Mozout

Generated on 24/11/2026 10:04

KNOWLEDGE BASE · PUBLIC DOCUMENT

Your extras through the API: what each service has, certificate status, renewals and buying more

What You Can Automate

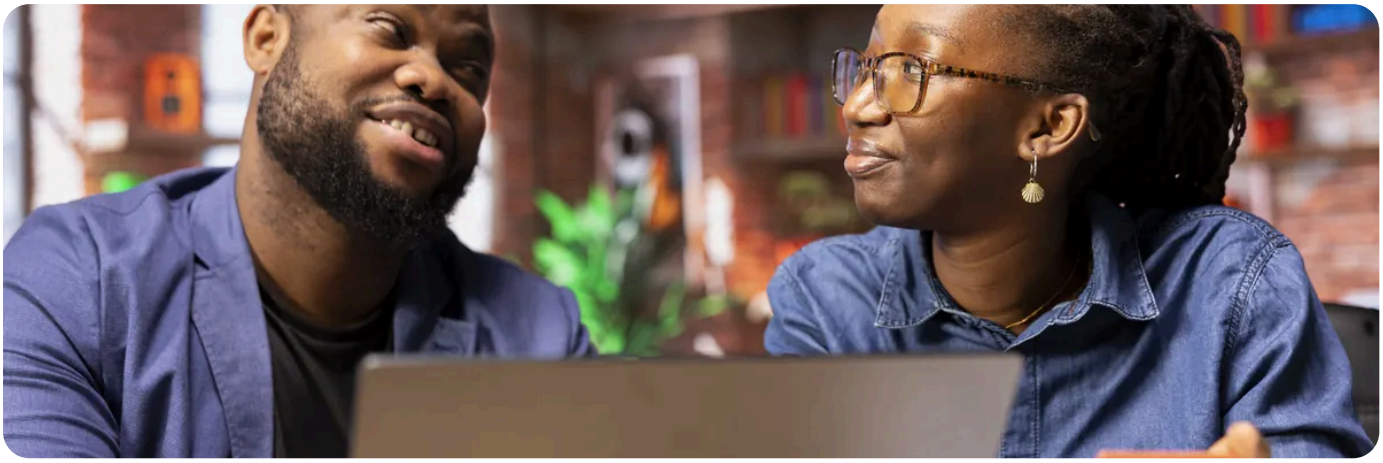
CODE	VERSION	COUNTRY · LANGUAGE
RLX-INT-KB-8454	v2026.11.24	International · English

MozOut International
RLX Holdings (Pty) Ltd. · Group RLX Holdings
Tax no. (SARS) 943 907 6234 · Registration no. (CIPC) 2025/667953/07

https://www.my.mozout.com/knowledgebase/8454

Your extras through the API: what each service has, certificate status, renewals and buying more

What You Can Automate Created on **November 24, 2026** 6 min read



The short answer: `GET /v1/extras` returns the extras the account already has: SSL certificates, SiteLock and other products, radio with its add-ons, and servers' extra IPs and disks. Each one comes with its status, the service it belongs to, the billing cycle, the amount, the next due date and the link to manage it in My MozOut. With that, your program knows which extras each service has, warns you before due dates and, for the products that allow it, buys more with your credit.

This article is about the extras you already have and buying safely. To show prices on your website, see **the catalogue and extras through the API**.

The id tells you what kind of extra it is

The id starts with	What it is	What else it carries
`s`	A service of its own, such as a certificate or SiteLock.	`parent_service_id` is `null`.
`a`	An add-on to a service, for example a radio add-on.	`parent_service_id`: the service it belongs to.
`v`	A server's extra IP or disk.	`quantity`, `invoice_id` and, in `parent_service_id`, the server.

Use the id exactly as it comes, letter included: `GET /v1/extras/s812` shows a single extra. Certificates, whether a service or an add-on, also carry `certificate`.

The fields you will use

Field	What it is for
`kind`	The type, such as `ssl`, `sitelock`, `radio`, `radio_addon`, `vps_ip` or `vps_disk`. It is also the list filter: `?kind=ssl`.
`status`	The status of the extra, for example `active`.
`site`	The site the extra relates to, when there is one. It can be `null`.
`billing_cycle`, `amount`	The billing cycle and the amount, in the account currency (`currency`).
`next_due_date`	The next due date.
`manage_url`	The link to manage the extra in My MozOut. Access details, such as the radio ones, never come through the API: you see them there.
`certificate`	Certificates only: `status`, for example `awaiting_configuration`, and `expires_at`, the date it expires.

The extras of a service you already have

1. **Find the service id** in `GET /v1/services`, with `hosting:read`. It works for servers too.
2. **Read every page** of `GET /v1/extras`, with `extras:read`. The list has no filter by service: you do that filtering on your side.
3. **Keep the ones whose `parent_service_id` matches the service id.** These are the add-ons of that service and, on a server, the extra IP and disk.
4. **Add the services of their own for the same site.** SiteLock carries the domain in `site`. A certificate bought through the API does not take a site and may show `site` as `null`: recognise it by its id and name.

```
curl "https://api.mozout.com/v1/extras?kind=sitelock&per_page=100" -H "Authorization: Bearer $MOZOUT_API_KEY"
```

Your hosting's own SSL is not an extra.
The certificate the hosting requests by itself (AutoSSL) does not appear in this list: you see it in `GET /v1/services/{id}/ssl`. See **hosting through the API**.

Warnings before due dates and for certificates still to configure

With `next_due_date` and `amount`, your program can warn you before each due date, with the amount. The invoices, once issued, can be seen and paid with credit through the API itself: see **invoices through the API**.

For certificates, look at `certificate`. With `status` set to `awaiting_configuration`, the certificate has been bought but not yet configured: open `manage_url` and finish it in My MozOut. `expires_at` tells you when it expires, so you can warn in good time.

Buying another extra, with no surprises

1. **See what is for sale** in `GET /v1/extras/catalog`, which also takes `?kind=`. Only products with `purchasable` set to `true` can be bought through the API, and `requires_site` says whether you must give the site. Each period carries `billing_cycle`, `months`, `price` and `setup_fee`.

2. **Check your credit and the limit.** `GET /v1/billing/credit`, with `billing:read`, returns the credit available, and `GET /v1/limits` returns the spending limit of the key and what it has already spent.

3. **Rehearse with a Sandbox key.** The purchase is simulated and nothing is charged. The answer carries the `amount`, `credit_sufficient` (whether your credit covers it) and `within_spending_limit` (whether it fits the spending limit of the key).

4. **Buy with a Production key** that has `billing:purchase` and `extras:write`, with the `Idempotency-Key` header. SiteLock takes `site`, the domain of the site. A certificate does not.

5. **Follow the delivery.** The answer is a `202` and the action ends with the payment. Follow the extra in `GET /v1/extras/{id}`, with the `extra_id` from the answer. For a certificate, finish the configuration in My MozOut, at `manage_url`.

```
curl https://api.mozout.com/v1/extras/purchase -X POST \
-H "Authorization: Bearer $MOZOUT_API_KEY" \
-H "Content-Type: application/json" \
-H "Idempotency-Key: 3c9e1f20-5b7a-4d2e-9f61-0a8b7c6d5e4f" \
-d '{"product_id": 96, "billing_cycle": "annually", "site": "yourcompany.com"}'
```

The 96 is only the example from the documentation: use the `id` the catalogue returns. Generate a new `Idempotency-Key` value for each purchase and, if you have to repeat the same request, repeat the same value too, so the purchase is never made twice. See **the Idempotency-Key**.

Always check the `simulated` field.
With a Sandbox key, or while the API is in preparation, purchases are simulated and nothing is charged, even with a `prod_` key. Only an answer with `simulated` set to `false` means the purchase was made.

When the purchase does not go through

Answer	What it means	What to do
`402 insufficient_credit`	Your credit does not cover it. Nothing was charged.	Top up your credit and try again.
`403 spending_limit_exceeded`	The purchase goes over the spending limit of the key. Nothing was charged.	Use a key with a higher limit.
`403 insufficient_scope`	The key lacks `billing:purchase` or `extras:write`.	Create a key with both.
`409 conflict`	The purchase cannot be made in the current state. The reason is in `details.reason`.	Read the reason. If the price changed, go back to the catalogue; if the account profile is incomplete, complete it in My MozOut.
`409 idempotency_conflict`	The `Idempotency-Key` was used on another request, or the first one is still running.	Wait and try again, or use a new value if it is a different purchase.
`422 validation_failed`	Some data is not valid, such as a period that is not in the catalogue. With `portal_only`, the product can only be bought in My MozOut.	Fix it with the help of `details`, or use `portal_url`.
`429 rate_limited`	Domain and extras purchases and invoice payments share a limit of 5 per minute, per account.	Wait for the time in `Retry-After`.

For the other codes, see **our API's error codes**.

Two keys, two jobs.
One with only `extras:read` and `hosting:read`, for the inventory and the warnings, unable to spend anything. Another for buying, with `billing:purchase`, `extras:write` and a spending limit to match. See **sensitive permissions and the spending limit**.

What stays in My MozOut: a server's extra IP and disk appear in the list, but they are bought on the server's page. Radio and the other catalogue products need choices that only exist there. The **Extra products** page of the documentation has the full examples.

RECOMMENDED

Want to follow the extras of each service from your own system? Start with a read-only key.

Create a key →

SEE ALSO

- **Catalogue and extras through the API: showing prices on your site and buying SSL or SiteLock**
- **Sensitive API permissions and the spending limit: paying, buying, reinstalling, cPanel access**
- **Idempotency-Key in our API: repeating a purchase, payment or ticket without doing it twice**
- **Long-running API operations: the 202 answer, the action_id and following the result**
- **Invoices through the API: listing, PDF, paying with credit, renewal cycle and cancellations**
- **Our API's error codes: what each one means, and which ones are worth retrying**

Mozout

MozOut International

RLX Holdings (Pty) Ltd. · Group RLX Holdings

TAX NO. (SARS)	943 907 6234
REGISTRATION NO. (CIPC)	2025/667953/07
REGISTERED OFFICE	1111 Olive Street Saint Louis 63101, Missouri, United States
CONTACT	info@mozout.com · +27 78 823 3325 · www.mozout.com
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

Online version of this document

RLX-INT-KB-8454 · v2026.11.24 · Generated on 24/11/2026 10:04