

Zipkin es un sistema de seguimiento distribuido que mediante la recopilación y análisis de datos de servicios, ayuda a solucionar problemas de latencia en las arquitecturas de servicios.

Si contamos con un ID de seguimiento en un archivo de registro, podemos acceder directamente a él. De lo contrario, podemos consultar en función de atributos como el servicio, el nombre de la operación, las etiquetas y la duración. Se brindan algunos datos interesantes, como el porcentaje de tiempo dedicado a un servicio y si las operaciones fallaron o no.

Con el jar descargado debemos abrir la consola del sistema operativo y ejecutar el jar según la locación del mismo:

Esto nos levantará el servidor de zipkin en el puerto 9411

D ■

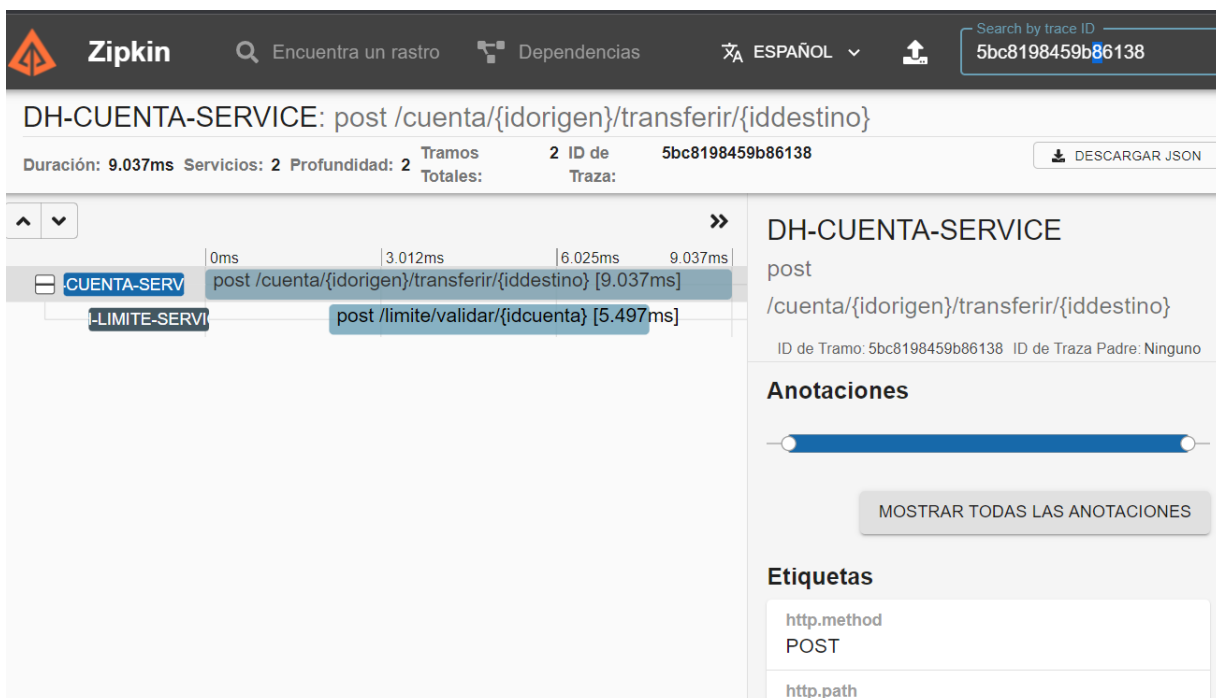


# Zipkin dashboard

Para acceder a Zipkin debemos acceder a su consola desde el puerto 9411



Podemos buscar la comunicación entre microservicios por el id de traza generado como vemos a continuación :





Las aplicaciones deben estar "instrumentadas" para informar los datos de seguimiento a Zipkin. Esto generalmente significa configurar un rastreador o una biblioteca de instrumentación. Las formas más populares de informar datos a Zipkin son a través de HTTP o Kafka, aunque existen muchas otras opciones, como Apache ActiveMQ, gRPC y RabbitMQ. Los datos proporcionados a la interfaz de usuario se almacenan en la memoria o de forma persistente con un back end compatible, como Apache Cassandra o Elasticsearch.

Si vamos a la opción de **dependencias** dentro del dashboard, podemos visualizar la dependencia generada entre los servicios que notifican información de trazabilidad a Zipkin, para esto tenemos que consultar la información dentro de un rango de fechas.

Si hago clic dentro de uno de los servicios involucrados puedo ver el historial de llamadas correctas y fallidas del servicio dado.

