



Design central

Astra Automation

NetApp
August 11, 2025

Índice

- Design central 1
 - Serviços web REST 1
 - Recursos e representação do Estado 1
 - Pontos de extremidade URI 1
 - Mensagens HTTP 1
 - Formatação JSON 2
 - Recursos e coleções 2
 - Atributos dos recursos do Astra 2
 - Estrutura comum para recursos do Astra 3
 - Detalhes HTTP 3
 - Transações de API e o modelo CRUD 4
 - Métodos HTTP 4
 - Cabeçalhos de solicitação e resposta 4
 - Parâmetros de consulta 5
 - Códigos de status HTTP 5
 - Formato de URL 6

Design central

Serviços web REST

Representational State Transfer (REST) é um estilo para a criação de aplicações web distribuídas. Quando aplicado ao design de uma API de serviços da Web, ele estabelece um conjunto de tecnologias e práticas recomendadas principais para expor recursos baseados em servidor e gerenciar seus estados. Embora O REST forneça uma base consistente para o desenvolvimento de aplicativos, os detalhes de cada API podem variar com base nas escolhas específicas de design. Você deve estar ciente das características da API REST do Astra Control antes de usá-la com uma implantação ativa.

Recursos e representação do Estado

Os recursos são os componentes básicos de um sistema baseado na Web. Ao criar um aplicativo REST de serviços da Web, as tarefas iniciais de design incluem:

- Identificação de recursos baseados em sistema ou servidor

Cada sistema usa e mantém recursos. Um recurso pode ser um arquivo, transação comercial, processo ou entidade administrativa. Uma das primeiras tarefas no projeto de um aplicativo baseado em serviços web REST é identificar os recursos.

- Definição de estados de recursos e operações de estado associadas

Os recursos estão sempre em um de um número finito de estados. Os estados, bem como as operações associadas usadas para afetar as mudanças de estado, devem ser claramente definidos.

Pontos de extremidade URI

Todos os recursos REST devem ser definidos e disponibilizados usando um esquema de endereçamento bem definido. Os endpoints onde os recursos estão localizados e identificados usam um URI (Uniform Resource Identifier). O URI fornece uma estrutura geral para criar um nome exclusivo para cada recurso na rede. O Uniform Resource Locator (URL) é um tipo de URI usado com serviços da Web para identificar e acessar recursos. Os recursos são normalmente expostos em uma estrutura hierárquica semelhante a um diretório de arquivos.

Mensagens HTTP

O Hypertext Transfer Protocol (HTTP) é o protocolo usado pelo cliente e servidor de serviços da Web para trocar mensagens de solicitação e resposta sobre os recursos. Como parte do projeto de um aplicativo de serviços da Web, os métodos HTTP são mapeados para os recursos e as ações de gerenciamento de estado correspondentes. HTTP está sem estado. Portanto, para associar um conjunto de solicitações e respostas relacionadas como parte de uma transação, informações adicionais devem ser incluídas nos cabeçalhos HTTP carregados com os fluxos de dados de solicitação e resposta.

Formatação JSON

Embora as informações possam ser estruturadas e transferidas entre um cliente e um servidor de serviços da Web de várias maneiras, a opção mais popular é JavaScript Object Notation (JSON). JSON é um padrão da indústria para representar estruturas de dados simples em texto simples e é usado para transferir informações de estado descrevendo os recursos. A API REST DO Astra Control usa JSON para formatar os dados transportados no corpo de cada solicitação e resposta HTTP.

Recursos e coleções

A API REST do Astra Control fornece acesso a instâncias de recursos e coleções de instâncias de recursos.



Conceitualmente um resource * REST é semelhante a um objeto * conforme definido com as linguagens e sistemas de programação orientada a objetos (OOP). Às vezes, esses termos são usados de forma intercambiável. Mas, em geral, "recurso" é preferível quando usado no contexto da API REST externa enquanto "objeto" é usado para os dados de instância com estado correspondente armazenados no servidor.

Atributos dos recursos do Astra

A API REST do Astra Control está em conformidade com os princípios de design RESTful. Cada instância de recurso Astra é criada com base em um tipo de recurso bem definido. Um conjunto de instâncias de recursos do mesmo tipo é chamado de **Collection**. As chamadas API atuam sobre recursos individuais ou coleções de recursos.

Tipos de recursos

Os tipos de recursos incluídos na API REST do Astra Control têm as seguintes características:

- Cada tipo de recurso é definido usando um esquema (normalmente em JSON)
- Cada esquema de recursos inclui o tipo de recurso e a versão
- Os tipos de recursos são globalmente únicos

Instâncias de recursos

As instâncias de recursos disponíveis pela API REST do Astra Control têm as seguintes características:

- As instâncias de recurso são criadas com base em um único tipo de recurso
- O tipo de recurso é indicado utilizando o valor tipo de material
- As instâncias são compostas por dados com estado que são mantidos pelo serviço Astra
- Cada instância é acessível através de uma URL única e de longa duração
- Nos casos em que uma instância de recurso pode ter mais de uma representação, diferentes tipos de Mídia podem ser usados para solicitar a representação desejada

Coleções de recursos

As coleções de recursos disponíveis pela API REST do Astra Control têm as seguintes características:

- O conjunto de instâncias de recursos de um único tipo de recurso é conhecido como uma coleção
- Coleções de recursos têm uma URL única e de longa duração

Identificadores de instância

Cada instância de recurso recebe um identificador quando é criada. Este identificador é um valor UUIDv4 de 128 bits. Os valores UUIDv4 atribuídos são globalmente únicos e imutáveis. Depois de emitir uma chamada de API que cria uma nova instância, um URL com o ID associado é retornado ao chamador em um `Location` cabeçalho da resposta HTTP. Você pode extrair o identificador e usá-lo em chamadas subsequentes quando se refere à instância de recurso.



O identificador de recurso é a chave primária usada para coleções.

Estrutura comum para recursos do Astra

Todos os recursos do Astra Control são definidos usando uma estrutura comum.

Dados comuns

Cada recurso Astra contém os valores-chave mostrados na tabela a seguir.

| Chave | Descrição |
|-----------|---|
| tipo | Um tipo de recurso globalmente único que é conhecido como tipo de recurso . |
| versão | Um identificador de versão que é conhecido como a versão resource . |
| id | Um identificador global único que é conhecido como resource identifier . |
| metadados | Um objeto JSON contendo várias informações, incluindo rótulos de usuário e sistema. |

Objeto de metadados

O objeto JSON de metadados incluído com cada recurso Astra contém os valores-chave mostrados na tabela a seguir.

| Chave | Descrição |
|------------------------|---|
| etiquetas | Matriz JSON de rótulos especificados pelo cliente associados ao recurso. |
| CriaçãoTimestamp | String JSON contendo um carimbo de data/hora indicando quando o recurso foi criado. |
| Alteração do Timestamp | String JSON contendo um carimbo de data/hora formatado ISO-8601 indicando quando o recurso foi alterado pela última vez. |
| CreatedBy | String JSON contendo o identificador UUIDv4 do ID do usuário que criou o recurso. Se o recurso foi criado por um componente interno do sistema e não houver UUID associado à entidade criadora, o UUID null é usado. |

Estado do recurso

Recursos selecionados um `state` valor que é usado para orquestrar transições de ciclo de vida e controlar o acesso.

Detalhes HTTP

A API REST do Astra Control usa HTTP e parâmetros relacionados para agir nas instâncias e coleções de recursos. Detalhes da implementação HTTP são apresentados abaixo.

Transações de API e o modelo CRUD

A API REST do Astra Control implementa um modelo transacional com operações bem definidas e transições de estado.

Transação de API de solicitação e resposta

Cada chamada de API REST é executada como uma solicitação HTTP para o serviço Astra. Cada solicitação gera uma resposta associada de volta ao cliente. Esse par de solicitação-resposta pode ser considerado uma transação de API.

Suporte para o modelo operacional CRUD

Cada uma das instâncias e coleções de recursos disponíveis por meio da API REST do Astra Control é acessada com base no modelo **CRUD**. Existem quatro operações, cada uma delas mapeia para um único método HTTP. As operações incluem:

- Criar
- Leia
- Atualização
- Eliminar

Para alguns recursos do Astra, apenas um subconjunto dessas operações é suportado. Você deve revisar o ["Referência de API online"](#) para obter mais informações sobre uma chamada de API específica.

Métodos HTTP

Os métodos HTTP ou verbos suportados pela API são apresentados na tabela abaixo.

| Método | CRUD | Descrição |
|----------|-------------|--|
| OBTER | Leia | Recupera propriedades de objeto para uma instância ou coleção de recursos. Isso é considerado uma operação list quando usado com uma coleção. |
| POST | Criar | Cria uma nova instância de recurso com base nos parâmetros de entrada. O URL de longo prazo é retornado em um <code>Location</code> cabeçalho de resposta. |
| COLOQUE | Atualização | Atualiza uma instância de recurso inteira com o corpo de solicitação JSON fornecido. Os valores-chave que não são modificáveis pelo usuário são preservados. |
| ELIMINAR | Eliminar | Exclui uma instância de recurso existente. |

Cabeçalhos de solicitação e resposta

A tabela a seguir resume os cabeçalhos HTTP usados com a API REST do Astra Control.



"[RFC 7232](#)" Consulte e "[RFC 7233](#)" para obter mais informações.

| Colhedor | Tipo | Notas de utilização |
|---------------------|----------|---|
| Aceitar | Pedido | Se o valor for "/" ou não for fornecido, <code>application/json</code> será retornado no cabeçalho de resposta <code>Content-Type</code> . Se o valor estiver definido para o tipo de Mídia de recurso Astra, o mesmo tipo de Mídia será retornado no cabeçalho <code>Content-Type</code> . |
| Autorização | Pedido | Token de portador com a chave API para o usuário. |
| Tipo de conteúdo | Resposta | Devolvido com base no <code>Accept</code> cabeçalho da solicitação. |
| ETAG | Resposta | Incluído com um sucesso como definido com RFC 7232. O valor é uma representação hexadecimal do valor MD5 para todo o recurso JSON. |
| If-Match | Pedido | Um cabeçalho de solicitação de pré-condição implementado conforme descrito na seção 3,1 RFC 7232 e suporte para solicitações PUT . |
| Se-modificado-desde | Pedido | Um cabeçalho de solicitação de pré-condição implementado conforme descrito na seção 3,4 RFC 7232 e suporte para solicitações PUT . |
| If-Unmodified-since | Pedido | Um cabeçalho de solicitação de pré-condição implementado conforme descrito na seção 3,4 RFC 7232 e suporte para solicitações PUT . |
| Localização | Resposta | Contém a URL completa do recurso recém-criado. |

Parâmetros de consulta

Os seguintes parâmetros de consulta estão disponíveis para uso com coleções de recursos. Consulte ["Trabalhe com coleções"](#) para obter mais informações.

| Parâmetro de consulta | Descrição |
|-----------------------|--|
| incluir | Contém os campos que devem ser retornados ao ler uma coleção. |
| filtro | Indica os campos que devem corresponder para que um recurso seja retornado ao ler uma coleção. |
| Ordenar | Determina a ordem de classificação dos recursos retornados ao ler uma coleção. |
| limite | Limita o número máximo de recursos retornados ao ler uma coleção. |
| ignorar | Define o número de recursos para passar e pular ao ler uma coleção. |
| contar | Indica se o número total de recursos deve ser retornado no objeto metadados. |

Códigos de status HTTP

Os códigos de status HTTP usados pela API REST do Astra Control são descritos abaixo.



A API REST do Astra Control também usa o padrão **Detalhes do problema para APIs HTTP**. Consulte ["Diagnóstico e suporte"](#) para obter mais informações.

| Código | Significado | Descrição |
|--------|----------------------|---|
| 200 | OK | Indica sucesso para chamadas que não criam uma nova instância de recurso. |
| 201 | Criado | Um objeto é criado com sucesso e o cabeçalho de resposta de localização inclui o identificador exclusivo para o objeto. |
| 204 | Nenhum conteúdo | A solicitação foi bem-sucedida, embora nenhum conteúdo tenha sido retornado. |
| 400 | Pedido incorreto | A entrada de solicitação não é reconhecida ou é inadequada. |
| 401 | Não autorizado | O usuário não está autorizado e deve autorizar. |
| 403 | Proibido | O acesso é negado devido a um erro de autorização. |
| 404 | Não encontrado | O recurso referido na solicitação não existe. |
| 409 | Conflito | Uma tentativa de criar um objeto falhou porque o objeto já existe. |
| 500 | Erro interno | Ocorreu um erro interno geral no servidor. |
| 503 | Serviço indisponível | O serviço não está pronto para lidar com a solicitação por algum motivo. |

Formato de URL

A estrutura geral da URL usada para acessar uma instância ou coleção de recursos através da API REST é composta por vários valores. Esta estrutura reflete o modelo de objeto subjacente e o design do sistema.

Conta como raiz

A raiz do caminho do recurso para cada endpoint REST é a conta Astra. E assim todos os caminhos no URL começam com `/account/{account_id}` onde `account_id` é o valor UUIDv4 exclusivo para a conta. Estrutura interna isso reflete um design em que todo o acesso a recursos é baseado em uma conta específica.

Categoria de recurso de endpoint

Os pontos de extremidade de recursos do Astra se enquadram em três categorias diferentes:

- (`/core`Núcleo`)
- Aplicativo gerenciado (`/k8s`)
- Topologia (`/topology`)

Consulte "[Recursos](#)" para obter mais informações.

Versão da categoria

Cada uma das três categorias de recursos tem uma versão global que controla a versão dos recursos acessados. Por convenção e definição, mover para uma nova versão principal de uma categoria de recurso (como, de `/v1` para `/v2`) introduzirá alterações de quebra na API.

Instância ou coleção de recursos

Uma combinação de tipos de recursos e identificadores pode ser usada no caminho, com base no acesso a uma instância ou coleção de recursos.

Exemplo

- Caminho do recurso

Com base na estrutura apresentada acima, um caminho típico para um endpoint é:

`/accounts/{account_id}/core/v1/users.`

- URL completo

O URL completo para o endpoint correspondente é:

`https://astra.netapp.io/accounts/{account_id}/core/v1/users.`

Informações sobre direitos autorais

Copyright © 2025 NetApp, Inc. Todos os direitos reservados. Impresso nos EUA. Nenhuma parte deste documento protegida por direitos autorais pode ser reproduzida de qualquer forma ou por qualquer meio — gráfico, eletrônico ou mecânico, incluindo fotocópia, gravação, gravação em fita ou storage em um sistema de recuperação eletrônica — sem permissão prévia, por escrito, do proprietário dos direitos autorais.

O software derivado do material da NetApp protegido por direitos autorais está sujeito à seguinte licença e isenção de responsabilidade:

ESTE SOFTWARE É FORNECIDO PELA NETAPP "NO PRESENTE ESTADO" E SEM QUAISQUER GARANTIAS EXPRESSAS OU IMPLÍCITAS, INCLUINDO, SEM LIMITAÇÕES, GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO E ADEQUAÇÃO A UM DETERMINADO PROPÓSITO, CONFORME A ISENÇÃO DE RESPONSABILIDADE DESTES DOCUMENTOS. EM HIPÓTESE ALGUMA A NETAPP SERÁ RESPONSÁVEL POR QUALQUER DANO DIRETO, INDIRETO, INCIDENTAL, ESPECIAL, EXEMPLAR OU CONSEQUENCIAL (INCLUINDO, SEM LIMITAÇÕES, AQUISIÇÃO DE PRODUTOS OU SERVIÇOS SOBRESSALIENTES; PERDA DE USO, DADOS OU LUCROS; OU INTERRUPÇÃO DOS NEGÓCIOS), INDEPENDENTEMENTE DA CAUSA E DO PRINCÍPIO DE RESPONSABILIDADE, SEJA EM CONTRATO, POR RESPONSABILIDADE OBJETIVA OU PREJUÍZO (INCLUINDO NEGLIGÊNCIA OU DE OUTRO MODO), RESULTANTE DO USO DESTES DOCUMENTOS, MESMO SE ADVERTIDA DA RESPONSABILIDADE DE TAL DANO.

A NetApp reserva-se o direito de alterar quaisquer produtos descritos neste documento, a qualquer momento e sem aviso. A NetApp não assume nenhuma responsabilidade nem obrigação decorrentes do uso dos produtos descritos neste documento, exceto conforme expressamente acordado por escrito pela NetApp. O uso ou a compra deste produto não representam uma licença sob quaisquer direitos de patente, direitos de marca comercial ou quaisquer outros direitos de propriedade intelectual da NetApp.

O produto descrito neste manual pode estar protegido por uma ou mais patentes dos EUA, patentes estrangeiras ou pedidos pendentes.

LEGENDA DE DIREITOS LIMITADOS: o uso, a duplicação ou a divulgação pelo governo estão sujeitos a restrições conforme estabelecido no subparágrafo (b)(3) dos Direitos em Dados Técnicos - Itens Não Comerciais no DFARS 252.227-7013 (fevereiro de 2014) e no FAR 52.227- 19 (dezembro de 2007).

Os dados aqui contidos pertencem a um produto comercial e/ou serviço comercial (conforme definido no FAR 2.101) e são de propriedade da NetApp, Inc. Todos os dados técnicos e software de computador da NetApp fornecidos sob este Contrato são de natureza comercial e desenvolvidos exclusivamente com despesas privadas. O Governo dos EUA tem uma licença mundial limitada, irrevogável, não exclusiva, intransferível e não sublicenciável para usar os Dados que estão relacionados apenas com o suporte e para cumprir os contratos governamentais desse país que determinam o fornecimento de tais Dados. Salvo disposição em contrário no presente documento, não é permitido usar, divulgar, reproduzir, modificar, executar ou exibir os dados sem a aprovação prévia por escrito da NetApp, Inc. Os direitos de licença pertencentes ao governo dos Estados Unidos para o Departamento de Defesa estão limitados aos direitos identificados na cláusula 252.227-7015(b) (fevereiro de 2014) do DFARS.

Informações sobre marcas comerciais

NETAPP, o logotipo NETAPP e as marcas listadas em <http://www.netapp.com/TM> são marcas comerciais da NetApp, Inc. Outros nomes de produtos e empresas podem ser marcas comerciais de seus respectivos proprietários.