

Propiedades

Bienvenido al **inventario digital** de tu inmobiliaria. En esta sección te explicamos cómo conectar tus herramientas externas para gestionar, consultar y mantener sincronizada toda la información de tus inmuebles.

¿Qué puedes hacer en este módulo?

1. Exploración y consulta detallada: Extrae tu catálogo completo utilizando filtros estratégicos (como fechas de creación o actualización) o profundiza en los datos de una propiedad específica. Podrás acceder a sus características precisas, material multimedia (fotografías y videos) y la información de sus propietarios.

2. Gestión segura de estados: ¡No solo puedes leer información, también puedes tomar acción! Mantén tu inventario al día actualizando la disponibilidad de tus inmuebles. Para tu tranquilidad, nuestro sistema cuenta con validaciones inteligentes que protegen tu información y previenen errores accidentales (por ejemplo, bloqueando cambios automáticos si la propiedad ya figura como "Arrendada").

¡Explora las opciones a continuación y sácale el máximo provecho a la gestión automatizada de tu portafolio!

- [Listar Propiedades](#)
- [Buscar Propiedad por Código](#)
- [Actualizar Estado de la Propiedad](#)
- [Crear Propiedad](#)

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
last_modified_end_date	string	No	—	Fecha de fin del rango de última modificación del inmueble. Formato: YYYY-MM-DD.

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 (>=)** a la indicada.
- Si envías **solo el end_date**: se retornarán registros con fecha **menor o igual (<=)** 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": "6.248300000000000:-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_alb2c3d4e5f6a7b8c9d0e1f2a3b4c5d6.jpeg",
        "imagen": "https://mi-
inmobiliaria.nuby.app/img/fotos/1920x1080_alb2c3d4e5f6a7b8c9d0e1f2a3b4c5d6.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> : <code>"1"</code> (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> → valor_texto <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	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. — <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	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>

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=2024-01-01" \
-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
```

Buscar Propiedad por Código

Permite obtener la información completa de una propiedad específica a partir de su código único. La respuesta incluye datos generales, ubicación, valores económicos, características, propietarios, imágenes, videos y códigos de portales inmobiliarios.

¿Para qué sirve este servicio?

Úsalo cuando ya conoces el código del inmueble y necesitas consultar toda su información detallada. Es ideal para mostrar la ficha completa de una propiedad en tu sitio web, sincronizar datos de un inmueble puntual o verificar la información registrada en el sistema.

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, y **{{code}}** por el código numérico del inmueble que desees consultar.

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

¿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. Construye la URL con el código de la propiedad como parte de la ruta e incluye los encabezados requeridos:

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.

Parámetro de ruta:

Parámetro	Tipo	Requerido	Descripción
code	integer	Sí	Código numérico único del inmueble que se desea consultar. Debe ser un número entero positivo.

Ejemplos de peticiones:

```
GET https://{{instancia}}/service/v2/public/properties/137
```

```
GET https://{{instancia}}/service/v2/public/properties/2045
```

3. La Respuesta (¿Qué te entregaremos?)

El sistema te devolverá un **objeto JSON** con toda la información de la propiedad solicitada. Si el código no corresponde a ningún inmueble registrado, se devolverá un arreglo vacío `[]`.

Respuesta como objeto, no como lista

A diferencia del servicio **Listar Propiedades** que devuelve un arreglo de objetos `[{...}, {...}]`, este servicio devuelve directamente un único objeto `{...}` con la información del inmueble consultado.

La respuesta exitosa 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": "6.248300000000000:-75.56120000000000",
"valor_arriendo1": "1600000",
"valor_arriendo2": "0",
"valor_venta1": "0",
"valor_venta2": "0",
"valor_administracion": "250000",
"avaluo_catastral": "185000000",
"impuesto_predial": "1250000.00",
"area": "75.00",
"observaciones": "Inmueble ubicado en zona residencial tranquila, cerca de centros
comerciales y transporte público.",
"propiedad_destacada": "No",
"llaves_en": "oficina",
"llaves_otro": null,
"paga_cuota_sost": "propietario",
"folio_matricula": "001-123456",
"referencia_catastral": "05001010203040",
"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": "5",
    "descripcion": "Nº De Piso",
    "tipo_campo": "numeric",
    "orden": "3",
    "grupo": "Características del inmueble",
    "valor": "1"
  },
  {
    "id": "4",
    "descripcion": "Antigüedad del Inmueble",
    "tipo_campo": "numeric",
    "orden": "5",
    "grupo": "Características del Inmueble",
    "valor": "3"
  },
  {
    "id": "14",
    "descripcion": "Cocina Integral",
    "tipo_campo": "checkbox",
    "orden": "5",
    "grupo": "Características Internas",
```

```
        "valor": "1"
    },
    {
        "id": "34",
        "descripcion": "Sala",
        "tipo_campo": "checkbox",
        "orden": "6",
        "grupo": "Características del Inmueble",
        "valor": "1"
    },
    {
        "id": "31",
        "descripcion": "Red de gas",
        "tipo_campo": "select",
        "orden": "5",
        "grupo": "Características del Inmueble",
        "valor": "si",
        "valor_texto": "Si"
    },
    {
        "id": "63",
        "descripcion": "Garaje",
        "tipo_campo": "checkbox",
        "orden": "10",
        "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"
    },
    {
        "posicion": "3",
        "size": "19201080",
        "img": "img/fotos/1920x1080_b3c4d5e6f7a8b9c0d1e2f3a4b5c6d7e8.jpeg",
        "imagen": "https://mi-
inmobiliaria.nuby.app/img/fotos/1920x1080_b3c4d5e6f7a8b9c0d1e2f3a4b5c6d7e8.jpeg"
    }
],
"videos": [
    {
        "url": "dQw4w9WgXcQ",
        "tipo": "youtube",
        "descripcion": null,
        "posicion": "1"
    }
],
"codigos_portales": [
    {
        "nombre_portal": "metrocuadrado",
        "id_portal": "MC-98765",
        "tipo_servicio": "arriendo"
    }
]
}

```

Cuando el código no corresponde a ningún inmueble, la respuesta será:

```
[ ]
```

Campos principales de la propiedad

Esta tabla enumera las claves presentes en el JSON de la 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. Cadena vacía 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.
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.

Clave	Descripción
valor_venta2	Valor secundario de venta. 0 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 null .
propiedad_destacada	Indica si la propiedad está marcada como destacada. Valores: Si o No .
llaves_en	Lugar donde se encuentran las llaves del inmueble (por ejemplo: "oficina").
llaves_otro	Ubicación alternativa de las llaves, si aplica. Puede ser null .
paga_cuota_sost	Quién paga la cuota de sostenimiento (por ejemplo: "propietario"). Puede ser null .
folio_matricula	Número de folio de matrícula inmobiliaria. Puede ser null .
referencia_catastral	Referencia catastral del inmueble. Puede ser null .
edificio_unidad	Tipo de agrupamiento del inmueble (por ejemplo: "urbanizacion", "edificio", "conjunto").
estado	Identificador numérico del estado del inmueble. Valores: 0 (Arrendada), 1 (Activa), 2 (Inactiva), 3 (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: YYYY-MM-DD HH:MM:SS .
ultima_fecha_modificacion	Fecha y hora de la última modificación del registro. Formato: YYYY-MM-DD HH:MM:SS .
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
<code>id</code>	Identificador de la característica. Corresponde al <code>id</code> del servicio Listar Características .
<code>descripcion</code>	Nombre descriptivo de la característica (por ejemplo: "Nº De Habitaciones").
<code>tipo_campo</code>	Tipo de campo: <code>numeric</code> , <code>checkbox</code> , <code>select</code> , entre otros.
<code>orden</code>	Posición de la característica dentro de su grupo.
<code>grupo</code>	Grupo al que pertenece la característica.
<code>valor</code>	Valor asignado a la característica para esta propiedad. Para <code>numeric</code> : un número. Para <code>checkbox</code> : <code>"1"</code> (marcado). Para <code>select</code> : el valor técnico de la opción seleccionada.
<code>valor_texto</code>	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
<code>id</code>	Identificador del propietario (tercero) en el sistema.
<code>documento</code>	Número de documento de identificación del propietario.
<code>nombres</code>	Nombres del propietario.
<code>apellidos</code>	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.

Clave	Descripción
imagen	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
url	Identificador o URL del video (por ejemplo, el ID de YouTube: <code>"dQw4w9WgXcQ"</code>).
tipo	Plataforma del video (por ejemplo: <code>"youtube"</code>).
descripcion	Descripción del video. Puede ser <code>null</code> .
posicion	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
nombre_portal	Nombre del portal inmobiliario (por ejemplo: "metrocuadrado", "fincaraiz").
id_portal	Identificador de la propiedad en el portal externo.
tipo_servicio	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 de ruta:

Cuando el código es numérico pero no corresponde a ningún inmueble registrado, el sistema **no devuelve un error**. En su lugar, responde con un HTTP `200` y un arreglo vacío:

```
[ ]
```

Importante

Asegúrate de validar en tu integración si la respuesta es un arreglo vacío `[ ]`

para manejar correctamente el caso en que la propiedad no exista.

Código HTTP	Descripción
200	Respuesta exitosa. Contiene el objeto JSON de la propiedad, o un arreglo vacío [] si no se encontró ningún inmueble con ese código.
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>
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>
404	La ruta no fue encontrada. Ocurre cuando el valor de <code>code</code> no es numérico (por ejemplo: <code>/properties/abc</code>).

5. Ejemplos de integración

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

cURL

```
# Consultar propiedad por código
curl -X GET "https://{{instancia}}/service/v2/public/properties/137" \
-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
$code = 137; // Código de la propiedad a consultar

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

$ch = curl_init($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) {
        $propiedad = json_decode($response, true);

        if (!empty($propiedad)) {
            echo "Propiedad encontrada:\n";
            echo "  Código: { $propiedad['codigo'] }\n";
            echo "  Título: { $propiedad['titulo'] }\n";
            echo "  Estado: { $propiedad['estado_texto'] }\n";
            echo "  Dirección: { $propiedad['direccion'] }\n";
            echo "  Imágenes: { $propiedad['cantidad_images'] }\n";
        } else {
            echo "No se encontró ninguna propiedad con el código { $code }.\n";
        }
    } else {
        echo "Error al consultar la propiedad.\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
codigo = 137 # Código de la propiedad a consultar

# 2. Prepara la dirección de la petición
url = f"https://{instancia}/service/v2/public/properties/{codigo}"

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

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

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

    if response.status_code == 200:
        propiedad = response.json()

        if propiedad:
            print(f"Propiedad encontrada:")
            print(f"  Código: {propiedad['codigo']}")
            print(f"  Título: {propiedad['titulo']}")
            print(f"  Estado: {propiedad['estado_texto']}")
            print(f"  Dirección: {propiedad['direccion']}")
            print(f"  Características: {len(propiedad['caracteristicas'])}")
```

```
        else:
            print(f"No se encontró ninguna propiedad con el código {codigo}.")
    else:
        print("Error al consultar la propiedad.")
        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
const codigo = 137; // Código de la propiedad a consultar

// 2. Prepara la dirección de la petición
const url = `https://${instancia}/service/v2/public/properties/${codigo}`;

// 3. Función para consultar la propiedad
async function consultarPropiedad() {
    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 propiedad = await response.json();

            if (propiedad && !Array.isArray(propiedad)) {
                console.log(`Propiedad encontrada:`);
            }
        }
    }
}
```

```

        console.log(`  Código: ${propiedad.codigo}`);
        console.log(`  Título: ${propiedad.titulo}`);
        console.log(`  Estado: ${propiedad.estado_texto}`);
        console.log(`  Dirección: ${propiedad.direccion}`);
        console.log(`  Imágenes: ${propiedad.cantidad_images}`);
    } else {
        console.log(`No se encontró ninguna propiedad con el código ${codigo}.`);
    }
} else {
    console.log('Error al consultar la propiedad.');
```

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
consultarPropiedad();
```

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
    codigo = "137", // Código de la propiedad a consultar

    // 2. Prepara la dirección de la petición
    url = "https://" & instancia & "/service/v2/public/properties/" & codigo,

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



```
        1
    1),

    // 4. Decodifica la respuesta JSON y conviértela en registro
    jsonResponse = Json.Document(response),
    resultado = Record.ToTable(jsonResponse),
    expandido = Table.Pivot(resultado, List.Distinct(resultado[Name]), "Name", "Value")
in
    expandido
```

Actualizar Estado de la Propiedad

Permite actualizar el estado de una propiedad específica a partir de su código único. Este servicio es útil para cambiar el estado de un inmueble entre los estados permitidos por el sistema.

¿Para qué sirve este servicio?

Úsalo cuando necesites cambiar el estado de un inmueble desde tu sistema externo, por ejemplo: marcar una propiedad como inactiva cuando se retira del mercado, reactivarla cuando vuelve a estar disponible, o registrar que fue vendida.

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, y `{{code}}` por el código numérico del inmueble cuyo estado deseas actualizar.

```
PATCH https://{{instancia}}/service/v2/public/properties/{{code}}/status
```

¿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?)

Debes enviar una petición **PATCH** con el nuevo estado en el cuerpo de la solicitud en formato JSON:

Método	PATCH
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 `update` para poder consumir este endpoint.

Parámetro de ruta:

Parámetro	Tipo	Requerido	Descripción
code	integer	Sí	Código numérico único del inmueble cuyo estado se desea actualizar.

Cuerpo de la petición (Body):

Campo	Tipo	Requerido	Descripción
status	integer	Sí	Nuevo estado que se desea asignar a la propiedad. Los valores válidos son: 1 = Activa 2 = Inactiva 3 = Vendida

Restricción importante sobre el estado "Arrendada" (0)

Aunque el estado `0` (Arrendada) existe en el sistema, **no es posible asignar este estado a través de la API**. El estado "Arrendada" se gestiona internamente por el sistema cuando se registra un contrato de arrendamiento asociado al inmueble.

Estados válidos del sistema

ID	Estado	Descripción
0	Arrendada	El inmueble tiene un contrato de arrendamiento vigente. No se puede asignar vía API.

ID	Estado	Descripción
1	Activa	El inmueble está disponible en el mercado para arriendo o venta.
2	Inactiva	El inmueble fue retirado temporalmente del mercado.
3	Vendida	El inmueble fue vendido.

Reglas de transición de estado

El sistema solo permite cambiar el estado de una propiedad cuando su estado actual es **Activa (1)** o **Inactiva (2)**. Las transiciones posibles son:

Estado actual	Puede cambiar a
Activa (1)	2 (Inactiva) o 3 (Vendida)
Inactiva (2)	1 (Activa) o 3 (Vendida)
Arrendada (0)	No se permite cambiar el estado vía API.
Vendida (3)	No se permite cambiar el estado vía API.

¿Por qué no se puede modificar el estado de una propiedad "Arrendada" o "Vendida"?

Estos son estados terminales que representan transacciones completadas. El cambio de estos estados se gestiona internamente a través de los procesos de negocio del sistema (por ejemplo, al finalizar un contrato de arrendamiento o revertir una venta).

Ejemplo de petición:

```
PATCH https://{instancia}/service/v2/public/properties/137/status

{
  "status": 2
}
```

3. La Respuesta (¿Qué te entregaremos?)

Si la actualización fue exitosa, el sistema responderá con HTTP 200 y un mensaje de confirmación:

```
{
  "message": "Se actualizo el estado de la propiedad correctamente"
}
```

Clave	Descripción
message	Mensaje indicando que el estado de la propiedad fue actualizado correctamente.

4. Seguridad y Posibles Errores

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

Código HTTP	Descripción	Ejemplo de respuesta
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. El campo <code>status</code> no fue enviado o no es numérico.	<pre>{ "error": "El estado(status) es requerido" }</pre>
401	El token ha expirado. Debes generar uno nuevo consumiendo el servicio de Login .	<code>"JWT Token expired."</code>
403	El cliente OAuth no tiene el scope necesario. Para endpoints PATCH se requiere el scope <code>update</code> .	<code>"Insufficient scope. Required: 'update', granted: '{scope_actual}'."</code>
404	No se encontró ninguna propiedad con el código indicado.	<pre>{ "error": "No se encontró la propiedad" }</pre>

Código HTTP	Descripción	Ejemplo de respuesta
422	El valor de <code>status</code> no corresponde a un estado válido del sistema (no es 0, 1, 2 o 3).	<pre>{ "error": "El estado no es válido" }</pre>
422	El estado actual de la propiedad no permite la modificación (por ejemplo: la propiedad está en estado "Arrendada" o "Vendida").	<pre>{ "error": "El estado actual de la propiedad no permite este tipo de modificación." }</pre>
500	Error interno del servidor. No se pudo ejecutar la actualización en la base de datos.	<pre>{ "error": "No se pudo actualizar el estado de la propiedad" }</pre>

5. Ejemplos de integración

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

cURL

```
# Cambiar estado de propiedad 137 a Inactiva (2)
curl -X PATCH "https://{{instancia}}/service/v2/public/properties/137/status" \
-H "Content-Type: application/json" \
-H "Authorization: Bearer TU_TOKEN_AQUI" \
-d '{"status": 2}'

# Cambiar estado de propiedad 2045 a Vendida (3)
curl -X PATCH "https://{{instancia}}/service/v2/public/properties/2045/status" \
-H "Content-Type: application/json" \
```

```
-H "Authorization: Bearer TU_TOKEN_AQUI" \  
-d '{"status": 3}'  
  
# Reactivar propiedad 890 (de Inactiva a Activa)  
curl -X PATCH "https://{{instancia}}/service/v2/public/properties/890/status" \  
-H "Content-Type: application/json" \  
-H "Authorization: Bearer TU_TOKEN_AQUI" \  
-d '{"status": 1}'
```

PHP

```
<?php  
  
$instance = 'tu_instancia'; // Reemplaza con tu instancia real  
$token = 'TU_TOKEN_AQUI'; // Token obtenido del servicio Login  
$code = 137; // Código de la propiedad  
$nuevoEstado = 2; // 1 = Activa, 2 = Inactiva, 3 = Vendida  
  
$url = "https://{$instance}/service/v2/public/properties/{$code}/status";  
  
$body = json_encode([  
    'status' => $nuevoEstado  
]);  
  
$ch = curl_init($url);  
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);  
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'PATCH');  
curl_setopt($ch, CURLOPT_POSTFIELDS, $body);  
curl_setopt($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";

        $data = json_decode($response, true);

        if ($http_code === 200) {
            echo "Éxito: " . $data['message'] . "\n";
        } else {
            echo "Error: " . ($data['error'] ?? 'Error desconocido') . "\n";
        }
    }
    curl_close($ch);

?>

```

Python

```

import requests
import json

# 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
codigo = 137 # Código de la propiedad
nuevo_estado = 2 # 1 = Activa, 2 = Inactiva, 3 = Vendida

# 2. Prepara la dirección y el cuerpo de la petición
url = f"https://{instancia}/service/v2/public/properties/{codigo}/status"

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

body = {
    "status": nuevo_estado
}

```



```

}

# 3. Envía la petición PATCH y procesa la respuesta
try:
    response = requests.patch(url, headers=headers, json=body)

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

    data = response.json()

    if response.status_code == 200:
        print(f"Éxito: {data['message']}")
    else:
        error_msg = data.get('error', 'Error desconocido')
        print(f"Error: {error_msg}")

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
const codigo = 137; // Código de la propiedad
const nuevoEstado = 2; // 1 = Activa, 2 = Inactiva, 3 = Vendida

// 2. Prepara la dirección de la petición
const url = `https://${instancia}/service/v2/public/properties/${codigo}/status`;

// 3. Función para actualizar el estado
async function actualizarEstadoPropiedad() {
    try {
        const response = await fetch(url, {
            method: 'PATCH',
            headers: {
                'Content-Type': 'application/json',

```

```

        'Authorization': `Bearer ${token}`
    },
    body: JSON.stringify({
        status: nuevoEstado
    })
});

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

const data = await response.json();

if (response.ok) {
    console.log(`Éxito: ${data.message}`);
} else {
    console.log(`Error: ${data.error || 'Error desconocido'}`);
}
} catch (error) {
    console.error(`Ocurrió un error de conexión: ${error}`);
}
}

// 4. Ejecutamos la función
actualizarEstadoPropiedad();

```

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
    codigo = "137", // Código de la propiedad
    nuevoEstado = 2, // 1 = Activa, 2 = Inactiva, 3 = Vendida

    // 2. Prepara la dirección y el cuerpo de la petición
    url = "https://" & instancia & "/service/v2/public/properties/" & codigo & "/status",
    body = Json.FromValue([status = nuevoEstado]),

```

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

// 4. Decodifica la respuesta JSON
jsonResponse = Json.Document(response)
in
jsonResponse
```

Crear Propiedad

Permite registrar una nueva propiedad en el sistema de forma integral. A través de este servicio, es posible guardar la información básica de la propiedad, asociar propietarios (existentes o creados dinámicamente en la misma petición), registrar características personalizadas, coordenadas geográficas, videos de recorridos, fotos del inmueble y relacionar otras propiedades.

¿Para qué sirve este servicio?

Úsalo cuando necesites automatizar el ingreso de propiedades a nuby desde portales externos, aplicaciones móviles o sistemas de captación propios. Este endpoint está diseñado bajo un modelo transaccional robusto, asegurando que si algún paso crítico falla (por ejemplo, validación de un propietario o falta de características obligatorias del tipo de propiedad), toda la operación se cancele de forma automática para evitar datos inconsistentes en tu base de datos.

1. El Endpoint (La dirección web)

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

```
POST 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 estructura la petición con un cuerpo (Body) en formato JSON que contenga las secciones detalladas a continuación:

Método	POST
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 `write` para poder consumir este endpoint.

Cuerpo de la petición (JSON Body - Estructura General)

La raíz del JSON enviado debe estructurarse con las siguientes llaves principales:

Elemento Raíz	Tipo	Requerido	Descripción
Propiedad	object	Sí	Contiene la información básica del inmueble (Título, tipo, ubicación, valores, etc.).
Propietarios	array	Sí	Lista de propietarios asociados. Puede ser un arreglo de IDs existentes o de objetos con datos de nuevos propietarios.
Características	object	No	Mapeo de características de la propiedad en formato llave-valor <code>{"id_caracteristica": "valor"}</code> .
Videos	array	No	Listado de enlaces a videos del inmueble.
Fotos	array	No	Listado de imágenes del inmueble en formato base64 o URLs públicas.
PropiedadesRelacionadas	array	No	Arreglo de códigos numéricos de propiedades que se desea relacionar de forma recíproca.

Elemento Raíz	Tipo	Requerido	Descripción
Coordenadas	string object	No	Coordenadas de latitud y longitud. Ejemplo de texto: "6.2089, -75.5678".

Estructura Detallada de las Secciones

2.1 Sección "Propiedad" ?

Contiene la información comercial y técnica base del inmueble. Los siguientes campos son validados estrictamente bajo validación rápida (Fail-Fast):

Campo	Tipo	Requerido	Descripción
txtTitulo	string	Sí	Título descriptivo o publicitario del inmueble. Máximo 255 caracteres.
tipo_id	integer	Sí	ID de la clase de propiedad (Ej: apartamento, casa, local). Debe ser una clase existente en la maestra de clases de inmueble.
propiedad_tipo	string	Sí	Modalidad de comercialización. Valores válidos: "arriendo", "venta", o "venta y arriendo".
municipio_id	integer	Sí	ID de la ciudad o municipio donde se ubica el inmueble (debe existir en la maestra de municipios).
direccion	string	Sí	Dirección exacta del inmueble. El sistema validará que no exista otra propiedad con la misma dirección en el mismo municipio para evitar duplicados accidentales.
estrato	integer	Sí	Estrato socioeconómico del inmueble. Debe ser un número entero entre 1 y 6.
barrio_id	integer	No	ID del barrio (debe existir en la maestra de barrios de la ciudad).
urbanizacion	string	No	Nombre del edificio, conjunto cerrado o urbanización.

Campo	Tipo	Requerido	Descripción
valor_arriendo	numeric	No	Canon mensual del arriendo (monto mayor o igual a 0).
valor_venta	numeric	No	Precio de venta solicitado (monto mayor o igual a 0).
valor_administracion	numeric	No	Costo de la cuota de administración de la copropiedad.
propiedad_area	numeric	No	Área privada del inmueble en metros cuadrados. Se envía estrictamente con el nombre físico de columna.
observaciones	string	No	Descripción comercial extensa del inmueble.
llaves_en	string	No	Ubicación física de las llaves. Valores válidos: "oficina", "propiedad", o "otro".
paga_cuota_sost	string	No	Establece quién asume el pago de la cuota de administración. Valores válidos: "propietario" o "inquilino".
folio_matricula	string	No	Número de matrícula inmobiliaria del inmueble.
referencia_catastral	string	No	Número de referencia catastral asignado por el municipio.

2.2 Sección "Propietarios" ?

Cada propiedad en nuby requiere obligatoriamente tener asociado al menos un propietario. El arreglo acepta dos formatos de datos, los cuales pueden combinarse:

Formato A: Propietario Existente (Por ID)

Si el propietario ya se encuentra registrado en el ERP nuby, simplemente envía su identificador único (ID de tercero) como un entero dentro del arreglo:

```
"Propietarios": [1528, 4390]
```

Formato B: Creación Dinámica de Propietario (Objeto de Datos)

Si el propietario no existe, puedes enviar sus datos completos estructurados en un objeto. El sistema validará su información, lo creará automáticamente en la base de datos de nuby, y lo asociará al inmueble bajo una misma transacción comercial:

Campo	Tipo	Requerido	Descripción
persona	integer	Sí	Naturaleza jurídica del tercero: 1 = Persona Natural, 2 = Persona Jurídica.
documento	string	Sí	Número de documento de identidad o NIT.
tipo_doc_id	integer	Sí	ID de la maestra de tipos de documentos (Cédula, NIT, Pasaporte, etc.).
nombre1	string	Sí	Primer nombre del propietario o razón social completa.
apellido1	string	Sí (para Natural)	Primer apellido del propietario (requerido si persona = 1).
direccion1	string	Sí	Dirección de domicilio del propietario.
municipio_id	integer	Sí	ID del municipio de residencia (debe existir en la base de datos).
email	string	Sí	Correo electrónico para envío de facturas and notificaciones. Debe tener formato de email válido.
telefono	string	Sí	Número telefónico o celular de contacto.
tipo_persona_id	integer	Sí (para Natural)	ID de tipo de persona tributaria (requerido si persona = 1).
regimen_id	integer	Sí (para Natural)	ID del régimen tributario del propietario (requerido si persona = 1).
resp_fiscal_id	integer	Sí (para Natural)	ID de la responsabilidad fiscal (requerido si persona = 1).

2.3 Sección "Características" ??

Permite registrar valores para las características personalizadas del inmueble. Se envía en formato de objeto clave-valor, donde la clave es el ID numérico de la característica y el valor es el contenido a asignar:

```
"Caracteristicas": {
  "1": "3",    // ID 1 (Habitaciones) = 3
  "2": "2",    // ID 2 (Baños) = 2
  "31": "si",  // ID 31 (Red de gas) = si (Obligatoria)
  "63": "1"    // ID 63 (Garaje) = 1 (Checkbox de Garaje activo)
}
```

Validación de Características Obligatorias por Tipo

El motor de validación del sistema nuby verificará qué características están configuradas en el ERP como **obligatorias** para el tipo de propiedad seleccionado (`tipo_id`). Si omites alguna de estas características obligatorias en la petición, o el formato del valor es erróneo (por ejemplo, enviar texto en un campo netamente numérico), el endpoint devolverá un código de estado `422 Unprocessable Entity` y se revertirá todo el proceso de guardado.

Nota: En una configuración estándar, el apartamento (`1247`) y la casa (`1249`) exigen de forma obligatoria las características de: N° de Habitaciones (ID 1), N° de Baños (ID 2) y Red de gas (ID 31).

2.4 Sección "Fotos" ?

Permite adjuntar imágenes del inmueble de forma directa. El procesamiento de imágenes está automatizado y realiza optimizaciones de rendimiento y de presentación visual (redimensionamiento, corrección de orientación por metadatos EXIF, e inserción de la marca de agua corporativa configurada en el sistema).

Campo	Tipo	Requerido	Descripción
url	string	Sí	Contenido de la imagen. Puede enviarse como una cadena Base64 válida (<code>data:image/jpeg;base64,...</code>) o como una URL accesible públicamente para su descarga.
nombre	string	No	Nombre del archivo o descripción física para almacenar el documento.

2.5 Sección "Videos" ?

Permite registrar recorridos en video de las propiedades (por ejemplo, cargados en plataformas como YouTube o Vimeo):

Campo	Tipo	Requerido	Descripción
video_url	string	Sí	URL completa o código físico del video. Ejemplo: <code>"https://www.youtube.com/watch?v=..."</code> .
video_tipo	string	No	Tipo o plataforma emisora del video (Ej: <code>"youtube"</code> o <code>"vimeo"</code>).
video_descripcion	string	No	Descripción del video.
video_posicion	integer	No	Posición de visualización de este video.

2.6 Sección "PropiedadesRelacionadas" y "Coordenadas" ?

- **PropiedadesRelacionadas** → Permite enlazar propiedades entre sí (ideal para bodegas subdivididas, oficinas del mismo centro de negocios, etc.). Recibe una lista de códigos de inmuebles: `[137, 1042]`. El sistema nuby creará una asociación **recíproca**: la propiedad creada apuntará a estas asociadas, y estas asociadas apuntarán automáticamente a la nueva propiedad.
- **Coordenadas** → Coordenadas de geolocalización. Se recomienda enviar como una cadena de texto separada por comas `"latitud,longitud"` (Ej: `"6.2089, -75.5678"`). El motor de nuby se encarga de analizar la cadena, validar su rango físico geográfico y guardarlo bajo el formato propietario del sistema.

3. La Respuesta (¿Qué te entregaremos?)

Esta sección describe la respuesta que recibirás del sistema cuando la creación del inmueble sea completamente exitosa.

Respuesta Exitosa (201 Created)

Se genera cuando la propiedad y todos sus componentes asociados han sido validados e insertados con éxito. Se retorna el identificador único físico asignado al inmueble:

```
{
  "error": false,
  "type": "success",
  "msg": "La propiedad ha sido creada exitosamente.",
}
```

```
"propiedad_id": 7786,  
"warnings": []  
}
```

¿Qué contiene la sección "warnings"?

Si se incluyen fotos o videos y el almacenamiento en la nube o procesamiento de marcas de agua tiene algún inconveniente no letal (por ejemplo, formato no soportado o redimensión fallida de una imagen en particular), el inmueble **se creará con éxito**, pero se devolverán advertencias en esta sección para que puedas corregir los archivos de forma manual. Esto no interrumpe el registro del inmueble.

4. Seguridad y Posibles Errores

El sistema realiza validaciones de autenticación, permisos (scopes) y estructura de datos. Si alguna falla, devolverá un error con su respectivo código HTTP y un mensaje descriptivo:

Código HTTP	Significado y Solución
400 Bad Request	Causa: La estructura de la petición es incorrecta o faltan datos esenciales. Ocurre antes de que se intente procesar la lógica de negocio. Ejemplos comunes: • No se envió el objeto <code>Propiedad</code> o el array <code>Propietarios</code>. • El array <code>Propietarios</code> está vacío. • Se omitieron campos obligatorios dentro de la sección <code>Propiedad</code> (ej: <code>municipio_id</code>, <code>direccion</code>). • Se envió un tipo de dato incorrecto (ej: texto en un campo numérico como <code>estrato</code>). Solución: Revisa el cuerpo (body) de tu petición JSON y compáralo con la estructura definida en la sección 2 de esta guía. Asegúrate de que todos los campos requeridos estén presentes y tengan el tipo de dato correcto.
401 Unauthorized	Causa: El Token JWT de autenticación no es válido o ha expirado. Solución: Vuelve a consumir el servicio de Login para generar un nuevo token de acceso y úsalo en el encabezado <code>Authorization</code>.

Código HTTP	Significado y Solución
403 Forbidden	Causa: El token es válido, pero el cliente OAuth con el que fue generado no tiene los permisos (scopes) necesarios para esta operación. Solución: Verifica la configuración de tu cliente OAuth en nuby y asegúrate de que tenga asignado el scope <code>write</code>.
422 Unprocessable Entity	Causa: La petición es sintácticamente correcta, pero incumple una regla de negocio del sistema. Ejemplos comunes: - La <code>direccion</code> enviada ya existe para otra propiedad en el mismo <code>municipio_id</code>. - Se omitió una característica marcada como obligatoria para el <code>tipo_id</code> de la propiedad. - Se intentó crear un propietario persona natural omitiendo los campos fiscales requeridos. - Se usó un ID de un modelo que no existe (ej: un <code>municipio_id</code> o <code>tipo_id</code> inválido). Solución: Lee con atención el <code>msg</code> del JSON de error. Te indicará exactamente qué regla de negocio se infringió para que puedas corregir los datos enviados.

Operación Atómica y de Transacción Segura

nuby implementa transacciones anidadas en sus modelos de negocio. Si el endpoint responde con un código `422` (o un fallo 500 del servidor), **toda la creación física del inmueble es cancelada de forma automática (ROLLBACK)**. No tendrás propiedades huérfanas sin propietario ni registros incompletos en las tablas del sistema.

5. Ejemplos de integración

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

Ejemplos de JSON Body

Ejemplo 1: Creación de propiedad con propietario existente

```
{
  "Propiedad": {
    "txtTitulo": "Penthouse Duplex El Poblado con Terraza",
    "tipo_id": 1247,
    "propiedad_tipo": "arriendo",
```

```

    "municipio_id": 1,
    "direccion": "Carrera 35 # 10B - 120, Apartamento 1201",
    "estrato": 1260,
    "barrio_id": 5,
    "valor_arriendo": 4800000,
    "valor_administracion": 650000,
    "urbanizacion": "Torres de San Lucas",
    "observaciones": "Espectacular penthouse con vista de 360 grados, tina de hidromasajes
en terraza principal, 3 alcobas cada una con baño.",
    "propiedad_area": 185.4,
    "llaves_en": "oficina",
    "paga_cuota_sost": "propietario"
  },
  "Propietarios": [
    1247
  ],
  "Caracteristicas": {
    "1": "3",    // Alcobas
    "2": "4",    // Baños
    "31": "si",  // Red de gas (Obligatoria)
    "63": "1"    // Garaje (Checkbox de Garaje - ID 63 en nuby)
  },
  "Coordenadas": "6.205210,-75.561240"
}

```

Ejemplo 2: Creación de propiedad con propietario nuevo, videos y fotos

```

{
  "Propiedad": {
    "txtTitulo": "Casa de Campo en Llanogrande",
    "tipo_id": 1249,
    "propiedad_tipo": "venta",
    "municipio_id": 2,
    "direccion": "Vía Llanogrande Kilómetro 4, Parcelación La Sofía",
    "estrato": 1259,
    "valor_venta": 12500000000,
    "valor_administracion": 300000,
    "observaciones": "Hermosa casa de un solo nivel, amplias zonas verdes, deck con zona
BBQ, acabados campestres modernos.",

```

```
"propiedad_area": 320.0,
"paga_cuota_sost": "propietario"
},
"Propietarios": [
  {
    "persona": 1,
    "documento": "1024567890",
    "tipo_doc_id": 1,
    "nombre1": "Alejandro",
    "apellido1": "Restrepo",
    "direccion1": "Transversal 39B # 4G - 85",
    "municipio_id": 1,
    "email": "alejandro.restrepo@email.com",
    "telefono": "3104567890",
    "tipo_persona_id": 1,
    "regimen_id": 2,
    "resp_fiscal_id": 11
  }
],
"Caracteristicas": {
  "1": "4",
  "2": "5",
  "31": "si", // Red de gas (Obligatoria)
  "63": "1", // Garaje (Checkbox de Garaje - ID 63 en nuby)
  "8": "Sí"
},
"Videos": [
  {
    "video_url": "https://www.youtube.com/watch?v=abc123xyz",
    "video_tipo": "youtube",
    "video_descripcion": "Video aéreo con Dron de la parcelación"
  }
],
"Fotos": [
  {
    "url":
"
DhgGAWjR9awAAAABJRU5ErkJggg==",
    "nombre": "foto_fachada_casa_campo.jpg"
```

```
}  
]  
}
```

cURL

```
# Define tu token y tu instancia  
TOKEN="TU_TOKEN_AQUI"  
INSTANCIA="tu-inmobiliaria.nuby.app"  
  
# Prepara el cuerpo de la petición utilizando estrictamente los nombres de campos físicos  
PAYLOAD='{  
  "Propiedad": {  
    "txtTitulo": "Apartamento para prueba cURL",  
    "tipo_id": 1247,  
    "propiedad_tipo": "arriendo",  
    "municipio_id": 127,  
    "direccion": "Calle Falsa 123 via cURL",  
    "estrato": 1258,  
    "propiedad_area": 97  
  },  
  "Propietarios": [1],  
  "Caracteristicas": {  
    "1": "3",  
    "2": "3",  
    "31": "si",  
    "63": "1"  
  }  
}'  
  
curl -X POST "https://${INSTANCIA}/service/v2/public/properties" \  
-H "Content-Type: application/json" \  
-H "Authorization: Bearer ${TOKEN}" \  
-d "${PAYLOAD}"
```

PHP

```

<?php

$instancia = 'tu-inmobiliaria.nuby.app';
$token = 'TU_TOKEN_AQUI';

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

$payload = [
    'Propiedad' => [
        'txtTitulo' => 'Propiedad creada desde PHP',
        'tipo_id' => 1247,
        'propiedad_tipo' => 'venta',
        'municipio_id' => 127,
        'direccion' => 'Avenida Siempreviva 742, PHP',
        'estrato' => 1258,
        'propiedad_area' => 97 // Nombre exacto de la columna física
    ],
    'Propietarios' => [1], // ID de un propietario existente
    'Caracteristicas' => [
        '1' => '3',    // Habitaciones
        '2' => '3',    // Baños
        '31' => 'si',  // Red de gas (Obligatoria)
        '63' => '1'    // Garaje (Checkbox de Garaje - ID 63 en nuby)
    ]
];

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

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

curl_close($ch);

```



```
echo "Código de estado HTTP: {$http_code}\n";  
echo "Respuesta del servidor:\n";  
print_r($response);  
  
?>
```

Python

```
import requests  
import json  
  
instancia = 'tu-inmobiliaria.nuby.app'  
token = 'TU_TOKEN_AQUI'  
url = f"https://{instancia}/service/v2/public/properties"  
  
payload = {  
    "Propiedad": {  
        "txtTitulo": "Propiedad Creada desde Python",  
        "tipo_id": 1247,  
        "propiedad_tipo": "venta",  
        "municipio_id": 127,  
        "direccion": "Calle de Python, 101",  
        "estrato": 1258,  
        "propiedad_area": 97  
    },  
    "Propietarios": [1], # ID de un propietario existente  
    "Caracteristicas": {  
        "1": "3",  
        "2": "3",  
        "31": "si",  
        "63": "1"  
    }  
}  
  
headers = {  
    "Content-Type": "application/json",
```

```

    "Authorization": f"Bearer {token}"
}

try:
    response = requests.post(url, headers=headers, data=json.dumps(payload))

    print(f"Código de estado HTTP: {response.status_code}")
    print("Respuesta del servidor:")
    print(response.json())

except requests.exceptions.RequestException as e:
    print(f"Ocurrió un error en la petición: {e}")

```

JavaScript (Fetch API)

```

const instancia = 'tu-inmobiliaria.nuby.app';
const token = 'TU_TOKEN_AQUI';
const url = `https://${instancia}/service/v2/public/properties`;

const payload = {
  "Propiedad": {
    "txtTitulo": "Propiedad Creada desde JavaScript",
    "tipo_id": 1247,
    "propiedad_tipo": "arriendo",
    "municipio_id": 127,
    "direccion": "Avenida JavaScript, Lote 5",
    "estrato": 1258,
    "propiedad_area": 97
  },
  "Propietarios": [1], // ID de un propietario existente
  "Caracteristicas": {
    "1": "3",
    "2": "3",
    "31": "si",
    "63": "1"
  }
}

```

```

};

async function crearPropiedad() {
  try {
    const response = await fetch(url, {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
        'Authorization': `Bearer ${token}`
      },
      body: JSON.stringify(payload)
    });

    const data = await response.json();

    console.log(`Código de estado HTTP: ${response.status}`);
    console.log('Respuesta del servidor:');
    console.log(data);

  } catch (error) {
    console.error('Error en la petición:', error);
  }
}

crearPropiedad();

```

Power Query M (Excel / Power BI)

```

let
    instancia = "tu-inmobiliaria.nuby.app",
    token = "TU_TOKEN_AQUI",
    url = "https://" & instancia & "/service/v2/public/properties",

    payload = [
        Propiedad = [
            txtTitulo = "Propiedad desde Power Query",
            tipo_id = 1247,
            propiedad_tipo = "venta",

```

```

        municipio_id = 127,
        direccion = "Calle Power BI, 4.0",
        estrato = 1258,
        propiedad_area = 97
    ],
    Propietarios = {1}, // ID de un propietario existente
    Caracteristicas = [
        #"1" = "3",
        #"2" = "3",
        #"31" = "si",
        #"63" = "1"
    ]
],

jsonPayload = Json.FromValue(payload),

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

jsonResponse = Json.Document(response)
in
jsonResponse

```