



Invoicing in your own name and on behalf of your clients

Integrator guide · Almendro FEC · Costa Rica electronic vouchers v4.4

With **one token** and **one endpoint** you can issue in two ways: in your own company's name and in the name of every client you manage. The only difference in the request body is a single field: `managed_contributor_id`. The URL, the token, the headers and the voucher structure are identical. This document explains why you are allowed to do this, what has to be set up once per client, and shows a complete example of each case with the differences marked in red.

1. The two ways you can issue

You do not have two integrations or two systems. You have the same code calling the same endpoint. What decides whose name the invoice goes out under is whether or not you send one field:

	Direct mode (your own name)	Integrator mode (on behalf of a client)
Who sells	Your company	Your client
Issuer before Hacienda	Your company	Your client, never you
Extra field in the JSON	None	<code>managed_contributor_id</code>
Signing certificate	Your company's	The client's (you never see it)
Consecutive number	Your company's	The client's, on the terminal Almendro assigned to you
Monthly quota	Yours	Yours as well
Token	Yours	Yours. You never need the client's token
URL	The same in both cases: <code>POST /api/v1/public/vouchers</code>	

The key point: if a client asks you to invoice from your POS, your e-commerce or your ERP, you do not ask them for their token or their certificate. You ask them to accept your management request once, and from then on you issue with your own token by sending the client's UUID.

2. Why you may issue on behalf of a client

This is the part that usually raises doubts, and the answer is short: **you are never the issuer of your client's invoice**. The issuer before Hacienda is always the client. You are the system that sends the data to Almendro.

Specifically, when you issue with `managed_contributor_id`:

- The 50 digit key is generated with the **client's tax ID**.
- The XML is signed with the **client's .p12 certificate**, which Almendro holds in custody. You never download it or handle it.
- The consecutive number advances in the **client's** counters.

- Hacienda receives the client as the issuer. Your company does not appear in the XML.
- Almendro internally records who triggered the issuance (audit log), and that shows up in the response as `emitted_by`. It is audit data, not fiscal data.

For Almendro to accept that issuance, four conditions must hold. If any one is missing, the request is rejected with `422` before anything is generated: no consecutive number is spent, no quota is spent, nothing reaches Hacienda.

1. Your plan allows managed clients (Integrator plan or equivalent).
2. There is an **active** management relationship between you and that client.
3. The client has an active `.p12` certificate **for the environment you are calling**. Sandbox and production use different certificates.
4. The client's account is not suspended.

2.1 The client's consent is the axis of everything

The client authorizes the management once and can withdraw it whenever they want, without asking your permission, from their own panel (`DELETE /my-integrators/{id}`). Doing so revokes your access in cascade and you stop being able to issue for them. Vouchers already issued are not affected in any way: they still belong to the client and remain valid.

The same client can be managed by several integrators at once (for example their POS, their online store and their accountant), each with its own assigned terminal, without the consecutive numbers colliding with each other.

2.2 Legal basis

These are the references the platform implements:

- **Electronic Vouchers Regulation, article 5:** every taxpayer signs with their own cryptographic key. Paragraph 5 requires keeping a record of the clients the service is provided to, which is exactly what the management relationship records.
- **Article 16:** the taxpayer's sovereignty over their key. Hence the client can revoke your access at any time and you cannot delete their certificates.
- **Article 21:** the voucher is sent to the correct environment, which is why the certificate check is per environment.
- **Resolution MH-DGT-RES-0027-2024, article 6:** audit log of operations, which is the immutable record of the relationship and of who issued each voucher.

These references describe the design of the platform. For a formal query before the Tax Administration, confirm them with your tax advisor.

3. What has to be set up, once per client

3.1 If the client does not yet have an Almendro account

You create it:

```
POST /api/v1/public/my-contributors
```

On creation the relationship is left active and Almendro assigns you a terminal for that client. The first integrator terminal is `00002`, because `00001` stays reserved for when the client invoices on their own. The client receives a welcome link to enter their panel; if it is lost or expires, resend it with `POST /my-contributors/{id}/resend-magic-link`.

3.2 If the client already exists in Almendro

You cannot create them again. You request access and they decide:

```
POST /api/v1/public/access-requests
```

The client receives an email and a notification in their panel, and can accept or reject. The acceptance can be **partial**: sandbox only, production only, or both. Until they accept, you cannot issue for them. The status is checked with `GET /access-requests/sent`, and it can be cancelled with `POST /access-requests/{id}/cancel`.

3.3 The client's certificate

One is uploaded per environment:

```
POST /api/v1/public/my-contributors/{id}/certificates (multipart/form-data)
```

You cannot delete the client's certificates: that endpoint answers `403` on purpose. If you want to end the relationship from your side, give up your access with `DELETE /my-access/{grantId}`.

3.4 If the client comes from another invoicing provider

Before issuing the first voucher, check which number each type is at and adjust it. If you do not, Hacienda rejects it for a duplicate consecutive number.

```
GET /api/v1/public/my-contributors/{id}/sequences      see the current state
POST /api/v1/public/my-contributors/{id}/sequences    initialize a type
PUT /api/v1/public/my-contributors/{id}/sequences/{seqId} adjust the number
```

3.5 Getting the client's UUID

With `GET /api/v1/public/my-contributors` you get your portfolio. The `id` field of each client is exactly the value that goes into `managed_contributor_id`.

Do all of this in sandbox first. It is the same JSON and the same token: the only thing that changes is that the URL carries `/sandbox` before `/vouchers`. Sandbox vouchers have no fiscal value and do not spend production numbering or quota.

4. Example 1: invoice in your own name

Scenario. Sistemas Tico S.A. (you, corporate ID 3101999888) invoices Hotel Las Palmas S.A. for the development of a website.

```
POST /api/v1/public/vouchers
Authorization: Bearer {YOUR_TOKEN}
Content-Type: application/json
```

```
{
  "voucher_type": "01",
  "situation": "1",
  "issued_at": "2026-07-31T10:30:00-06:00",
  "issuer_activity_code": "6201.0",
  "sale_condition": "01",
  "currency_code": "CRC",
  "exchange_rate": 1.00000,
  "receiver": {
    "id_type": "02",
    "id_number": "3101704304",
    "name": "Hotel Las Palmas S.A.",
    "emails": ["recepcion@hotellaspalmas.cr"]
  },
  "line_items": [
    {
      "line_number": 1,
      "cabys_code": "8511000000000",
      "quantity": 1,
      "unit_of_measure": "Sp",
      "detail": "Website development",
      "unit_price": 500000.00000,
      "total_amount": 500000.00000,
      "sub_total": 500000.00000,
      "taxes": [
        { "codigo": "01", "codigoTarifa": "08", "tarifa": 13, "monto": 65000.00000 }
      ],
      "impuesto_neto": 65000.00000,
      "total_line_amount": 565000.00000
    }
  ],
  "payment_methods": [
    { "tipo": "04" }
  ]
}
```

202 Accepted response (trimmed):

```
{
  "success": true,
  "data": {
    "voucher_key": "506310726...(50 digits, with YOUR tax ID)",
    "consecutive_number": "00100001010000000001",
    "status": "pending",
    "is_via_integrator": false,
    "emitted_by": null,
    "emitter": null
  }
}
```

The 20 digit consecutive number reads like this: 001 branch, 00001 terminal, 01 voucher type, 0000000001 number. Issuing in your own name, the terminal is yours.

5. Example 2: the same call, on behalf of a client

Scenario. Ferretería El Roble S.A. (your managed client, corporate ID 3101555777) sells materials to Hotel Las Palmas S.A. from the point of sale you operate. The invoice has to be issued by the hardware store, not by you.

Same endpoint, same token, same headers:

```
POST /api/v1/public/vouchers
Authorization: Bearer {YOUR_TOKEN}
Content-Type: application/json
```

■ In red, what changes because you are issuing on behalf of the client. The rest is the data of that sale.

```
{
  "managed_contributor_id": "019d867d-0241-7288-8ece-fd64da75616d",
  "voucher_type": "01",
  "situation": "1",
  "issued_at": "2026-07-31T10:30:00-06:00",
  "issuer_activity_code": "4752.0",
  "sale_condition": "01",
  "currency_code": "CRC",
  "exchange_rate": 1.00000,
  "receiver": {
    "id_type": "02",
    "id_number": "3101704304",
    "name": "Hotel Las Palmas S.A.",
    "emails": ["recepcion@hotellaspalmas.cr"]
  },
  "line_items": [
    {
      "line_number": 1,
      "cabys_code": "2394001000000",
      "quantity": 10,
      "unit_of_measure": "Unid",
      "detail": "50 kg cement bag",
      "unit_price": 7500.00000,
      "total_amount": 75000.00000,
      "sub_total": 75000.00000,
      "taxes": [
        { "codigo": "01", "codigoTarifa": "08", "tarifa": 13, "monto": 9750.00000 }
      ],
      "impuesto_neto": 9750.00000,
      "total_line_amount": 84750.00000
    }
  ],
  "payment_methods": [
    { "tipo": "01" }
  ]
}
```

202 Accepted response (trimmed):

```
{
  "success": true,
  "data": {
    "voucher_key": "506310726...(50 digits, with the CLIENT's tax ID)",
    "consecutive_number": "00100002010000000001",
    "status": "pending",
    "is_via_integrator": true,
    "emitted_by": { "contributor_id": "...", "name": "Sistemas Tico S.A." },
    "emitter": { "name": "Ferretería El Roble S.A.", "id_number": "3101555777" }
  }
}
```

Look at the consecutive number: `001` branch, `00002` terminal (the one Almendro assigned to you over that client), `01` type, `0000000001` number. That own terminal is what prevents two integrators of the same client from generating duplicate consecutive numbers.

The CABYS codes in the examples are illustrative. Always use the real code of the good or service according to the Hacienda catalog. The same applies to `issuer_activity_code` : it has to be a registered economic activity of the issuer.

6. Differences, field by field

■ In red, the only thing that changes between one mode and the other.

Element	In your own name	On behalf of the client
URL	Identical: <code>POST /api/v1/public/vouchers</code>	
Token and headers	Identical. Always your integrator account token	
<code>managed_contributor_id</code>	Not sent	UUID of the managed client
<code>issuer_activity_code</code>	Your company's economic activity	The client's economic activity. If you send yours, it errors out
<code>receiver</code>	The same in both cases: it is the buyer of that sale	
<code>line_items</code> , <code>payment_methods</code> , <code>references</code>	Same structure and same calculation rules	
<code>client_id</code> and <code>item_id</code>	Your account's catalog	Also your account's catalog, not the client's
<code>local_code</code> and <code>terminal_code</code>	Your branches and registers	Do not send them. The terminal resolves on its own
50 digit key	With your tax ID	With the client's tax ID
Digital signature	Your certificate	The client's certificate
Consecutive terminal	<code>00001</code>	<code>00002</code> onwards, the one assigned to you
Monthly quota and rate limit	Always against your integrator plan	
Response	<code>is_via_integrator: false</code>	<code>is_via_integrator: true</code> , plus <code>emitted_by</code> and <code>emitter</code>

Two details that cause avoidable errors. First: `client_id` and `item_id` are always resolved against the token's catalog, that is, yours. When issuing on behalf of a client, the predictable move is to send explicit `receiver` and `line_items` . Second: do not send `local_code` or `terminal_code` in managed mode, unless you have created branches and registers for that client with `/my-contributors/{id}/branches` .

7. Errors you will see and what they mean

HTTP	Message	What happened and how to fix it
422	Your plan does not allow issuing on behalf of managed clients	Your plan has no managed client allowance. It requires the Integrator plan
422	The managed contributor does not exist, is not active, or has no active managed relationship with your account	The UUID is wrong, the access request was never accepted, or the client unlinked you. Check with <code>GET /my-contributors</code>
422	The managed contributor "X" is suspended	The client's account is suspended. The client or the administrator resolves it
422	The managed contributor "X" has no active .p12 certificate for the "production" environment	The certificate for that environment is missing. Typical case: the sandbox one was uploaded and production is being called. Upload it with <code>POST /my-contributors/{id}/certificates</code>
422	Issuer economic activity error	You sent your own <code>issuer_activity_code</code> instead of the one that corresponds to the client
429	Monthly voucher limit reached	It is your quota, not the client's. Everything you issue for clients counts against your plan

None of the `422` responses consumes a consecutive number or quota: validation happens before the voucher is generated and signed. You can fix and retry at no cost.

8. Checklist before the first real submission

1. Your plan has a managed client allowance.
2. The client accepted your access request, or you created their account.
3. The client's certificate is loaded for the environment you are going to use.
4. You took the client's `id` with `GET /my-contributors` and you are going to send it in `managed_contributor_id`.
5. You are using the **client's** economic activity in `issuer_activity_code`.
6. If the client came from another invoicing provider, you already adjusted the consecutive numbers.
7. You tested the full flow on `/sandbox` and Hacienda accepted the test voucher.
8. You have the webhook configured, or you are going to poll the status with `GET /vouchers/{key}`.

When the eight points are ready, remove `/sandbox` from the URL. There is nothing else to change.