

Plugin tool_monitoring

Beliebige [Moodle](#) Instanzen mit einem Monitoring Setup (wie [Prometheus](#) & [Grafana](#)) verbinden.

Grafana Dashboard

Das Plugin findet man auf [Github](#) und im [Moodle Marketplace](#).

Features

- **☐ Sofort einsatzbereit:** Von Haus aus direkt mit [sinnvollen Moodle Metriken](#) und einem [Prometheus Endpoint](#).
- **☐ Erweiterbar:** Um eine eigene Metrik zu definieren, einfach von der `metric` Klasse erben und [registrieren](#).
- **☐ Generisch:** Agnostisch gegenüber der verwendeten Monitoring Software; unterstützt benutzerdefinierte [Exporter](#).
- **☐☐☐ Admin-freundlich:** Bequemes [Admin Dashboard](#) um registrierte Metriken und Exporter zu verwalten und zu konfigurieren.
- **☐ Flexibel:** Für höhere Ansprüche können Metriken [konfiguriert](#) und sogar mittels der [Moodle Tag API](#) getaggt werden.

Admin Dashboard

Das Admin Dashboard findet man unter `/admin/tool/monitoring` oder indem man über *Website-Administration > Plugins > Dienstprogramme > Monitoring > Übersicht* navigiert.

Metrics Overview

Siehe [Dokumentation](#) für die vorinstallierten Metriken und die Labels nach denen diese aufgeteilt sind.

Metric Config users_online

Der Prometheus Endpoint selbst ist direkt erreichbar und kann über die Route `/monitoringexporter_prometheus/metrics` abgerufen werden.

“ **Warnung: Der Endpoint ist standardmäßig ungesichert.** Solange kein `prometheus_token` gesetzt ist, sieht jeder, der die URL aufrufen kann, *alle*

aktivierten Metriken ohne Zugangsbeschränkung. Obwohl standardmäßig alle neu registrierten Metriken *deaktiviert* sind, sollte zur Sicherheit ein Token gesetzt werden.

Metrics mit Tags gruppieren (optional)

`tool_monitoring` erlaubt es, Metriken mittels der [Moodle's Tag API](#) ganz leicht zu gruppieren. Man kann jeder Metrik beliebige Tags zuweisen (die ihren eigenen Schlagwortbereich haben). Exporter können dann über einen Parameter die auszuliefernden Metriken nach ihren Tags filtern. Zum Beispiel akzeptiert der mitgelieferte Prometheus Exporter einen `tag` Query Parameter in der URL. Wird ein einzelner Tag Name übergeben, filtert dieser all Metriken raus, die nicht mit diesem Tag versehen sind. Wird eine kommagetrennte **Liste von Tags** übergeben, werden nur die Metriken exportiert, die mit **allen** diesen Tags versehen sind. (Zu beachten ist die **UND**-Verknüpfung, die hier angewendet wird.) Für Details zur Implementierung, siehe die Prometheus [exporter](#) Klasse.

Um eine Metrik mit Tags zu versehen, muss man auf ihre [Konfigurationsseite](#) gehen und dann einfach die Tag Namen in das "Tags" Feld eingeben.

Metric Config Tags users_online

Metrics Overview Tags

Ein Klick auf einen Tag filtert die Liste der Metriken entsprechend.

Metrics Overview Tags filtered

Prometheus configuration

```
scrape_configs:
  # Einen beliebigen einzigartigen Job Namen wählen.
  - job_name: moodle
    # Standard ist HTTP.
    scheme: https
    # Wenn ein Zugangstoken gesetzt wurde, diesen hier angeben.
    authorization:
      credentials: 'super-secure-secret'
    # Den vollständigen Endpoint Pfad angeben. Standard ist sonst einfach '/metrics'.
    # Wenn das Moodle Routing nicht vollständig konfiguriert wurde, muss noch '/r.php'
    vorangestellt werden.
    metrics_path: /monitoringexporter_prometheus/metrics
    # Ziel Host angeben.
    static_configs:
      - targets: ['example.com']
```

Ein Grafana Dashboard aufsetzen

Dies setzt voraus, dass eine Prometheus Konfiguration wie oben bereits besteht. Ein beispielhaftes Grafana Dashboard kann von [docs/examples/grafana_dashboard.json](https://grafana.com/docs/examples/grafana_dashboard.json) in Grafana importiert werden:

Dazu nach "Dashboards" > "Neu" navigieren und "importieren" auswählen und die json-Datei auswählen oder deren Inhalt dort hineinkopieren. Dann die Prometheus Datenquellen auswählen und auf "importieren" klicken.

Dokumentation

Die Installationsanweisungen, Anleitungen zum Hinzufügen eigener Metriken und Exporter sub-Plugins und die vollständige Dokumentation findet man in der [README](#).

Copyright

© 2025 Daniel Fainberg, Martin Gauk, Sebastian Rupp, Malte Schmitz, Melanie Treitinger

Code, Tests und Dokumentation von und für Menschen geschrieben. ☐☐☐

Autor: <[Daniel Fainberg](#), TU Berlin>

Version #4

Erstellt: 2026-08-12 10:11:50 UTC von Daniel Fainberg

Zuletzt aktualisiert: 2026-08-18 14:24:43 UTC von Daniel Fainberg