

POSTMAN REQUESTS

Click Here ➤ [PostMan Collections](#)

The screenshot shows the Postman application interface. On the left, there's a sidebar with various sections like Collections, Environments, History, APIs, Specs, Mock servers, Monitors, Flows, Insights, and Files. The main area displays a collection named 'Infollion'. Under this collection, there are several requests: 'GET Get data', 'POST Post data', 'GET New Request', and another 'GET New Request'. The 'GET Get data' request is currently selected. The request details show a GET method for 'http://localhost:3000/route'. The 'Headers' tab is active, displaying 14 headers. One header, 'X-Forwarded-For', is checked and has a value of '10.0.0.6'. Other headers listed include '19.45.72.1', '192.168.1.10', '25.14.26.23', '10.0.0.1', and '10.0.0.6'. Below the headers, the 'Body' tab is visible, showing a JSON response with the key 'ip' and value '10.0.0.6', and 'routedTo' with the value 'Node-B'. The status bar at the bottom right indicates a '200 OK' response with a size of 74.

The screenshot shows a terminal window with the number '163' in the top-left corner. The terminal tabs are PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS. The terminal content shows the following output:

```
PS D:\Load-Balancer> node server.js
Load Balancer API running on port 3000
Incoming IP: 10.0.0.6 → Routed to: Node-B
```

2K22 EP 40 DINESH BISHT's Works... New Import GET Get data • | GET New Request • | GET New Request • | GET http://localhost:3000/route | GET New Request + ↻

Collections + Q Search collections Infollion GET Get data POST Post data GET New Request GET New Request

environments History APIs Specs lock servers Monitors Flows Insights Files BETA

HTTP Infollion / Get data

GET http://localhost:3000/route

Docs Params Authorization Headers (14) Body Scripts Settings

Headers 6 hidden

Key	Value	Description	...
<input type="checkbox"/> X-Forwarded-For	10.0.0.6		
<input checked="" type="checkbox"/> X-Forwarded-For	19.45.72.1		
<input type="checkbox"/> X-Forwarded-For	192.168.1.10		
<input type="checkbox"/> X-Forwarded-For	25.14.26.23		
<input type="checkbox"/> X-Forwarded-For	10.0.0.1		
<input type="checkbox"/> X-Forwarded-For	10.0.0.6		

Body Cookies Headers (7) Test Results (1/1) 200 OK 74 ms 272 B

{ } JSON Preview Visualize

```

1  {
2   "ip": "10.0.0.6",
3   "routedTo": "Node-B"
4 }
```

LOAD-BALANCER

- > node_modules
- {} package-lock.json JA, U
- {} package.json JA, U
- ① README.md IM
- JS server.js IA, U

JS server.js > ...

```

138 app.get("/route", rateLimiter , (req, res) => {
145   routedto: node
146 });
147 );
148
149
150 // Metrics endpoint
151 app.get("/metrics", (req, res) => {
152   res.json(metrics);
153 });
154
155
156 // Simulate traffic via API
157 app.get("/simulate/:count", (req, res) => {
158   const count = parseInt(req.params.count) || 5;
159   simulateTraffic(count);
160   res.json({ message: `${count} requests simulated` });
161 });
162
163
164 app.listen(PORT, () => {
165   console.log(`Load Balancer API running on port ${PORT}`);
166 });


```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Load-Balancer> node server.js
Load Balancer API running on port 3000
Incoming IP: 10.0.0.6 → Routed to: Node-B
Incoming IP: 19.45.72.1 → Routed to: Node-C

	Key	Value
<input type="checkbox"/>	X-Forwarded-For	10.0.0.6
<input checked="" type="checkbox"/>	X-Forwarded-For	19.45.72.1
<input type="checkbox"/>	X-Forwarded-For	192.168.1.10
<input type="checkbox"/>	X-Forwarded-For	25.14.26.23
<input type="checkbox"/>	X-Forwarded-For	10.0.0.1
<input type="checkbox"/>	X-Forwarded-For	10.0.0.6

Body Cookies Headers (7) Test Results (1/1) ⏱

{ } JSON ▾ ▷ Preview ⚡ Visualize ▾

```
1  {
2    "ip": "19.45.72.1",
3    "routedTo": "Node-C"
4 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\Load-Balancer> node server.js
Load Balancer API running on port 3000
Incoming IP: 10.0.0.6 → Routed to: Node-B
Incoming IP: 19.45.72.1 → Routed to: Node-C
Incoming IP: 19.45.72.1 → Routed to: Node-C
[]
```

GET

▼ http://localhost:3000/route

Docs Params Authorization Headers (14) Body Scripts • Settings

Headers (6 hidden)

	Key	Value	Description
<input type="checkbox"/>	X-Forwarded-For	10.0.0.6	
<input type="checkbox"/>	X-Forwarded-For	19.45.72.1	
<input type="checkbox"/>	X-Forwarded-For	192.168.1.10	
<input checked="" type="checkbox"/>	X-Forwarded-For	25.14.26.23	
<input type="checkbox"/>	X-Forwarded-For	10.0.0.1	
<input type="checkbox"/>	X-Forwarded-For	10.0.0.6	

Body Cookies Headers (7) Test Results (1/1) ↻

200 OK

{ } JSON ▾ Preview Visualize ▾

```

1  {
2    "ip": "25.14.26.23",
3    "routedTo": "Node-B"
4  }

```

```

5 157  app.get("/simulate/:count", (req, res) => {
5 158  const count = parseInt(req.params.count) || 5;
5 159  simulateTraffic(count);
5 160  res.json({ message: `${count} requests simulated` });
5 161  });
5 162
5 163
5 164  app.listen(PORT, () => {
5 165  console.log(`Load Balancer API running on port ${PORT}`);
5 166  });

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Load-Balancer> node server.js
 Load Balancer API running on port 3000
 Incoming IP: 10.0.0.6 → Routed to: Node-B
 Incoming IP: 19.45.72.1 → Routed to: Node-C
 Incoming IP: 19.45.72.1 → Routed to: Node-C
 Incoming IP: 25.14.26.23 → Routed to: Node-B

SIMULATION OF 15 IP ADDRESSES

The screenshot shows a REST API testing interface. At the top, there's a header bar with 'GET' selected, a URL input field containing 'http://localhost:3000/simulate/15', and a 'Send' button. Below the header are tabs for 'Docs', 'Params' (which is active), 'Authorization', 'Headers (6)', 'Body', 'Scripts', and 'Settings'. A 'Cookies' tab is also present. Under 'Params', there's a table titled 'Query Params' with one row: 'Key' and 'Value'. The main body of the interface shows a table with columns 'Body', 'Cookies', 'Headers (7)', 'Test Results', and a status indicator. The 'Test Results' section shows a green '200 OK' status with a response time of '22 ms' and a size of '270 B'. The response body is displayed as JSON: { "message": "15 requests simulated" }. There are also icons for preview, visualize, and save response.

The screenshot shows a terminal window with the following content:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Load-Balancer> node server.js
Load Balancer API running on port 3000
Incoming IP: 5.238.200.54 → Routed to: Node-C
Incoming IP: 213.56.62.54 → Routed to: Node-B
Incoming IP: 79.132.168.153 → Routed to: Node-B
Incoming IP: 224.208.139.91 → Routed to: Node-C
Incoming IP: 127.157.186.40 → Routed to: Node-A
Incoming IP: 40.192.81.20 → Routed to: Node-A
Incoming IP: 118.194.154.191 → Routed to: Node-A
Incoming IP: 140.85.181.114 → Routed to: Node-B
Incoming IP: 140.119.16.177 → Routed to: Node-C
Incoming IP: 153.27.254.221 → Routed to: Node-B
Incoming IP: 9.0.250.85 → Routed to: Node-C
Incoming IP: 65.211.85.230 → Routed to: Node-A
Incoming IP: 139.18.77.82 → Routed to: Node-B
Incoming IP: 78.228.246.137 → Routed to: Node-C
Incoming IP: 47.56.166.128 → Routed to: Node-B
```

15 IP Addresses Simulated

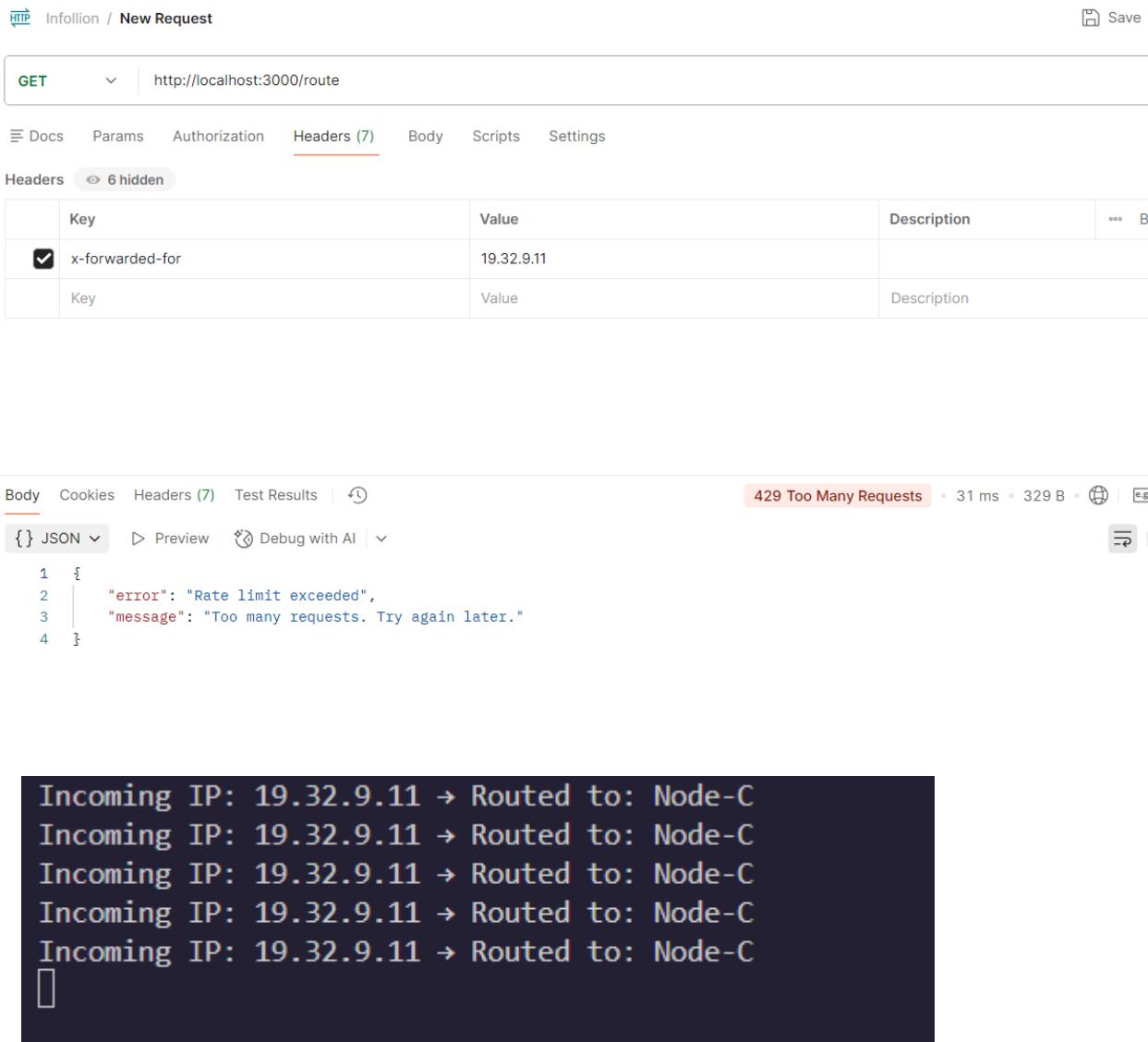
METRICS DASHBOARD

The screenshot shows the Infolion interface for managing APIs. On the left, there's a sidebar with collections like 'Infolion' containing 'Get data', 'Post data', and 'New Request'. The main area shows a request configuration for 'GET /metrics' to 'http://localhost:3000/metrics'. The 'Params' tab is selected, showing a table with one row: 'Key' (empty) and 'Value' (empty). Below this, the 'Body' tab is selected, showing a JSON response:

```
1 {  
2     "totalRequests": 15,  
3     "perNode": {  
4         "Node-C": 5,  
5         "Node-B": 6,  
6         "Node-A": 4  
7     }  
8 }
```

The status bar at the bottom indicates a 200 OK response with 150 ms latency and 300 B size.

Limit Reach if more than 5 requests are sent with same IP ADDRESSES WITHIN ONE MINUTE



The screenshot shows the Infollion API testing interface. At the top, it displays a 'New Request' configuration for a 'GET' method to 'http://localhost:3000/route'. The 'Headers' tab is selected, showing a table with one row: 'x-forwarded-for' set to '19.32.9.11'. Below this, the 'Body' tab is selected, showing a JSON response body:

```
1 {  
2   "error": "Rate limit exceeded",  
3   "message": "Too many requests. Try again later."  
4 }
```

In the bottom right corner of the interface, there is a log window displaying the following text:

```
Incoming IP: 19.32.9.11 → Routed to: Node-C  
Incoming IP: 19.32.9.11 → Routed to: Node-C
```

Only 5 request have been received