About Server Statistics

In this article

You can use Server Statistics to analyze your own aggregate data from GitHub Enterprise Server, and help us improve GitHub products.

> **Who can use this feature**
> Enterprise owners can enable Server Statistics.

About the benefits of Server Statistics

Server Statistics can help you anticipate the needs of your organization, understand how your team works, and show the value you get from GitHub Enterprise Server.

Once enabled, Server Statistics collects aggregate data on how much certain features are used on your instance over time. Unlike other Admin Stats API endpoints, which only return data for the last day, Server Statistics provides historical data of all Server Statistics metrics collected since the day you enabled the feature. For more information, see "Enabling Server Statistics for your enterprise."

When you enable Server Statistics, you're helping to build a better GitHub. The aggregated data you'll provide gives us insights into how GitHub adds value to our customers. This information allows GitHub to make better and more informed product decisions, ultimately benefiting you.

About data security

We respect your data. We will never transmit data from your GitHub Enterprise Server instance unless you have first given us permission to do so.

We collect no personal data. We also don't collect any GitHub content, such as code, issues, comments, or pull request content.

Only owners of the connected enterprise account or organization on GitHub Enterprise Cloud can access the data.

Only certain aggregate metrics are collected on repositories, issues, pull requests, and other features. To see the list of aggregate metrics collected, see "Server Statistics data collected."

Any updates to the collected metrics will happen in future feature releases of GitHub Enterprise Server and will be described in the GitHub Enterprise Server release notes. In

addition, we will update this article with all metric updates.

For a better understanding of how we store and secure Server Statistics data, see "GitHub Security."

About data retention and deletion

GitHub collects Server Statistics data for as long as your GitHub Enterprise Server license is active and the Server Statistics feature is enabled.

If you would like to delete your data, you may do so by contacting GitHub Support, your GitHub account representative, or your Customer Success Manager. Generally, we delete data in the timeframe specified in our privacy statement. For more information, see GitHub's privacy statement in the GitHub.com documentation.

About data portability

As an organization owner or enterprise owner on GitHub Enterprise Cloud, you can access Server Statistics data by exporting the data in a CSV or JSON file or through the Server Statistics REST API. For more information, see "Requesting Server Statistics using the REST API" or "Exporting Server Statistics."

About disabling data collection

You can disable the Server Statistics feature at any time. For more information, see "Enabling Server Statistics for your enterprise."

Server Statistics data collected

After you enable Server Statistics, metrics are collected through a daily job that runs on your GitHub Enterprise Server instance. The aggregate metrics are stored on your organization or enterprise account on GitHub Enterprise Cloud and are not stored on your GitHub Enterprise Server instance.

The following aggregate metrics will be collected and transmitted on a daily basis and represent the total counts for the day.

CSV column	Name	Description
A	`github_connect.features_enabled`	Array of GitHub Connect features that are enabled for your instance (see "About GitHub Connect")
B	`host_name`	The hostname for your instance
C	`dormant_users.dormancy_threshold`	The length of time a user must be inactive to be considered dormant
D	`dormant_users.total_dormant_users`	Number of dormant user accounts
E	`ghes_version`	The version of GitHub Enterprise Server that your instance is running
F	`server_id`	The UUID generated for your instance

G	`collection_date`	The date the metrics were collected
H	`schema_version`	The version of the database schema used to store this data
I	`ghe_stats.comments.total_commit_comments`	Number of comments on commits
J	`ghe_stats.comments.total_gist_comments`	Number of comments on gists
K	`ghe_stats.comments.total_issue_comments`	Number of comments on issues
L	`ghe_stats.comments.total_pull_request_comments`	Number of comments on pull requests
M	`ghe_stats.gists.total_gists`	Number of gists (both secret and public)
N	`ghe_stats.gists.private_gists`	Number of secret gists
O	`ghe_stats.gists.public_gists`	Number of public gists
P	`ghe_stats.hooks.total_hooks`	Number of pre-receive hooks (both active and inactive)
Q	`ghe_stats.hooks.active_hooks`	Number of active pre-receive hooks
R	`ghe_stats.hooks.inactive_hooks`	Number of inactive pre-receive hooks
S	`ghe_stats.issues.total_issues`	Number of issues (both open and closed)
T	`ghe_stats.issues.open_issues`	Number of open issues
U	`ghe_stats.issues.closed_issues`	Number of closed issues
V	`ghe_stats.milestones.total_milestones`	Number of milestones (both open and closed)
W	`ghe_stats.milestones.open_milestones`	Number of open milestones
X	`ghe_stats.milestones.closed_milestones`	Number of closed milestones
Y	`ghe_stats.orgs.total_orgs`	Number of organizations (both enabled and disabled)
Z	`ghe_stats.orgs.disabled_orgs`	Number of disabled organizations
AA	`ghe_stats.orgs.total_teams`	Number of teams
AB	`ghe_stats.orgs.total_team_members`	Number of team members
AC	`ghe_stats.pages.total_pages`	Number of GitHub Pages sites

AD	`ghe_stats.pulls.total_pulls`	Number of pull requests
AE	`ghe_stats.pulls.merged_pulls`	Number of merged pull requests
AF	`ghe_stats.pulls.mergeable_pulls`	Number of pull requests that are currently mergeable
AG	`ghe_stats.pulls.unmergeable_pulls`	Number of pull requests that are currently unmergeable
AH	`ghe_stats.repos.total_repos`	Number of repositories (both upstream repositories and forks)
AI	`ghe_stats.repos.root_repos`	Number of upstream repositories
AJ	`ghe_stats.repos.fork_repos`	Number of forks
AK	`ghe_stats.repos.org_repos`	Number of repositories owned by organizations
AL	`ghe_stats.repos.total_pushes`	Number of pushes to repositories
AM	`ghe_stats.repos.total_wikis`	Number of wikis
AN	`ghe_stats.users.total_users`	Number of user accounts
AO	`ghe_stats.users.admin_users`	Number of user accounts that are site administrators
AP	`ghe_stats.users.suspended_users`	Number of user accounts that are suspended
AQ	`actions_stats.number_of_repos_using_actions`	Number of repositories using GitHub Actions
AR	`actions_stats.percentage_of_repos_using_actions`	Percentage of repositories using GitHub Actions
AS	`packages_stats.registry_enabled`	Whether GitHub Packages with repository-scoped packages is enabled for your GitHub Enterprise Server instance
AT	`packages_stats.registry_v2_enabled`	Whether GitHub Packages with granular permissions is enabled for your GitHub Enterprise Server instance
AU	`packages_stats.ecosystems.docker.registry_enabled`	Whether Docker is enabled for GitHub Packages
AV	`packages_stats.ecosystems.docker.published_packages_count`	Number of published Docker images (private, public, and internal)
AW	`packages_stats.ecosystems.docker.private_packages_count`	Number of private Docker images
AX	`packages_stats.ecosystems.dock`	Number of public Docker

AX	`packages_stats.ecosystems.docker.public_packages_count`	Number of public Docker images
AY	`packages_stats.ecosystems.docker.internal_packages_count`	Number of internal Docker images
AZ	`packages_stats.ecosystems.docker.user_packages_count`	Number of Docker images owned by users
BA	`packages_stats.ecosystems.docker.organization_packages_count`	Number of Docker images owned by organizations
BB	`packages_stats.ecosystems.docker.daily_download_count`	Number of downloads of Docker images
BC	`packages_stats.ecosystems.docker.daily_update_count`	Number of Docker images updated
BD	`packages_stats.ecosystems.docker.daily_delete_count`	Number of Docker images deleted
BE	`packages_stats.ecosystems.docker.daily_create_count`	Number of Docker images created
BF	`packages_stats.ecosystems.maven.registry_enabled`	Whether Maven is enabled for GitHub Packages
BG	`packages_stats.ecosystems.maven.published_packages_count`	Number of published Maven packages (private, public, and internal)
BH	`packages_stats.ecosystems.maven.private_packages_count`	Number of private Maven packages
BI	`packages_stats.ecosystems.maven.public_packages_count`	Number of public Maven packages
BJ	`packages_stats.ecosystems.maven.internal_packages_count`	Number of internal Maven packages
BK	`packages_stats.ecosystems.maven.user_packages_count`	Number of Maven packages owned by user accounts
BL	`packages_stats.ecosystems.maven.organization_packages_count`	Number of Maven packages owned by organizations
BM	`packages_stats.ecosystems.maven.daily_download_count`	Number of downloads of Maven packages
BN	`packages_stats.ecosystems.maven.daily_update_count`	Number of Maven packages updated
BO	`packages_stats.ecosystems.maven.daily_delete_count`	Number of Maven packages deleted
BP	`packages_stats.ecosystems.maven.daily_create_count`	Number of Maven packages created
BQ	`packages_stats.ecosystems.npm.registry_enabled`	Whether npm is enabled for GitHub Packages
BR	`packages_stats.ecosystems.npm.published_packages_count`	Number of published npm packages (private, public, and internal)

BS	`packages_stats.ecosystems.npm.private_packages_count`	Number of private npm packages
BT	`packages_stats.ecosystems.npm.public_packages_count`	Number of public npm packages
BU	`packages_stats.ecosystems.npm.internal_packages_count`	Number of internal npm packages
BV	`packages_stats.ecosystems.npm.user_packages_count`	Number of npm packages owned by user accounts
BW	`packages_stats.ecosystems.npm.organization_packages_count`	Number of npm packages owned by organizations
BX	`packages_stats.ecosystems.npm.daily_download_count`	Number of downloads of npm packages
BY	`packages_stats.ecosystems.npm.daily_update_count`	Number of npm packages updated
BZ	`packages_stats.ecosystems.npm.daily_delete_count`	Number of npm packages deleted
CA	`packages_stats.ecosystems.npm.daily_create_count`	Number of npm packages created
CB	`packages_stats.ecosystems.nuget.registry_enabled`	Whether NuGet is enabled for GitHub Packages
CC	`packages_stats.ecosystems.nuget.published_packages_count`	Number of published NuGet packages (private, public, and internal)
CD	`packages_stats.ecosystems.nuget.private_packages_count`	Number of private NuGet packages
CE	`packages_stats.ecosystems.nuget.public_packages_count`	Number of public NuGet packages
CF	`packages_stats.ecosystems.nuget.internal_packages_count`	Number of internal NuGet packages
CG	`packages_stats.ecosystems.nuget.user_packages_count`	Number of NuGet packages owned by user accounts
CH	`packages_stats.ecosystems.nuget.organization_packages_count`	Number of NuGet packages owned by organizations
CI	`packages_stats.ecosystems.nuget.daily_download_count`	Number of downloads of Nuget packages
CJ	`packages_stats.ecosystems.nuget.daily_update_count`	Number of NuGet packages updated
CK	`packages_stats.ecosystems.nuget.daily_delete_count`	Number of NuGet packages deleted
CL	`packages_stats.ecosystems.nuget.daily_create_count`	Number of NuGet packages created
CM	`packages_stats.ecosystems.ruby_gems.registry_enabled`	Whether Rubygems is enabled for GitHub Packages

CN	`packages_stats.ecosystems.ruby_gems.published_packages_count`	Number of published Rubygems packages (private, public, and internal)
CO	`packages_stats.ecosystems.ruby_gems.private_packages_count`	Number of private Rubygems packages
CP	`packages_stats.ecosystems.ruby_gems.public_packages_count`	Number of public Rubygems packages
CQ	`packages_stats.ecosystems.ruby_gems.internal_packages_count`	Number of internal Rubygems packaes
CR	`packages_stats.ecosystems.ruby_gems.user_packages_count`	Number of Rubygems packages owned by user accounts
CS	`packages_stats.ecosystems.ruby_gems.organization_packages_count`	Number of Rubygems packages owned by organizations
CT	`packages_stats.ecosystems.ruby_gems.daily_download_count`	Number of downloads of Rubygems packages
CU	`packages_stats.ecosystems.ruby_gems.daily_update_count`	Number of Rubygems packages updated
CV	`packages_stats.ecosystems.ruby_gems.daily_delete_count`	Number of Rubygems packages deleted
CW	`packages_stats.ecosystems.ruby_gems.daily_create_count`	Number of Rubygems packages created
CX	`packages_stats.ecosystems.containers.registry_enabled`	Whether Container registry is enabled for GitHub Packages
CY	`packages_stats.ecosystems.containers.published_packages_count`	Number of published container images (private, public, and internal)
CZ	`packages_stats.ecosystems.containers.private_packages_count`	Number of private container images
DA	`packages_stats.ecosystems.containers.public_packages_count`	Number of public container images
DB	`packages_stats.ecosystems.containers.internal_packages_count`	Number of internal container images
DC	`packages_stats.ecosystems.containers.user_packages_count`	Number of container images owned by user accounts
DD	`packages_stats.ecosystems.containers.organization_packages_count`	Number of container images owned by organizations
DE	`packages_stats.ecosystems.containers.daily_download_count`	Number of downloads of container images
DF	`packages_stats.ecosystems.containers.daily_update_count`	Number of container images updated
DG	`packages_stats.ecosystems.containers.daily_delete_count`	Number of container images deleted

DH	`packages_stats.ecosystems.containers.daily_create_count`	Number of container images created

Server Statistics data examples

To see an example of the headings included in the CSV export for Server Statistics, download the Server Statistics CSV example.

To see an example of the response payload for the Server Statistics API, see "Requesting Server Statistics using the REST API."