

# Listar Asesores

Permite obtener la lista completa de asesores registrados en el sistema. Cada elemento de la respuesta contiene los datos básicos de contacto de cada asesor: documento de identidad, nombre completo, teléfono, correo electrónico y fotografía.

## ¿Para qué sirve este servicio?

Úsalo cuando necesites conocer los asesores disponibles en la inmobiliaria, por ejemplo para mostrar un directorio de agentes en tu sitio web, poblar un selector de asesores en un formulario de contacto, o sincronizar la información de los agentes con tu sistema externo.

## 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/agents
```

## ¿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 ni parámetros adicionales. 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 asesores registrados (excluyendo usuarios internos del sistema). Los asesores se devuelven ordenados alfabéticamente por nombre. La respuesta se verá similar a esta:

```
[
  {
    "id": "3",
    "documento": "52498731",
    "nombre": "ANDREA MILENA CASTRO ROJAS",
    "telefono": "3104567890",
    "email": "andrea.castro@miredinmobiliaria.com",
    "path_photo": "https://mi-inmobiliaria.nuby.app/pic/usuarios/foto_andrea_castro.jpg"
  },
  {
    "id": "7",
    "documento": "80213654",
    "nombre": "CARLOS EDUARDO PINEDA VARGAS",
    "telefono": "3209876543",
    "email": "carlos.pineda@miredinmobiliaria.com",
    "path_photo": "https://mi-inmobiliaria.nuby.app/pic/usuarios/foto_carlos_pineda.jpg"
  },
  {
    "id": "12",
    "documento": "1098745231",
    "nombre": "JULIANA PATRICIA RÍOS MENDOZA",
    "telefono": "3156781234",
    "email": "juliana.rios@miredinmobiliaria.com",
    "path_photo": "https://mi-inmobiliaria.nuby.app/pic/iconos/user_photo.png"
  }
]
```

## Descripción de cada campo:

Campo	Tipo	Descripción
<code>id</code>	string	Identificador único del usuario en el sistema.
<code>documento</code>	string	Número de documento de identidad del asesor.
<code>nombre</code>	string	Nombre completo del asesor (compuesto por primer nombre, segundo nombre, primer apellido y segundo apellido).
<code>telefono</code>	string	Número de teléfono del asesor.
<code>email</code>	string	Dirección de correo electrónico del asesor.
<code>path_photo</code>	string	URL completa de la fotografía del asesor. Si el asesor no tiene foto registrada, se devuelve la imagen genérica del sistema.

### Nota sobre la respuesta

El listado excluye automáticamente los usuarios internos del sistema (cuentas de administración y soporte). Solo se devuelven los asesores reales de la inmobiliaria. Si no hay asesores registrados, la respuesta será un arreglo vacío `[]`.

## 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/agents" \
-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

$url = "https://{instance}/service/v2/public/agents";
```

```

$ch = curl_init($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) {
        $asesores = json_decode($response, true);
        echo "Asesores encontrados: " . count($asesores) . "\n";
        foreach ($asesores as $asesor) {
            echo " [{$asesor['id']}] {$asesor['nombre']} - {$asesor['email']}\n";
        }
    } else {
        echo "Error al consultar los asesores.\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 de la petición
url = f"https://{instancia}/service/v2/public/agents"

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:
        asesores = response.json()
        print(f"Asesores encontrados: {len(asesores)}")
        for asesor in asesores:
            print(f" [{asesor['id']}] {asesor['nombre']} - {asesor['email']}")
    else:
        print("Error al consultar los asesores.")
        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 de la petición
const url = `https://${instancia}/service/v2/public/agents`;

```

```

// 3. Función para consultar los asesores
async function consultarAsesores() {
  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 asesores = await response.json();
      console.log(`Asesores encontrados: ${asesores.length}`);
      asesores.forEach(asesor => {
        console.log(` [${asesor.id}] ${asesor.nombre} - ${asesor.email}`);
      });
    } else {
      console.log('Error al consultar los asesores.');
      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
consultarAsesores();

```

## 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
url = "https://" & instancia & "/service/v2/public/agents",

// 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",
    {"id", "documento", "nombre", "telefono", "email", "path_photo"},
    {"ID", "Documento", "Nombre", "Telefono", "Email", "Foto"})
in
expandido
```

Revisión #12

Creado el 24 mayo 2024 16:49:04

Actualizado el 16 abril 2026 17:39:52 por Isabel Higuita