

Propiedades / Listar Clases de Inmueble

Permite obtener la lista completa de clases de inmueble disponibles en el sistema. Cada elemento de la respuesta contiene el identificador único de la clase y su nombre descriptivo. Este servicio es útil para poblar selectores de tipo de inmueble en tus integraciones.

¿Para qué sirve este servicio?

Úsalo para conocer los valores válidos del campo `clase` de inmueble al momento de filtrar o registrar propiedades a través de la API. Así te aseguras de enviar siempre un valor reconocido por el sistema. Las clases de inmueble disponibles dependen de la configuración de cada instancia.

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/masters/properties/property-classes
```

¿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:

Método	GET
Content-Type	application/json
Authorization	Bearer token , Token obtenido al consumir el servicio Login

¿Aún no tienes tu Token de acceso?

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

3. La Respuesta (¿Qué te entregaremos?)

Si tu Token es válido, el sistema te devolverá una lista con todas las clases de inmueble configuradas en tu instancia. La respuesta se verá similar a esta:

```
[
  {
    "id": "1367",
    "clase": "Amoblados"
  },
  {
    "id": "1253",
    "clase": "Apartaestudio"
  },
  {
    "id": "1247",
    "clase": "Apartamento"
  },
  {
    "id": "1248",
    "clase": "Bodega"
  },
  {
    "id": "1249",
    "clase": "Casa"
  }
]
```

Nota: Los valores de `id` y la cantidad de clases disponibles pueden variar entre instancias, ya que dependen de la configuración particular de cada inmobiliaria. La lista se devuelve ordenada alfabéticamente por el nombre de la clase.

Tabla con la descripción de cada campo del JSON:

Clave	Descripción
<code>id</code>	Identificador único de la clase de inmueble (tipo texto). Utiliza este valor cuando necesites filtrar propiedades por clase de inmueble a través de la API.
<code>clase</code>	Nombre descriptivo de la clase de inmueble (por ejemplo: "Apartamento", "Casa", "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."
  }
}
```

Otros posibles errores:

Código HTTP	Descripción
400	Token no enviado o con formato incorrecto. Asegúrate de incluir el encabezado <code>Authorization: Bearer TU_TOKEN</code> .
401	Token expirado o inválido. Solicita uno nuevo a través del servicio de Login.

Código HTTP	Descripción
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:

cURL

```
curl -X GET "https://{{instancia}}/service/v2/public/masters/properties/property-classes" \
-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

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

$ch = curl_init($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) {
        $clases = json_decode($response, true);
        echo "Clases de inmueble:\n";
        foreach ($clases as $clase) {
            echo "  ID: {$clase['id']} - {$clase['class']}\n";
        }
    } else {
        echo "Error al consultar las clases de inmueble.\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

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

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:
        clases = response.json()
        print("Clases de inmueble:")
        for clase in clases:
            print(f" ID: {clase['id']} - {clase['clase']}")
    else:
        print("Error al consultar las clases de inmueble.")
        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

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

// 3. Función para consultar las clases de inmueble
async function consultarClasesDeInmueble() {
    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 clases = await response.json();
  console.log('Clases de inmueble:');
  clases.forEach(clase => {
    console.log(` ID: ${clase.id} - ${clase.clase}`);
  });
} else {
  console.log('Error al consultar las clases de inmueble. ');
  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
consultarClasesDeInmueble();

```

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

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

  // 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", "clase"}, {"ID",  
"Clase"})
```

```
in
```

```
    expandido
```

Revisión #12

Creado el 15 julio 2024 15:56:19

Actualizado el 16 abril 2026 17:34:01 por Isabel Higuita