Instrumenting MCP

Four Lessons Learned

MCP is JSON-RPC 2.0 With Extra* Steps

* extra good steps tbh

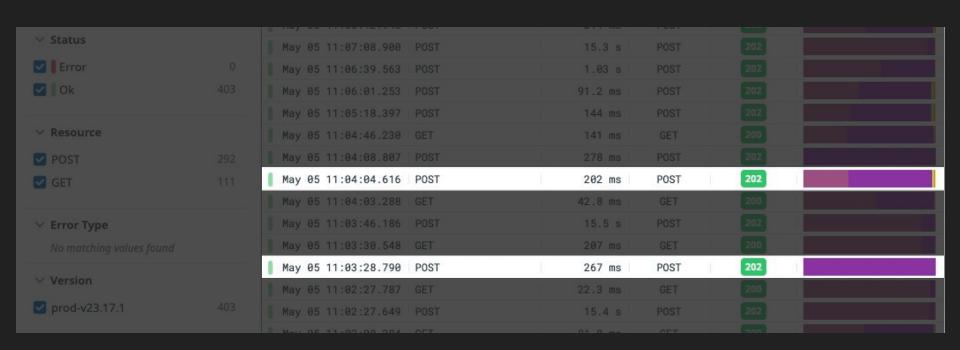
Errors Are In The Payload

Status: 200 but...

```
1 {
2  "jsonrpc": "2.0",
3  "error": {
4     "code": -32601,
5     "message": "Method not found"
6     },
7     "id": "1"
8 }
```

The Errors Never Make It To Your Observability Tools

enterior — Protections		11a, 00 11.00.12.710	1001	011 #	1 001		
∨ Status	0 403	May 05 11:07:08.900	POST	15.3	s POST	202	
✓ Error		May 05 11:06:39.563	POST	1.03	s POST	202	
Ok		May 05 11:06:01.253	POST	91.2 m	ns POST	202	
		May 05 11:05:18.397	POST	144 m	ns POST	202	(3)
∨ Resource		May 05 11:04:46.230	GET	141 m	ns GET	200	
✓ POST	292	May 05 11:04:08.807	POST	278 п	ns POST	202	
☑ GET	111	May 05 11:04:04.616	POST	202 m	ns POST	202	
		May 05 11:04:03.288	GET	42.8 m	is GET	200	
∨ Error Type		May 05 11:03:46.186	POST	15.5	s POST	202	
No matching values found		May 05 11:03:30.548	GET	207 m	is GET	200	
✓ Version ✓ prod-v23.17.1		May 05 11:03:28.790	POST	267 m	ns POST	202	
		May 05 11:02:27.787	GET	22.3 m	is GET	200	
	403	May 05 11:02:27.649	POST	15.4	s POST	202	
		May 05 11,02,00 204	CET	01.0	CET	200	



Streamed Responses are "202 Accepted"

It Is Hard to "Tap" Into a Library

- Provide access to the raw request
- Provide a dependency injection option

Rely on exceptions / runtime errors



Building It Live on Twitch



Docs, Cloud Options, & More