

# 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:

<b>Método</b>	GET
<b>Content-Type</b>	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
<b>id</b>	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.
<b>estado</b>	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
```

Revisión #9

Creado el 15 julio 2024 15:05:11

Actualizado el 16 abril 2026 17:33:32 por Isabel Higuaita