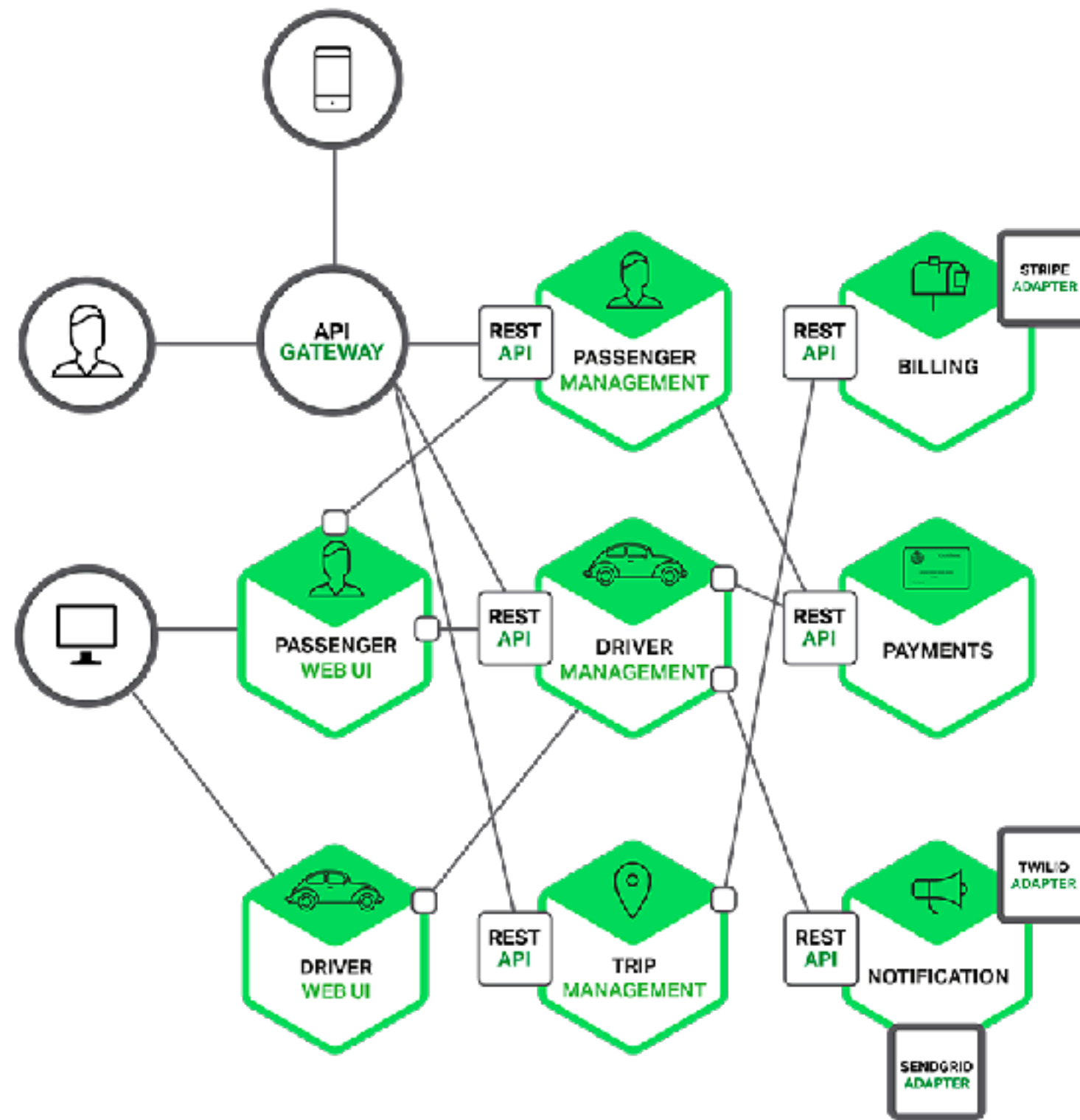


Monitoring with Mocrservice



Microservice



Monitoring



Monitoring

Fatals



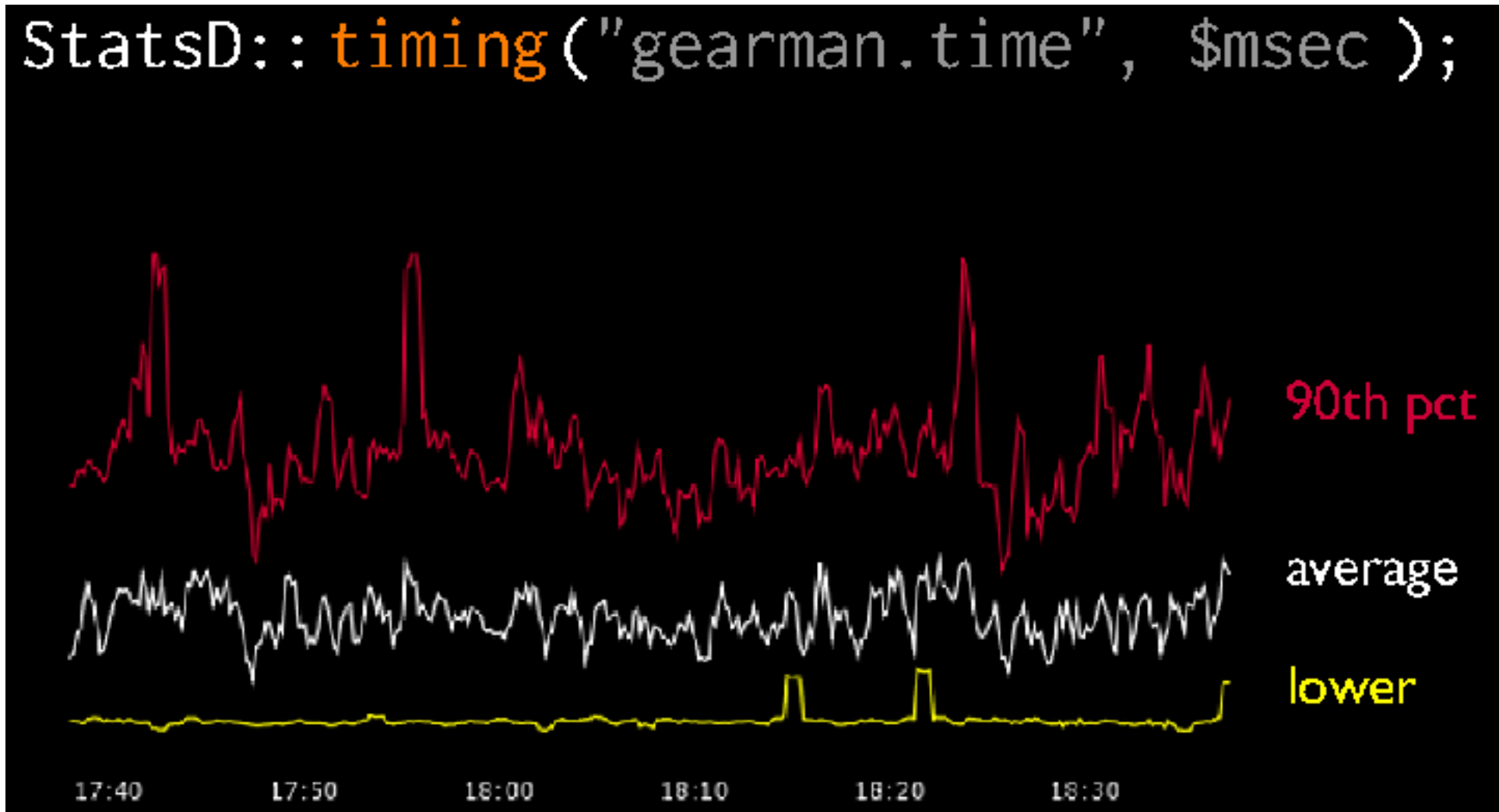
Errors



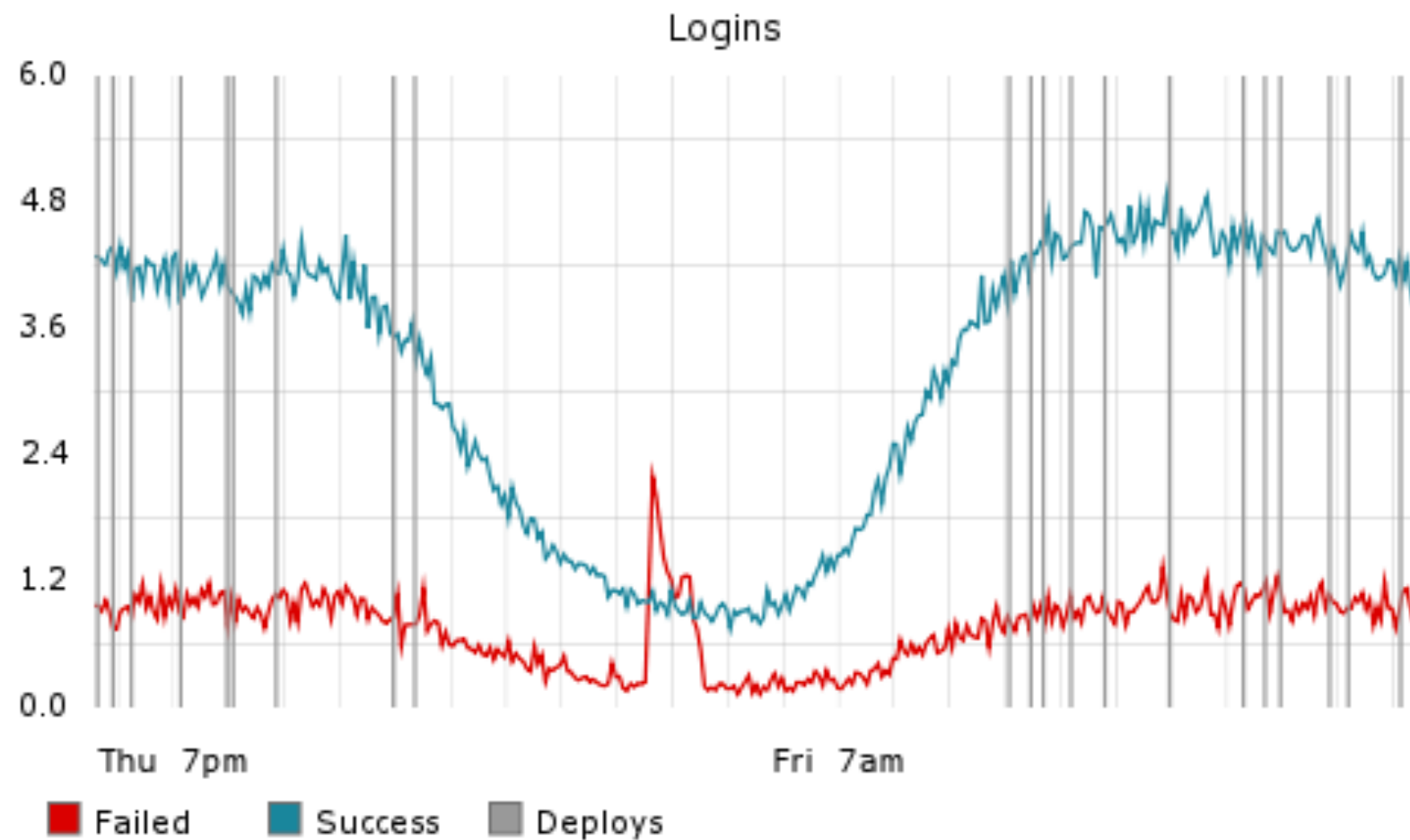
Warnings



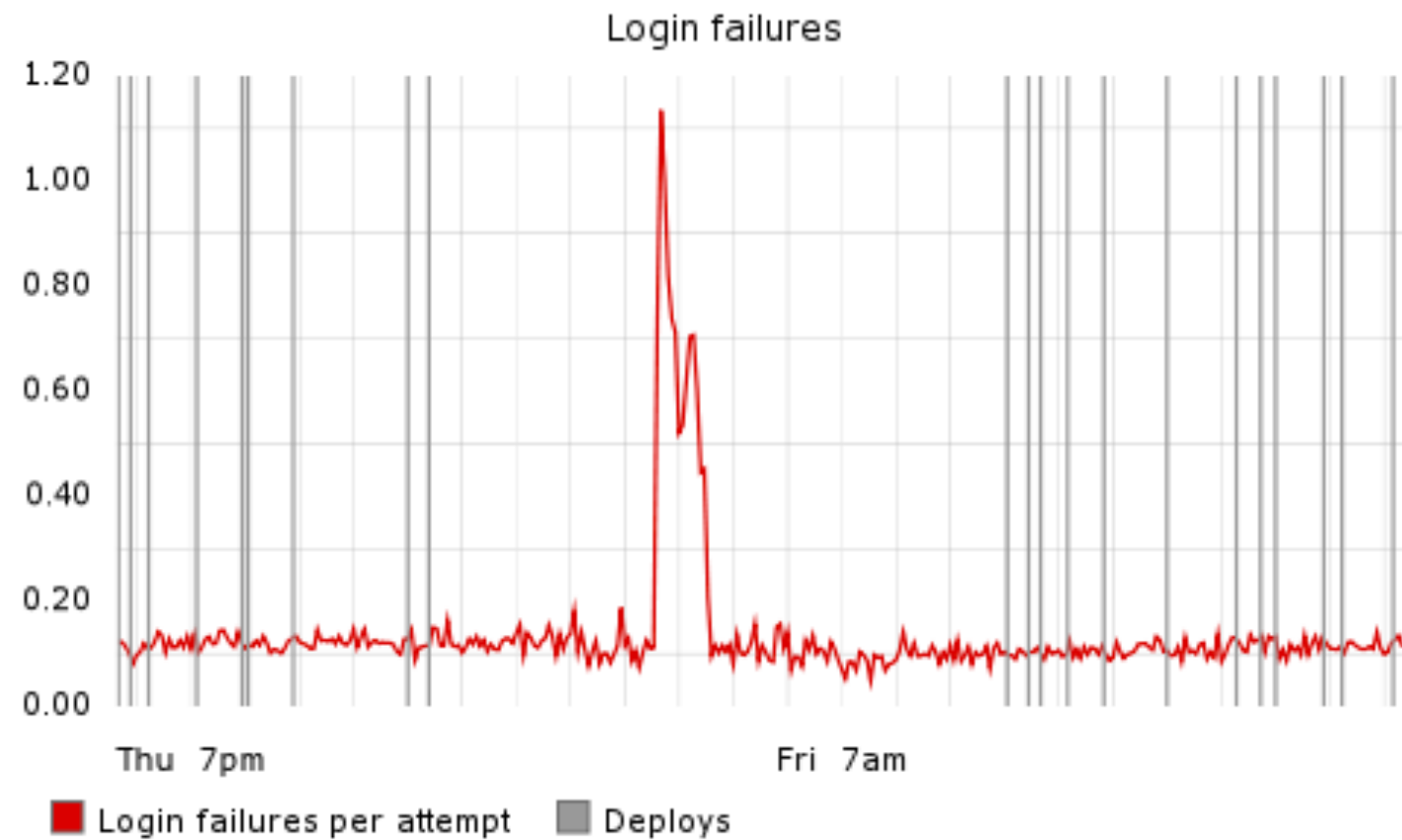
Monitoring



Monitoring



Monitoring



Spring Boot Actuator



Enable Actuator

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-actuator</artifactId>  
</dependency>
```



List of endpoint

/health

/info

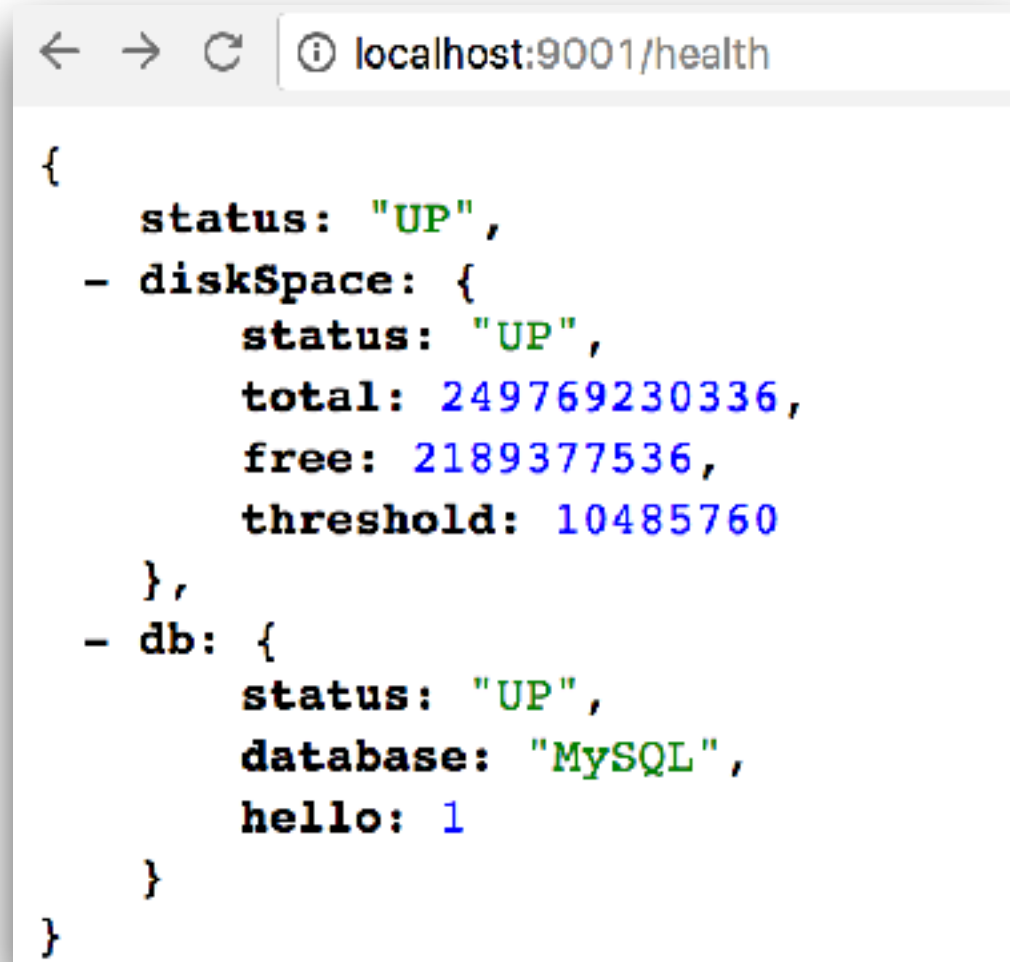
/metrics

/trace



Health

Show application health information

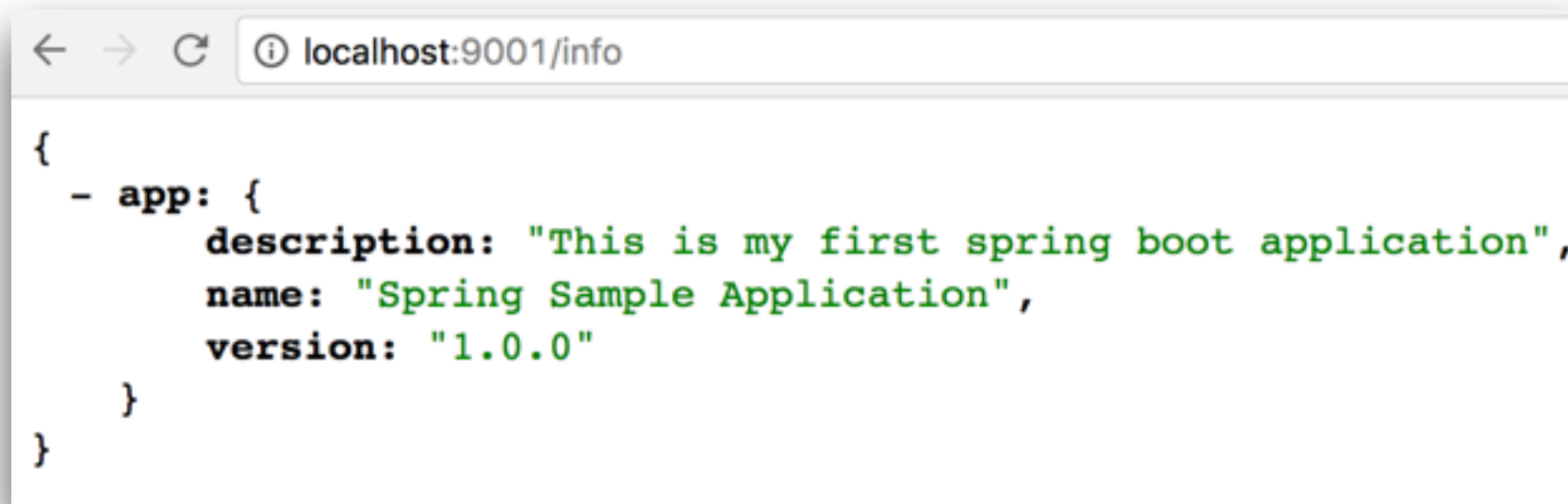
A screenshot of a web browser window. The address bar shows 'localhost:9001/health'. The main content area displays a JSON object representing the application's health status. The JSON is formatted with syntax highlighting: strings are in green, numbers in blue, and property names in black. The structure is a root object with 'status' and two array elements: 'diskSpace' and 'db'. 'diskSpace' is an object with 'status', 'total', 'free', and 'threshold' properties. 'db' is an object with 'status', 'database', and 'hello' properties.

```
{
  status: "UP",
  - diskSpace: {
    status: "UP",
    total: 249769230336,
    free: 2189377536,
    threshold: 10485760
  },
  - db: {
    status: "UP",
    database: "MySQL",
    hello: 1
  }
}
```



Info

Information of your application

A screenshot of a web browser window. The address bar shows 'localhost:9001/info'. The main content area displays a JSON object representing application information.

```
{  
  - app: {  
    description: "This is my first spring boot application",  
    name: "Spring Sample Application",  
    version: "1.0.0"  
  }  
}
```



Info

Update application.properties

info.app.name=Spring Sample Application

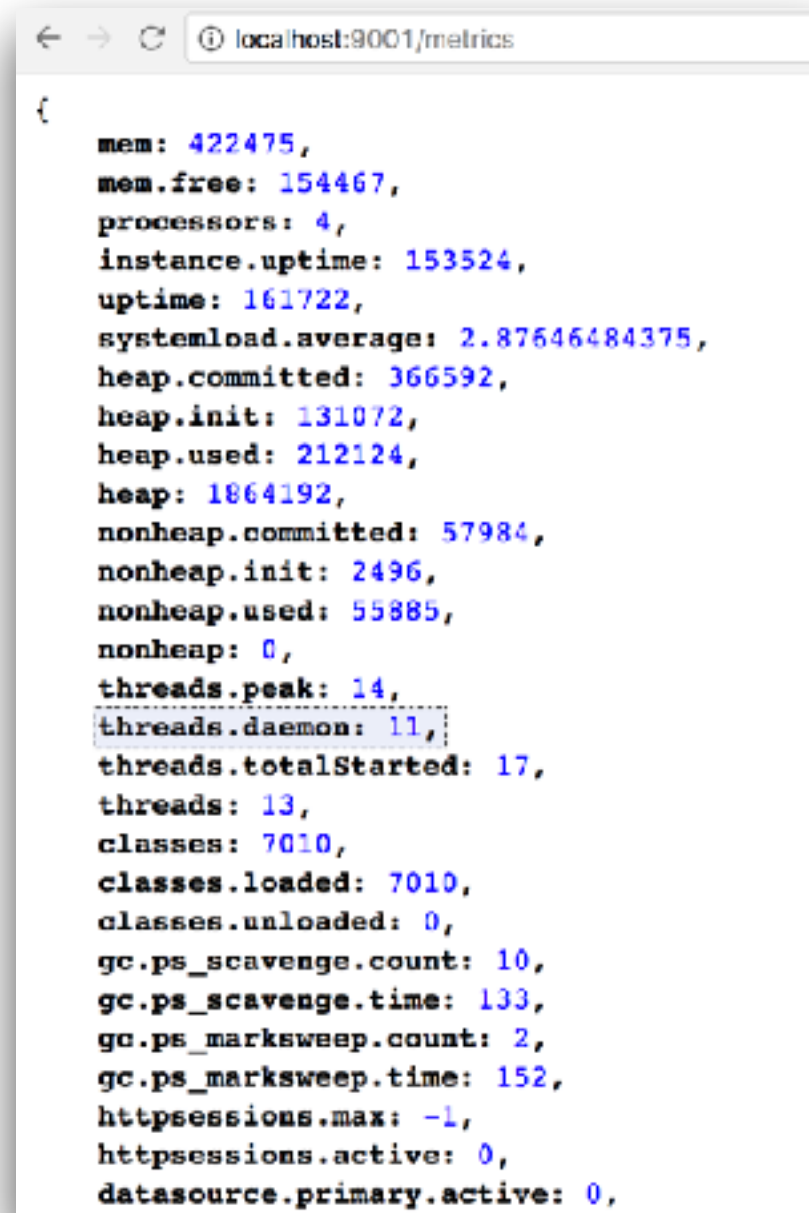
info.app.description=This is my first spring boot application

info.app.version=1.0.0



Metrics

Show metrics of your application



```
{
  mem: 422475,
  mem.free: 154467,
  processors: 4,
  instance.uptime: 153524,
  uptime: 161722,
  systemload.average: 2.87646484375,
  heap.committed: 366592,
  heap.init: 131072,
  heap.used: 212124,
  heap: 1864192,
  nonheap.committed: 57984,
  nonheap.init: 2496,
  nonheap.used: 55885,
  nonheap: 0,
  threads.peak: 14,
  threads.daemon: 11,
  threads.totalStarted: 17,
  threads: 13,
  classes: 7010,
  classes.loaded: 7010,
  classes.unloaded: 0,
  gc.ps_scavenge.count: 10,
  gc.ps_scavenge.time: 133,
  gc.ps_marksweep.count: 2,
  gc.ps_marksweep.time: 152,
  httpsessions.max: -1,
  httpsessions.active: 0,
  datasource.primary.active: 0,
```



Trace

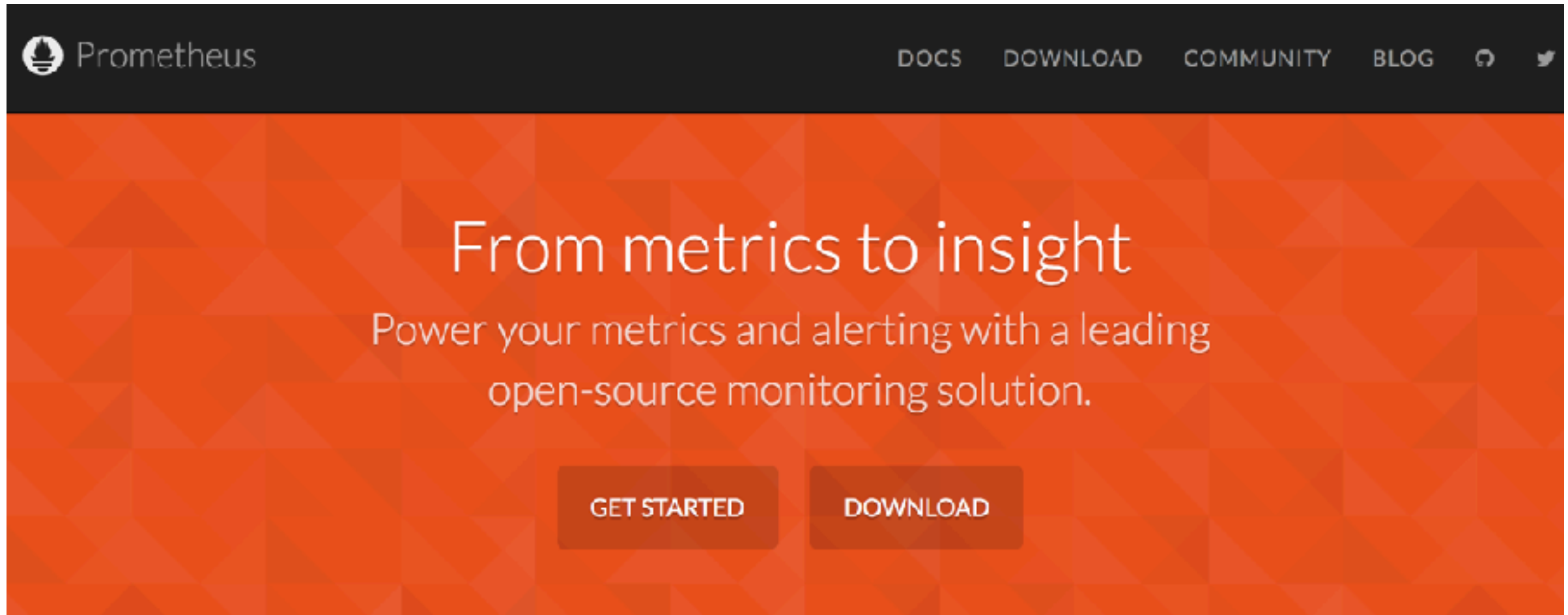
Show trace information via HTTP request



A screenshot of a web browser window displaying trace information. The address bar shows 'localhost:9001/trace'. The main content area displays a JSON array of three trace entries. Each entry contains a timestamp, an 'info' object with 'method' and 'path', and a truncated 'headers' object. The first two entries are for GET requests to '/user/1' and '/favicon.ico'. The third entry is for a GET request to '/user/1' with a dashed box around the 'headers' field.

```
[
  - {
    timestamp: 1491870774795,
    - info: {
      method: "GET",
      path: "/user/1",
      + headers: {...}
    }
  },
  - {
    timestamp: 1491870774775,
    - info: {
      method: "GET",
      path: "/favicon.ico",
      + headers: {...}
    }
  },
  - {
    timestamp: 1491870774557,
    - info: {
      method: "GET",
      path: "/user/1",
      + headers: {...}
    }
  },
],
```

Working with Prometheus



<https://prometheus.io/>



Download

prometheus

The Prometheus monitoring system and time series database. [prometheus/prometheus](#)

2.0.0-alpha.0 / 2017-04-10 [Pre-release](#) [Release notes](#)

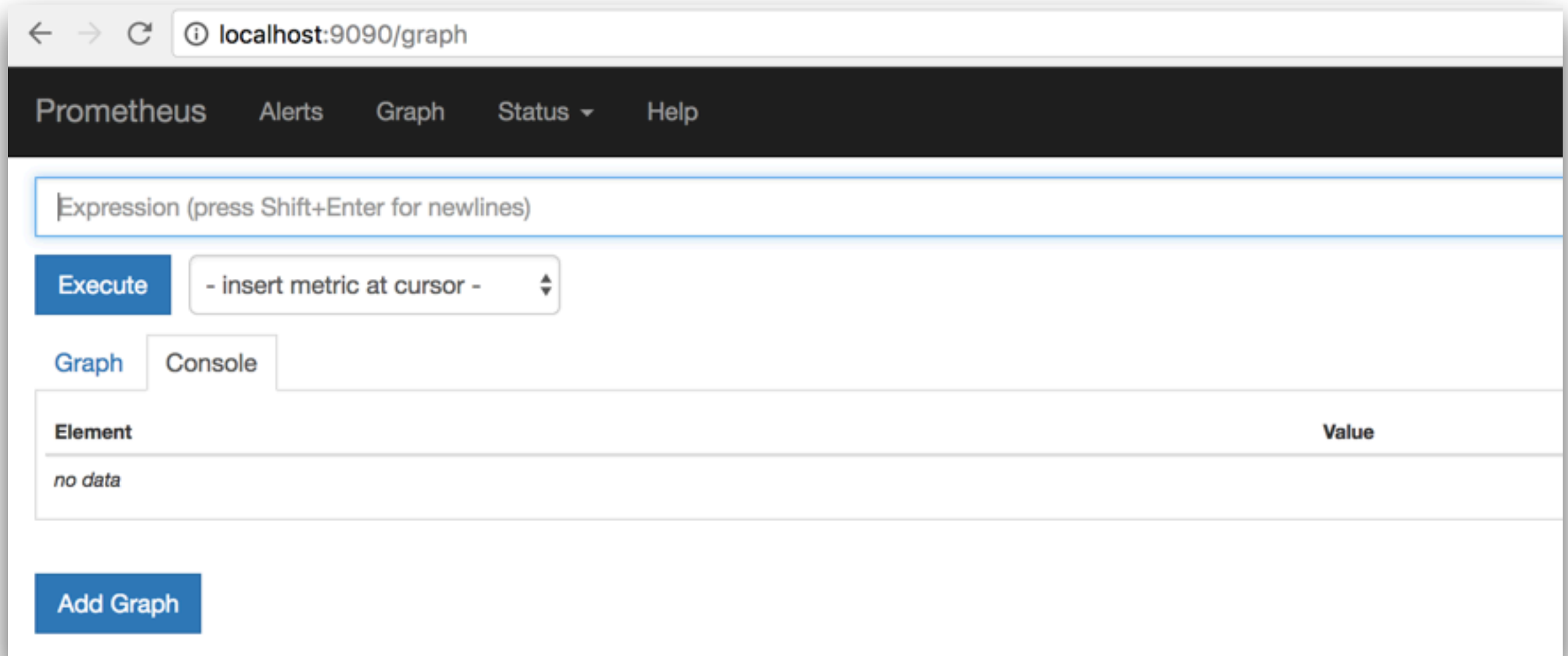
File name	OS	Arch	Size
prometheus-2.0.0-alpha.0.darwin-amd64.tar.gz	darwin	amd64	16.20 MiB
prometheus-2.0.0-alpha.0.linux-amd64.tar.gz	linux	amd64	16.22 MiB
prometheus-2.0.0-alpha.0.windows-amd64.tar.gz	windows	amd64	16.16 MiB

<https://prometheus.io/download/>



Start server

./prometheus



Start workshop



Spring boot and Prometheus

<dependency>

<groupId>org.springframework.boot</groupId>

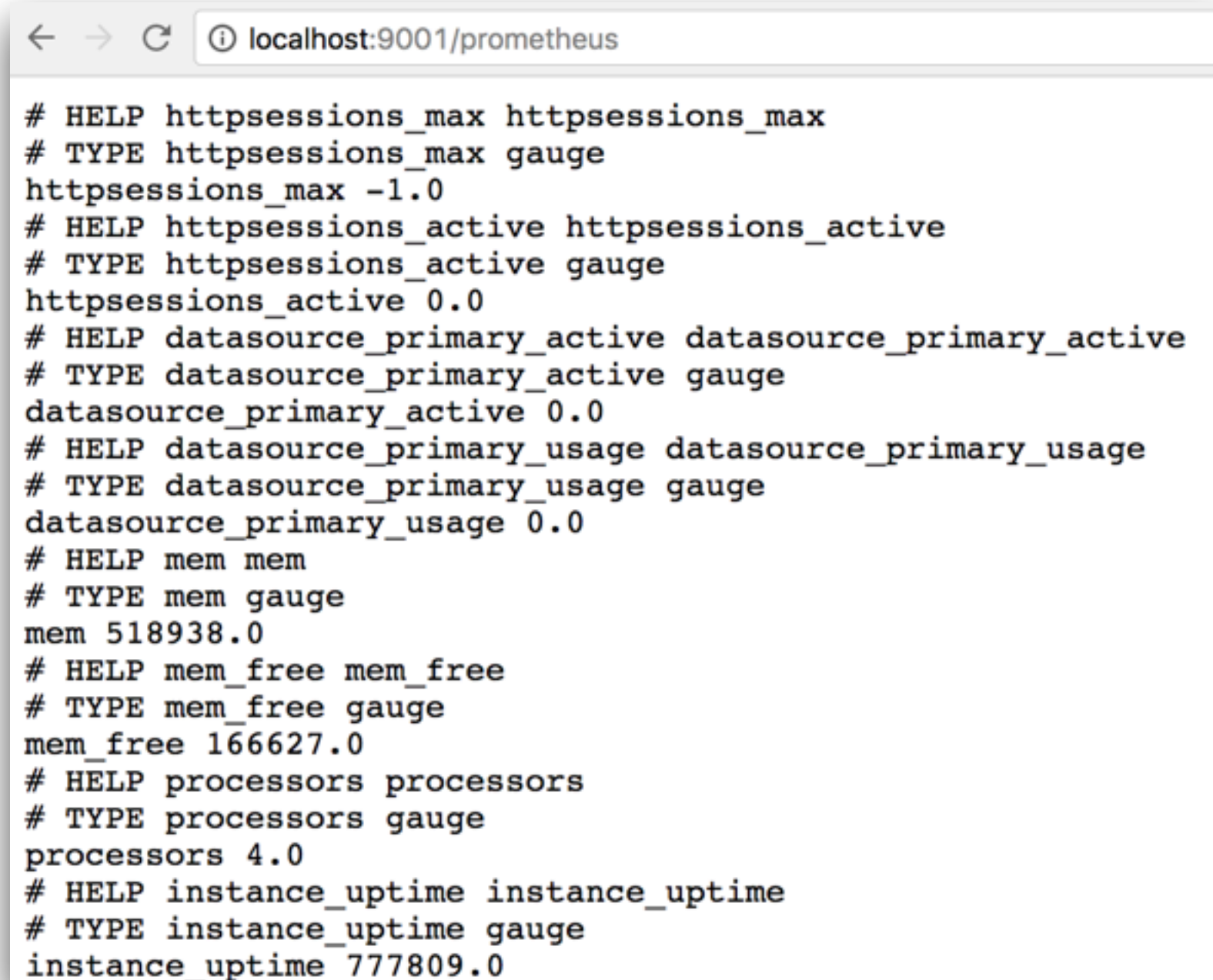
<artifactId>**prometheus-spring-boot-starter**</artifactId>

<version>**1.0.1**</version>

</dependency>



Prometheus endpoint



A screenshot of a web browser window displaying the Prometheus endpoint output. The browser's address bar shows 'localhost:9001/prometheus'. The page content is a text-based response containing several lines of Prometheus scrape data. Each line represents a different metric, including 'httpsessions_max', 'httpsessions_active', 'datasource_primary_active', 'datasource_primary_usage', 'mem', 'mem_free', 'processors', and 'instance_uptime'. Each line includes a help comment, the metric name and type, and the current value.

```
< > ↻ ⓘ localhost:9001/prometheus

# HELP httpsessions_max httpsessions_max
# TYPE httpsessions_max gauge
httpsessions_max -1.0
# HELP httpsessions_active httpsessions_active
# TYPE httpsessions_active gauge
httpsessions_active 0.0
# HELP datasource_primary_active datasource_primary_active
# TYPE datasource_primary_active gauge
datasource_primary_active 0.0
# HELP datasource_primary_usage datasource_primary_usage
# TYPE datasource_primary_usage gauge
datasource_primary_usage 0.0
# HELP mem mem
# TYPE mem gauge
mem 518938.0
# HELP mem_free mem_free
# TYPE mem_free gauge
mem_free 166627.0
# HELP processors processors
# TYPE processors gauge
processors 4.0
# HELP instance_uptime instance_uptime
# TYPE instance_uptime gauge
instance_uptime 777809.0
```



Add new target

./prometheus.yml

scrape_configs:

- **job_name: 'user-service'**

scrape_interval: 5s

metrics_path: '/prometheus'

static_configs:

- **targets: ['localhost:9001']**



Start server

./prometheus

Targets

prometheus

Endpoint	State	Labels	Last Scrape	Error
http://localhost:9090/metrics	UP	instance="localhost:9090"	14.839s ago	

user-service

Endpoint	State	Labels	Last Scrape	Error
http://localhost:9001/prometheus	UP	instance="localhost:9001"	448ms ago	

Working with graph

