

Listar Facturas

Permite obtener la lista paginada de todas las facturas registradas en el sistema. Cada elemento incluye el número de factura, resolución, fechas, valor total, saldo pendiente, tercero facturado, estado, información DIAN y datos de auditoría.

¿Para qué sirve este servicio?

Úsalo para sincronizar la información de facturación con tu sistema externo, generar reportes de cartera, alimentar dashboards con indicadores de facturación, validar el estado de cobro de los contratos o exportar la información a Excel y Power BI.

1. El Endpoint (La dirección web)

Apunta tu sistema a la siguiente dirección. Recuerda reemplazar **{{instancia}}** por la dirección web completa que utilizas para ingresar a tu plataforma.

```
GET https://{{instancia}}/service/v2/public/invoices/list
```

¿Qué debes colocar en {{instancia}}?

Es muy sencillo: corresponde a la **dirección web principal** que utilizas a diario para ingresar a tu plataforma (incluyendo la terminación `.nuby.app` o `.arrendasoft.co`).

Por ejemplo, si para entrar a tu sistema escribes `inmobiliaria.nuby.app` o `inmobiliaria.arrendasoft.co` en tu navegador, esa será exactamente tu instancia. Solo asegúrate de no incluir el "https://" ni barras diagonales ("/") al final.

2. La Petición (¿Qué debes enviarnos?)

Este servicio no requiere cuerpo en la petición. Envía los encabezados requeridos y opcionalmente los parámetros de paginación en la URL:

Método	GET
Content-Type	application/json

Authorization	Bearer token , Token obtenido al consumir el servicio Login
----------------------	---

¿Aún no tienes tu Token de acceso?

Para consumir este servicio necesitas un Token vigente. Consulta el servicio de [Login](#) para aprender cómo obtenerlo.

Parámetros de consulta (query string):

Parámetro	Tipo	Requerido	Por defecto	Descripción
<code>page</code>	integer	No	1	Número de página de resultados que se desea recuperar. Debe ser un número positivo.
<code>page_size</code>	integer	No	10	Número máximo de facturas por página. Debe ser un número positivo. Valor máximo permitido: 1000 .

Ejemplos de petición:

```
GET https://mi-inmobiliaria.nuby.app/service/v2/public/invoices/list
GET https://mi-inmobiliaria.nuby.app/service/v2/public/invoices/list?page=2
GET https://mi-inmobiliaria.nuby.app/service/v2/public/invoices/list?page_size=50
GET https://mi-inmobiliaria.nuby.app/service/v2/public/invoices/list?page=3&page_size=100
```

3. La Respuesta (¿Qué te entregaremos?)

Si tu Token es válido, el sistema te devolverá un objeto JSON con el estado de la respuesta, el listado de facturas y la información de paginación. Los resultados están ordenados por fecha de factura descendente (más recientes primero). La respuesta se verá similar a esta:

```
{
  "status": 200,
  "message": null,
  "body": [
    {
      "factura_id": 15230,
```

```
"factura_numero": 8745,
"resolucion": "FECO - 18764000002851",
"fecha_factura": "2025-04-01",
"fecha_vencimiento": "2025-04-06",
"valor_total": "2850000.00",
"saldo": "2850000.00",
"documento_tercero": "80213654",
"nombre_tercero": "CARLOS EDUARDO PINEDA VARGAS",
"estado": "Facturada",
"estado_dian": "Aceptada",
"estado_cliente_dian": "Recibida",
"notas": 0,
"creado_por": "Sandra",
"fecha_envio_dian": "2025-04-01 08:30:15",
"fecha_creacion": "2025-04-01 07:45:22"
},
{
  "factura_id": 15229,
  "factura_numero": 8744,
  "resolucion": "FECO - 18764000002851",
  "fecha_factura": "2025-04-01",
  "fecha_vencimiento": "2025-04-06",
  "valor_total": "4500000.00",
  "saldo": "0.00",
  "documento_tercero": "1098745231",
  "nombre_tercero": "JULIANA PATRICIA RÍOS MENDOZA",
  "estado": "Pagada",
  "estado_dian": "Aceptada",
  "estado_cliente_dian": "Recibida",
  "notas": 0,
  "creado_por": "Sandra",
  "fecha_envio_dian": "2025-04-01 08:31:02",
  "fecha_creacion": "2025-04-01 07:46:10"
},
{
  "factura_id": 15210,
  "factura_numero": 8730,
  "resolucion": "0 - 18764000002851",
  "fecha_factura": "2025-03-01",
  "fecha_vencimiento": "2025-03-06",
```

```
"valor_total": "1200000.00",
"saldo": "1200000.00",
"documento_tercero": "43876123",
"nombre_tercero": "DIANA MARCELA OSPINA VELÁSQUEZ",
"estado": "Facturada",
"estado_dian": null,
"estado_cliente_dian": null,
"notas": 1,
"creado_por": "Administrador",
"fecha_envio_dian": null,
"fecha_creacion": "2025-03-01 09:12:33"
},
{
  "factura_id": 14850,
  "factura_numero": 8520,
  "resolucion": "FECO - 18764000002851",
  "fecha_factura": "2025-02-01",
  "fecha_vencimiento": "2025-02-06",
  "valor_total": "3950000.00",
  "saldo": "3950000.00",
  "documento_tercero": "900456789",
  "nombre_tercero": "INVERSIONES HORIZONTE S.A.S.",
  "estado": "Anulada por NC",
  "estado_dian": "Aceptada",
  "estado_cliente_dian": null,
  "notas": 1,
  "creado_por": "Sandra",
  "fecha_envio_dian": "2025-02-01 10:05:44",
  "fecha_creacion": "2025-02-01 08:20:15"
}
],
"pagination": {
  "total_records": 15230,
  "total_pages": 305,
  "current_page": 1,
  "page_size": 50,
  "current_page_records": 4,
  "has_next_page": true,
  "has_previous_page": false
}
```

```
}
```

Estructura principal de la respuesta

Campo	Tipo	Descripción
<code>status</code>	integer	Código del estado de la respuesta. <code>200</code> para exitoso, <code>400</code> para error de validación, <code>500</code> para error interno.
<code>message</code>	string null	Cuando la respuesta es exitosa vale <code>null</code> . Si no se encuentran facturas, contiene el mensaje informativo. Si hay error, describe el problema.
<code>body</code>	array null	Listado de facturas devueltas. Arreglo vacío <code>[]</code> si no hay resultados. <code>null</code> en caso de error.
<code>pagination</code>	object null	Información de paginación para recorrer los resultados. <code>null</code> en caso de error.

Campos de cada factura (body)

Cada elemento dentro del arreglo `body` contiene las siguientes claves:

Campo	Tipo	Descripción
<code>factura_id</code>	integer	Identificador único de la factura en el sistema.
<code>factura_numero</code>	integer	Número consecutivo de la factura.
<code>resolucion</code>	string	Prefijo y número de la resolución de facturación en formato <code>PREFIJO - NÚMERO</code> (ej. <code>"FECO - 18764000002851"</code>).
<code>fecha_factura</code>	string	Fecha de emisión de la factura en formato <code>YYYY-MM-DD</code> .
<code>fecha_vencimiento</code>	string	Fecha de vencimiento de la factura en formato <code>YYYY-MM-DD</code> .
<code>valor_total</code>	string	Valor total de la factura con decimales (ej. <code>"2850000.00"</code>).
<code>saldo</code>	string	Saldo pendiente de pago de la factura con decimales (ej. <code>"0.00"</code> cuando está pagada).
<code>documento_tercero</code>	string	Número de documento de identidad del tercero/cliente facturado.
<code>nombre_tercero</code>	string	Nombre completo del tercero/cliente facturado.

Campo	Tipo	Descripción
estado	string	Estado textual de la factura. Valores posibles: <code>Facturada</code> , <code>Anulada</code> , <code>Pagada</code> , <code>Borrador</code> , <code>Anulada por NC</code> , <code>Contabilizada</code> .
estado_dian	string null	Estado de la factura en la DIAN cuando es factura electrónica (ej. <code>"Aceptada"</code>). <code>null</code> si no aplica o no se ha enviado.
estado_cliente_dian	string null	Estado de aceptación/rechazo del cliente ante la DIAN cuando es factura electrónica. <code>null</code> si no aplica.
notas	integer	Cantidad de notas débito o crédito asociadas a la factura.
creado_por	string	Nombre del usuario que creó la factura en el sistema.
fecha_envio_dian	string null	Fecha y hora de envío de la factura a la DIAN en formato <code>YYYY-MM-DD HH:MM:SS</code> . <code>null</code> si no se ha enviado.
fecha_creacion	string	Fecha y hora de creación del registro en formato <code>YYYY-MM-DD HH:MM:SS</code> .

Campos de paginación (pagination)

El objeto `pagination` contiene las siguientes claves:

Campo	Tipo	Descripción
total_records	integer	Total de facturas registradas en el sistema.
total_pages	integer	Cantidad total de páginas disponibles (total de registros dividido por el tamaño de página).
current_page	integer	Número de la página actual consultada.
page_size	integer	Tamaño de la página (cantidad de registros solicitados por consulta).
current_page_records	integer	Cantidad de registros efectivamente devueltos en la página actual.
has_next_page	boolean	<code>true</code> si existen páginas posteriores que se pueden consultar.
has_previous_page	boolean	<code>true</code> si existen páginas anteriores que se pueden consultar.

Nota sobre facturas sin resultados

Si no existen facturas registradas en el sistema, la respuesta tendrá HTTP `200` con `body` vacío (`[]`), un `message` informativo: "No se encontraron facturas en el sistema." y los valores de paginación en cero.

4. Seguridad y Posibles Errores

¡Tu pase tiene fecha de caducidad!

Por medidas de seguridad, el `token` que te entregamos **solo dura 1 hora**. Una vez transcurrido ese tiempo, el pase expirará y el sistema te bloqueará el acceso devolviéndote un error `401`. Cuando esto ocurra, tu sistema simplemente debe volver a consumir el servicio de [Login](#) para pedir un pase nuevo y continuar trabajando.

Así se ve el error que te devolverá el sistema cuando tu token se haya vencido o intentes usar un pase inválido:

```
{
  "statusCode": 401,
  "error": {
    "type": "SERVER_ERROR",
    "description": "JWT Token expired."
  }
}
```

Si alguno de los parámetros de paginación no cumple con las validaciones, el sistema devolverá un error `400`. Por ejemplo:

```
{
  "status": 400,
  "message": "El parámetro \"page_size\" debe ser un número positivo.",
  "body": null,
  "pagination": null
}
```

Posibles errores:

Código HTTP	Descripción
-------------	-------------

<p>400</p>	<p>Token faltante o inválido. Posibles causas:</p> <ul style="list-style-type: none"> — No se envió el encabezado <code>Authorization</code>. Mensaje: <code>"JWT Token required."</code> — El encabezado no tiene el formato <code>Bearer {token}</code>. Mensaje: <code>"JWT Token not send."</code> — El token no fue encontrado en el sistema. Mensaje: <code>"JWT Token not found."</code> <p>Parámetros inválidos. Posibles causas:</p> <ul style="list-style-type: none"> — <code>page</code> no es numérico o no es positivo. Mensaje: <code>"El parámetro \"page\" debe ser un número positivo."</code> — <code>page_size</code> no es numérico o no es positivo. Mensaje: <code>"El parámetro \"page_size\" debe ser un número positivo."</code>
<p>401</p>	<p>El token ha expirado. Debes generar uno nuevo consumiendo el servicio de Login. Mensaje: <code>"JWT Token expired."</code></p>
<p>403</p>	<p>El cliente OAuth no tiene el scope necesario para esta operación. Para endpoints GET se requiere el scope <code>read</code>. Mensaje: <code>"Insufficient scope. Required: 'read', granted: '{scope_actual}'."</code></p>
<p>500</p>	<p>Error interno del servidor al procesar la consulta de facturas. Mensaje: <code>"Error interno del servidor al obtener las facturas."</code></p>

5. Ejemplos de integración

Aquí tienes ejemplos de código listos para que tus desarrolladores los adapten a tu plataforma:

cURL

```
# Listar facturas con paginación por defecto
curl -X GET "https://{{instancia}}/service/v2/public/invoices/list" \
-H "Content-Type: application/json" \
-H "Authorization: Bearer TU_TOKEN_AQUI"

# Listar facturas página 3 con 100 registros por página
curl -X GET "https://{{instancia}}/service/v2/public/invoices/list?page=3&page_size=100" \
-H "Content-Type: application/json" \
-H "Authorization: Bearer TU_TOKEN_AQUI"
```

PHP

```
<?php

$instance = 'tu_instancia'; // Reemplaza con tu instancia real
$token = 'TU_TOKEN_AQUI'; // Token obtenido del servicio Login

// Parámetros de paginación
$page = 1;
$page_size = 50;

$queryParams = http_build_query([
    'page' => $page,
    'page_size' => $page_size
]);

$url = "https://{instance}/service/v2/public/invoices/list?{$queryParams}";

$ch = curl_init($url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, [
    'Content-Type: application/json',
    "Authorization: Bearer {$token}"
]);

$response = curl_exec($ch);
$http_code = curl_getinfo($ch, CURLINFO_HTTP_CODE);

if (curl_errno($ch)) {
    echo 'Error: ' . curl_error($ch);
} else {
    echo "Código de estado HTTP: " . $http_code . "\n";

    $info = json_decode($response, true);
    $body = $info['body'] ?? [];

    if ($http_code === 200 && !empty($body)) {
        $pagination = $info['pagination'];
        echo "Facturas encontradas: {$pagination['total_records']} (página
{$pagination['current_page']} de {$pagination['total_pages']})\n";
    }
}
```

```
        foreach ($body as $factura) {
            echo " Factura #{$factura['factura_numero']} - {$factura['nombre_tercero']} -
{$factura['estado']} - Total: \${$factura['valor_total']} - Saldo:
\${$factura['saldo']}\n";
        }
    } else {
        echo "Mensaje: " . ($info['message'] ?? 'Sin resultados') . "\n";
    }
}
curl_close($ch);

?>
```

Python

```
import requests

# 1. Configura tus datos de acceso
instancia = 'mi-inmobiliaria.nuby.app' # Reemplaza con tu dirección web completa
token = 'TU_TOKEN_AQUI' # Token obtenido del servicio Login

# 2. Prepara la dirección y los parámetros
url = f"https://{instancia}/service/v2/public/invoices/list"

params = {
    "page": 1,
    "page_size": 50
}

headers = {
    "Content-Type": "application/json",
    "Authorization": f"Bearer {token}"
}

# 3. Envía la petición GET y procesa la respuesta
try:
    response = requests.get(url, params=params, headers=headers)
```

```

print(f"Código de estado HTTP: {response.status_code}")

if response.status_code == 200:
    info = response.json()
    body = info.get('body', [])
    pagination = info.get('pagination', {})

    if body:
        print(f"Facturas: {pagination['total_records']} total (página
{pagination['current_page']} de {pagination['total_pages']})")
        for factura in body:
            print(f"  #{factura['factura_numero']} - {factura['nombre_tercero']} -
{factura['estado']} - Total: ${factura['valor_total']} - Saldo: ${factura['saldo']}")
        else:
            print(f"Mensaje: {info.get('message', 'Sin resultados')}")
    else:
        print("Error al consultar las facturas.")
        print(f"Detalle del error: {response.text}")

except Exception as e:
    print(f"Ocurrió un error de conexión: {e}")

```

JavaScript

```

// 1. Configura tus datos de acceso
const instancia = 'mi-inmobiliaria.nuby.app'; // Reemplaza con tu dirección web completa
const token = 'TU_TOKEN_AQUI'; // Token obtenido del servicio Login

// 2. Prepara la dirección con parámetros
const params = new URLSearchParams({
  page: 1,
  page_size: 50
});

const url = `https://${instancia}/service/v2/public/invoices/list?${params}`;

```

```

// 3. Función para consultar las facturas
async function consultarFacturas() {
  try {
    const response = await fetch(url, {
      method: 'GET',
      headers: {
        'Content-Type': 'application/json',
        'Authorization': `Bearer ${token}`
      }
    });

    console.log(`Código de estado HTTP: ${response.status}`);

    if (response.ok) {
      const info = await response.json();
      const body = info.body || [];
      const pagination = info.pagination || {};

      if (body.length > 0) {
        console.log(`Facturas: ${pagination.total_records} total (página
${pagination.current_page} de ${pagination.total_pages})`);
        body.forEach(factura => {
          console.log(` #${factura.factura_numero} - ${factura.nombre_tercero}
- ${factura.estado} - Total: $$${factura.valor_total} - Saldo: $$${factura.saldo}`);
        });
      } else {
        console.log(`Mensaje: ${info.message || 'Sin resultados'}`);
      }
    } else {
      console.log('Error al consultar las facturas.');
```

```

      const errorData = await response.text();
      console.log(`Detalle del error: ${errorData}`);
    }
  } catch (error) {
    console.error(`Ocurrió un error de conexión: ${error}`);
  }
}

// 4. Ejecutamos la función

```

```
consultarFacturas();
```

Power Query M (Excel / Power BI)

```
let
    // 1. Configura tus datos de acceso
    instancia = "mi-inmobiliaria.nuby.app", // Reemplaza con tu dirección web completa
    token = "TU_TOKEN_AQUI", // Token obtenido del servicio Login

    // 2. Prepara la dirección de la petición con parámetros
    url = "https://" & instancia &
    "/service/v2/public/invoices/list?page=1&page_size=1000",

    // 3. Envía la petición GET
    response = Web.Contents(url, [
        Headers = [
            #"Content-Type" = "application/json",
            #"Authorization" = "Bearer " & token
        ]
    ]),

    // 4. Decodifica la respuesta JSON
    jsonResponse = Json.Document(response),
    body = jsonResponse[body],

    // 5. Convierte el listado en tabla
    tabla = Table.FromList(body, Splitter.SplitByNothing(), null, null,
    ExtraValues.Error),
    expandido = Table.ExpandRecordColumn(tabla, "Column1",
        {"factura_id", "factura_numero", "resolucion", "fecha_factura",
        "fecha_vencimiento", "valor_total", "saldo", "documento_tercero", "nombre_tercero",
        "estado", "estado_dian", "notas", "creado_por", "fecha_creacion"},
        {"FacturaID", "Numero", "Resolucion", "FechaFactura", "FechaVencimiento",
        "ValorTotal", "Saldo", "DocTercero", "NombreTercero", "Estado", "EstadoDian", "Notas",
        "CreadoPor", "FechaCreacion"})
in
    expandido
```

Revisión #16

Creado el 1 agosto 2025 19:06:57 por Jorge Iván González

Actualizado el 17 abril 2026 01:32:47 por Isabel Higuira