



Web



Monitoring



WEB &gt; GUIDES &gt; PSPDFKIT SERVER &gt; MONITORING

# PSPDFKit Server metrics reference



PSPDFKit Server has been deprecated and replaced by [Document Engine](#). To migrate to Document Engine and unlock advanced document processing capabilities, refer to our [migration guide](#). Learn more about these enhancements on our [blog](#).

This is a reference page for all internal metrics exported by PSPDFKit Server.

Metrics follow the [DogStatsD protocol](#) format, a variant of the popular StatsD protocol. Server sends a metric update to the compatible monitoring agent either when some event happens (e.g. an HTTP response is sent), or when a measurement is collected periodically (e.g. the memory used is sampled). The agent aggregates metrics in fixed time windows and forwards them to the monitoring system, where they are persisted for further analysis. How the metric is aggregated depends on its type and the agent implementation (e.g. [Telegraf](#) might perform different aggregations than [CloudWatch agent](#) does). Refer to our [Integration](#) guide to learn how to export Server metrics in different environments and deployment settings.

## Metric types

Server exports three types of metrics:

- ⚙ Counters — Each metric update carries a value that increases a counter by that value. An example of this is when a file system cache hits metric, where a counter increment is sent every time an item is found in the cache.



ASK AI

- ⌘ Gauges — A metric update carries the most recent value of a particular measurement. This is a very common type for metrics gathered periodically, e.g. memory usage.
- ⌘ Timings — Each metric update carries the duration of a particular event. An example of this is an HTTP request handling duration. Usually agents aggregate timings by calculating statistics based on measurements falling into the time window, e.g. count, minimum, maximum, mean, percentiles, etc. Timings are often used when we need to both count the events and measure their duration.

# Tags

Apart from the metric name and value, each metric update includes a set of tags that allow you to group and filter measurements belonging to the same metric when analyzing them.

This is a list of common tags attached to every metric exported by Server:

TAG	DESCRIPTION
host	The hostname of the Server container
node	The unique ID of the Server node in the cluster
family	This is always set to <code>pspdfkit-server</code>

# Metrics reference

## HTTP performance

NAME	TYPE
http_server.req_end	timing

This is the duration it takes PSPDFKit Server to process the HTTP request and respond.

TAG	DESCRIPTION
status	HTTP response status
method	HTTP request method
group	Either <code>standard</code> for regular HTTP requests, or <code>long_poll</code> for long polling requests

When analyzing HTTP performance metrics, make sure to separate metrics based on the `group`. By definition, long polling requests take a long time to complete because the client keeps the connection open to allow Server to send a response only when it has data ready. In most situations, you're most likely interested in metrics with the `group` set to `standard`.

## PostgreSQL performance

NAME	TYPE
<code>pg_client.query</code>	timing
<code>pg_client.queue</code>	timing
<code>pg_client.decode</code>	timing
<code>pg_client.result_size</code>	gauge

These measurements concern the performance of SQL queries made by Server against PostgreSQL:

- ❖ `pg_client.query` tells you how long it took to actually execute the query.
- ❖ `pg_client.queue` tells you how long the query waited for the connection to be available from the pool.
- ❖ `pg_client.decode` tells you how long it took to decode the query results.
- ❖ `pg_client.result_size` is a measurement that tells you how many rows were returned per query.

To get insight into the total time it takes to complete the database query, you need to take the sum of the `query`, `queue`, and `decode` measurements.

TAG	DESCRIPTION
result	Either <code>success</code> or <code>error</code> , indicating if the query succeeded.
command	The SQL command that was executed. One of <code>select</code> , <code>update</code> , <code>delete</code> , <code>insert</code> ,
error_code	PostgreSQL error code, only set when <code>result</code> is <code>error</code> .
severity	Error severity, only set when <code>result</code> is <code>error</code>

## Asset storage

NAME	TYPE
assets.fetch_asset	timing
assets.store_asset	timing

These measurements track the time it takes to retrieve or store the asset in the asset storage. Note that the asset is only fetched from the storage if it's not found in the cache.

TAG	DESCRIPTION
result	Either <code>success</code> or <code>error</code> , indicating if the operation was successful.

## File system cache

NAME	TYPE
cache.fs_hit	counter
cache.fs_miss	counter
cache.fs_size	gauge
cache.fs_free	timing

These measurements are related to the Server file system cache used for document source files:

- ❖ The `cache.fs_hit` and `cache.fs_miss` measurements count cache hits and misses.
- ❖ The `cache.fs_size` measurement reports the current size of the file system cache. The cache size is limited by the `ASSET_STORAGE_CACHE_SIZE` configuration option.
- ❖ The `cache.fs_free` measurement tells you how long it took to clear a full cache.

## In-memory cache

NAME	TYPE
cache.memory_hit	counter
cache.memory_miss	counter

These measurements are related to the in-memory cache for PDF metadata. `cache.fs_hit` and `cache.fs_miss` measurements count cache hits and misses.

## Redis cache

NAME	TYPE
cache.redis_hit	timing
cache.redis_miss	timing
cache.redis_set	timing
cache.redis_error	timing

These measurements are related to the optional Redis cache used for caching rendering results between multiple PSPDFKit Server instances.

- ❖ `cache.redis_hit` indicates how long it took to fetch an item from Redis when there was a cache hit.
- ❖ `cache.redis_miss` indicates how long the request to Redis took when there was a cache miss.
- ❖ `cache.redis_set` indicates how long it took to store an item in Redis.

❖ `cache_redis_error` indicates how long a Redis operation that errored out took.

TAG	DESCRIPTION
op	The Redis operation that was performed. Only set for the <code>cache.redis_error</code> metric.

## Remote documents

NAME	TYPE
remote_doc.response_start	timing
remote_doc.response_end	timing

These measurements concern the time it takes the Server to fetch documents from remote URLs.

- ❖ `remote_doc.response_start` tells you the time between when the Server sent the request and when the first byte of data has been received.
- ❖ `remote_doc.response_end` tells you how long the actual data transfer took after the remote server started responding.

To get the total remote document response time, sum up both metrics.

TAG	DESCRIPTION
result	<code>success</code> , <code>error</code> , or <code>timeout</code> , indicating if fetching the remote document succeeded.

## Document conversion

NAME	TYPE
document_conversion.convert	timing

The duration of the Office documents conversion.

TAG	DESCRIPTION
result	Either <code>success</code> or <code>error</code> , indicating the conversion result.

## PDF processing

NAME	TYPE
Nutrientd.queue	timing
Nutrientd.exec	timing

These measurements concern all Server operations that involve working with PDFs, including rendering, extracting content, and preparing PDFs for being downloaded.

- ❖ `pspdfkit.queue` tells you how long an operation had to wait until there was a worker available to execute it.
- ❖ `pspdfkit.exec` tells you how long the operation actually took.

TAG	DESCRIPTION
request	The PDF operation performed

## Signing service

NAME	TYPE
signing_service.sign	timing

How long it took the signing service to respond to the signing request.

TAG	DESCRIPTION
result	Either <code>success</code> or <code>error</code> , indicating if the call to the signing service succeeded.

# Instant

NAME	TYPE
layer.sync.before_changes	timing
layer.sync.before_commit	timing
layer.sync.after_commit	timing
layer.sync.total	timing

These metrics track the duration of Instant sync phases. For operations that fail, Server emits only `layer.sync.total`.

TAG	DESCRIPTION
result	Only set for <code>layer.sync.total</code> . Either <code>success</code> or <code>error</code> , indicating if Instant sync succ

# Memory total

NAME	TYPE
vm_memory.total	gauge

The total amount of memory allocated by the Server process. Note that the amount of memory taken by the Server *container* is usually larger than this number, since there are also other processes running inside the container.

# Compute resources utilization

NAME	TYPE
vm_scheduler_wall_time.active	timing
vm_scheduler_wall_time.total	timing



❖ `vm_scheduler_wall_time.active` tells you how much time the Erlang VM spent being active in the last interval.

❖ `vm_scheduler_wall_time.total` tells you the total uptime of the Erlang VM.

If you divide the active time by the total time, the resulting number indicates the utilization of compute resources assigned to Server. In other words, it says which percentage of the time the Server was busy doing work.

Note that, as with memory, this only concerns the Server *process* — the CPU utilization of the container may be different, as there are other processes running inside it as well.

TAG	DESCRIPTION
scheduler_number	The internal Server scheduler number

Server starts as many schedulers as there are logical CPU cores available. In most cases, you'll want to take the average of metrics described here across schedulers.

Was this helpful?

☒ YES

☐ NO

Questions? [Contact us](#)

