moodleload documentation

moodleload is an online load testing service for Moodle. It lets you measure the real capacity of your Moodle installation **without installing any load testing software**: we provide the engines, the injectors and the test resources, you describe what your users do and when, and you get the results back to tune your technical environment.

> **In a nutshell**
>
> You register your Moodle site, install a small companion plugin on it, pick or write scenarios (log in, open a course, attempt a quiz…), then schedule a job. moodleload fires virtual users at your site from its own injectors and collects response times and errors for you.

How it works

A load test always involves three parties:

Party	Role
Your Moodle site	The target. The `tool_moodleload` plugin, installed on it, gives the service a controlled access (API keys, test users).
The moodleload engine	Plans and drives the jobs: it starts the injectors, follows their progress and stores the results.
The injectors	Machines that play the scenarios as many simultaneous virtual users, each with its own Moodle session.

Each site is tested within the limits of your plan (number of sites, virtual users, duration, available features).

Getting started

1. **Install the load test plugin** (`tool_moodleload`) on your production infrastructure.
2. **Configure the plugin** and define the API keys.
3. **Check the load link** between the test service and your site: in *My sites*, its status goes from *Not verified* to *Authorized*.
4. **Create a test**, or a mix if your plan includes it, from the standard scenarios or from your own scenarios (depending on your plan).
5. Optionally **add your tests and mixes to a campaign**.
6. **Schedule a job** to start load testing your server.
7. **Get your results**.

Key concepts

- **Site**: a Moodle installation you own, identified by its wwwroot, linked to the service by the `tool_moodleload` plugin. All the data attached to a site belong to its owner only.
- **Site variables**: values used by the scenarios on a given site (course ids, module ids…), so that a scenario can be reused from one site to another.
- **Scenario**: the sequence of steps a virtual user plays. Standard scenarios are provided for everyone; you can write your own.
- **Step**: one action of a virtual user (HTTP request, file upload, web service call), with its checks and captures.
- **Envelope**: the shape of the load in time (ADSHR curve or staircase).
- **Test**: a scenario played against one site with an audience size and an envelope. A **mix** plays several scenarios at the same time; a **campaign** plays tests and mixes one after the other.
- **Job**: one scheduled execution (a *shot*) of a test, a mix or a campaign, fired by the injectors. Its status goes from *Pending* to *Running* then *Done*, and its results are available once *Compiled*.

Data confidentiality

Tokens, passwords and scenario engineering are confidential. Every user, administrators included, only sees and edits the data of **their own** sites. Support teams never browse your data directly: when assistance is needed, it is done at your request, in your own account.

Where to go next

- Sites: registering a site, the load link, site variables, site preparation.
- Scenarios: scopes, phases, resources, testing a scenario.
- Steps: step types, checks, variables and captures.
- Envelopes: ADSHR and staircase load shapes.
- Tests, mixes and campaigns: audience, envelope, rating.
- Jobs: scheduling, safety checks, lifecycle, results.

The service itself is available at moodleload.moogwai.org.

From:
https://docsen.moodleload.moogwai.org/ -

Moodleload Load Test Service Online

Permanent link:
https://docsen.moodleload.moogwai.org/doku.php?id=start

Last update: 2026/11/09 13:44