

Tu primera consulta (ejemplos prácticos)

En este paso realizarás tu **primera consulta exitosa** al API del Catálogo para obtener la lista de servicios. Aquí tienes ejemplos.

📋 Requisitos previos

1. Ya tienes un **token de acceso válido** (consulta [Obtener acceso](#))
2. Estás usando correctamente el **entorno sandbox** o **producción**, según corresponda

REST API y los Datos relacionales

De forma predeterminada, Catalogo solo recupera el valor de referencia de un campo relacional en los elementos. Para recuperar también datos anidados de un campo relacional, se puede utilizar el parámetro `fields` en REST. Esto le permite recuperar datos referenciados de un servicio.

Para definir el ámbito de los campos que se devuelven por tipo de colección, puede usar la sintaxis `<field>:<scope>` del parámetro `fields` de la siguiente manera:

```
GET /items/services
[]?fields[]=name
[]&fields[]=description
[]&fields[]=legal_framework_ids.description
[]&fields[]=services_electronic_gov_type_ids.electronic_gov_type_id.type
```

Otra alternativa sería:

```
GET /items/services
[]?fields[]=name,description,legal_framework_ids.description,services_electronic_gov_type_ids.electronic_gov_type
```

Método HTTP SEARCH

Al usar la API de REST para leer varios elementos mediante filtros (muy) avanzados, es posible que se encuentre con el problema de que la dirección URL simplemente no puede contener suficientes datos para incluir la estructura de consulta completa. En esos casos, puede usar el método HTTP SEARCH como reemplazo directo de GET, donde se le permite colocar la consulta en el cuerpo de la solicitud de la siguiente manera:

Antes:

```
GET /items/services?filter[name][_eq]=Hello World
```

Después:

```
SEARCH /items/services
```

```
{
  "query": {
    "filter": {
      "name": {
        "_eq": "Hello World"
      }
    }
  }
}
```

Hay mucha discusión sobre si se debe o no poner un cuerpo en una solicitud GET, usar POST para crear consultas de búsqueda o confiar en un método completamente diferente. A partir de ahora, hemos optado por alinearnos con la especificación del método HTTP SEARCH de IETF.

Lectura útil:

- [Método de búsqueda HTTP \(IETF, 2021\)](#)
- [Definición de un nuevo método HTTP: HTTP SEARCH \(Tim Perry, 2021\)](#)
- [HTTP GET con cuerpo de solicitud \(StackOverflow, 2009 y en curso\)](#)
- [Uso del cuerpo de Elastic Search GET \(elástico, s.f.\)](#)
- [Dropbox comienza a usar POST y por qué se trata de un diseño deficiente de la API. \(Evert Pot, 2015\)](#)

Códigos de error

A continuación, se muestran los códigos de error globales utilizados en Catalogo y lo que significan.

Código de error	Estado	Descripción
FAILED_VALIDATION	400	Error en la validación de este elemento en particular
FORBIDDEN	403	No se le permite realizar la acción actual
INVALID_TOKEN	403	El token proporcionado no es válido
TOKEN_EXPIRED	401	El token proporcionado es válido, pero ha caducado

Código de error	Estado	Descripción
INVALID_CREDENTIALS	401	El nombre de usuario/contraseña o el token de acceso son incorrectos
INVALID_IP	401	Su dirección IP no está en la lista de permitidos para ser utilizada con este usuario
INVALID_OTP	401	Se proporcionó una OTP incorrecta
INVALID_PAYLOAD	400	La carga útil proporcionada no es válida
INVALID_QUERY	400	No se pueden utilizar los parámetros de consulta solicitados
UNSUPPORTED_MEDIA_TYPE	415	El formato de carga útil o el encabezado proporcionados no son compatibles <code>Content-Type</code>
REQUESTS_EXCEEDED	429	Alcanza el límite de velocidad
ROUTE_NOT_FOUND	404	El punto de conexión no existe
SERVICE_UNAVAILABLE	503	No se pudo usar el servicio externo
UNPROCESSABLE_CONTENT	422	Intentaste hacer algo ilegal

“ Seguridad

Para evitar que se filtren los elementos existentes, todas las acciones de los elementos no existentes devolverán un error FORBIDDEN.

Para conocer más sobre los [items](#), [consultas](#) y [filtrado](#) en catalogo visita los enlaces correspondientes.

Revisión #5

Creado 2 julio 2025 14:11:22 por Luis Matos

Actualizado 23 julio 2025 02:47:03 por Luis Matos