



Instrumenting Node.js Apps

Thomas Watson
Principal Software Engineer
Elastic APM Node.js Agent
@wa7son



Who am I?

- Thomas Watson
- Open Source developer at github.com/watson
- Node.js Engineer at Elastic
- Elastic APM Node.js agent
- Member of the Diagnostics Working Group under the Node.js Foundation
- Tweets as @wa7son



Introduction to Elastic APM

elastic.co/webinars/elasticsearch-apm-overview

Setup & Configuration

“

The Elastic APM agent philosophy is simple: Convention over configuration and don't be in the way.

```
npm install elastic-apm-node --save
```

Configuration

Inline

Config file

Environment variables

```
require('elastic-apm-node').start({  
  serviceName: 'my-node-app'  
})
```

```
require('elastic-apm-node').start({  
  serviceName: 'my-node-app',  
  secretToken: 'secret!'  
})
```

```
require('elastic-apm-node').start({  
  serviceName: 'my-node-app',  
  secretToken: 'secret!',  
  serverUrl: 'https://example.com:8200'  
})
```

```
require('elastic-apm-node').start({  
  serviceName: 'my-node-app',  
  secretToken: 'secret!',  
  serverUrl: 'https://example.com:8200',  
  active: process.env.NODE_ENV === 'production'  
})
```

```
NODE_ENV=production  
ELASTIC_APM_SERVICE_NAME=my-node-app  
ELASTIC_APM_SECRET_TOKEN=secret!  
ELASTIC_APM_SERVER_URL=https://example.com:8200
```

```
require('elastic-apm-node').start({  
  active: process.env.NODE_ENV === 'production'  
})
```

```
{  
  "serviceName": "my-node-app",  
  "secretToken": "secret!",  
  "serverUrl": "https://example.com:8200"  
}
```

```
module.exports = {  
  serviceName: 'my-node-app',  
  secretToken: 'secret!',  
  serverUrl: 'https://example.com:8200',  
  active: process.env.NODE_ENV === 'production'  
}  
  
-----
```

```
require('elastic-apm-node').start()
```

```
ELASTIC_APM_CONFIG_FILE=/path/to/elastic-apm-node.js
```

```
const agent = require('elastic-apm-node').start()
```

```
const agent = require('elastic-apm-node')
```

“

*Put the call to the “start” function at the
very top of your main app file - before
requiring any other modules*

node server.js

```
const path = require('path')
const express = require('express')
const afterAll = require('after-all-results')
const db = require('./lib/db')
const redis = require('./lib/redis')
const accounting = require('./lib/accounting')

const app = express()
// etc...
```

```
const agent = require('elastic-apm-node').start({...})
const path = require('path')
const express = require('express')
const afterAll = require('after-all-results')
const db = require('./lib/db')
const redis = require('./lib/redis')
const accounting = require('./lib/accounting')

const app = express()
// etc...
```

```
const conf = require('./lib/config')
const agent = require('elastic-apm-node').start(conf.apm)
const path = require('path')
const express = require('express')
const afterAll = require('after-all-results')
const db = require('./lib/db')
const redis = require('./lib/redis')
const accounting = require('./lib/accounting')

const app = express()
// etc...
```

```
import conf from './server/config'  
import agent from 'elastic-apm-node'  
import path from 'path'  
import express from 'express'  
import afterAll from 'after-all-results'  
import db from './lib/db'  
import redis from './lib/redis'  
import accounting from './lib/accounting'
```

agent.start(conf.apm)

var app = express()
// etc...

```
import agent from 'elastic-apm-node/start'
```

```
node -r elastic-apm-node/start server.js
```



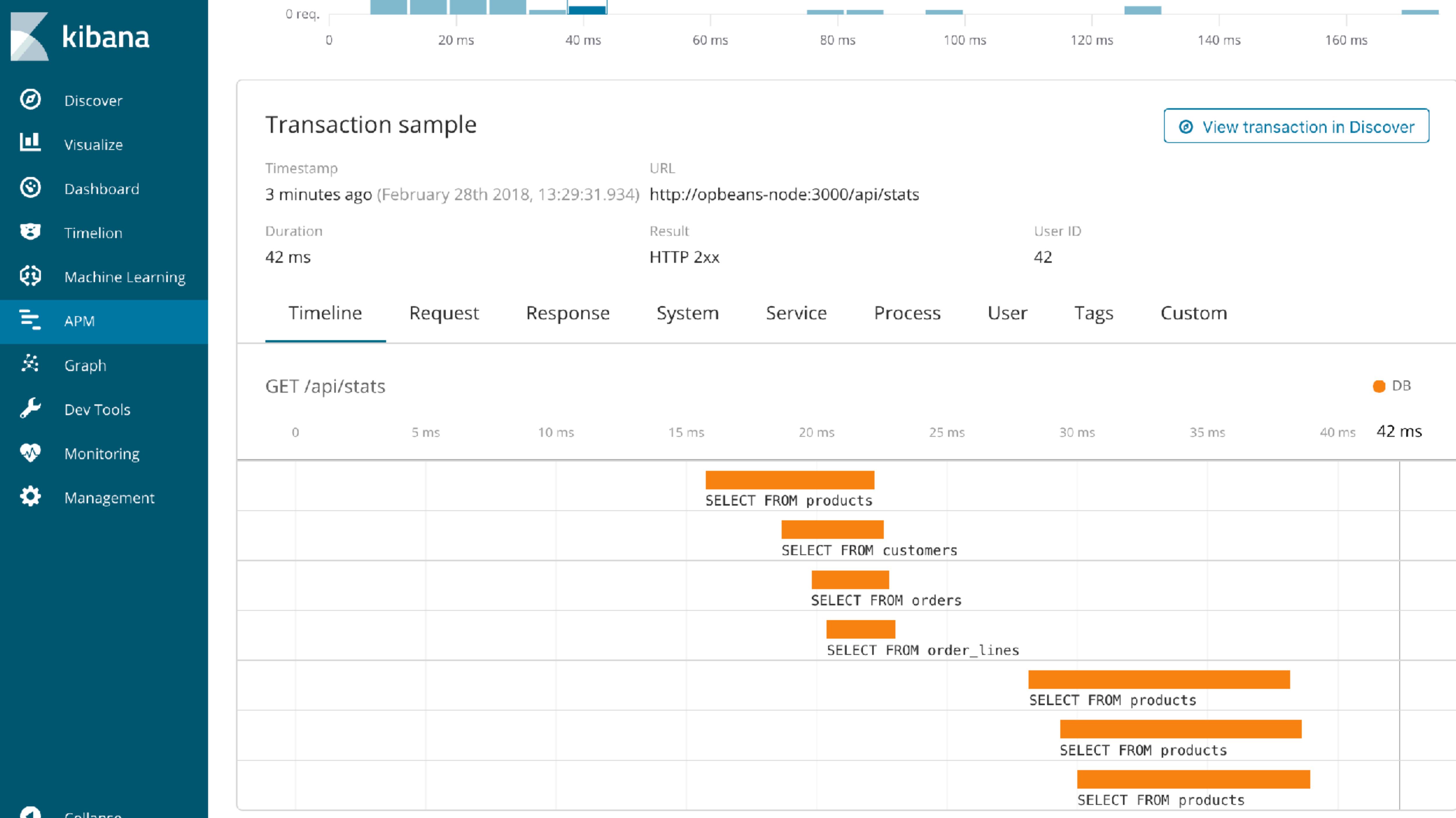
Data Scrubbing

filterHTTPHeaders: true

Custom Filters

```
agent.addFilter((payload) => {
  if (payload.context.request &&
    payload.context.request.headers)
  {
    // redact sensitive data
    payload.context.request.headers['x-secret'] = '[REDACTED]'
  }

  // remember to return the modified payload
  return payload
})
```



Timeline

Request

Response

System

Service

Process

GET /api/stats

0

5 ms

10 ms

15 ms

20 ms

25 ms



SELECT FROM products



SELECT FROM customers



SELECT FROM orders



SELECT FROM order_l:





A pixelated, black and white image of a cat's face, looking slightly to the right. The cat has a white patch on its forehead and a pink nose. It is set against a dark blue background with some white highlights.

Custom Spans

```
const data = JSON.parse(req.body)
```

```
const span = agent.startSpan('Parsing JSON')
const data = JSON.parse(req.body)
```

```
const span = agent.startSpan('Parsing JSON')
const data = JSON.parse(req.body)
if (span) span.end()
```

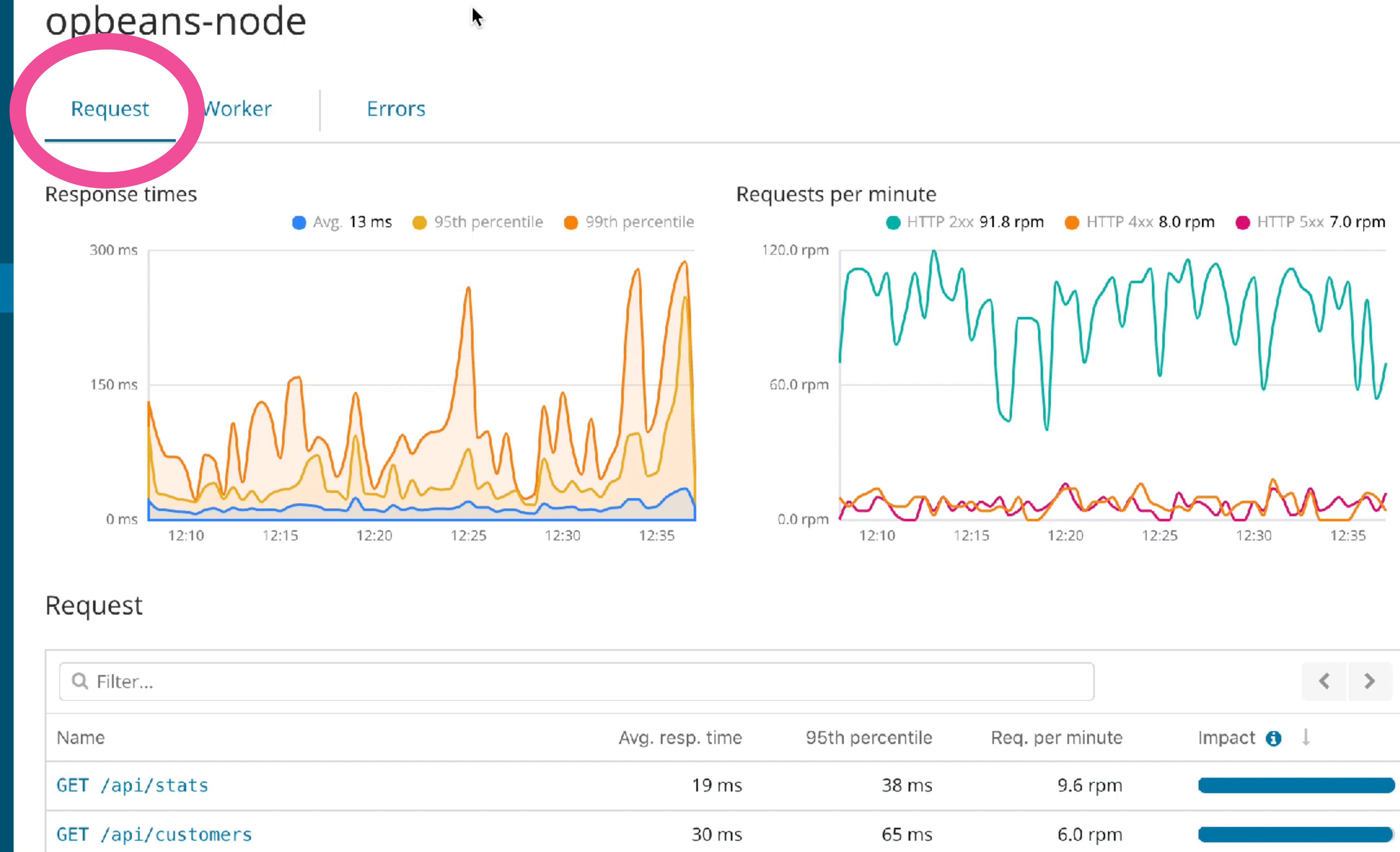
How to instrument non HTTP server apps



kibana

- Discover
- Visualize
- Dashboard
- Timeline
- Machine Learning
- APM
- Graph
- Dev Tools
- Monitoring
- Management

Collapse





kibana

Discover

Visualize

Dashboard

Timeline

Machine Learning

APM

Graph

Dev Tools

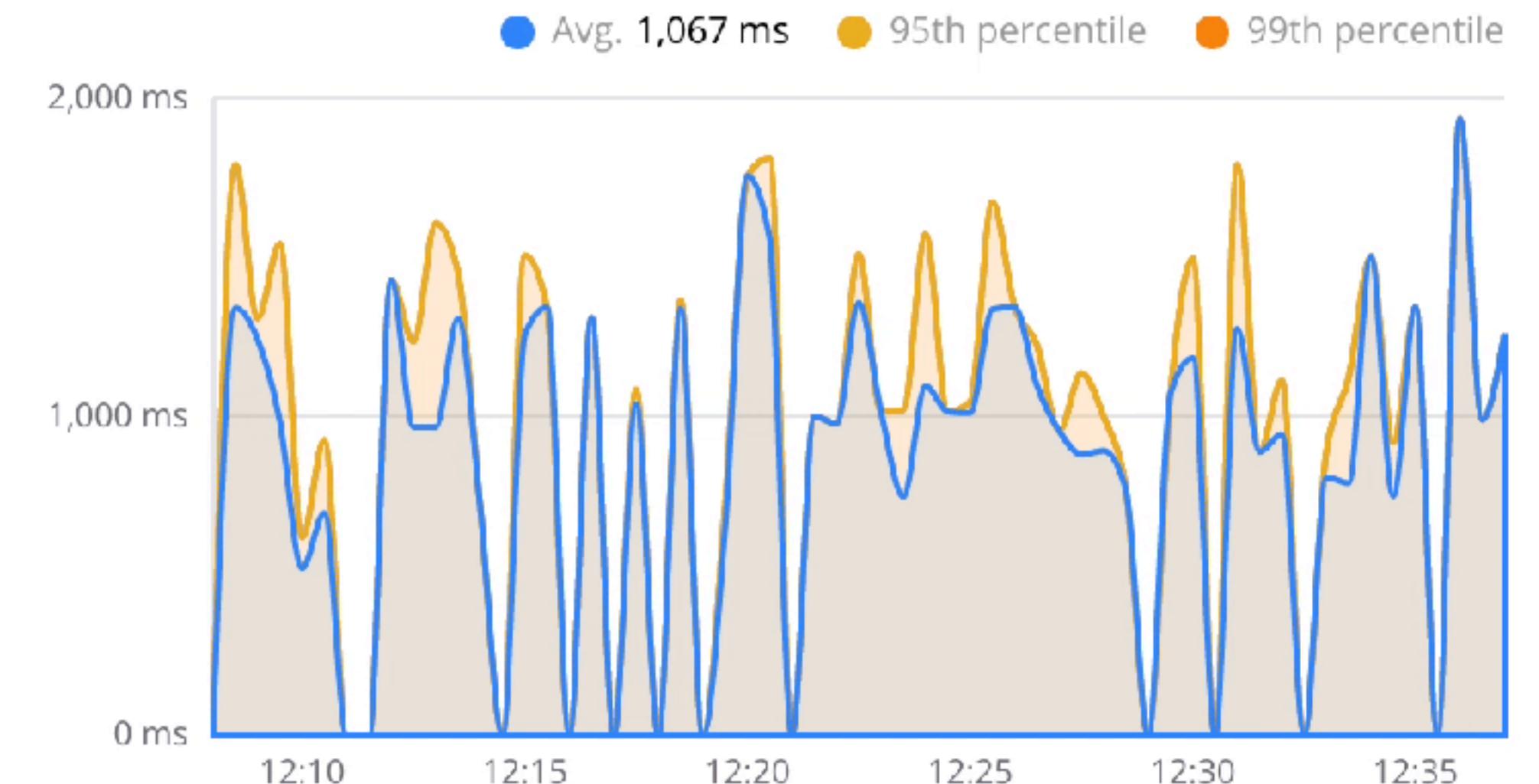
Monitoring

Management

opbeans-node

[Request](#)[Worker](#)[Errors](#)

Response times



Transactions per minute



Worker

Worker				
Name	Avg. resp. time	95th percentile	Trans. per minute	Impact
Process completed order	1,124 ms	1,774 ms	1.1 tpm	<div style="width: 100%; background-color: #007bff;"></div>
Update shipping status	1,051 ms	1,756 ms	1.0 tpm	<div style="width: 100%; background-color: #007bff;"></div>



Collapse

```
const queue = require('./queue')

queue.on('job', (job, cb) => {
  processJob(job, cb)
})
```

```
const agent = require('elastic-apm-node').start()  
const queue = require('./queue')  
  
queue.on('job', (job, cb) => {  
  processJob(job, cb)  
})
```

```
const agent = require('elastic-apm-node').start()
const queue = require('./queue')

queue.on('job', (job, cb) => {
  const name = 'Job ' + job.type
  const type = 'job'
  const trans = agent.startTransaction(name, type)

  processJob(job, cb)
})
```

```
const agent = require('elastic-apm-node').start()
const queue = require('./queue')

queue.on('job', (job, cb) => {
  const name = 'Job ' + job.type
  const type = 'job'
  const trans = agent.startTransaction(name, type)

  processJob(job, (err) => {
    if (err) trans.result = 'error'
    trans.end()
    cb(err)
  })
})
})
```

```
const queue = require('./queue')

queue.on('job', (job, cb) => {
  const name = 'Job ' + job.type
  const type = 'job'
  const trans = agent.startTransaction(name, type)

  processJob(job, (err) => {
    if (err) trans.result = 'error'
    trans.end()
    cb(err)
  })
})
```

```
const queue = require('./queue')

queue.on('job', (job, cb) => {
  const name = 'Job ' + job.type
  const type = 'job'
  const trans = agent.startTransaction(name, type)

  processJob(job, (err) => {
    if (err) {
      trans.result = 'error'
      agent.captureError(err)
    }
    trans.end()
    cb(err)
  })
})
})
```

Tuning Elastic APM

In-memory Queue

Config option	Default	Description
flushInterval	10	How often the queue is flushed (in seconds)
maxQueueSize	-1	Max number of items in the queue before its flushed (unlimited: -1)



kibana

Discover

Visualize

Dashboard

Timelion

Machine Learning

APM

Graph

Dev Tools

Monitoring

Management

Error group 7446e

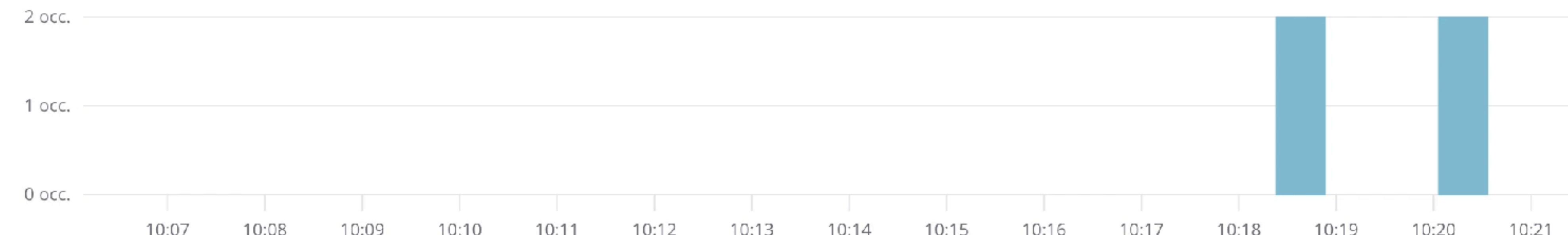
Exception message

passwordBuf must be Buffer

Culprit

<anonymous> (server/routes.js)

Occurrences



Error occurrence

[View 4 occurrences in Discover](#)

Timestamp

a minute ago (February 28th 2018, 10:20:31.617)

URL

<http://localhost:3001/api/login>

Request method

POST

Handled

N/A

User ID

N/A

[Exception stacktrace](#)[Request](#)[Response](#)[System](#)[Service](#)[Process](#)[User](#)[Tags](#)[Custom](#)

Collapse



kibana

- Discover
 - Visualize
 - Dashboard
 - Timelion
 - Machine Learning
 - APM
 - Graph
 - Dev Tools
 - Monitoring
 - Management
- Collapse

Transaction sample

[View transaction in Discover](#)

Timestamp

16 minutes ago (February 28th 2018, 10:10:08.348) http://localhost:3001/api/stats

URL

Duration

154 ms

Result

HTTP 2xx

User ID

42

Timeline Request Response System Service Process User Tags Custom

GET /api/stats

DB

0 20 ms 40 ms 60 ms 80 ms 100 ms 120 ms 140 ms 154 ms

SELECT FROM products

SELECT FROM customers

SELECT FROM orders

SELECT FROM order_lines

SELECT FROM products

SELECT FROM products

SELECT FROM products

Source Code Context

Config option	Default	Description
sourceLinesErrorAppFrames	5	Lines of source code to collect around in-app frames on error stack traces
sourceLinesErrorLibraryFrames	5	Lines of source code to collect around library frames on error stack traces
sourceLinesSpanAppFrames	0	Lines of source code to collect around in-app frames for spans
sourceLinesSpanLibraryFrames	0	Lines of source code to collect around library frames for spans

`captureSpanStackTraces: true`

stackTraceLimit: 50

`transactionMaxSpans: Infinity`

transactionSampleRate: 1.0

```
require('elastic-apm-node').start({  
  ignoreUrls: [  
    '/ping',  
    /^\/admin\/i  
  ]  
})
```

```
require('elastic-apm-node').start({  
  ignoreUserAgents: [  
    'curl/',  
    /pingdom/i  
  ]  
})
```

More information

elastic.co/guide/en/apm/get-started

elastic.co/guide/en/apm/agent/nodejs

Getting involved





[github.com / elastic / apm-agent-nodejs](https://github.com/elastic/apm-agent-nodejs)

[github.com / elastic / apm-server](https://github.com/elastic/apm-server)

Q&A

elasticon AMA booth

&

discuss.elastic.co

Thank you

