

Listar Propiedades

Permite obtener una lista paginada de propiedades registradas en la inmobiliaria, incluyendo información detallada de cada una: datos generales, ubicación, valores económicos, características, propietarios, imágenes, videos y códigos de portales inmobiliarios.

¿Para qué sirve este servicio?

Úsalo para sincronizar el inventario de propiedades con tu sistema externo, alimentar un sitio web, generar reportes o integrar la información de inmuebles con plataformas de terceros. El endpoint soporta paginación y filtros por rangos de fecha para consultas eficientes.

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/properties
```

¿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 requiere autenticación mediante un Token JWT. Envía los encabezados requeridos y opcionalmente los parámetros de filtrado en la URL:

Método	GET
Content-Type	application/json

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

Autenticación requerida

Este servicio requiere un Token de autenticación válido. Debes incluir el encabezado `Authorization: Bearer TU_TOKEN` en cada petición. El token se obtiene consumiendo el servicio de **Login**. Adicionalmente, el cliente OAuth debe contar con el scope `read` para poder consumir este endpoint.

Parámetros de consulta (query string):

Parámetro	Tipo	Requerido	Por defecto	Descripción
page	integer	No	1	Número de página de resultados que se desea recuperar.
limit	integer	No	10	Número máximo de propiedades por página. Valor máximo permitido: 50 .
listing_start_date	string	No	—	Fecha de inicio del rango de consignación del inmueble. Formato: <code>YYYY-MM-DD</code> (ejemplo: <code>2024-03-01</code>).
listing_end_date	string	No	—	Fecha de fin del rango de consignación del inmueble. Formato: <code>YYYY-MM-DD</code> .
created_start_date	string	No	—	Fecha de inicio del rango de creación del inmueble en el sistema. Formato: <code>YYYY-MM-DD</code> .
created_end_date	string	No	—	Fecha de fin del rango de creación del inmueble en el sistema. Formato: <code>YYYY-MM-DD</code> .
last_modified_start_date	string	No	—	Fecha de inicio del rango de última modificación del inmueble. Formato: <code>YYYY-MM-DD</code> .

Parámetro	Tipo	Requerido	Por defecto	Descripción
<code>last_modified_end_date</code>	string	No	—	Fecha de fin del rango de última modificación del inmueble. Formato: <code>YYYY-MM-DD</code> .

Comportamiento de los filtros de fecha

Para los parámetros de tipo fecha que se componen de un rango (por ejemplo `listing_start_date` y `listing_end_date`):

- Si envías **solo el start_date**: se retornarán registros con fecha **mayor o igual (\geq)** a la indicada.
- Si envías **solo el end_date**: se retornarán registros con fecha **menor o igual (\leq)** a la indicada.
- Si envías **ambos**: se retornarán registros dentro de ese rango de fechas (BETWEEN).
- La fecha de inicio **no puede ser mayor** que la fecha de fin; de lo contrario se devolverá un error de validación.

Ejemplos de peticiones:

```
GET https://{{instancia}}/service/v2/public/properties
```

```
GET https://{{instancia}}/service/v2/public/properties?page=1&limit=10
```

```
GET
https://{{instancia}}/service/v2/public/properties?page=1&limit=10&listing_start_date=2024-03-26
```

```
GET
https://{{instancia}}/service/v2/public/properties?page=1&limit=10&listing_start_date=2024-03-01&listing_end_date=2024-03-31
```

```
GET
https://{{instancia}}/service/v2/public/properties?page=2&limit=20&created_start_date=2024-01-01&created_end_date=2024-06-30
```

3. La Respuesta (¿Qué te entregaremos?)

El sistema te devolverá una lista con las propiedades que coincidan con los filtros aplicados. Cada propiedad incluye datos generales, ubicación, valores económicos, características, propietarios, imágenes, videos y códigos de portales inmobiliarios. La respuesta se verá similar a esta:

```
[
  {
    "codigo": "137",
    "titulo": "Apartamento amplio con vista al parque central",
    "clase_id": "1247",
    "clase_inmueble": "Apartamento",
    "tipo_servicio_id": "arriendo",
    "tipo_servicio": "Arriendo",
    "estrato": "1258",
    "estrato_texto": "Cuatro",
    "fecha_consignacion": "2024-04-26",
    "asesor_id": "5",
    "asesor": "María López Ramírez",
    "pais_id": "1",
    "pais": "COLOMBIA",
    "departamento_id": "5",
    "departamento": "Antioquia",
    "municipio_id": "1",
    "municipio": "Medellin",
    "barrio_id": "3",
    "barrio": "Los Rosales",
    "direccion": "CALLE 45 # 32 - 18",
    "coordenadas": "6.2483000000000000;-75.56120000000000",
    "valor_arriendo1": "1600000",
    "valor_arriendo2": "0",
    "valor_venta1": "0",
    "valor_venta2": "0",
    "valor_administracion": "0",
    "avaluo_catastral": "0",
    "impuesto_predial": "0.00",
    "area": "75.00",
    "observaciones": null,
    "propiedad_destacada": "No",
    "llaves_en": "oficina",
    "llaves_otro": null,
  }
]
```

```
"paga_cuota_sost": "propietario",
"folio_matricula": null,
"referencia_catastral": null,
"edificio_unidad": "urbanizacion",
"estado": "1",
"estado_texto": "Activa",
"cantidad_images": "3",
"cantidad_videos": "1",
"fecha_creacion": "2024-04-26 10:30:00",
"ultima_fecha_modificacion": "2024-05-15 14:22:00",
"caracteristicas": [
  {
    "id": "1",
    "descripcion": "Nº De Habitaciones",
    "tipo_campo": "numeric",
    "orden": "1",
    "grupo": "Características del inmueble",
    "valor": "4"
  },
  {
    "id": "2",
    "descripcion": "Nº De Baños",
    "tipo_campo": "numeric",
    "orden": "2",
    "grupo": "Características del inmueble",
    "valor": "3"
  },
  {
    "id": "31",
    "descripcion": "Red de gas",
    "tipo_campo": "select",
    "orden": "5",
    "grupo": "Características del Inmueble",
    "valor": "si",
    "valor_texto": "Si"
  },
  {
    "id": "14",
    "descripcion": "Cocina Integral",
    "tipo_campo": "checkbox",
```

```
        "orden": "5",
        "grupo": "Características Internas",
        "valor": "1"
    }
],
"propietarios": [
    {
        "id": "272",
        "documento": "1020304050",
        "nombres": "ANA MARÍA",
        "apellidos": "PÉREZ RODRÍGUEZ"
    }
],
"imagenes": [
    {
        "posicion": "1",
        "size": "19201080",
        "img": "img/fotos/1920x1080_a1b2c3d4e5f6a7b8c9d0e1f2a3b4c5d6.jpeg",
        "imagen": "https://mi-
inmobiliaria.nuby.app/img/fotos/1920x1080_a1b2c3d4e5f6a7b8c9d0e1f2a3b4c5d6.jpeg"
    },
    {
        "posicion": "2",
        "size": "19201080",
        "img": "img/fotos/1920x1080_f6e5d4c3b2a1f6e5d4c3b2a1f6e5d4c3.jpeg",
        "imagen": "https://mi-
inmobiliaria.nuby.app/img/fotos/1920x1080_f6e5d4c3b2a1f6e5d4c3b2a1f6e5d4c3.jpeg"
    }
],
"videos": [
    {
        "url": "dQw4w9WgXcQ",
        "tipo": "youtube",
        "descripcion": null,
        "posicion": "1"
    }
],
"codigos_portales": [
    {
        "nombre_portal": "metrocuadrado",
```

```

        "id_portal": "MC-12345",
        "tipo_servicio": "arriendo"
    }
]
}
]

```

Campos principales de la propiedad

Esta tabla enumera las claves presentes en el JSON de cada propiedad y proporciona una descripción de cada una:

Clave	Descripción
codigo	Código único del inmueble en el sistema.
titulo	Título del anuncio del inmueble.
clase_id	Identificador de la clase de inmueble. Corresponde al <code>id</code> del servicio Listar Clases de Inmueble .
clase_inmueble	Nombre de la clase de inmueble (por ejemplo: "Apartamento", "Casa", "Finca").
tipo_servicio_id	Identificador del tipo de servicio. Valores posibles: <code>"arriendo"</code> , <code>"venta"</code> , <code>"venta y arriendo"</code> .
tipo_servicio	Nombre descriptivo del tipo de servicio (por ejemplo: "Arriendo", "Venta").
estrato	Identificador del estrato socioeconómico del inmueble.
estrato_texto	Texto descriptivo del estrato (por ejemplo: "Cuatro", "Tres").
fecha_consignacion	Fecha en que el inmueble fue captado/consignado para su gestión. Formato: <code>YYYY-MM-DD</code> .
asesor_id	Identificador del asesor asignado al inmueble. Vacío si no tiene asesor asignado.
asesor	Nombre completo del asesor asignado.
pais_id	Identificador del país donde se ubica el inmueble.
pais	Nombre del país.
departamento_id	Identificador del departamento/estado/provincia.
departamento	Nombre del departamento.
municipio_id	Identificador del municipio/ciudad.
municipio	Nombre del municipio.
barrio_id	Identificador del barrio. Puede ser <code>null</code> si no se ha asignado barrio.

Clave	Descripción
barrio	Nombre del barrio.
direccion	Dirección física del inmueble.
coordenadas	Coordenadas geográficas del inmueble en formato <code>latitud:longitud</code> .
valor_arriendo1	Valor principal de arriendo (canon mensual).
valor_arriendo2	Valor secundario de arriendo. <code>0</code> si no aplica.
valor_venta1	Valor principal de venta.
valor_venta2	Valor secundario de venta. <code>0</code> si no aplica.
valor_administracion	Valor de la cuota de administración.
avaluo_catastral	Avalúo catastral del inmueble.
impuesto_predial	Valor del impuesto predial.
area	Área del inmueble en metros cuadrados.
observaciones	Observaciones o descripción adicional del inmueble. Puede ser <code>null</code> .
propiedad_destacada	Indica si la propiedad está marcada como destacada. Valores: <code>"Si"</code> o <code>"No"</code> .
llaves_en	Lugar donde se encuentran las llaves del inmueble (por ejemplo: "oficina").
llaves_otro	Ubicación alternativa de las llaves, si aplica. Puede ser <code>null</code> .
paga_cuota_sost	Quién paga la cuota de sostenimiento (por ejemplo: "propietario"). Puede ser <code>null</code> .
folio_matricula	Número de folio de matrícula inmobiliaria. Puede ser <code>null</code> .
referencia_catastral	Referencia catastral del inmueble. Puede ser <code>null</code> .
edificio_unidad	Tipo de agrupamiento del inmueble (por ejemplo: "urbanizacion", "edificio", "conjunto").
estado	Identificador numérico del estado del inmueble. Valores: <code>0</code> (Arrendada), <code>1</code> (Activa), <code>2</code> (Inactiva), <code>3</code> (Vendida).
estado_texto	Texto descriptivo del estado (por ejemplo: "Activa", "Arrendada").
cantidad_images	Cantidad de imágenes asociadas al inmueble.
cantidad_videos	Cantidad de videos asociados al inmueble.
fecha_creacion	Fecha y hora de creación del registro en el sistema. Formato: <code>YYYY-MM-DD HH:MM:SS</code> .
ultima_fecha_modificacion	Fecha y hora de la última modificación del registro. Formato: <code>YYYY-MM-DD HH:MM:SS</code> .

Clave	Descripción
caracteristicas	Lista de características del inmueble (ver tabla detallada abajo).
propietarios	Lista de propietarios del inmueble (ver tabla detallada abajo).
imagenes	Lista de imágenes del inmueble (ver tabla detallada abajo).
videos	Lista de videos del inmueble (ver tabla detallada abajo).
codigos_portales	Lista de códigos de publicación en portales inmobiliarios externos (ver tabla detallada abajo).

Características

Cada elemento dentro de la lista `caracteristicas` contiene la información de una característica del inmueble y su valor asignado:

Clave	Descripción
id	Identificador de la característica. Corresponde al <code>id</code> del servicio Listar Características .
descripcion	Nombre descriptivo de la característica (por ejemplo: "Nº De Habitaciones").
tipo_campo	Tipo de campo: <code>numeric</code> , <code>checkbox</code> , <code>select</code> , entre otros.
orden	Posición de la característica dentro de su grupo.
grupo	Grupo al que pertenece la característica.
valor	Valor asignado a la característica para esta propiedad. Para <code>numeric</code> : un número. Para <code>checkbox</code> : "1" (marcado). Para <code>select</code> : el valor técnico de la opción seleccionada.
valor_texto	Descripción legible del valor seleccionado. Solo presente cuando <code>tipo_campo</code> es <code>select</code> (por ejemplo: valor <code>"si"</code> → <code>valor_texto</code> <code>"Si"</code>).

Propietarios

Cada elemento dentro de la lista `propietarios` contiene la información de un propietario del inmueble:

Clave	Descripción
id	Identificador del propietario (tercero) en el sistema.
documento	Número de documento de identificación del propietario.
nombres	Nombres del propietario.
apellidos	Apellidos del propietario.

Imágenes

Cada elemento dentro de la lista `imagenes` contiene la información de una fotografía del inmueble:

Clave	Descripción
<code>posicion</code>	Posición de ordenamiento de la imagen (1 = principal).
<code>size</code>	Resolución de la imagen (por ejemplo: "19201080").
<code>img</code>	Ruta relativa de la imagen en el servidor.
<code>imagen</code>	URL completa de la imagen, lista para consumir directamente.

Videos

Cada elemento dentro de la lista `videos` contiene la información de un video asociado al inmueble:

Clave	Descripción
<code>url</code>	Identificador o URL del video (por ejemplo, el ID de YouTube: <code>"608AV8w6gL0"</code>).
<code>tipo</code>	Plataforma del video (por ejemplo: <code>"youtube"</code>).
<code>descripcion</code>	Descripción del video. Puede ser <code>null</code> .
<code>posicion</code>	Posición de ordenamiento del video.

Códigos de Portales

Cada elemento dentro de la lista `codigos_portales` contiene la información de publicación del inmueble en un portal inmobiliario externo:

Clave	Descripción
<code>nombre_portal</code>	Nombre del portal inmobiliario (por ejemplo: "metrocuadrado", "fincaraiz").
<code>id_portal</code>	Identificador de la propiedad en el portal externo.
<code>tipo_servicio</code>	Tipo de servicio bajo el cual fue publicada la propiedad en ese portal.

4. Seguridad y Posibles Errores

El sistema realiza validaciones de autenticación, scopes y parámetros. Si alguna falla, devolverá un error con su respectivo código HTTP:

Código HTTP	Descripción
400	<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. — <code>limit</code> no es numérico o es mayor a 50. — Las fechas no tienen el formato <code>YYYY-MM-DD</code>. — La fecha de inicio es mayor que la fecha de fin en un rango.
401	<p>El token ha expirado. Debes generar uno nuevo consumiendo el servicio de Login. Mensaje: <code>"JWT Token expired."</code></p>
403	<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>

5. Ejemplos de integración

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

cURL

```
# Listar propiedades con paginación
curl -X GET "https://{{instancia}}/service/v2/public/properties?page=1&limit=10" \
-H "Content-Type: application/json" \
-H "Authorization: Bearer TU_TOKEN_AQUI"

# Filtrar por rango de fecha de consignación
curl -X GET
"https://{{instancia}}/service/v2/public/properties?page=1&limit=10&listing_start_date=202
4-03-01&listing_end_date=2024-03-31" \
-H "Content-Type: application/json" \
-H "Authorization: Bearer TU_TOKEN_AQUI"

# Filtrar por fecha de creación
```

```
curl -X GET
"https://{{instancia}}/service/v2/public/properties?page=1&limit=20&created_start_date=202
4-01-01" \
-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 consulta
$page = 1;
$limit = 10;
$listingStartDate = '2024-03-01';
$listingEndDate = '2024-03-31';

$queryParams = http_build_query([
    'page' => $page,
    'limit' => $limit,
    'listing_start_date' => $listingStartDate,
    'listing_end_date' => $listingEndDate
]);

$url = "https://{$instance}/service/v2/public/properties?{$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";

    if ($http_code === 200) {
        $propiedades = json_decode($response, true);
        echo "Propiedades encontradas: " . count($propiedades) . "\n";
        foreach ($propiedades as $propiedad) {
            echo " [{$propiedad['codigo']}] {$propiedad['titulo']} -
{$propiedad['estado_texto']}\n";
        }
    } else {
        echo "Error al consultar las propiedades.\n";
        echo $response . "\n";
    }
}
curl_close($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/properties"

params = {
    "page": 1,
    "limit": 10,
    "listing_start_date": "2024-03-01",

```

```
    "listing_end_date": "2024-03-31"
}

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:
        propiedades = response.json()
        print(f"Propiedades encontradas: {len(propiedades)}")
        for prop in propiedades:
            print(f"  [{prop['codigo']}] {prop['titulo']} - {prop['estado_texto']}")
    else:
        print("Error al consultar las propiedades.")
        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,
    limit: 10,
    listing_start_date: '2024-03-01',
```

```

    listing_end_date: '2024-03-31'
  });

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

// 3. Función para consultar las propiedades
async function consultarPropiedades() {
  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 propiedades = await response.json();
      console.log(`Propiedades encontradas: ${propiedades.length}`);
      propiedades.forEach(prop => {
        console.log(` [${prop.codigo}] ${prop.titulo} - ${prop.estado_texto}`);
      });
    } else {
      console.log('Error al consultar las propiedades. ');
      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
consultarPropiedades();

```

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/properties?page=1&limit=50",

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

    // 4. Decodifica la respuesta JSON y conviértela en tabla
    jsonResponse = Json.Document(response),
    tabla = Table.FromList(jsonResponse, Splitter.SplitByNothing(), null, null,
ExtraValues.Error),
    expandido = Table.ExpandRecordColumn(tabla, "Column1",
        {"codigo", "titulo", "clase_inmueble", "tipo_servicio", "estado_texto",
"direccion", "municipio", "departamento", "area", "valor_arriendo1", "valor_venta1",
"fecha_consignacion"},
        {"Codigo", "Titulo", "ClaseInmueble", "TipoServicio", "Estado", "Direccion",
"Municipio", "Departamento", "Area", "ValorArriendo", "ValorVenta", "FechaConsignacion"})
in
    expandido

```

Revisión #26

Creado el 24 mayo 2024 14:51:54

Actualizado el 16 abril 2026 17:35:51 por Isabel Higuita