



# Implementación de REST de Astra

## Astra Automation

NetApp  
August 11, 2025

# Tabla de contenidos

- Implementación de REST de Astra ..... 1
  - Diseño del núcleo..... 1
    - Servicios web REST ..... 1
    - Recursos y colecciones ..... 2
    - Detalles de HTTP..... 3
    - Formato de URL..... 6
  - Recursos y extremos ..... 7
    - Resumen de recursos DE ASTRA Control REST ..... 7
    - Recursos adicionales y extremos ..... 10
- Consideraciones adicionales ..... 10
  - Seguridad RBAC ..... 10
  - Trabajar con colecciones ..... 11
  - Diagnóstico y soporte..... 12
  - Revocar un token de API ..... 12

# Implementación de REST de Astra

## Diseño del núcleo

### Servicios web REST

La transferencia de estado representacional (REST) es un estilo para crear aplicaciones web distribuidas. Cuando se aplica al diseño de una API de servicios web, establece un conjunto de tecnologías generales y prácticas recomendadas para exponer recursos basados en servidor y administrar sus estados. Aunque REST proporciona una base consistente para el desarrollo de aplicaciones, los detalles de cada API pueden variar en función de las opciones de diseño específicas. Debe conocer las características de la API REST de Astra Control antes de utilizarla con una implementación en directo.

### Recursos y representación estatal

Los recursos son los componentes básicos de un sistema basado en la Web. Al crear una aplicación DE SERVICIOS web DE REST, las tareas de diseño más tempranas incluyen:

- Identificación de recursos basados en sistemas o servidores

Cada sistema utiliza y mantiene los recursos. Un recurso puede ser un archivo, una transacción comercial, un proceso o una entidad administrativa. Una de las primeras tareas en el diseño de una aplicación basada en servicios web DE REST es identificar los recursos.

- Definición de estados de recursos y operaciones estatales asociadas

Los recursos siempre se encuentran en uno de un número limitado de estados. Los estados, así como las operaciones asociadas utilizadas para afectar los cambios de estado, deben estar claramente definidos.

### Extremos de URI

Todos los recursos REST deben definirse y ponerse a disposición mediante un esquema de direccionamiento bien definido. Los extremos en los que se encuentran e identifican los recursos utilizan un identificador uniforme de recursos (URI). El URI proporciona un marco general para crear un nombre único para cada recurso de la red. El Localizador uniforme de recursos (URL) es un tipo de URI que se utiliza con los servicios web para identificar y acceder a los recursos. Los recursos normalmente se exponen en una estructura jerárquica similar a un directorio de archivos.

### Mensajes HTTP

El Protocolo de transferencia de hipertexto (HTTP) es el protocolo utilizado por el cliente y servidor de servicios web para intercambiar mensajes de solicitud y respuesta sobre los recursos. Como parte del diseño de una aplicación de servicios web, los métodos HTTP se asignan a los recursos y a las correspondientes acciones de administración del estado. HTTP no tiene estado. Por lo tanto, para asociar un conjunto de solicitudes y respuestas relacionadas como parte de una transacción, se debe incluir información adicional en los encabezados HTTP transportados con los flujos de datos de solicitud y respuesta.

## Formato JSON

Aunque la información se puede estructurar y transferir entre un cliente de servicios web y un servidor de varias maneras, la opción más popular es la notación de objetos JavaScript (JSON). JSON es un estándar del sector para representar estructuras de datos simples en texto sin formato y se utiliza para transferir información de estado que describe los recursos. La API REST de Astra Control utiliza JSON para dar formato a los datos transportados en el cuerpo de cada solicitud y respuesta HTTP.

## Recursos y colecciones

La API REST de Astra Control proporciona acceso a las instancias de recursos y las colecciones de instancias de recursos.



Conceptualmente UN RESTO **recurso** es similar a un **objeto** tal y como se define con los lenguajes y sistemas de programación orientada a objetos (OOP). A veces estos términos se utilizan indistintamente. Pero, en general, se prefiere "recurso" cuando se utiliza en el contexto de la API de REST externa mientras que "objeto" se utiliza para los datos de instancia con estado correspondientes almacenados en el servidor.

### Atributos de los recursos de Astra

La API REST de Astra Control cumple los principios de diseño RESTful. Cada instancia de recurso de Astra se crea según un tipo de recurso bien definido. Un conjunto de instancias de recursos del mismo tipo se denomina **colección**. Las llamadas API actúan sobre recursos individuales o colecciones de recursos.

### Tipos de recursos

Los tipos de recursos incluidos con la API REST de Astra Control tienen las siguientes características:

- Cada tipo de recurso se define mediante un esquema (normalmente en JSON)
- Cada esquema de recursos incluye el tipo de recurso y la versión
- Los tipos de recursos son globalmente únicos

### Instancias de recursos

Las instancias de recursos disponibles a través de la API REST de Astra Control tienen las siguientes características:

- Las instancias de recursos se crean en función de un único tipo de recurso
- El tipo de recurso se indica mediante el valor Tipo de soporte
- Las instancias se componen de datos con estado que el servicio Astra mantiene
- Se puede acceder a cada instancia mediante una dirección URL única y de larga duración
- En los casos en que una instancia de recurso puede tener más de una representación, se pueden utilizar diferentes tipos de medios para solicitar la representación deseada

### Colecciones de recursos

Las colecciones de recursos disponibles a través de la API REST de Astra Control tienen las siguientes características:

- El conjunto de instancias de recursos de un único tipo de recurso se conoce como una colección
- Las colecciones de recursos tienen una URL única y de larga duración

## Identificadores de instancia

Cada instancia de recurso tiene asignado un identificador cuando se crea. Este identificador es un valor UUIDv4 de 128 bits. Los valores UUIDv4 asignados son globalmente únicos e inmutables. Después de emitir una llamada API que crea una nueva instancia, se devuelve una URL con el ID asociado al llamante en un `Location` Encabezado de la respuesta HTTP. Puede extraer el identificador y utilizarlo en llamadas posteriores cuando haga referencia a la instancia del recurso.



El identificador de recurso es la clave principal utilizada para las colecciones.

## Estructura común para los recursos de Astra

Cada recurso de Astra Control se define mediante una estructura común.

### Datos comunes

Cada recurso de Astra contiene los valores clave que se muestran en la siguiente tabla.

Clave	Descripción
tipo	Un tipo de recurso único global que se conoce como el <b>tipo de recurso</b> .
versión	Un identificador de versión que se conoce como <b>versión de recurso</b> .
id	Identificador único global que se conoce como el <b>identificador de recurso</b> .
metadatos	Un objeto JSON que contiene diversa información, incluidas las etiquetas de usuario y del sistema.

### Objeto de metadatos

El objeto JSON de metadatos incluido con cada recurso de Astra contiene los valores clave que se muestran en la siguiente tabla.

Clave	Descripción
etiquetas	Cabina JSON de etiquetas especificadas por el cliente asociadas con el recurso.
CreationTimestamp	Cadena JSON que contiene una Marca de tiempo que indica cuándo se creó el recurso.
Modificación.Marca de hora	Cadena JSON que contiene una Marca de tiempo con formato ISO-8601 que indica cuándo se modificó por última vez el recurso.
CreatedBy	Cadena JSON que contiene el identificador UUIDv4 del ID de usuario que creó el recurso. Si el recurso fue creado por un componente interno del sistema y no hay un UUID asociado con la entidad de creación, se utiliza el UUID <b>nulo</b> .

### Estado del recurso

Recursos seleccionados a `state` valor que se utiliza para orquestar transiciones de ciclo de vida y controlar el acceso.

## Detalles de HTTP

La API REST de Astra Control utiliza HTTP y los parámetros relacionados para actuar en las instancias y colecciones de recursos. A continuación se presentan los detalles de la implementación HTTP.

## Transacciones API y modelo CRUD

La API REST de Astra Control implementa un modelo transaccional con operaciones bien definidas y transiciones de estado.

### Transacción de API de solicitud y respuesta

Cada llamada de API REST se realiza como una solicitud HTTP al servicio Astra. Cada solicitud genera una respuesta asociada al cliente. Este par de solicitud-respuesta puede considerarse una transacción API.

### Compatibilidad con el modelo operativo CRUD

Se accede a cada una de las instancias y colecciones de recursos disponibles a través de la API REST de Astra Control basándose en el modelo **CRUD**. Hay cuatro operaciones, cada una de las cuales se asigna a un único método HTTP. Entre las operaciones se incluyen:

- Cree
- Lea
- Actualizar
- Eliminar

Para algunos de los recursos de Astra, sólo se admite un subconjunto de estas operaciones. Debe revisar el ["Referencia de API en línea"](#) Para obtener más información acerca de una llamada API específica.

### Métodos HTTP

Los métodos o verbos HTTP soportados por la API se presentan en la tabla siguiente.

Método	CRUD	Descripción
OBTENGA	Lea	Recupera propiedades de objeto para una instancia o colección de recursos. Esto se considera una operación de <b>lista</b> cuando se utiliza con una colección.
PUBLICAR	Cree	Crea una nueva instancia de recurso basada en los parámetros de entrada. La URL a largo plazo se devuelve en un <code>Location</code> encabezado de respuesta.
PUESTO	Actualizar	Actualiza una instancia de recurso completa con el cuerpo de solicitud JSON proporcionado. Se conservan los valores clave que no pueden modificarse el usuario.
ELIMINAR	Eliminar	Elimina una instancia de recurso existente.

### Encabezados de solicitud y respuesta

En la siguiente tabla se resumen los encabezados HTTP utilizados con la API REST de Astra Control.



Consulte ["RFC 7232"](#) y.. ["RFC 7233"](#) si quiere más información.

Encabezado	Tipo	Notas de uso
Accept	Solicitud	Si el valor es "/" o no se proporciona, <code>application/json</code> . Se devuelve en el encabezado de respuesta <code>Content-Type</code> . Si el valor se establece en Tipo de medio de recurso de Astra, se devuelve el mismo tipo de medio en el encabezado Tipo de contenido.
Autorización	Solicitud	Token del portador con la clave API para el usuario.
Tipo de contenido	Respuesta	Devuelto en función del <code>Accept</code> solicite el encabezado.
ETag	Respuesta	Se incluye con un éxito según se define con RFC 7232. El valor es una representación hexadecimal del valor MD5 para todo el recurso JSON.
Coincidencia IF	Solicitud	Cabecera de solicitud de condición previa implementada como se describe en la sección 3.1 RFC 7232 y soporte para solicitudes <b>PUT</b> .
If-Modified-Since	Solicitud	Cabecera de solicitud de condición previa implementada como se describe en la sección 3.4 RFC 7232 y soporte para solicitudes <b>PUT</b> .
If-Unmodified-since	Solicitud	Cabecera de solicitud de condición previa implementada como se describe en la sección 3.4 RFC 7232 y soporte para solicitudes <b>PUT</b> .
Ubicación	Respuesta	Contiene la dirección URL completa del recurso recién creado.

## Parámetros de consulta

Los siguientes parámetros de consulta están disponibles para su uso con colecciones de recursos. Consulte ["Trabajar con colecciones"](#) si quiere más información.

Parámetro de consulta	Descripción
incluya	Contiene los campos que se deben devolver al leer una colección.
filtro	Indica los campos que deben coincidir para que se devuelva un recurso al leer una colección.
OrderBy	Determina el orden de los recursos devueltos al leer una colección.
límite	Limita el número máximo de recursos devueltos al leer una colección.
omitir	Establece el número de recursos que se van a transferir y omitir al leer una colección.
cuenta	Indica si se debe devolver el número total de recursos en el objeto de metadatos.

## códigos de estado HTTP

A continuación se describen los códigos de estado HTTP utilizados por la API DE REST de Astra Control.



La API REST de Astra Control también utiliza el estándar **Detalles del problema para API de HTTP**. Consulte ["Diagnóstico y soporte"](#) si quiere más información.

Codificación	Significado	Descripción
200	DE ACUERDO	Indica que las llamadas que no crean una nueva instancia de recurso tienen éxito.
201	Creado	Se ha creado correctamente un objeto y el encabezado de respuesta de ubicación incluye el identificador único del objeto.
204	Sin contenido	La solicitud se ha realizado correctamente aunque no se ha devuelto ningún contenido.
400	Solicitud incorrecta	La entrada de la solicitud no se reconoce o no es apropiada.
401	No autorizado	El usuario no está autorizado y debe autenticar.
403	Prohibido	Se deniega el acceso debido a un error de autorización.
404	No encontrado	El recurso al que se hace referencia en la solicitud no existe.
409	Conflicto	Error al intentar crear un objeto porque el objeto ya existe.
500	Error interno	Se ha producido un error interno general en el servidor.
503	Servicio no disponible	El servicio no está listo para atender la solicitud por algún motivo.

## Formato de URL

La estructura general de la URL utilizada para acceder a una instancia de recurso o a una colección a través de la API DE REST está compuesta por varios valores. Esta estructura refleja el modelo de objeto subyacente y el diseño del sistema.

### Cuenta como la raíz

La raíz de la ruta de recursos a cada extremo DE REST es la cuenta Astra. Por lo tanto, todas las rutas de la URL comienzan con `/account/{account_id}` donde `account_id` Es el valor único UUIDv4 de la cuenta. Estructura interna esto refleja un diseño en el que todo el acceso a los recursos se basa en una cuenta específica.

### Categoría de recurso de extremo

Los extremos de recursos de Astra se dividen en tres categorías diferentes:

- Núcleo (`/core`)
- Aplicación gestionada (`/k8s`)
- Topología (`/topology`)

Consulte "[Recursos](#)" si quiere más información.

### Versión de categoría

Cada una de las tres categorías de recursos tiene una versión global que controla la versión de los recursos a los que se tiene acceso. Por convención y definición, pasar a una nueva versión principal de una categoría de recursos (como, por ejemplo, de `/v1` para `/v2`) Introducirá cambios de ruptura en la API.

### Instancia o colección de recursos

Se puede usar una combinación de tipos de recursos e identificadores en la ruta de acceso, en función de si se accede a una instancia de recurso o a una recopilación.



## Ejemplo

- Ruta de recursos

En función de la estructura presentada anteriormente, una ruta típica a un punto final es:  
`/accounts/{account_id}/core/v1/users.`

- Complete la dirección URL

La dirección URL completa del punto final correspondiente es:  
`https://astra.netapp.io/accounts/{account_id}/core/v1/users.`

## Recursos y extremos

Puede acceder a los recursos proporcionados a través de la API REST de Astra Control para automatizar una implementación de Astra. Cada recurso está disponible mediante uno o varios extremos. A continuación se proporciona una introducción a los recursos DE REST que se pueden utilizar como parte de una puesta en marcha de automatización.



El formato de la ruta y la dirección URL completa que se utiliza para acceder a los recursos de Astra Control se basa en varios valores. Consulte ["Formato de URL"](#) si quiere más información. Consulte también ["Referencia de API en línea"](#) Para obtener más información sobre el uso de los recursos y puntos finales de Astra.

## Resumen de recursos DE ASTRA Control REST

Los extremos de recursos principales proporcionados en la API REST de Astra Control se organizan en tres categorías. Se puede tener acceso a cada recurso con el conjunto completo de operaciones CRUD (crear, leer, actualizar, eliminar) excepto donde se indique.

La columna **Release** indica la versión Astra cuando se introdujo el recurso por primera vez. Este campo está negrita para los recursos que se han añadido más recientemente a la API DE REST.

### Recursos básicos

Los principales extremos de recursos proporcionan los servicios básicos necesarios para establecer y mantener el entorno de tiempo de ejecución de Astra.

Recurso	Liberar	Descripción
Cuenta	21.12	Los recursos de la cuenta le permiten gestionar los inquilinos aislados dentro del entorno de implementación Multitenant Astra Control.
ASUP	21.08	Los recursos de ASUP representan los paquetes de AutoSupport que se envían al soporte de NetApp.
Certificado	22.08	Los recursos de certificado representan los certificados instalados que se utilizan para la autenticación segura de las conexiones salientes.
Credencial	21.04	Los recursos de credenciales contienen información relacionada con la seguridad que se puede utilizar con los usuarios de Astra, los clústeres, los bloques y los back-ends de almacenamiento.

<b>Recurso</b>	<b>Liberar</b>	<b>Descripción</b>
Prestaciones	21.08	Los recursos de derechos representan las características y capacidades disponibles para una cuenta en función de las licencias y suscripciones activas.
Evento	21.04	Los recursos de eventos representan todos los eventos que se producen en el sistema, incluido el subconjunto clasificado como notificaciones.
Gancho de ejecución	21.12	Los recursos del enlace de ejecución representan secuencias de comandos personalizadas que se pueden ejecutar antes o después de realizar una instantánea de una aplicación administrada.
Función	21.08	Los recursos de la función representan las funciones Astra seleccionadas que puede consultar para determinar si están habilitadas o deshabilitadas en el sistema. El acceso está limitado a solo lectura.
Grupo	22.08	Los recursos de grupo representan los grupos Astra y los recursos asociados. Solo los grupos LDAP son compatibles con la versión actual.
Fuente de gancho	21.12	Los recursos de origen de enlace representan el código fuente real utilizado con un enlace de ejecución. Separar el código fuente del control de ejecución tiene varias ventajas, como permitir que se compartan las secuencias de comandos.
Grupo LDAP	22,1	Puede enumerar los grupos dentro del servidor LDAP configurado. El acceso a los grupos LDAP es de solo lectura.
Usuario LDAP	22,11	Puede enumerar los usuarios dentro del servidor LDAP configurado. El acceso a los usuarios LDAP es de solo lectura.
Licencia	21.08	Los recursos de licencia representan las licencias disponibles para una cuenta de Astra.
Notificación	21.04	Los recursos de notificación representan eventos Astra que tienen un destino de notificación. El acceso se proporciona por usuario.
Paquete	22.04	Los recursos del paquete proporcionan registro y acceso a definiciones de paquetes. Los paquetes de software constan de varios componentes, incluidos archivos, imágenes y otros artefactos.
Permiso	23,06	Los recursos de permisos representan los permisos relacionados con las operaciones dentro del sistema. La API ofrece acceso de solo lectura a los permisos.
Función	23,06	Los recursos de rol representan roles disponibles en el sistema. La API ofrece acceso de solo lectura a los roles.
Vinculación de roles	21.04	Los recursos de vinculación de roles representan las relaciones entre pares específicos de usuarios y cuentas. Además del vínculo entre los dos, se especifica un conjunto de permisos para cada uno mediante una función específica.
Ajuste	21.08	Los recursos de configuración representan una colección de pares clave-valor que describen una función para una cuenta Astra específica.
Suscripción	21.08	Los recursos de suscripción representan las suscripciones activas de una cuenta Astra.

Recurso	Liberar	Descripción
Tarea	22,11	Los recursos de la tarea proporcionan acceso de sólo lectura a la tarea administrada y se pueden utilizar para mostrar el estado de las tareas internas de ejecución prolongada.
Token	21.04	Los recursos de token representan los tokens disponibles para acceder mediante programación a la API REST de Astra Control.
Notificación sin leer	21.04	Los recursos de notificación no leídos representan notificaciones asignadas a un usuario específico pero aún no leídas.
Renovar	22.04	Los recursos de actualización proporcionan acceso a los componentes de software y la capacidad de iniciar actualizaciones.
Usuario	21.04	Los recursos de usuario representan a los usuarios de Astra capaces de acceder al sistema en función de su función definida.

### Recursos de aplicaciones gestionados

Los extremos de recursos de la aplicación gestionada proporcionan acceso a las aplicaciones de Kubernetes gestionadas.

Recurso	Liberar	Descripción
Activo de aplicación	21.04	Los recursos de activos de aplicación representan colecciones internas de información de estado necesarias para gestionar las aplicaciones Astra.
Backup de aplicaciones	21.04	Los recursos de backup de la aplicación representan backups de las aplicaciones gestionadas.
Instantánea de aplicación	21.04	Los recursos de instantánea de la aplicación representan instantáneas de las aplicaciones gestionadas.
Anulación de gancho de ejecución	21.12	Los recursos de anulación del enlace para la ejecución le permiten deshabilitar los enlaces de ejecución predeterminados de NetApp preinstalados para aplicaciones específicas según sea necesario.
Programación	21.04	Los recursos de programación representan operaciones de protección de datos programadas para las aplicaciones gestionadas como parte de una política de protección de datos.

### Recursos de topología

Los extremos de recursos de topología proporcionan acceso a las aplicaciones no administradas y a los recursos de almacenamiento.

Recurso	Liberar	Descripción
Recurso de API	22,11	Los extremos de recursos de la API proporcionan acceso de solo lectura a los recursos de Kubernetes en un clúster gestionado específico.
APL	21.04	Los recursos de aplicaciones representan todas las aplicaciones de Kubernetes, incluidas las que no son gestionadas por Astra.
AppMirror	22.08	Los recursos de AppMirror representan los recursos de AppMirror para gestionar las relaciones de duplicación de aplicaciones.

Recurso	Liberar	Descripción
Cucharón	21.08	Los recursos de bucket representan los bloques cloud de S3 que se utilizan para almacenar backups de las aplicaciones que gestiona Astra.
Cloud	21.08	Los recursos cloud representan las nubes a las que los clientes de Astra pueden conectarse con el fin de gestionar clústeres y aplicaciones.
Clúster	21.08	Los recursos del clúster representan los clústeres de Kubernetes que no gestiona Kubernetes.
Nodo del clúster	21.12	Los recursos del nodo de clúster proporcionan una resolución adicional al permitirle acceder a cada nodo dentro de un clúster Kubernetes.
Clúster gestionado	21.08	Los recursos de clúster gestionados representan los clústeres de Kubernetes que gestiona actualmente Kubernetes.
Espacio de nombres	21.12	Los recursos de espacio de nombres ofrecen acceso a los espacios de nombres que se usan en un clúster de Kubernetes.
Back-end de almacenamiento	21.08	Los recursos de back-end de almacenamiento representan proveedores de servicios de almacenamiento que pueden utilizar los clústeres y aplicaciones gestionados de Astra.
Clase de almacenamiento	21.08	Los recursos de la clase de almacenamiento representan diferentes clases o tipos de almacenamiento detectados y disponibles para un clúster gestionado específico.
Volumen	21.04	Los recursos de volúmenes representan los volúmenes de almacenamiento de Kubernetes asociados con las aplicaciones gestionadas.

## Recursos adicionales y extremos

Existen varios recursos y puntos finales adicionales que puede utilizar para dar soporte a una implementación de Astra.



Estos recursos y extremos no se incluyen actualmente con la documentación de referencia de la API REST de Astra Control.

### Openapi

Los extremos de OpenAPI proporcionan acceso al documento JSON de OpenAPI actual y a otros recursos relacionados.

### OpenMetrics

Los extremos OpenMetrics proporcionan acceso a las métricas de la cuenta mediante el recurso OpenMetrics. Existe soporte para el modelo de puesta en marcha de Astra Control Center.

## Consideraciones adicionales

### Seguridad RBAC

La API REST de Astra admite el control de acceso basado en roles (RBAC) para otorgar y restringir el acceso a las funciones del sistema.

## Funciones de Astra

Cada usuario de Astra se asigna a una sola función que determina las acciones que se pueden realizar. Los roles se organizan en una jerarquía tal y como se describe en la tabla siguiente.

Función	Descripción
Propietario	Tiene todos los permisos de la función Admin y también puede eliminar cuentas Astra.
Admin	Tiene todos los permisos de la función Miembro y también puede invitar a los usuarios a unirse a una cuenta.
Miembro	Puede gestionar completamente la aplicación Astra y los recursos informáticos.
Visionador	Limitado a sólo ver recursos.

## RBAC mejorado con granularidad de espacio de nombres



Esta función se introdujo con la versión 22.04 de la API ASTRA REST.

Cuando se establece un enlace de roles para un usuario específico, se puede aplicar una restricción para limitar los espacios de nombres a los que el usuario tiene acceso. Hay varias formas de definir esta restricción como se describe en la tabla siguiente. Consulte el parámetro `roleConstraints` En la API de enlace de roles para obtener más información.

Espacios de nombres	Descripción
Todo	El usuario puede acceder a todos los espacios de nombres mediante el parámetro comodín "*". Este es el valor predeterminado para mantener la compatibilidad con versiones anteriores.
Ninguno	La lista de restricciones se especifica aunque esté vacía. Esto indica que el usuario no puede acceder a ningún espacio de nombres.
Lista Namespace	Se incluye el UUID de un espacio de nombres que restringe al usuario al espacio de nombres único. También se puede utilizar una lista separada por comas para permitir el acceso a varios espacios de nombres.
Etiqueta	Se especifica una etiqueta y se permite el acceso a todos los espacios de nombres coincidentes.

## Trabajar con colecciones

La API REST de Astra Control proporciona varias formas diferentes de acceder a las colecciones de recursos a través de los parámetros de consulta definidos.

### Selección de valores

Puede especificar qué parejas de clave-valor deben devolverse para cada instancia de recursos mediante el `include` parámetro. Todas las instancias se devuelven en el cuerpo de respuesta.

### Filtrado

El filtrado de recursos de recopilación permite al usuario de API especificar condiciones que determinan si un recurso se devuelve en el cuerpo de respuesta. La `filter` el parámetro se utiliza para indicar la condición de filtrado.

## Ordenación

La ordenación de recursos de recopilación permite al usuario de API especificar el orden en el que se devuelven los recursos en el cuerpo de respuesta. La `orderBy` el parámetro se utiliza para indicar la condición de filtrado.

## Paginación

Puede aplicar la paginación restringiendo el número de instancias de recursos devueltas en una solicitud mediante `limit` parámetro.

## Cuenta

Si incluye el parámetro booleano `count` establezca en `true`, el número de recursos de la matriz devuelta para una respuesta determinada se proporciona en la sección de metadatos.

## Diagnóstico y soporte

Hay varias funciones de soporte disponibles con la API REST de Astra Control que se pueden utilizar para diagnósticos y depuración.

## Recursos de API

Existen varias funciones de Astra expuestas a través de los recursos de API que proporcionan información y soporte de diagnóstico.

Tipo	Descripción
Evento	Actividades del sistema que se registran como parte del procesamiento de Astra.
Notificación	Subconjunto de los eventos que se consideran lo suficientemente importantes como para ser presentados al usuario.
Notificación sin leer	Las notificaciones que el usuario debe leer o recuperar aún.

## Revocar un token de API

Puede revocar un token de API en la interfaz web de Astra cuando ya no sea necesario.

### Antes de empezar

Necesitará credenciales para iniciar sesión en la interfaz de usuario web de Astra para su puesta en marcha. También debe identificar los tokens que desea revocar.

### Acerca de esta tarea

Después de que se revoque un token, éste se puede utilizar de forma inmediata y permanente.

### Pasos

1. Inicia sesión en Astra con las credenciales de tu cuenta de la siguiente manera:
  - a. Servicio Astra Control: "<https://astra.netapp.io>"
  - b. Astra Control Center: Utilice la URL de su entorno local tal y como se establece durante la instalación
2. Haga clic en el icono de figura situado en la parte superior derecha de la página y seleccione **acceso API**.
3. Seleccione el token o tokens que desea revocar.
4. En el cuadro desplegable **acciones**, haga clic en **revocar tokens**.

## Información de copyright

Copyright © 2025 NetApp, Inc. Todos los derechos reservados. Imprimido en EE. UU. No se puede reproducir este documento protegido por copyright ni parte del mismo de ninguna forma ni por ningún medio (gráfico, electrónico o mecánico, incluidas fotocopias, grabaciones o almacenamiento en un sistema de recuperación electrónico) sin la autorización previa y por escrito del propietario del copyright.

El software derivado del material de NetApp con copyright está sujeto a la siguiente licencia y exención de responsabilidad:

ESTE SOFTWARE LO PROPORCIONA NETAPP «TAL CUAL» Y SIN NINGUNA GARANTÍA EXPRESA O IMPLÍCITA, INCLUYENDO, SIN LIMITAR, LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN CONCRETO, CUYA RESPONSABILIDAD QUEDA EXIMIDA POR EL PRESENTE DOCUMENTO. EN NINGÚN CASO NETAPP SERÁ RESPONSABLE DE NINGÚN DAÑO DIRECTO, INDIRECTO, ESPECIAL, EJEMPLAR O RESULTANTE (INCLUYENDO, ENTRE OTROS, LA OBTENCIÓN DE BIENES O SERVICIOS SUSTITUTIVOS, PÉRDIDA DE USO, DE DATOS O DE BENEFICIOS, O INTERRUPCIÓN DE LA ACTIVIDAD EMPRESARIAL) CUALQUIERA SEA EL MODO EN EL QUE SE PRODUJERON Y LA TEORÍA DE RESPONSABILIDAD QUE SE APLIQUE, YA SEA EN CONTRATO, RESPONSABILIDAD OBJETIVA O AGRAVIO (INCLUIDA LA NEGLIGENCIA U OTRO TIPO), QUE SURJAN DE ALGÚN MODO DEL USO DE ESTE SOFTWARE, INCLUSO SI HUBIEREN SIDO ADVERTIDOS DE LA POSIBILIDAD DE TALES DAÑOS.

NetApp se reserva el derecho de modificar cualquiera de los productos aquí descritos en cualquier momento y sin aviso previo. NetApp no asume ningún tipo de responsabilidad que surja del uso de los productos aquí descritos, excepto aquello expresamente acordado por escrito por parte de NetApp. El uso o adquisición de este producto no lleva implícita ninguna licencia con derechos de patente, de marcas comerciales o cualquier otro derecho de propiedad intelectual de NetApp.

Es posible que el producto que se describe en este manual esté protegido por una o más patentes de EE. UU., patentes extranjeras o solicitudes pendientes.

LEYENDA DE DERECHOS LIMITADOS: el uso, la copia o la divulgación por parte del gobierno están sujetos a las restricciones establecidas en el subpárrafo (b)(3) de los derechos de datos técnicos y productos no comerciales de DFARS 252.227-7013 (FEB de 2014) y FAR 52.227-19 (DIC de 2007).

Los datos aquí contenidos pertenecen a un producto comercial o servicio comercial (como se define en FAR 2.101) y son propiedad de NetApp, Inc. Todos los datos técnicos y el software informático de NetApp que se proporcionan en este Acuerdo tienen una naturaleza comercial y se han desarrollado exclusivamente con fondos privados. El Gobierno de EE. UU. tiene una licencia limitada, irrevocable, no exclusiva, no transferible, no sublicenciable y de alcance mundial para utilizar los Datos en relación con el contrato del Gobierno de los Estados Unidos bajo el cual se proporcionaron los Datos. Excepto que aquí se disponga lo contrario, los Datos no se pueden utilizar, desvelar, reproducir, modificar, interpretar o mostrar sin la previa aprobación por escrito de NetApp, Inc. Los derechos de licencia del Gobierno de los Estados Unidos de América y su Departamento de Defensa se limitan a los derechos identificados en la cláusula 252.227-7015(b) de la sección DFARS (FEB de 2014).

## Información de la marca comercial

NETAPP, el logotipo de NETAPP y las marcas que constan en <http://www.netapp.com/TM> son marcas comerciales de NetApp, Inc. El resto de nombres de empresa y de producto pueden ser marcas comerciales de sus respectivos propietarios.