

Activating Web Services (ZSK Termine API)

This guide outlines the steps required to successfully set up the ZSK Termine API, including common pitfalls (e.g., incorrect admin URL or shell command).

Prerequisites

- ZSK Termine plugin version 1.3.0 (` 2025061100`) or newer installed and updated via Site administration → Notifications.
- Active Pro license (REST and Webhooks are Pro features).
- Access as a primary administrator (site manager).

Step 1: Enable web services globally

1. Open Site administration.
2. Go to *Advanced features*.

Direct URL: ` /admin/settings.php?section=optionalsubsystems`

3. Set "Enable web services" (` enablewebservices`) to "Yes".
4. Save changes.

Without this setting, many web service menus will not be accessible.

Step 2: Enable the REST protocol

1. Search for "REST" or "Protocols" using the admin search bar at the top, or navigate directly to:
` /admin/webservice/protocols.php`
2. Click the gear icon next to the REST server.
3. Enable the REST server → Save.

Step 3: Plugin upgrade (register service)

1. Site administration → Notifications

2. Run the upgrade for ZSK Termine until no pending notifications remain.

The external service "ZSK Termine API" (plugin: `local\_zsk\_termine`) will then appear.

Step 4: Verify the external service

1. Admin search: "Integrated services" / "External services", or:

`/admin/settings.php?section=externalservices`

2. "ZSK Termine API" must appear in the list (shortname: `zsk\_termine`). 3. Check via the "Functions" link that the following are present:

- `local\_zsk\_termine\_get\_events` – Event list (JSON)

- `local\_zsk\_termine\_get\_ical` – iCal content (JSON)

The service is enabled by default in the plugin; the "All users" setting is fine for the initial setup.

Step 5: Create token

Important: Correct URL

Token management is not located at `settings.php?section=webservicetokens`.

This URL often results in an "area error."

Correct URL:

`/admin/webservice/tokens.php`

Alternatively: Search the admin area for "Manage tokens" or "Create token."

Create token

1. Add token / Create token

2. Fields:

- Name: e.g., `zsk\_termine\_api` (optional, for reference)
- User: API user or administrator (using a dedicated technical user is recommended for production)
- Service: ZSK Termine API
- IP restriction: Leave blank (for testing) or restrict later
- Valid until: Leave blank = no expiration date

3. Save changes

4. Copy the token immediately – from Moodle 4.3 onwards, it is no longer displayed after leaving the page.

The token is a long alphanumeric key (e.g., `fd73bf06b4865d7903d7e9d0ab9495d5`).

The numbers in the token table (IDs) are not the API key.

Permissions

The API functions require the capability `local/zsk\_termine:view` .

Site administrators usually have this automatically. For a separate API user, add this permission to the "Authenticated user" role. Step 6: Test the REST call

Browser (simplest test)

Enter the URL into the address bar (no spaces, replace `WSTOKEN`):

```text

http://YOUR-MOODLE-  
URL/webservice/rest/server.php?wstoken=WSTOKEN&wsfunction=local\_zsk\_termine\_get\_events&moodlewsrestformat=json&limit=5

```

Success: JSON containing ` "events":[...]` .

Shell using curl (server / SSH)

Do not enter the URL directly into the shell—Bash interprets ``http://...`` as a file path, and ``&`` starts background jobs.

```
` `` `bash

curl -G "http://YOUR-MOODLE-URL/webservice/rest/server.php" \
--data-urlencode "wstoken=WSTOKEN" \
--data-urlencode "wsfunction=local_zsk_termine_get_events" \
--data-urlencode "moodlewsrestformat=json" \
--data-urlencode "limit=5"
` `` `
```

Example of a successful response:

```
` `` `json

{
  "events": [
    {
      "id": 2,
      "title": "Virtual Coffee",
      "categoryid": 3,
      "categoryname": "Social Life",
      "location": "Cafeteria",
      "timestart": 1781859420,
      "timeend": 0,
      "cancelled": false,
      "courseid": 0,
      "url": "http://YOUR-MOODLE-URL/local/zsk_termine/view.php?id=2",
      "datetime": "19.06.2026 09:57"
    }
  ]
}
` `` `
```

After the first successful call, the "Manage tokens" section will display a timestamp (instead of "Never") under "Last access". iCal via REST (optional)

```
` `` ` bash

curl -G "http://YOUR-MOODLE-URL/webservice/rest/server.php" \
--data-urlencode "wstoken=WSTOKEN" \
--data-urlencode "wsfunction=local_zsk_termine_get_ical" \
--data-urlencode "moodlewsrestformat=json"
` `` `
```

Step 7: Pro Settings (Webhook / Reminder)

Under Site administration → ZSK plugins → ZSK Appointments – Settings (after upgrade):

| Setting | Meaning |

| -|-|

| Reminder (days before appointment starts) | Automatic reminder email (Cron, Pro) |

| Webhook URL (outbound) | JSON POST upon appointment creation/modification/cancellation |

| Webhook secret (optional) | `X-ZSK-Signature` header (HMAC-SHA256) |

These fields are independent of the REST token. Webhook = Moodle sends data to an external URL; REST token = external system queries Moodle.

iCal subscription without REST token

With an active Pro license, the iCal subscription URL appears in the same settings:

```
` /local/zsk_termine/ical.php?token=... `
```

This URL can be added as a calendar subscription in Outlook or Google.

Troubleshooting

| Problem | Cause / Solution |

|||

| Range error at `...?section=webservice/tokens` | Incorrect URL → use
`/admin/webservice/tokens.php` |

| `No such file or directory` in Bash | URL entered without `curl` |

| `[1] 1240...` background jobs in Bash | `&` in the URL – use `curl -G` with quotes |

| `invalidtoken` | Incorrect token or token deleted → create a new one |

| `pro\_feature\_required` | Check Pro license |

| `Access control exception` | Grant `local/zsk\_termine:view` to the API user |

| ZSK Appointments API service missing | Plugin upgrade (Step 3) |

| Token ID used instead of key | Use only the long key from the creation screen | Admin search (if menu paths are unclear)

On the Site administration page, search for:

- `Enable web services`
- `REST`
- `External services`
- `Manage tokens`

The menu structure varies depending on the Moodle version (e.g., under Server or Plugins).
Direct URLs and the search function are more reliable than fixed click paths.

Security

- Treat REST tokens like passwords; do not share them in chats, repositories, or screenshots.
- For production: use a dedicated API user, optionally apply IP restrictions to the token, and delete the test token after setup to generate a new one.
- Only set a webhook secret if the receiver verifies the signature.

Quick checklist

- [] Enable web services (`enablewebservice`)
- [] Enable REST server
- [] ZSK Termine plugin upgrade
- [] ZSK Termine API service visible with both functions
- [] Token generated and saved at ` /admin/webservice/tokens.php`
- [] ` curl` or browser test returns JSON containing ` events`
- [] Optional: Webhook URL / iCal subscription configured