

Maestras

Bienvenido a la **base estructural** de tus integraciones. En esta sección detallamos los servicios disponibles para consultar las "entidades maestras" en la API REST de nuby.

¿Qué son exactamente las entidades maestras? Para evitar términos técnicos complejos, piensa en ellas como los **diccionarios predefinidos**, los catálogos base o las listas desplegables de tu sistema.

¿Por qué son tan importantes para tu conexión?

- **Estandarización total:** Son listas fijas que organizan la información para que todo se mantenga ordenado y sin ambigüedades (garantizando, por ejemplo, que el sistema identifique correctamente una "Casa" frente a un "Apartamento").
- **Sincronización de lenguaje:** Permiten que tus plataformas externas lean, entiendan y clasifiquen los datos usando exactamente los mismos parámetros y reglas que utiliza tu inmobiliaria.

¡Explora las opciones a continuación para sincronizar tus catálogos y asegurar que todos tus sistemas hablen el mismo idioma!

- [Propiedades / Listar Estados](#)
- [Propiedades / Listar Clases de Inmueble](#)
- [Propiedades / Listar Tipos de Servicios](#)
- [Propiedades / Listar Características](#)
- [Propiedades / Listar Características por Clase de Inmueble](#)

Propiedades / Listar Estados

Permite obtener la lista completa de estados posibles que puede tener una propiedad en el sistema. Cada elemento de la respuesta contiene el identificador numérico del estado y su descripción en texto. Este servicio es útil para poblar selectores o filtros de estado en tus integraciones.

¿Para qué sirve este servicio?

Úsalo para conocer los valores válidos del campo `estado` al momento de filtrar, crear o actualizar propiedades a través de la API. Así te aseguras de enviar siempre un valor reconocido por 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.

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

¿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 todos los estados posibles de una propiedad. La respuesta se verá así:

```
[
  {
    "id": 1,
    "estado": "Activa"
  },
  {
    "id": 0,
    "estado": "Arrendada"
  },
  {
    "id": 2,
    "estado": "Inactiva"
  },
  {
    "id": 3,
    "estado": "Vendida"
  }
]
```

Tabla con la descripción de cada campo del JSON:

Clave	Descripción
id	Identificador numérico del estado. Utiliza este valor cuando necesites filtrar propiedades por estado o actualizar el estado de una propiedad a través de la API.
estado	Nombre descriptivo del estado en texto legible (por ejemplo: "Activa", "Arrendada").

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.
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

```
curl -X GET "https://{{instancia}}/service/v2/public/masters/properties/states" \  
-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/states";  
  
$ch = curl_init($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) {  
        $estados = json_decode($response, true);  
        echo "Estados de propiedades:\n";  
        foreach ($estados as $estado) {  
            echo "  ID: {$estado['id']} - {$estado['estado']}\n";  
        }  
    } else {  
        echo "Error al consultar los estados.\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/states"  
  
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:  
        estados = response.json()  
        print("Estados de propiedades:")  
        for estado in estados:  
            print(f"  ID: {estado['id']} - {estado['estado']}")  
    else:  
        print("Error al consultar los estados.")  
        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/states`;

// 3. Función para consultar los estados de propiedades
async function consultarEstados() {
    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 estados = await response.json();
            console.log('Estados de propiedades:');
            estados.forEach(estado => {
                console.log(`  ID: ${estado.id} - ${estado.estado}`);
            });
        } else {
            console.log('Error al consultar los estados.');
            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
consultarEstados();
```

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/states",

    // 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", "estado"}, {"ID",
        "Estado"})
in
    expandido
```

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: <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

```
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['clase']}\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
```

Propiedades / Listar Tipos de Servicios

Permite obtener la lista completa de tipos de servicio que puede tener una propiedad en el sistema. Cada elemento de la respuesta contiene el identificador del tipo de servicio y su nombre descriptivo. Este servicio es útil para poblar selectores o filtros de tipo de servicio en tus integraciones.

¿Para qué sirve este servicio?

Úsalo para conocer los valores válidos del campo de tipo de servicio al momento de filtrar o registrar propiedades a través de la API. Así te aseguras de enviar siempre un valor reconocido por el sistema. Los nombres descriptivos pueden variar ligeramente entre instancias según la configuración de cada inmobiliaria, pero los identificadores (`id`) son siempre los mismos.

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/service-types
```

¿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 los tres tipos de servicio disponibles para propiedades. La respuesta se verá similar a esta:

```
[
  {
    "id": "arriendo",
    "servicio": "Arriendo"
  },
  {
    "id": "venta",
    "servicio": "Venta"
  },
  {
    "id": "venta y arriendo",
    "servicio": "Venta y Arriendo"
  }
]
```

Nota: El campo `id` es de tipo texto (string) y sus valores son fijos: `"arriendo"`, `"venta"` y `"venta y arriendo"`. El campo `servicio` muestra el nombre descriptivo, que puede variar ligeramente entre instancias según la configuración de cada inmobiliaria.

Tabla con la descripción de cada campo del JSON:

Clave	Descripción
-------	-------------

id	Identificador del tipo de servicio (tipo texto). Valores posibles: "arriendo", "venta", "venta y arriendo". Utiliza este valor cuando necesites filtrar propiedades por tipo de servicio a través de la API.
servicio	Nombre descriptivo del tipo de servicio (por ejemplo: "Arriendo", "Venta", "Venta y Arriendo").

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.
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/service-types" \  
-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/service-types";  
  
$ch = curl_init($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) {  
        $servicios = json_decode($response, true);  
        echo "Tipos de servicio:\n";  
        foreach ($servicios as $servicio) {
```

```

        echo "  ID: {$servicio['id']} - {$servicio['servicio']}\n";
    }
} else {
    echo "Error al consultar los tipos de servicio.\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/service-types"

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:
        servicios = response.json()
        print("Tipos de servicio:")
        for servicio in servicios:

```

```

        print(f" ID: {servicio['id']} - {servicio['servicio']}")
    else:
        print("Error al consultar los tipos de servicio.")
        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/service-types`;

// 3. Función para consultar los tipos de servicio
async function consultarTiposDeServicio() {
    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 servicios = await response.json();
            console.log('Tipos de servicio:');
            servicios.forEach(servicio => {
                console.log(` ID: ${servicio.id} - ${servicio.servicio}`);
            });
        } else {

```

```

        console.log('Error al consultar los tipos de servicio.');
```

```

        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
consultarTiposDeServicio();
```

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/service-types",

    // 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", "servicio"}, {"ID",
        "Servicio"})
in
```

expandido

Propiedades / Listar Características

Permite obtener la lista completa de características que pueden tener las propiedades en el sistema. Cada elemento de la respuesta contiene la información detallada de una característica, incluyendo su tipo de campo, opciones disponibles y la clase de inmueble a la que pertenece. Este servicio es útil para construir formularios dinámicos de propiedades o para conocer qué datos se pueden registrar en cada tipo de inmueble.

¿Para qué sirve este servicio?

Úsalo para obtener el catálogo de características configuradas en la plataforma. Cada característica está vinculada a una clase de inmueble específica (Apartamento, Casa, Bodega, etc.) y tiene un tipo de campo que indica cómo debe capturarse la información (numérico, checkbox, lista desplegable, etc.). Si necesitas consultar solo las características de una clase de inmueble en particular, utiliza el endpoint de **Listar Características por Clase 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.

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

¿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 características configuradas para las propiedades. 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": "1367",
    "clase_inmueble": "Amoblados"
  },
  {
    "id": "2",
    "descripcion": "Nº De Baños",
    "tipo_campo": "numeric",
    "opciones_valores": null,
    "opciones_descripciones": null,
    "orden": "2",
    "grupo": "Características del inmueble",
```

```
"clase_inmueble_id": "1367",
"clase_inmueble": "Amoblados"
},
{
  "id": "54",
  "descripcion": "Oficina",
  "tipo_campo": "checkbox",
  "opciones_valores": "",
  "opciones_descripciones": "",
  "orden": "3",
  "grupo": "Características del Inmueble",
  "clase_inmueble_id": "1248",
  "clase_inmueble": "Bodega"
},
{
  "id": "13",
  "descripcion": "Baño Auxiliar",
  "tipo_campo": "checkbox",
  "opciones_valores": null,
  "opciones_descripciones": null,
  "orden": "4",
  "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, sus grupos y las clases de inmueble asociadas 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.

Tabla con la descripción de cada campo del JSON:

Clave	Descripción
id	Identificador único de la característica (tipo texto).
descripcion	Nombre descriptivo de la característica (por ejemplo: "Nº De Habitaciones", "Parqueadero").
tipo_campo	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.
opciones_valores	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.
opciones_descripciones	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> .
orden	Posición numérica que indica el orden de presentación de la característica dentro de su grupo.
grupo	Nombre del grupo al que pertenece la característica (por ejemplo: "Características del Inmueble"). Permite agrupar visualmente las características en un formulario.
clase_inmueble_id	Identificador de la clase de inmueble a la que está asociada esta característica. Corresponde al <code>id</code> devuelto por el servicio Listar Clases de Inmueble .
clase_inmueble	Nombre descriptivo de la clase de inmueble (por ejemplo: "Amoblados", "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.
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. Endpoint relacionado: Filtrar por Clase de Inmueble

Si necesitas obtener únicamente las características de una clase de inmueble específica, puedes usar el siguiente endpoint en lugar de filtrar la respuesta completa:

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

Donde `{id}` es el identificador numérico de la clase de inmueble (obtenido del servicio **Listar Clases de Inmueble**). La respuesta tiene exactamente la misma estructura que este endpoint, pero filtrada a la clase indicada.

Si el parámetro `{id}` está vacío o no es numérico, el sistema devolverá un error `400`:

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

6. Ejemplos de integración

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

cURL

```
# Listar todas las características
curl -X GET "https://{{instancia}}/service/v2/public/masters/properties/features" \
-H "Content-Type: application/json" \
-H "Authorization: Bearer TU_TOKEN_AQUI"

# Listar características de una clase de inmueble específica (ej. ID 1249 = Casa)
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

// Para todas las características:
$url = "https://{$instance}/service/v2/public/masters/properties/features";

// Para filtrar por clase de inmueble (descomentar y reemplazar el ID):
// $claseInmuebleId = 1249; // Ejemplo: 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 propiedades:\n";
        foreach ($caracteristicas as $car) {
            echo "    [{$car['clase_inmueble']}] {$car['descripcion']} (tipo:
{$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

# 2. Prepara la dirección y los encabezados
# Para todas las características:

```

```

url = f"https://{instancia}/service/v2/public/masters/properties/features"

# Para filtrar por clase de inmueble (descomentar y reemplazar el ID):
# clase_inmueble_id = 1249 # Ejemplo: Casa
# 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:
        características = response.json()
        print("Características de propiedades:")
        for car in características:
            print(f"  [{car['clase_inmueble']}] {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

```

```

// 2. Prepara la dirección
// Para todas las características:
const url = `https://${instancia}/service/v2/public/masters/properties/features`;

// Para filtrar por clase de inmueble (descomentar y reemplazar el ID):
// const claseInmuebleId = 1249; // Ejemplo: Casa
// const url =
`https://${instancia}/service/v2/public/masters/properties/features/property-
class/${claseInmuebleId}`;

// 3. Función para consultar las características
async function consultarCaracteristicas() {
  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 propiedades:');
      caracteristicas.forEach(car => {
        console.log(`  [${car.clase_inmueble}] ${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  
consultarCaracteristicas();
```

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/features",  
  
    // 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
```

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:

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

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 Listar Clases de Inmueble .

¿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
id	Identificador único de la característica (tipo texto).
descripcion	Nombre descriptivo de la característica (por ejemplo: "Nº De Habitaciones", "Parqueadero").
tipo_campo	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.
opciones_valores	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.
opciones_descripciones	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> .
orden	Posición numérica que indica el orden de presentación de la característica dentro de su grupo.
grupo	Nombre del grupo al que pertenece la característica (por ejemplo: "Características del Inmueble"). Permite agrupar visualmente las características en un formulario.
clase_inmueble_id	Identificador de la clase de inmueble a la que está asociada esta característica. Coincidirá con el <code>id</code> enviado en la URL.
clase_inmueble	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
```