interweb

Generated on 25/11/2026 12:54

KNOWLEDGE BASE · PUBLIC DOCUMENT

Invoices, receipts and payment reminders by transactional e-mail: what to send, when and with which fields

SMTP Relay & Transactional

CODE	VERSION	COUNTRY · LANGUAGE
RLX-AO-KB-2798	v2026.11.25	Angola · English

Interweb Angola
RLX Angola, Lda. · Group RLX Holdings
Tax ID (NIF) 500 260 5724 · Commercial registration no. 26464-25/250724

https://www.meu.interweb.ao/knowledgebase/2798

Invoices, receipts and payment reminders by transactional e-mail: what to send, when and with which fields

SMTP Relay & Transactional · Created on **November 25, 2026** · 3 min read



The short answer: there are three e-mails, each at its own moment. The **invoice** goes out when your system issues it, with the PDF attached. The **receipt** goes out when the payment comes in. The **reminder** is scheduled with `send_at` to go out before the due date, and cancelled if the customer pays first. All three go through `POST /v1/email/messages`, from an address on your verified domain, such as `invoices@yourcompany.ao`.

The three e-mails and when each one goes out

E-mail	When it goes out	What it carries
Invoice	When your system issues it	Number, amount, due date and the PDF attached
Receipt	When the payment comes in	Receipt number, amount paid and the invoice it settles
Reminder	Scheduled a few days before the due date	Amount still due, due date and how to pay

The invoice, with the PDF attached

Save the subject and the text in a template, with the variables `{{name}}`, `{{number}}`, `{{amount}}` and `{{date}}`. When you send, give the `template_id`, this invoice's values in `variables` and the PDF in `attachments`:

```
curl https://api.interweb.ao/v1/email/messages \
-H "Authorization: Bearer sandbox_..." \
-H "Content-Type: application/json" \
-H "Idempotency-Key: invoice-FT-2026-214" \
-d '{"from": "Your Company <invoices@yourcompany.ao>",
"to": "joao.fernandes@example.ao",
"template_id": 12,
```

```
"variables": {"name": "João", "number": "FT 2026/214",
"amount": "125,000.00 Kz", "date": "30/11/2026"},
"attachments": [{"filename": "invoice-FT2026-214.pdf",
"content": "JVBERi0xLjQK...",
"content_type": "application/pdf"}]}'
```

The PDF goes in base64, up to 5 MB in total per e-mail, and executable files are refused. The details are in **attaching invoices to a transactional e-mail** and in **templates with variables**.

The receipt, when the payment comes in

Send the receipt from the same point in your system that marks the invoice as paid, with another template and the same sender. Use its own `Idempotency-Key`, such as `receipt-FT-2026-214`: if your system repeats the request, after a timeout or a payment notice that arrives twice, the customer does not get two receipts. See **retries and idempotency**.

The reminder before the due date

1. **When you issue the invoice, schedule the reminder** with `send_at`, in ISO 8601 with the time zone, for example `2026-11-27T09:00:00+01:00`. It can be up to 30 days ahead.
2. **Keep the `id`** from the answer next to the invoice, in your system.
3. **If the customer pays first**, cancel the reminder with `POST /v1/email/messages/{id}/cancel`. Credits are only deducted when the e-mail goes out, so a cancelled reminder is not paid for.
4. **If the due date is more than 30 days away**, schedule the reminder later from your own system, for example in a daily task.

More about the scheduled time in **scheduling an e-mail with send_at**.

Knowing whether the invoice arrived

Check the status with `GET /v1/email/messages/{id}` or receive it in your system through the **webhook**, with the `email.delivered` and `email.bounced` events. A bounced invoice is a customer who did not get it: your system can alert the team to check the address. Addresses with a permanent bounce go on the **suppression list** and stop receiving.

> **The log does not replace your invoice archive.**
> The content of each e-mail is kept for 30 days; after that only the record of the send remains. The invoice stays in your system, with the e-mail's `id` next to it, so you can find it later. See **the Email Center log**.

Rehearse all three before going live

with a `sandbox_` key: the request is checked, nothing goes out and nothing is charged. Send to `bounce@example.com` to see what your system does with a bounced invoice. See **testing without sending**.

RECOMMENDED

Want invoices, receipts and reminders to go out from your system on their own? See the Transactional E-mail page.

See Transactional E-mail →

SEE ALSO

- **Attaching invoices and tickets to a transactional e-mail: base64, size limit and refused files**
- **Transactional e-mail templates with variables: write once, fill in each customer's details**
- **Scheduling a transactional e-mail with send_at, and cancelling it before it goes out**
- **Email Center webhooks: receiving delivery, bounce and complaint events in your system**
- **Timeouts, retries and idempotency: how not to charge a customer twice**
- **Password reset, order and invoice e-mails: what a good transactional message says, with examples**

interweb

Interweb Angola

RLX Angola, Lda. · Group RLX Holdings

TAX ID (NIF)	500 260 5724
COMMERCIAL REGISTRATION NO.	26464-25/250724
REGISTERED OFFICE	Rua Direita do Patriota, Bairro Dangereux Belas, Luanda, Angola
CONTACT	info@interweb.ao · +244 942 011 191 · www.interweb.ao
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

RLX-AO-KB-2798 · v2026.11.25 · Generated on 25/11/2026 12:54

Online version of this document