


# Distributed tracing

## Overview

Distributed tracing is a **method of tracking application requests as they flow from frontend devices to backend services and databases**. It is primarily used for debugging the distributed systems. Distributed tracing allows seeing in more detail how exactly a particular user request is executed and what type of interactions between the backend services, frontend, and backend, etc. happen once it's been made. For example, in the screenshot below it is possible to see how distributed tracing helps differentiate between the REDIS, search-service, and ElasticSearch requests.



 You can watch the original video materials on distributed tracing [here](#).

## In this article:

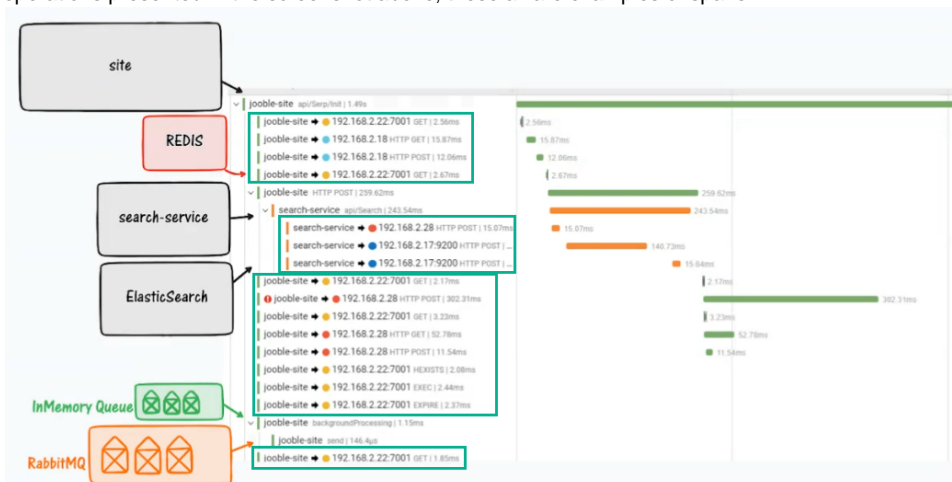
- [Overview](#)
- [Trace vs span](#)
- [Where to find the trace ID](#)
  - [Response headers](#)
  - [In the logs](#)
  - [Rabbit error queue](#)
  - [Using search functionality in Grafana](#)
- [Trace sampling](#)

## Read next:

- [How to add tracing to services, related decision-making](#)
- [How to search for traces using Grafana](#)

## Trace vs span

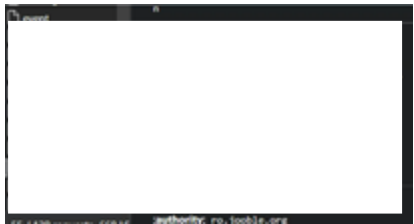
When talking about distributed tracing, it is important to differentiate between a **trace** and a **span**. A **trace** is the **tree of operations** related to one user request (as seen in the screenshot above). A span, on the other hand, is **one of the requests from this tree of operations**. For example, for the tree of operations presented in the screenshot above, these all are examples of spans.



## Where to find the trace ID

### Response headers

The first option is to check the **response headers** as shown in the example below:



In the logs

The latest version of Jooble's **logging library** stores the trace IDs among other information in the logs. For example:

ProcessID	1
ProcessName	Botnet
RequestID	00000000-00000000-00000000-00000000
RequestPath	/api/Error
Source	site-core
SourceContext	Jooble.Common.Log.Contexts.JsonError
SpanID	aab375271f1e125e
Tags	production
TraceID	5005411b3b0747a597080ea263e0f5

Rabbit error queue

It is also possible to find trace IDs in the **Rabbit error queue**, for example:



Using search functionality in Grafana

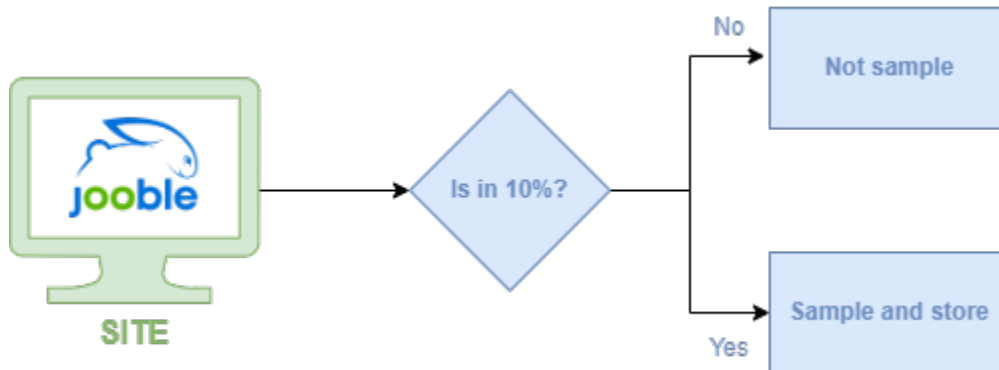
[Grafana](#) allows searching the stored **last 30 minutes of data**.

Trace ID	Trace name	Start time	Duration
<a href="#">65e7fd6a794e634e227233d5d72e940</a>	jooble-site api/Serp/Init	2022-12-12 15:19:35.059	3.08 s
<a href="#">f4de1b1045d20a6636e0e04a7c2ab863</a>	jooble-site Statistics/SaveImpressionsOnS...	2022-12-12 15:19:32.038	3.02 s
<a href="#">998c535f7hr17fe40q15d7d0q9qer6n</a>	<root span not yet received>	2022-12-12 15:19:27.652	3.36 s

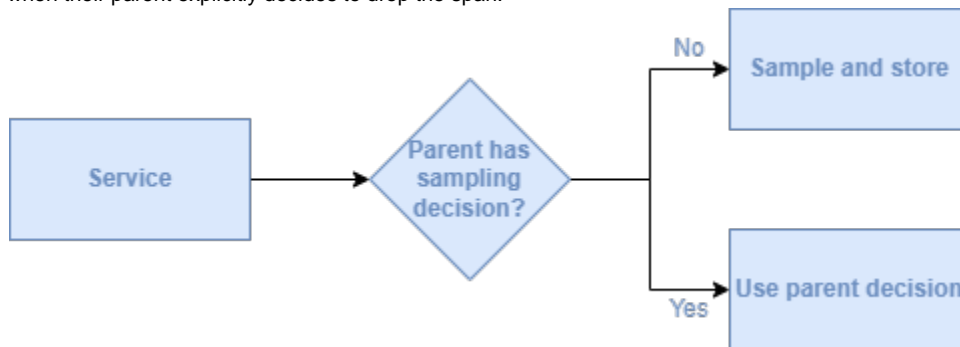
For details on how to use Grafana, refer to the [section below](#).

## Trace sampling

Due to storage space concerns, **all (100%) trace data** is stored for the **staging** environment and only **10% of trace data** is stored for the **production** environment. The percentage value for the production environment **can be changed** in [Cobra](#) site



By default logic, downstream services **depend on the sampling decision from their parent**. They do not sample and do not store the trace when their parent explicitly decides to drop the span.



You can **force the sampling** of your requests by going to [http://localhost:3000](#) and clicking the “Enable Force Tracing” button:

