriate error.

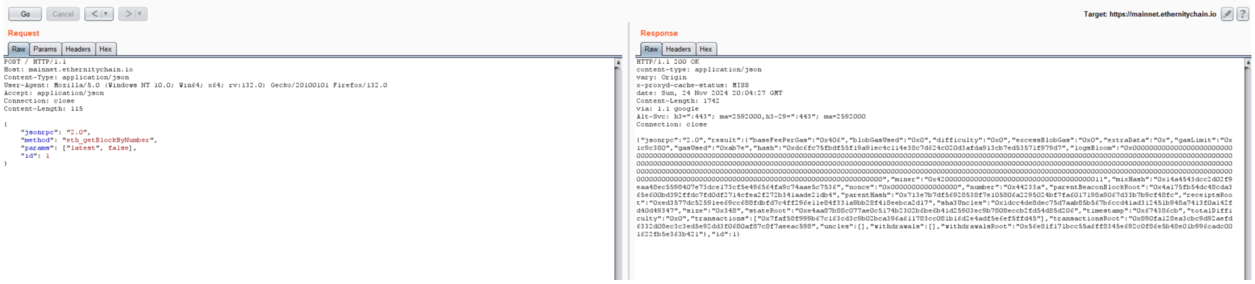


```
GET /?invalidParam=123 HTTP/1.1
Host: mainnet.ethernitychain.io
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:132.0) Gecko/20100101 Firefox/132.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: cross-site
Priority: u=0, i
Content-Length: 2
```

```
HTTP/1.0 400 Bad Request
Content-Type: text/html; charset=UTF-8
Referrer-Policy: no-referrer
Content-Length: 273
Date: Sun, 24 Nov 2024 20:01:54 GMT


<html><head>
<meta http-equiv="content-type" content="text/html;charset=utf-8">
<title>400 Bad Request</title>
</head>
<body text=#000000 bgcolor=#ffffff>
<h1>Error: Bad Request</h1>
<h2>Your client has issued a malformed or illegal request.</h2>
<h2></h2>
</body></html>
```

6. **Valid Payload Test: PASSED**
    - **Objective**: Ensure the endpoint processes a valid payload correctly.
    - **Steps**: Submit a properly formatted JSON payload (if applicable).

- **Expected Result**: The response contains the expected output without errors.



```
POST / HTTP/1.1
Host: mainnet.ethernitychain.io
Content-Type: application/json
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:132.0) Gecko/20100101 Firefox/132.0
Accept: application/json
Connection: close
Content-Length: 115

{
    "jsonrpc": "2.0",
    "method": "eth_getBlockByNumber",
    "params": ["latest", false],
    "id": 1
}
```

```
HTTP/1.1 200 OK
content-type: application/json
vary: Origin
x-proxyd-cache-status: MISS
date: Sun, 24 Nov 2024 20:04:27 GMT
Content-Length: 1742
via: 1.1 google
Alt-Svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
Connection: close

{"jsonrpc":"2.0","result":{"baseFeePerGas":"0x406","blobGasUsed":"0x0","difficulty":"0x0","excessBlobGas":"0x0","extraData":"0x","gasLimit":"0x
1c9c380","gasUsed":"0xab7e","hash":"0xdc6fc75fbdf55f19a91ec4c114e38c7d624c020d3afda913cb7ed53571f979d7","logsBloom":"0x000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000","miner":"0x4200000000000000000000000000000000000011","mixHash":"0x14a4543dcc2d02f9
eaa48ec5598407e73dce173cf5e496564fa9c74aae5c7536","nonce":"0x0000000000000000","number":"0x44233a","parentBeaconBlockRoot":"0x4a175fb54dc48cda3
65e600bd392ffdc7fd0df2714cfea2f272b341aade21db4","parentHash":"0x713e7b7df56928538f7e105806a2295024bf7fa6017198a9067d33b7b9cf48fc","receiptsRoo
t":"0xed3577dc52591ee69cc688fdbfd7c4ff296e11e84f331a9bb28f418eebca2d17","sha3Uncles":"0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142f
d40d49347","size":"0x348","stateRoot":"0xe4aa87b88c077ae0c5174b2302b6be6b41d25903ec9b7808eccb2fd54d85d206","timestamp":"0x674386cb","totalDiffi
culty":"0x0","transactions":["0x7faf58f999b67c163cd3c9b02bca396a611783cc081b16d2e4adf5e6ef5ffd45"],"transactionsRoot":"0x890fa128ea3cbc9d92aefd
6332d08ec3c3ed5e92dd3f0680af87c8f7aeeac598","uncles":[],"withdrawals":[],"withdrawalsRoot":"0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc00
1622fb5e363b421"},"id":1}
```