

Mailers: master data for your SMTP accounts

A mail needs an account to go out through. **Mailers** are that account as master data: one record per SMTP mailbox or relay, with its host, its credentials, the address it sends from, and the limits your provider actually enforces. Every mail VeloxFactory sends goes through one of these records.

The `MAIL_*` keys in the environment file are Laravel's own fallback mailer and are deliberately **not** used by report mailing. Mail configuration belongs in the application, where it can be changed, tested and audited without a deployment.

Screenshot placeholder: `mailer.index.png`

Mailers overview under Configuration with transport badges, the approval status and the Active column.

Where to find it

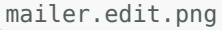
Mailers live in the main menu under **Configuration → Mailers**. The overview is a regular VeloxFactory lookup: search over name, host, sender address and description, filters for transport, active state, tested state, creator, updater and creation date, sortable columns, and the filter state kept in your session.

New opens the create form, the pencil on a row opens an existing record. Deleting is done from the edit page.

The fields of a mailer

Field	Required	Description
-------	----------	-------------

Name	yes	Unique, three to 55 characters, for example <code>Office SMTP</code> . This is what a render request may use instead of the ID.
Transport	yes	<code>smtp</code> for a real server, <code>sendmail</code> for the local binary, <code>log</code> to write the mail into the log instead of sending it. <code>log</code> is what makes a demo or a staging system harmless.
Host, Port	for smtp	The server and its port, for example <code>smtp.office365.com</code> and <code>587</code> .
Scheme	no	<code>smtp</code> for STARTTLS on port 587, <code>smtps</code> for implicit TLS on port 465. Left empty the transport decides by port, which is right for most servers.
Username, Password	no	Stored encrypted. The password is never returned by the API, and the edit form shows it empty: leave it empty to keep the stored one.
From Address	yes	The sender of every mail sent through this mailer. Many providers require it to match the authenticated mailbox.
From Name	no	The display name next to the address, for example <code>Shipping Department</code> .
Reply To	no	Where replies should go, when that is not the sender. Useful for a no-reply sender with a real support mailbox behind it.
Timeout	no	Seconds before the transport gives up. Relevant for synchronous sending, where the render request waits.
Local Domain	no	The EHLO name. Left empty the host part of the application URL is used. Some strict relays insist on a specific value.
Limit per minute / hour / day	no	The dispatch rate limits, see below. The hourly limit defaults to 500.
Description	no	Free note, for example which department the mailbox belongs to.
Active	-	On by default. An inactive mailer stays in the master data and keeps its history, but it can no longer be selected and no longer resolves in a render request.

Screenshot placeholder: 

Mailer edit form with connection fields, the three rate limit fields and the remaining-slots badges.

Testing a mailer

A new or changed mailer starts as **Unapproved**. The edit page has a **Send Test Mail** button that sends a short message through exactly the configuration you just saved, to your own address by default. On success the mailer flips to **Approved**, on failure the transport's own error message is shown, which is usually enough to see whether it was the port, the credentials or the TLS mode.

This mirrors how report connections are verified, and it works the same way: the status is information, not a gate. A mailer that has never been tested can still be used, it is simply not marked as proven.

i Changing a connection field resets the approval. Transport, host, port, scheme, credentials and sender address all invalidate the tested state, so a green badge always refers to the configuration that is actually stored.

A test mail consumes a rate limit slot like any other mail, so the counter stays honest. If the limit is exhausted, the test is refused with the waiting time instead of being sent.

Rate limits

Providers cap outbound mail, and they all cap differently. Exceeding the cap does not produce a polite error, it produces a blocked account, so the limits belong on the mailer, next to the credentials they protect.

Each mailer has three independent windows, each of them optional:

Window	Typical for
per minute	Relays that cap bursts, for example Office 365 at roughly 30 messages per minute.

Window	Typical for
per hour	Shared hosting mailboxes, typically 200 to 500 per hour. This is the one that defaults to 500.
per day	Mailbox providers, for example Gmail at 500 per day.

An empty field means the window is not enforced. All three empty means the mailer sends unthrottled. Check your provider's documentation and configure slightly below its cap, because the windows here are fixed rather than sliding.

A slot is spent per **message**, not per recipient: a mail to five people costs one slot, which is how providers count as well.

When a slot is not available, VeloxFactory does not drop the mail:

Path	What happens
Synchronous	The mail is converted into a delayed queued send. The response says so, the task stays <i>Pending</i> and shows when it is due.
Asynchronous	The queued job releases itself until the window opens. Waiting does not count against the retry budget, so throttling never turns into an error.

The edit page shows the remaining slots per configured window, so a run that is about to hit a cap is visible before it does.

⚠ **The queue fallback needs a running queue.** If a synchronous send is pushed into the queue because of the rate limit and no Horizon worker is running, the mail waits as *Pending* in the Mail Queue. Nothing is lost, but nothing is sent either.

Counters are per mailer and per installation. Two VeloxFactory instances sharing one SMTP account do not share a counter.

Name resolution and deletion

A render request may address a mailer by its numeric ID or by its unique name, and only **active** records resolve. An unknown or inactive name is refused with a clear message before anything is rendered, rather than silently falling back to some default mailer.

A mailer cannot be deleted while a mail task still references it, so the history of what was sent stays intact. To retire a mailbox, switch the record to inactive: it disappears from every picker and

stops resolving, while everything that was sent through it keeps its record.

The API

Mailers are a regular API resource under `/api/v1/mailler`, with the same envelope, the same audit segment and the same error format as every other resource.

Endpoint	Description
<code>GET /mailler</code>	List mailers, optionally narrowed with <code>limit</code> .
<code>GET /mailler/{id}</code>	A single mailer, with <code>withAudit=true</code> including its audit segment.
<code>POST /mailler</code>	Create a mailer. <code>name</code> , <code>transport</code> and <code>fromAddress</code> are required, <code>host</code> and <code>port</code> additionally for SMTP.
<code>POST /mailler/{id}/test</code>	Send a test mail, optionally to a given <code>to</code> address, and update the approval status.
<code>PATCH /mailler/{id}</code>	Partial update. An omitted or empty password keeps the stored one.
<code>DELETE /mailler/{id}</code>	Delete a mailer, refused while mail tasks still reference it.

A mailer resource is returned as:

```
{
```

```
        "rateLimitRemaining": {
            "minute": 28,
            "hour": 1,
            "day": 1
        }
    }
}
```

The password is never part of the response.

AI assistants

The MCP server exposes the same operations as tools, including the test. An assistant can create a mailer from a provider's documented settings and verify it in the same conversation.

Permissions

Mailers use the standard permission scheme and have their own **Mailers** section in the user editor.

Permission	Grants
mailer:read	See the menu entry, the overview and a mailer's detail page. Also required to pick a mailer when sending.
mailer:create	Create mailers.
mailer:update	Change mailers, including the active state, and send test mails.
mailer:delete	Delete mailers.
mailer:full	All of the above, like global:admin.

Credentials are readable to anyone with mailer:read in the sense that the username is shown; the password is not, on any path. Treat mailer:update as an administrative permission.