

Proyecto de paradigmas

Desarrollo de una API para el Procesamiento de Logs de una Aplicación Spring Boot utilizando Programación Funcional

Documentación del Proyecto - Procesamiento de Logs (ProyectoPP)

Participantes:

Kevin Beita Marín

Greivin Narváez Jiménez

Francisco Araya Leitón

Yutaro Cubero Abe

Curso:

Paradigmas de programación.

Profesor:

Allam Chaves Zamora

Universidad Nacional de Costa Rica

Sede Interuniversitaria de Alajuela

2025

Este proyecto implementa una API en Spring Boot para procesar logs de la aplicación. La API genera diferentes reportes analíticos (errores, tiempos de respuesta, uso de endpoints, alertas críticas y estado general) utilizando técnicas de programación funcional de Java; para así hacerla totalmente funcional para el que desee implementarla en sus propios proyectos.

A continuación, se procederá a dar las instrucciones para el uso y prueba del api y ejecutar la aplicación.

Instrucciones:

1. Abrir la carpeta del proyecto en un ide.
2. En la terminal integrada ejecutar: `mvn clean package`
3. Ejecutar: `mvn spring-boot:run`
4. Acceder a la aplicación en `http://localhost:8080`
5. Abrir Swagger UI en <http://localhost:8080/swagger-ui.html> o <http://localhost:8080/swagger-ui/index.html>

Siguiendo estos pasos, podrá ejecutar con éxito la aplicación.

Ahora bien, por motivos de documentación, se procederá a dar evidencia de cada endpoint del api, con sus respectivas capturas de pantalla. Estas capturas, son directamente desde Swagger UI, donde se puede mostrar de manera visual la ejecución de cada endpoint y su salida en formato JSON.

Reporte de errores:

reports-endpoint			^
GET	/api/reports/usage/top-bottom	Top 3 y Bottom 3 de endpoints	▼
GET	/api/reports/usage/by-method	Conteo de peticiones por método HTTP	▼
GET	/api/reports/status	Estado general de la aplicación	▼
GET	/api/reports/response/sum-by-endpoint	Suma de tiempos por endpoint	▼
GET	/api/reports/response/stats	Estadísticas de tiempos de respuesta	▼
GET	/api/reports/errors/top3	Top 3 errores	▼
GET	/api/reports/errors/peak-hours	Horas pico de errores	▼
GET	/api/reports/errors/by-code	Errores por código	▼
GET	/api/reports/critical/events	Eventos críticos	▼
GET	/api/reports/critical/count	Cantidad de eventos críticos	▼

GET

/api/reports/errors/by-code

Errores por código

⌵

Devuelve la lista de errores agrupados por código HTTP o nombre de excepción con su cantidad.

Parameters

Cancel

No parameters

Execute

Clear

Responses

Curl

curl -X 'GET' \n'http://localhost:8080/api/reports/errors/by-code' \n-H 'accept: application/json'

Request URL

http://localhost:8080/api/reports/errors/by-code

Code

Details

200

Response body

[]

Download

Response headers

connection: keep-alive
content-type: application/json
date: Sun, 21 Sep 2025 17:31:10 GMT
keep-alive: timeout=60
transfer-encoding: chunked

Responses

Code

Description

Links

200

Lista de errores agrupados por código

No links

Media type

application/json

Controls Accept header

Example Value | Schema

```
[  
  {  
    "error": "404",  
    "count": 15  
  },  
  {  
    "error": "500",  
    "count": 4  
  }  
]
```

Top3 mostrando los 3 errores más frecuentes:

GET

/api/reports/errors/top3

Top 3 errores

⌵

Retorna los 3 errores más frecuentes detectados en los logs.

Parameters

Cancel

No parameters

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/api/reports/errors/top3' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/api/reports/errors/top3
```

Server response

Code	Details
200	<div>Response body</div> <div>[]</div> <div>Response headers</div> <div>connection: keep-alive content-type: application/json date: Sun, 21 Sep 2025 17:32:20 GMT keep-alive: timeout=60 transfer-encoding: chunked</div>

Responses

Code	Description	Links
200	Lista con los 3 errores más frecuentes	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
[
  {
    "error": "404",
    "count": 15
  },
  {
    "error": "NullPointerException",
    "count": 7
  },
  {
    "error": "500",
    "count": 4
  }
]
```

Mostrando errores agrupados por hora:

GET

/api/reports/errors/peak-hours

Horas pico de errores

^

Devuelve la cantidad de errores agrupados por hora del día.

Parameters

Cancel

No parameters

Execute

Clear

Responses

Curl

curl -X 'GET' \ 'http://localhost:8080/api/reports/errors/peak-hours' \ -H 'accept: application/json'

Request URL

http://localhost:8080/api/reports/errors/peak-hours

Server response

Code

Details

200

Response body

[]

Response headers

connection: keep-alive
content-type: application/json
date: Sun, 21 Sep 2025 17:35:12 GMT
keep-alive: timeout=60
transfer-encoding: chunked

Responses

Code

Description

Links

200

Lista de errores agrupados por hora

No links

Media type

application/json

Controls Accept header.

Example Value | Schema

[{ "hour": 14, "count": 10 }, { "hour": 15, "count": 0 }]

Tiempo de respuestas:

Mostrando promedio, min, max, mediana:

GET

/api/reports/response/stats Estadísticas de tiempos de respuesta

Devuelve promedio, mínimo, máximo y mediana de los tiempos de respuesta.

Parameters

Cancel

No parameters

Execute

Clear

Responses


Curl

```
curl -X 'GET' \
'http://localhost:8080/api/reports/response/stats' \
-H 'accept: application/json'
```

Request URL

```
http://localhost:8080/api/reports/response/stats
```

Server response

Code	Details	
200	<div>Response body<div><pre>{ "avg": 64.28571428571429, "min": 5, "max": 228, "median": 40}</pre></div><div> Download</div></div> <div>Response headers<div><pre>connection: keep-alive content-type: application/json date: Sun, 21 Sep 2025 17:37:58 GMT keep-alive: timeout=60 transfer-encoding: chunked</pre></div></div>	
Responses		
Code	Description	Links
200	Estadísticas de tiempos de respuesta	No links
<div>Media type<div><div>application/json</div><div>▼</div></div><div>Controls Accept header.</div><div>Example Value Schema</div><div><pre>{ "average": 230.5, "min": 12, "max": 1200, "median": 210}</pre></div></div>		

GET

/api/reports/response/sum-by-endpoint

Suma de tiempos por endpoint

Devuelve la suma de tiempos de respuesta acumulados por cada endpoint.

Parameters

No parameters

Cancel

Execute

Clear

Responses

Curl

curl -X 'GET' \n'http://localhost:8080/api/reports/response/sum-by-endpoint' \n-H 'accept: application/json'

Request URL

http://localhost:8080/api/reports/response/sum-by-endpoint

Server response

Code

Details

200

Response body

```
[
  {
    "endpoint": "/v3/api-docs",
    "totalMs": 228
  },
  {
    "endpoint": "/api/reports/errors/by-code",
    "totalMs": 85
  },
  {
    "endpoint": "/swagger-ui/index.html",
    "totalMs": 51
  },
  {
    "endpoint": "/api/reports/response/stats",
    "totalMs": 48
  },
  {
    "endpoint": "/swagger-ui/swagger-initializer.js",
    "totalMs": 40
  },
  {
    "endpoint": "/v3/api-docs/swagger-config",
    "totalMs": 35
  },
  {
    "endpoint": "/api/reports/errors/top3",
```

Download

Response headers

```
connection: keep-alive
content-type: application/json
date: Sun, 21 Sep 2025 17:39:27 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

Responses

Responses

Code	Description	Links
200	Lista con tiempos de respuesta acumulados por endpoint	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
[
  {
    "endpoint": "/api/users",
    "sum": 4520
  },
  {
    "endpoint": "/api/login",
    "sum": 2800
  }
]
```

Mostrando top y least usados:

GET /api/reports/usage/top-bottom Top 3 y Bottom 3 de endpoints

Devuelve los 3 endpoints más usados y los 3 menos usados.

Parameters

Cancel

No parameters

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/api/reports/usage/top-bottom' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/api/reports/usage/top-bottom
```

Server response

Code

Details

200

Response body

```
{
  "top3": [
    {
      "endpoint": "/api/reports/errors/by-code",
      "total": 1
    },
    {
      "endpoint": "/api/reports/errors/peak-hours",
      "total": 1
    },
    {
      "endpoint": "/api/reports/errors/top3",
      "total": 1
    }
  ],
  "bottom3": [
    {
      "endpoint": "/swagger-ui/swagger-initializer.js",
      "total": 1
    },
    {
      "endpoint": "/v3/api-docs",
      "total": 1
    },
    {
      "endpoint": "/v3/api-docs/swagger-config",
      "total": 1
    }
  ]
}
```

Download

Response headers

```
connection: keep-alive
content-type: application/json
date: Sun, 21 Sep 2025 17:42:10 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

Responses

Code

Description

Links

200

Mapa con endpoints más y menos usados

No links

Media type

application/json

Controls Accept header.

Example Value |

```
{
  "mostUsed": [
    "/api/login",
    "/api/users"
  ],
  "leastUsed": [
    "/api/admin",
    "/api/status"
  ]
}
```


Mostrando conteo por método HTTP:

GET /api/reports/usage/by-method Conteo de peticiones por método HTTP

Devuelve cuántas peticiones se hicieron por método (GET, POST, PUT, DELETE).

Parameters

No parameters

Cancel

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:8080/api/reports/usage/by-method' \
-H 'accept: application/json'
```

Request URL

```
http://localhost:8080/api/reports/usage/by-method
```

Server response

Code

Details

200

Response body

```
{
  "method": "GET",
  "total": 10
}
```

 Download

Response headers

```
connection: keep-alive
content-type: application/json
date: Sun, 21 Sep 2025 17:43:58 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

Responses

Code

Description

Links

200

Lista de conteos por método HTTP

No links

Media type

application/json

Controls

Accept header.

Example Value

Schema

```
{
  "method": "GET",
  "count": 1200
},
{
  "method": "POST",
  "count": 300
}
```

Alertas y eventos críticos:

Mostrando lista de alertas críticas:

GET

/api/reports/critical/events

Eventos críticos

⌵

Devuelve la lista de eventos críticos detectados en los logs.

Parameters

Cancel

No parameters

Execute

Clear

Responses

Curl

curl -X 'GET' \n'http://localhost:8080/api/reports/critical/events' \n-H 'accept: application/json'

Request URL

http://localhost:8080/api/reports/critical/events

Server response

Code

Details

200

Response body

[]

Download

Response headers

connection: keep-alive\ncontent-type: application/json\ndate: Sun, 21 Sep 2025 17:45:39 GMT\nkeep-alive: timeout=60\ntransfer-encoding: chunked

Responses

Code

Description

Links

200

Lista de eventos críticos

No links

Media type

application/json

Controls Accept header.

Example Value

Schema

{\n "id": "A1",\n "type": "HIGH_ERROR_RATE",\n "description": "5 errores 500 en 1 minuto",\n "timestamp": "2025-09-20T14:22:00Z"\n}

GET /api/reports/critical/count Cantidad de eventos críticos

Devuelve el número total de eventos críticos detectados.

Parameters Cancel

No parameters

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:8080/api/reports/critical/count' \
-H 'accept: application/json'
```

Request URL

```
http://localhost:8080/api/reports/critical/count
```

Server response


Code

Details

200

Response body

```
{  
  "count": 0  
}
```

Download

Response headers

```
connection: keep-alive  
content-type: application/json  
date: Sun, 21 Sep 2025 17:46:59 GMT  
keep-alive: timeout=60  
transfer-encoding: chunked
```

Responses

Code	Description	Links
200	Cantidad total de eventos críticos	No links

Media type

application/json

▼

Controls Accept header.

Example Value |

```
{  
  "count": 3  
}
```

Estado de la aplicación:

Mostrando estado general:

GET

/api/reports/status Estado general de la aplicación

^

Devuelve un resumen con total de peticiones, errores y tiempo promedio de respuesta.

Parameters

Cancel

No parameters

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:8080/api/reports/status' \
-H 'accept: application/json'
```

Request URL

```
http://localhost:8080/api/reports/status
```

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "totalRequests": 13, "totalErrors": 0, "avgResponseTime": 40.61538461538461 }</pre></div><div>Download</div></div> <div><div>Response headers</div><div><pre>connection: keep-alive content-type: application/json date: Sun, 21 Sep 2025 17:48:06 GMT keep-alive: timeout=60 transfer-encoding: chunked</pre></div></div>
Responses	
Code	Description
200	<div>Estado de la aplicación</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "requestsProcessed": 1520, "errors": 26, "avgResponseTime": 245.6 }</pre></div>

Datos de prueba desde Postman

Entrada:

GET <http://localhost:8080/api/usage/by-method>

Salida:

Error 404 Not Found

Entrada:

POST <http://localhost:8080/api/reports/usage/by-method>

Salida:

Error 405 Method Not Allowed

Entrada:

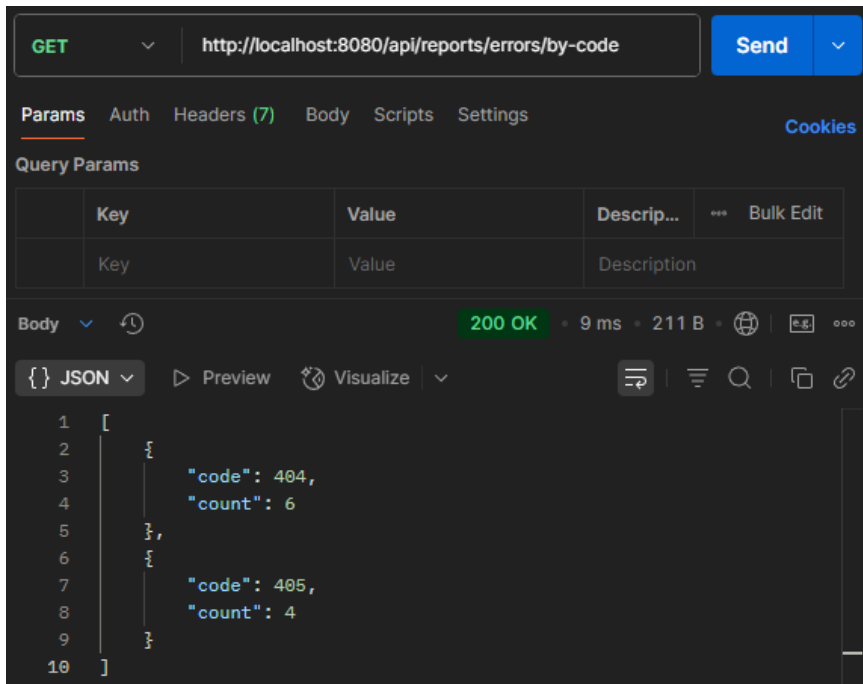
GET <http://localhost:8080/api/reports/errors/by-code>

Salida:

Código 200 OK

Nota: ‘Los datos iniciales puede reiterarse con el fin de aumentar los contadores de eventos en los endpoint’

Cantidad de errores por código



GET ▼ http://localhost:8080/api/reports/errors/by-code Send ▼

Params **Auth** **Headers (7)** **Body** **Scripts** **Settings** Cookies

Query Params

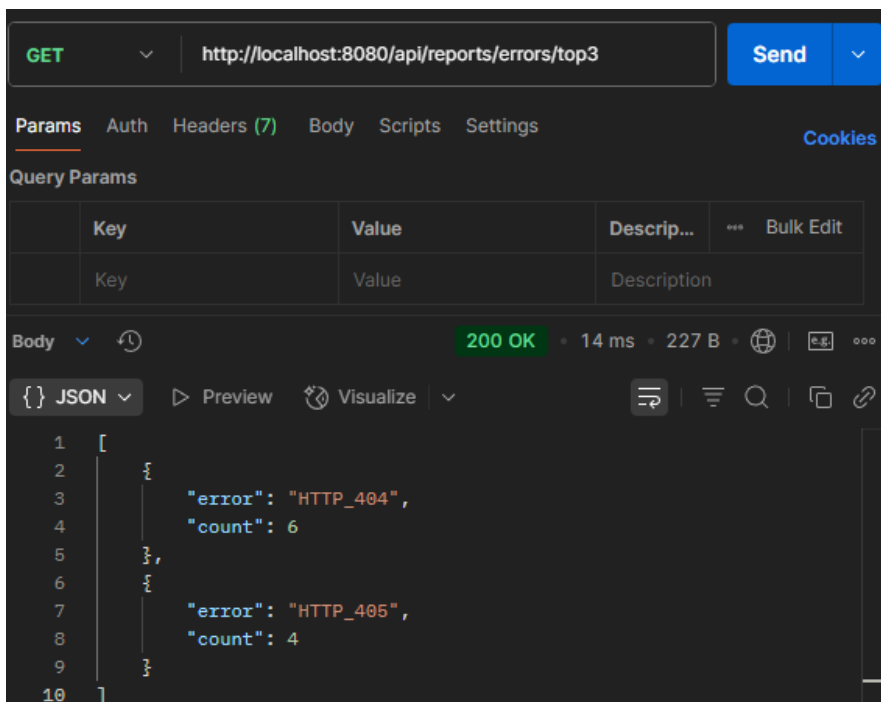
	Key	Value	Descrip...	...	Bulk Edit
	Key	Value	Description		

Body ▼ 🔄 200 OK • 9 ms • 211 B • 🌐 📄 ⋮

{} **JSON** ▼ ▶ **Preview** 🔄 **Visualize** ▼ 🔄 ☰ 🔍 📄 🔗

```
1  [
2    {
3      "code": 404,
4      "count": 6
5    },
6    {
7      "code": 405,
8      "count": 4
9    }
10 ]
```

Top 3 errores más frecuentes



GET ▼ http://localhost:8080/api/reports/errors/top3 Send ▼

Params **Auth** **Headers (7)** **Body** **Scripts** **Settings** Cookies

Query Params

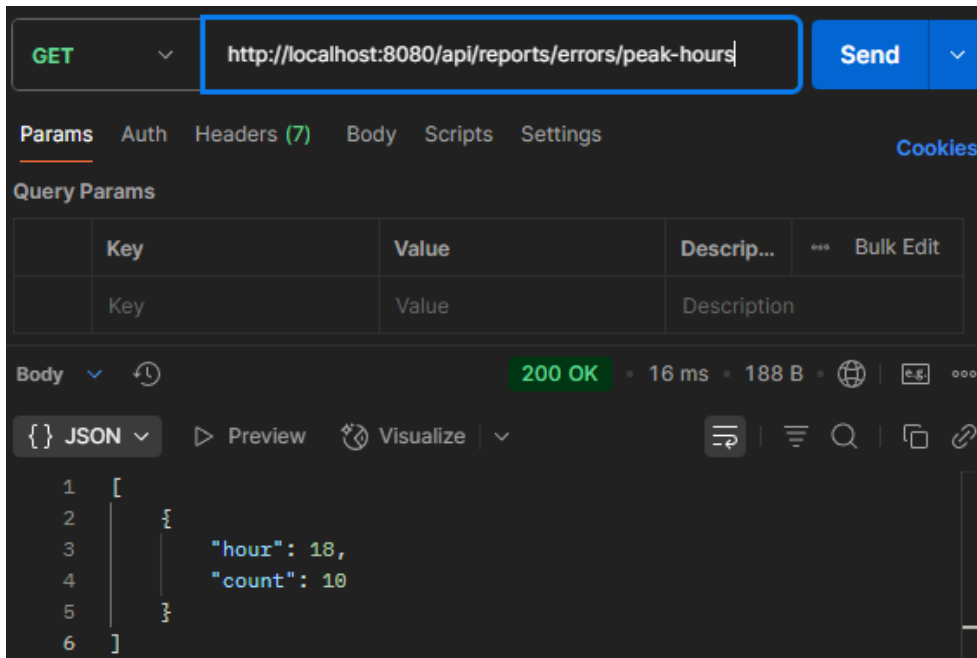
	Key	Value	Descrip...	...	Bulk Edit
	Key	Value	Description		

Body ▼ 🔄 200 OK • 14 ms • 227 B • 🌐 📄 ⋮

{} **JSON** ▼ ▶ **Preview** 🔄 **Visualize** ▼ 🔄 ☰ 🔍 📄 🔗

```
1  [
2    {
3      "error": "HTTP_404",
4      "count": 6
5    },
6    {
7      "error": "HTTP_405",
8      "count": 4
9    }
10 ]
```

Horas pico donde suceden más errores



GET <http://localhost:8080/api/reports/errors/peak-hours> Send

Params Auth Headers (7) Body Scripts Settings Cookies

Query Params

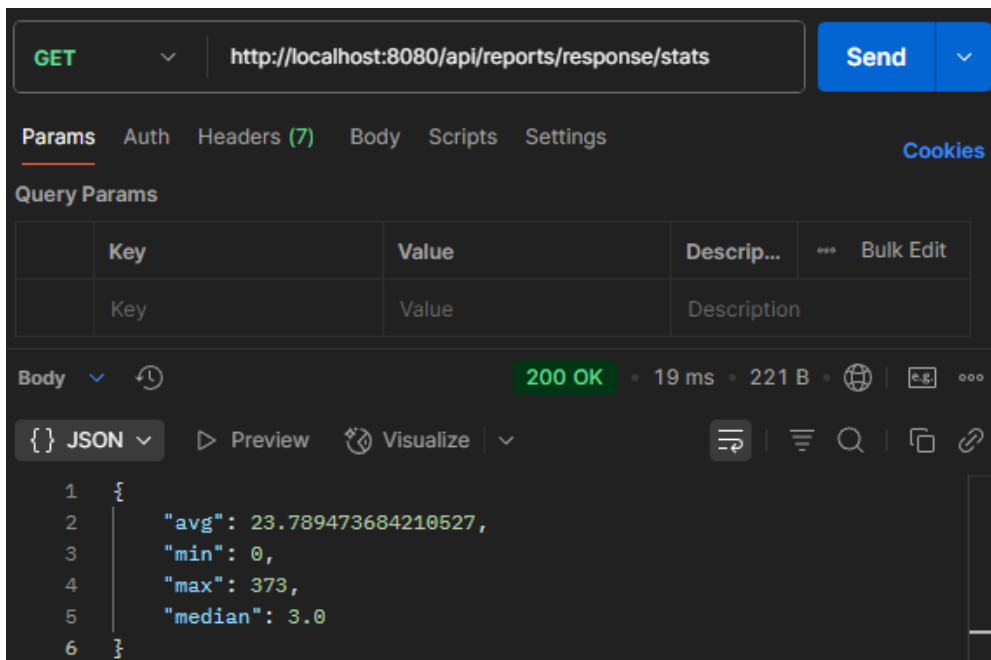
	Key	Value	Descrip...	Bulk Edit
	Key	Value	Description	

Body 200 OK • 16 ms • 188 B •

{} JSON Preview Visualize

```
1  [  
2    {  
3      "hour": 18,  
4      "count": 10  
5    }  
6  ]
```

Estadísticas de tiempos de respuesta



GET <http://localhost:8080/api/reports/response/stats> Send

Params Auth Headers (7) Body Scripts Settings Cookies

Query Params

	Key	Value	Descrip...	Bulk Edit
	Key	Value	Description	

Body 200 OK • 19 ms • 221 B •

{} JSON Preview Visualize

```
1  {  
2    "avg": 23.789473684210527,  
3    "min": 0,  
4    "max": 373,  
5    "median": 3.0  
6  }
```

Suma de milisegundos empleados por endpoint

GET

http://localhost:8080/api/reports/response/sum-by-endpoint

Send

Params

Auth

Headers (7)

Body

Scripts

Settings

Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body

200 OK

18 ms

607 B

Save Response

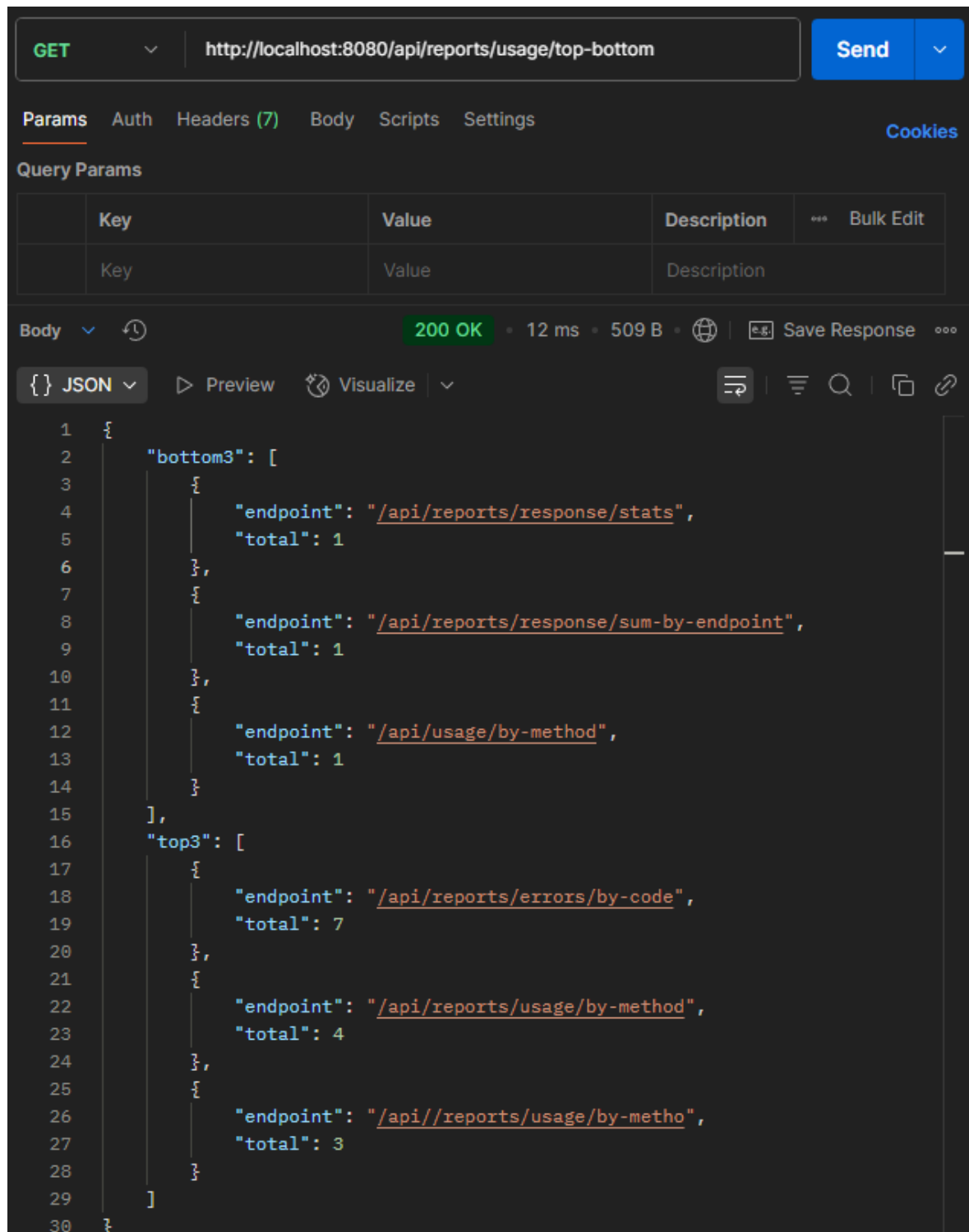
{ } JSON

Preview

Visualize

```
2  {
3      "endpoint": "/api/reports/usage/by-method",
4      "totalMs": 391
5  },
6  {
7      "endpoint": "/api/reports/errors/by-code",
8      "totalMs": 27
9  },
10 {
11     "endpoint": "/api/reports/response/stats",
12     "totalMs": 14
13 },
14 {
15     "endpoint": "/api/reports/errors/top3",
16     "totalMs": 10
17 },
18 {
19     "endpoint": "/api/reports/errors/peak-hours",
20     "totalMs": 8
21 },
22 {
23     "endpoint": "/api/usage/by-method",
24     "totalMs": 7
25 },
26 {
27     "endpoint": "/api//reports/usage/by-metho",
28     "totalMs": 6
29 },
30 {
31     "endpoint": "/api//reports/usage/by-method",
32     "totalMs": 3
33 }
34 ]
```


Top 3 endpoints más y menos utilizados



GET ⌵ http://localhost:8080/api/reports/usage/top-bottom Send ⌵

Params **Auth** **Headers (7)** **Body** **Scripts** **Settings** Cookies

Query Params

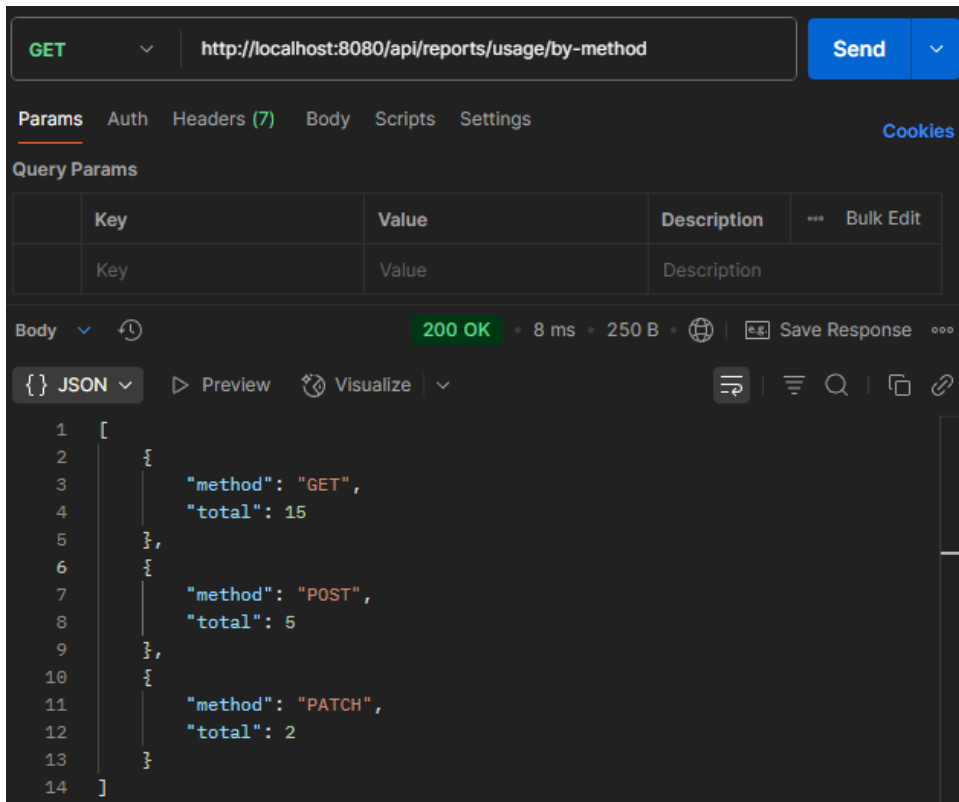
	Key	Value	Description	⋮ Bulk Edit
	Key	Value	Description	

Body ⌵ 🕒 200 OK • 12 ms • 509 B 🌐 📄 Save Response ⋮

{ } JSON ⌵ ▶ Preview 🔄 Visualize ⌵ 🔍 📄 🔗

```
1 {
2   "bottom3": [
3     {
4       "endpoint": "/api/reports/response/stats",
5       "total": 1
6     },
7     {
8       "endpoint": "/api/reports/response/sum-by-endpoint",
9       "total": 1
10    },
11    {
12      "endpoint": "/api/usage/by-method",
13      "total": 1
14    }
15  ],
16  "top3": [
17    {
18      "endpoint": "/api/reports/errors/by-code",
19      "total": 7
20    },
21    {
22      "endpoint": "/api/reports/usage/by-method",
23      "total": 4
24    },
25    {
26      "endpoint": "/api//reports/usage/by-metho",
27      "total": 3
28    }
29  ]
30 }
```

Cantidad de usos por método



GET ▼ http://localhost:8080/api/reports/usage/by-method Send ▼

Params Auth Headers (7) Body Scripts Settings Cookies

Query Params

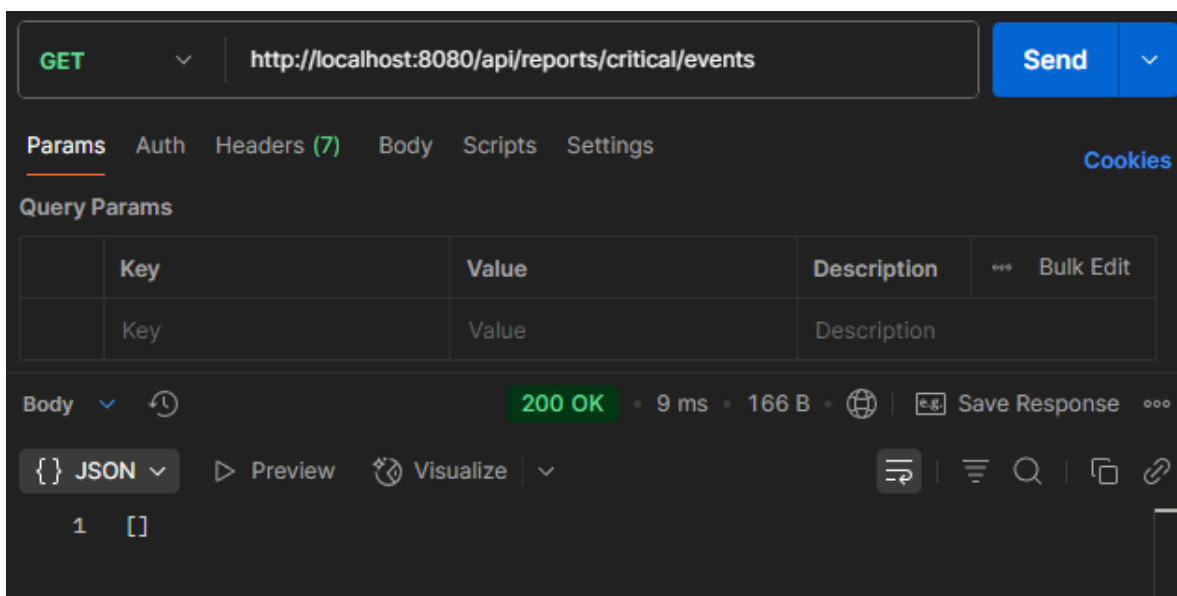
	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body ▼ 🕒 200 OK • 8 ms • 250 B • 🌐 📄 Save Response ...

{} JSON ▼ ▶ Preview 🔗 Visualize ▼ 🔄 ☰ 🔍 📄 🔗

```
1  [
2    {
3      "method": "GET",
4      "total": 15
5    },
6    {
7      "method": "POST",
8      "total": 5
9    },
10   {
11     "method": "PATCH",
12     "total": 2
13   }
14 ]
```

Eventos críticos



GET ▼ http://localhost:8080/api/reports/critical/events Send ▼

Params Auth Headers (7) Body Scripts Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body ▼ 🕒 200 OK • 9 ms • 166 B • 🌐 📄 Save Response ...

{} JSON ▼ ▶ Preview 🔗 Visualize ▼ 🔄 ☰ 🔍 📄 🔗

```
1  []
```

Contador de eventos críticos

GET ▼ http://localhost:8080/api/reports/critical/count Send ▼

Params **Auth** **Headers (7)** **Body** **Scripts** **Settings** Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body ▼ 🕒 200 OK • 11 ms • 175 B • 🌐 | 📄 Save Response ...

{} **JSON** ▼ ▶ Preview 🔄 Visualize ▼ ↺ | ☰ 🔍 | 📄 🔗

```
1 {
2   "count": 0
3 }
```

Estado general de la aplicación

GET ▼ http://localhost:8080/api/reports/status Send ▼

Params **Auth** **Headers (7)** **Body** **Scripts** **Settings** Cookies

Query Params

	Key	Value	Descript...	...	Bulk Edit
	Key	Value	Descrip	View more actions	

Body ▼ 🕒 200 OK • 11 ms • 223 B • 🌐 | 📄 Save Response ...

{} **JSON** ▼ ▶ Preview 🔄 Visualize ▼ ↺ | ☰ 🔍 | 📄 🔗

```
1 {
2   "totalRequests": 25,
3   "totalErrors": 10,
4   "avgResponseMs": 20.04
5 }
```

A su vez, también se procede a mostrar la programación funcional aplicada en la aplicación. Por lo que el proyecto utiliza programación funcional en Java para procesar colecciones de logs. Se emplean pipelines de Streams para filtrar, mapear y agrupar datos sin usar bucles tradicionales. Se promueve la inmutabilidad y se evita modificar colecciones externas dentro de las operaciones de stream.

Ejemplo incluido en el código (cálculo de mediana):

```
List<Long> sorted = durations.stream()  
    .sorted()  
    .collect(Collectors.toList());  
double median = (sorted.size() % 2 == 1)  
    ? sorted.get(sorted.size()/2)  
    : (sorted.get(sorted.size()/2 - 1) + sorted.get(sorted.size()/2)) / 2.0;
```

El proyecto de Procesamiento de Logs implementado en Spring Boot cumple con los objetivos planteados, ofreciendo una API robusta y funcional para el análisis de registros de aplicación. Gracias al uso de técnicas de programación funcional en Java, se logra un procesamiento eficiente de grandes volúmenes de datos, promoviendo la inmutabilidad y evitando efectos secundarios sobre colecciones externas.

Los diferentes reportes generados —errores, tiempos de respuesta, uso de endpoints, alertas críticas y estado general— permiten una visualización clara y estructurada de la información relevante para la gestión y monitoreo de la aplicación. Asimismo, la integración con Swagger UI facilita la interacción directa con los endpoints, proporcionando una experiencia de prueba y documentación accesible y completa.

En definitiva, esta implementación constituye una herramienta versátil y reutilizable para desarrolladores que requieran analizar logs de manera precisa y eficiente, destacando por su facilidad de uso, claridad en los reportes y adopción de buenas prácticas de programación funcional en Java.