

# Propiedades / Listar

## Características por Clase de Inmueble

Permite obtener la lista de características que puede tener una propiedad, filtrada por una clase de inmueble específica. Cada elemento de la respuesta contiene la información detallada de una característica perteneciente a la clase indicada. Este servicio es ideal cuando ya conoces la clase de inmueble y solo necesitas las características aplicables a ese tipo.

### ¿Para qué sirve este servicio?

Úsalo cuando necesites construir un formulario dinámico para un tipo de inmueble en particular (por ejemplo, solo las características de "Casa" o de "Apartamento"). Si prefieres obtener todas las características de todas las clases de una sola vez, utiliza el endpoint de **Listar Características**. Para conocer los identificadores válidos de clases de inmueble, consulta el servicio de **Listar Clases de Inmueble**.

## 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 **{{id}}** por el identificador numérico de la clase de inmueble que deseas consultar.

```
GET https://{{instancia}}/service/v2/public/masters/properties/features/property-class/{{id}}
```

### ¿Qué debes colocar en {{instancia}}?

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

Por ejemplo, si para entrar a tu sistema escribes `inmobiliaria.nuby.app` o `inmobiliaria.arrendasoft.co` en tu navegador, esa será exactamente tu

instancia. Solo asegúrate de no incluir el "https://" ni barras diagonales ("/") al final.

## 2. La Petición (¿Qué debes enviarnos?)

Este servicio no requiere cuerpo en la petición. Solo necesitas enviar los encabezados correctos y el parámetro de ruta:

<b>Método</b>	GET
<b>Content-Type</b>	application/json
<b>Authorization</b>	<b>Bearer token</b> , Token obtenido al consumir el servicio <a href="#">Login</a>

### Parámetro de ruta:

Parámetro	Tipo	Requerido	Descripción
id	Numérico	Sí	Identificador único de la clase de inmueble. Puedes obtener los valores válidos consultando el servicio de <a href="#">Listar Clases de Inmueble</a> .

### ¿Aún no tienes tu Token de acceso?

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

## 3. La Respuesta (¿Qué te entregaremos?)

Si tu Token es válido y el `id` de la clase de inmueble es correcto, el sistema te devolverá una lista con las características asociadas a esa clase. Por ejemplo, para la clase "Casa" (`id = 1249`), la respuesta se verá similar a esta:

```
[
  {
    "id": "1",
    "descripcion": "Nº De Habitaciones",
    "tipo_campo": "numeric",
    "opciones_valores": null,
    "opciones_descripciones": null,
```

```
"orden": "1",
"grupo": "Características del Inmueble",
"clase_inmueble_id": "1249",
"clase_inmueble": "Casa"
},
{
  "id": "31",
  "descripcion": "Red de gas",
  "tipo_campo": "select",
  "opciones_valores": "si,no",
  "opciones_descripciones": "Si,No",
  "orden": "5",
  "grupo": "Características del Inmueble",
  "clase_inmueble_id": "1249",
  "clase_inmueble": "Casa"
},
{
  "id": "22",
  "descripcion": "Balcón",
  "tipo_campo": "checkbox",
  "opciones_valores": null,
  "opciones_descripciones": null,
  "orden": "10",
  "grupo": "Características del Inmueble",
  "clase_inmueble_id": "1249",
  "clase_inmueble": "Casa"
},
{
  "id": "38",
  "descripcion": "Parqueadero",
  "tipo_campo": "select",
  "opciones_valores": "propio,visitantes",
  "opciones_descripciones": "Propio,Visitantes",
  "orden": "12",
  "grupo": "Características del Inmueble",
  "clase_inmueble_id": "1249",
  "clase_inmueble": "Casa"
}
]
```

**Nota:** Las características disponibles y sus grupos dependen de la configuración de cada instancia. La lista se devuelve ordenada por grupo, luego por orden dentro del grupo, y finalmente por descripción alfabéticamente. La estructura de la respuesta es idéntica a la del servicio de **Listar Características**, pero filtrada a la clase de inmueble indicada.

Tabla con la descripción de cada campo del JSON:

Clave	Descripción
<b>id</b>	Identificador único de la característica (tipo texto).
<b>descripcion</b>	Nombre descriptivo de la característica (por ejemplo: "Nº De Habitaciones", "Parqueadero").
<b>tipo_campo</b>	Tipo de campo que define cómo se captura la información. Valores posibles: <code>numeric</code> (campo numérico), <code>checkbox</code> (casilla de verificación), <code>select</code> (lista desplegable con opciones predefinidas), entre otros.
<b>opciones_valores</b>	Valores técnicos de las opciones disponibles, separados por coma. Aplica únicamente cuando <code>tipo_campo</code> es <code>select</code> . Será <code>null</code> o vacío para otros tipos de campo.
<b>opciones_descripciones</b>	Descripciones legibles de las opciones disponibles, separadas por coma. Cada descripción corresponde posicionalmente al valor en <code>opciones_valores</code> . Por ejemplo, si <code>opciones_valores</code> es <code>"propio,visitantes"</code> , las descripciones serán <code>"Propio,Visitantes"</code> .
<b>orden</b>	Posición numérica que indica el orden de presentación de la característica dentro de su grupo.
<b>grupo</b>	Nombre del grupo al que pertenece la característica (por ejemplo: "Características del Inmueble"). Permite agrupar visualmente las características en un formulario.
<b>clase_inmueble_id</b>	Identificador de la clase de inmueble a la que está asociada esta característica. Coincidirá con el <code>id</code> enviado en la URL.
<b>clase_inmueble</b>	Nombre descriptivo de la clase de inmueble (por ejemplo: "Casa", "Apartamento", "Bodega").

## 4. Seguridad y Posibles Errores

**¡Tu pase tiene fecha de caducidad!**

Por medidas de seguridad, el `token` que te entregamos **solo dura 1 hora**. Una vez transcurrido ese tiempo, el pase expirará y el sistema te bloqueará el acceso devolviéndote un error `401`. Cuando esto ocurra, tu sistema simplemente debe volver

a consumir el servicio de [Login](#) para pedir un pase nuevo y continuar trabajando.

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

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

Si envías un `id` vacío o no numérico, el sistema te devolverá un error de validación:

```
{
  "error": "El ID de la clase del inmueble no puede estar vacío y debe ser de tipo numérico"
}
```

Otros posibles errores:

Código HTTP	Descripción
400	Token no enviado, formato incorrecto, o el parámetro <code>id</code> está vacío o no es numérico.
401	Token expirado o inválido. Solicita uno nuevo a través del servicio de Login.
403	El cliente OAuth no tiene el scope necesario para esta operación. Para endpoints GET se requiere el scope <code>read</code> . Mensaje: "Insufficient scope. Required: 'read', granted: '{scope_actual}'."

## 5. Ejemplos de integración

Aquí tienes ejemplos de código listos para que tus desarrolladores los adapten a tu plataforma. En estos ejemplos se consulta la clase "Casa" con `id = 1249`:

### cURL

```
curl -X GET "https://{{instancia}}/service/v2/public/masters/properties/features/property-
class/1249" \
-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 de Login
$claseInmuebleId = 1249; // ID de la clase de inmueble (ej. Casa)

$url = "https://{$instance}/service/v2/public/masters/properties/features/property-
class/{$claseInmuebleId}";

$ch = curl_init($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) {
        $caracteristicas = json_decode($response, true);
        echo "Características de la clase de inmueble (ID {$claseInmuebleId}):\n";
        foreach ($caracteristicas as $car) {
            echo "  {$car['descripcion']} (tipo: {$car['tipo_campo']})\n";
        }
    }
}
```

```
    } else {
        echo "Error al consultar las características.\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 de Login
clase_inmueble_id = 1249 # ID de la clase de inmueble (ej. Casa)

# 2. Prepara la dirección y los encabezados
url = f"https://{instancia}/service/v2/public/masters/properties/features/property-
class/{clase_inmueble_id}"

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:
        caracteristicas = response.json()
        print(f"Características de la clase de inmueble (ID {clase_inmueble_id}):")
        for car in caracteristicas:
```

```
        print(f" {car['descripcion']} (tipo: {car['tipo_campo']})")
    else:
        print("Error al consultar las características.")
        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 de Login
const claseInmuebleId = 1249; // ID de la clase de inmueble (ej. Casa)

// 2. Prepara la dirección
const url = `https://${instancia}/service/v2/public/masters/properties/features/property-
class/${claseInmuebleId}`;

// 3. Función para consultar las características por clase de inmueble
async function consultarCaracteristicasPorClase() {
    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 caracteristicas = await response.json();
            console.log(`Características de la clase de inmueble (ID
${claseInmuebleId}):`);
            caracteristicas.forEach(car => {
```

```

        console.log(` ${car.descripcion} (tipo: ${car.tipo_campo})`);
    });
} else {
    console.log('Error al consultar las características.');
```

```

    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
consultarCaracteristicasPorClase();

```

## 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 de Login
    claseInmuebleId = "1249", // ID de la clase de inmueble (ej. Casa)

    // 2. Prepara la dirección de la petición
    url = "https://" & instancia &
"/service/v2/public/masters/properties/features/property-class/" & claseInmuebleId,

    // 3. Envía la petición GET con el token de autenticación
    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",
        {"id", "descripcion", "tipo_campo", "opciones_valores", "opciones_descripciones",
"orden", "grupo", "clase_inmueble_id", "clase_inmueble"},
        {"ID", "Descripcion", "TipoCampo", "OpcionesValores", "OpcionesDescripciones",
"Orden", "Grupo", "ClaseInmuebleID", "ClaseInmueble"})
in
    expandido
```

---

Revisión #7

Creado el 15 julio 2024 16:16:37

Actualizado el 16 abril 2026 17:35:10 por Isabel Higuita