

Buscar Asesor por Documento

Permite obtener la información de un asesor específico a partir de su número de documento de identidad. La respuesta contiene los datos básicos de contacto del asesor: documento, nombre completo, teléfono, correo electrónico y fotografía.

¿Para qué sirve este servicio?

Úsalo cuando ya conoces el número de documento del asesor y necesitas consultar su información detallada. Es ideal para mostrar la ficha de un agente en particular, verificar datos de un asesor asignado a una propiedad o sincronizar la información de un agente puntual 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, y **{{document}}** por el número de documento de identidad del asesor que desees consultar.

```
GET https://{{instancia}}/service/v2/public/agents/{{document}}
```

¿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 construir la URL con el número de documento del asesor como parte de la ruta e incluir los encabezados requeridos:

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.

Parámetro de ruta:

Parámetro	Tipo	Requerido	Descripción
<code>document</code>	string	Sí	Número de documento de identidad del asesor que se desea consultar. Se envía directamente en la URL.

Ejemplo de petición:

```
GET https://mi-inmobiliaria.nuby.app/service/v2/public/agents/52498731
```

3. La Respuesta (¿Qué te entregaremos?)

Si tu Token es válido y el documento corresponde a un asesor registrado, el sistema te devolverá un objeto JSON con la información del asesor. 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"
}
```

Descripción de cada campo:

Campo	Tipo	Descripción
-------	------	-------------

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

Importante: asesor no encontrado

Cuando el documento no corresponde a ningún asesor registrado, el sistema **no devuelve un error**. En su lugar, responde con un HTTP `200` y un arreglo vacío:

```
[ ]
```

Asegúrate de validar en tu integración si la respuesta es un arreglo vacío `[ ]` para manejar correctamente el caso en que el asesor no exista.

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
200	Respuesta exitosa. Contiene el objeto JSON del asesor, o un arreglo vacío <code>[]</code> si no se encontró ningún asesor con ese documento.
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

```
# Consultar asesor por documento
curl -X GET "https://{{instancia}}/service/v2/public/agents/52498731" \
-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
$documento = '52498731'; // Documento del asesor a consultar

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

$ch = curl_init($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) {
        $asesor = json_decode($response, true);

        if (!empty($asesor)) {
            echo "Asesor encontrado:\n";
            echo "  ID: { $asesor['id'] }\n";
            echo "  Nombre: { $asesor['nombre'] }\n";
            echo "  Email: { $asesor['email'] }\n";
            echo "  Teléfono: { $asesor['telefono'] }\n";
        } else {
            echo "No se encontró ningún asesor con el documento { $documento }.\n";
        }
    } else {
        echo "Error al consultar el asesor.\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
documento = '52498731' # Documento del asesor a consultar

# 2. Prepara la dirección de la petición
url = f"https://{instancia}/service/v2/public/agents/{documento}"

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:
        asesor = response.json()

        if asesor:
            print(f"Asesor encontrado:")
            print(f"  ID: {asesor['id']}")
            print(f"  Nombre: {asesor['nombre']}")
            print(f"  Email: {asesor['email']}")
            print(f"  Teléfono: {asesor['telefono']}")
        else:
            print(f"No se encontró ningún asesor con el documento {documento}.")
    else:
```

```
print("Error al consultar el asesor.")
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
const documento = '52498731'; // Documento del asesor a consultar

// 2. Prepara la dirección de la petición
const url = `https://${instancia}/service/v2/public/agents/${documento}`;

// 3. Función para consultar el asesor
async function consultarAsesor() {
    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 asesor = await response.json();

            if (asesor && !Array.isArray(asesor)) {
                console.log(`Asesor encontrado:`);
                console.log(`  ID: ${asesor.id}`);
                console.log(`  Nombre: ${asesor.nombre}`);
                console.log(`  Email: ${asesor.email}`);
            }
        }
    } catch (error) {
        console.error('Error al consultar al asesor:', error);
    }
}
```

```

        console.log(` Teléfono: ${asesor.telefono}`);
    } else {
        console.log(`No se encontró ningún asesor con el documento
${documento}.`);
    }
} else {
    console.log('Error al consultar el asesor.');
```

```

    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
consultarAsesor();

```

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
    documento = "52498731", // Documento del asesor a consultar

    // 2. Prepara la dirección de la petición
    url = "https://" & instancia & "/service/v2/public/agents/" & documento,

    // 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 registro

```

```
jsonResponse = Json.Document(response),
resultado = Record.ToTable(jsonResponse),
expandido = Table.Pivot(resultado, List.Distinct(resultado[Name]), "Name", "Value")
in
expandido
```

Revisión #9

Creado el 24 mayo 2024 16:57:37

Actualizado el 16 abril 2026 17:43:37 por Isabel Higuita