

Contratos

Bienvenido al **archivador legal digital** de tu inmobiliaria. En esta sección agrupamos todas las opciones de conexión para que tus plataformas externas interactúen con los **acuerdos de arrendamiento**.

¿Cuál es el propósito de este módulo? Facilitar la sincronización de tu ecosistema. Aquí encontrarás las herramientas necesarias para que tus sistemas consulten y extraigan automáticamente los detalles de cualquier contrato, eliminando por completo la necesidad de realizar búsquedas manuales o revisar documentos físicos en nuestra plataforma.

Explora el listado de abajo para descubrir cómo consultar términos, información financiera, estados y fechas de tus inmuebles alquilados.

- [Listar Contratos](#)

Listar Contratos

Permite obtener la lista paginada de todos los contratos de arrendamiento registrados en el sistema. Cada elemento incluye la información completa del contrato: propiedad asociada, propietarios, inquilinos, valores económicos, fechas, estado y observaciones.

¿Para qué sirve este servicio?

Úsalo para sincronizar el inventario de contratos con tu sistema externo, generar reportes de gestión inmobiliaria, alimentar dashboards con información contractual o consultar el estado general de los contratos administrados.

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/contracts/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
page	integer	No	1	Número de página de resultados que se desea recuperar. Debe ser un número positivo.
page_size	integer	No	10	Número máximo de contratos por página. Debe ser un número positivo. Valor máximo permitido: 1000 .

Ejemplos de petición:

```
GET https://mi-inmobiliaria.nuby.app/service/v2/public/contracts/list
GET https://mi-inmobiliaria.nuby.app/service/v2/public/contracts/list?page=2
GET https://mi-inmobiliaria.nuby.app/service/v2/public/contracts/list?page_size=50
GET https://mi-inmobiliaria.nuby.app/service/v2/public/contracts/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 contratos y la información de paginación. La respuesta se verá similar a esta:

```
{
  "status": 200,
  "message": null,
  "body": [
    {
      "contrato_id": 45,
      "consecutivo": "45",
      "propiedad_id": 312,
      "propiedad": "312 - Cll 78 # 12 - 34 Apto 501",
      "estrato_propiedad": "Cuatro",
```

```
"propietario": "[1] 52498731 - ANDREA MILENA CASTRO ROJAS",
"propietarios_id": "18",
"inquilino": "[1] 80213654 - CARLOS EDUARDO PINEDA VARGAS",
"inquilinos_id": "42",
"valor_canon_individual": "2850000",
"canon_total": 2850000,
"porcentaje_comision": "10.00 %",
"periodicidad": "Mensual",
"escenario": "CANON DE ARRENDAMIENTO SIN IVA, CON RETEFUENTE 3.5%, SIN RETEIVA,
SIN RETEICA",
"estado": "Activo",
"estado_id": 1,
"fecha_inicio": "2024-06-01",
"fecha_fin": "2025-05-31",
"fecha_terminacion": null,
"observaciones": "Contrato renovado automáticamente por un año adicional.",
"uso": "Vivienda",
"fecha_creacion": "2024-05-28 14:32:10",
"creado_por": "Sandra"
},
{
  "contrato_id": 46,
  "consecutivo": "46",
  "propiedad_id": 589,
  "propiedad": "589 - Cra 45 # 90 - 12 Local 3",
  "estrato_propiedad": "Comercial",
  "propietario": "[1] 900456789 - INVERSIONES HORIZONTE S.A.S.",
  "propietarios_id": "65",
  "inquilino": "[1] 1098745231 - JULIANA PATRICIA RÍOS MENDOZA",
  "inquilinos_id": "73",
  "valor_canon_individual": "4500000",
  "canon_total": 4500000,
  "porcentaje_comision": "8.50 %",
  "periodicidad": "Mensual",
  "escenario": "CANON DE ARRENDAMIENTO CON IVA 19%, CON RETEFUENTE 3.5%, CON RETEIVA
15%, SIN RETEICA",
  "estado": "Activo",
  "estado_id": 1,
  "fecha_inicio": "2023-09-01",
  "fecha_fin": "2025-08-31",
```

```
    "fecha_terminacion": null,
    "observaciones": null,
    "uso": "Comercial",
    "fecha_creacion": "2023-08-25 09:15:44",
    "creado_por": "Administrador"
  },
  {
    "contrato_id": 12,
    "consecutivo": "12",
    "propiedad_id": 104,
    "propiedad": "104 - Cll 23 # 56 - 78",
    "estrato_propiedad": "Tres",
    "propietario": "[1] 71654321 - HÉCTOR FABIO MORENO AGUILAR",
    "propietarios_id": "31",
    "inquilino": "[1] 43876123 - DIANA MARCELA OSPINA VELÁSQUEZ",
    "inquilinos_id": "55",
    "valor_canon_individual": "1200000",
    "canon_total": 1200000,
    "porcentaje_comision": "12.00 %",
    "periodicidad": "Mensual",
    "escenario": "CANON DE ARRENDAMIENTO SIN IVA, SIN RETEFUENTE, SIN RETEIVA, SIN
RETEICA",
    "estado": "Terminado",
    "estado_id": 2,
    "fecha_inicio": "2022-01-15",
    "fecha_fin": "2023-01-14",
    "fecha_terminacion": "2022-11-30",
    "observaciones": "Terminación anticipada por mutuo acuerdo.",
    "uso": "Vivienda",
    "fecha_creacion": "2021-12-20 16:40:33",
    "creado_por": "Sandra"
  }
],
"pagination": {
  "total_records": 347,
  "total_pages": 116,
  "current_page": 1,
  "page_size": 3,
  "current_page_records": 3,
  "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 contratos, contiene el mensaje informativo. Si hay error, describe el problema.
<code>body</code>	array	Listado de contratos devueltos. Arreglo vacío <code>[]</code> si no hay resultados.
<code>pagination</code>	object	Información de paginación para recorrer los resultados.

Campos de cada contrato (body)

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

Campo	Tipo	Descripción
<code>contrato_id</code>	integer	Identificador único del contrato en el sistema.
<code>consecutivo</code>	string	Número consecutivo interno del contrato.
<code>propiedad_id</code>	integer	Identificador de la propiedad asociada al contrato.
<code>propiedad</code>	string	Etiqueta descriptiva de la propiedad en formato <code>{id} - {dirección}</code> .
<code>estrato_propiedad</code>	string	Estrato socioeconómico asignado a la propiedad (ej. "Tres", "Cuatro", "Comercial").
<code>propietario</code>	string	Cadena con el consecutivo del detalle, documento y nombre completo del propietario o propietarios. Formato: <code>[n] documento - NOMBRE</code> . Si hay varios, van separados por coma.
<code>propietarios_id</code>	string	ID(s) del/los propietario(s) en el sistema. Si hay varios, van separados por coma.

Campo	Tipo	Descripción
<code>inquilino</code>	string	Cadena con el consecutivo del detalle, documento y nombre completo del inquilino o inquilinos. Mismo formato que <code>propietario</code> .
<code>inquilinos_id</code>	string	ID(s) del/los inquilino(s) en el sistema. Si hay varios, van separados por coma.
<code>valor_canon_individual</code>	string	Valor del canon por cada detalle del contrato. Si hay varios detalles, los valores van separados por coma.
<code>canon_total</code>	number null	Valor total del canon de arrendamiento del contrato.
<code>porcentaje_comision</code>	string	Porcentaje de comisión de la inmobiliaria con formato <code>"X.XX %"</code> .
<code>periodicidad</code>	string	Frecuencia de pago del canon (ej. "Mensual").
<code>escenario</code>	string	Descripción del escenario tributario y de facturación aplicado al contrato.
<code>estado</code>	string	Estado textual del contrato. Valores posibles: <code>Activo</code> , <code>Terminado</code> , <code>En Construcción</code> .
<code>estado_id</code>	integer	Identificador numérico del estado: <code>1</code> = Activo, <code>2</code> = Terminado, <code>3</code> = En Construcción.
<code>fecha_inicio</code>	string	Fecha de inicio del contrato en formato <code>YYYY-MM-DD</code> .
<code>fecha_fin</code>	string	Fecha de finalización prevista del contrato en formato <code>YYYY-MM-DD</code> .
<code>fecha_terminacion</code>	string null	Fecha de terminación efectiva del contrato (si aplica). <code>null</code> si el contrato sigue vigente.
<code>observaciones</code>	string null	Observaciones o notas asociadas al contrato. <code>null</code> si no hay.
<code>uso</code>	string	Tipo de uso de la propiedad: <code>Vivienda</code> o <code>Comercial</code> .
<code>fecha_creacion</code>	string	Fecha y hora de creación del registro en formato <code>YYYY-MM-DD HH:MM:SS</code> .
<code>creado_por</code>	string	Nombre del usuario que creó el contrato en el sistema.

Campos de paginación (pagination)

El objeto `pagination` contiene las siguientes claves:

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

Nota sobre contratos sin resultados

Si no existen contratos registrados en el sistema, la respuesta tendrá HTTP `200` con `body` vacío (`[]`), un `message` informativo 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` con la estructura estándar. Por ejemplo:

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

Posibles errores:

Código HTTP	Descripción
400	Token faltante o inválido. Posibles causas: — 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> Parámetros inválidos. Posibles causas: — <code>page</code> no es numérico o no es positivo. — <code>page_size</code> no es numérico o no es positivo.
401	El token ha expirado. Debes generar uno nuevo consumiendo el servicio de Login . Mensaje: <code>"JWT Token expired."</code>
403	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>
500	Error interno del servidor al procesar la consulta de contratos.

5. Ejemplos de integración

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

cURL

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

# Listar contratos página 3 con 100 registros por página
curl -X GET "https://{{instancia}}/service/v2/public/contracts/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/contracts/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 = $info['pagination'];
        echo "Contratos encontrados: {$pagination['total_records']} (página
{$pagination['current_page']} de {$pagination['total_pages']})\n";

        foreach ($body as $contrato) {
            echo "    [{$contrato['consecutivo']}] {$contrato['propiedad']} -
{$contrato['estado']} - Canon: \${$contrato['canon_total']}\n";
        }
    } else {
        echo "Mensaje: " . ($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/contracts/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"Contratos: {pagination['total_records']} total (página {pagination['current_page']} de {pagination['total_pages']})")
            for contrato in body:
                print(f" [{contrato['consecutivo']}] {contrato['propiedad']} - {contrato['estado']} - Canon: ${contrato['canon_total']}")
        else:
            print(f"Mensaje: {info.get('message', 'Sin resultados')}")
    else:
        print("Error al consultar los contratos.")
        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://${instancia}/service/v2/public/contracts/list?${params}`;

// 3. Función para consultar los contratos
async function consultarContratos() {
  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(`Contratos: ${pagination.total_records} total (página
${pagination.current_page} de ${pagination.total_pages})`);
      body.forEach(contrato => {
        console.log(`  [${contrato.consecutivo}] ${contrato.propiedad} -
${contrato.estado} - Canon: ${contrato.canon_total}`);
      });
    }
  }
}
```

```

        });
    } else {
        console.log(`Mensaje: ${info.message || 'Sin resultados'}`);
    }
} else {
    console.log('Error al consultar los contratos. ');
    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
consultarContratos();

```

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/contracts/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",
{"contrato_id", "consecutivo", "propiedad", "estrato_propiedad", "propietario",
"inquilino", "canon_total", "porcentaje_comision", "periodicidad", "estado",
"fecha_inicio", "fecha_fin", "fecha_terminacion", "uso", "creado_por"},
{"ContratoID", "Consecutivo", "Propiedad", "Estrato", "Propietario", "Inquilino",
"Canon", "Comision", "Periodicidad", "Estado", "FechaInicio", "FechaFin",
"FechaTerminacion", "Uso", "CreadoPor"})
in
expandido
```