Kerndesign

Astra Automation

NetApp
December 01, 2023

Inhalt

Kerndesign

REST-Web-Services

Representational State Transfer (REST) ist ein Stil für die Erstellung von verteilten Web-Anwendungen. Bei der Anwendung auf das Design einer Web-Services-API, stellt sie eine Reihe von Mainstream-Technologien und Best Practices für die Offenlegung serverbasierter Ressourcen und die Verwaltung ihrer Status. REST bietet eine konsistente Grundlage für die Applikationsentwicklung, doch je nach den jeweiligen Designoptionen können die Details jeder API variieren. Vor dem Einsatz in einer Live-Implementierung sollten Sie sich der Merkmale der Astra Control REST API bewusst sein.

Ressourcen- und Zustandsdarstellung

Ressourcen sind die Grundkomponenten eines webbasierten Systems. Beim Erstellen einer ANWENDUNG FÜR REST-Webservices umfassen die frühen Designaufgaben Folgendes:

- Identifizierung von System- oder serverbasierten Ressourcen

  Jedes System nutzt und verwaltet Ressourcen. Eine Ressource kann eine Datei, eine Geschäftstransaktion, ein Prozess oder eine administrative Einheit sein. Eine der ersten Aufgaben bei der Entwicklung einer auf REST-Webservices basierenden Applikation ist die Identifizierung der Ressourcen.

- Definition von Ressourcenstatus und zugehörigen Statusoperationen

  Die Ressourcen befinden sich immer in einer endlichen Anzahl von Staaten. Die Zustände sowie die damit verbundenen Operationen, die zur Auswirkung der Statusänderungen verwendet werden, müssen klar definiert werden.

URI-Endpunkte

Jede REST-Ressource muss definiert und über ein gut definiertes Adressierungssystem verfügbar gemacht werden. Die Endpunkte, in denen die Ressourcen gefunden und identifiziert werden, verwenden einen einheitlichen Resource Identifier (URI). Der URI bietet ein allgemeines Framework zum Erstellen eines eindeutigen Namens für jede Ressource im Netzwerk. Der Uniform Resource Locator (URL) ist ein URI-Typ, der mit Webservices zur Identifizierung und zum Zugriff von Ressourcen verwendet wird. Ressourcen werden in der Regel in einer hierarchischen Struktur ausgesetzt, die einem Dateiverzeichnis ähnelt.

HTTP-Meldungen

Hypertext Transfer Protocol (HTTP) ist das Protokoll, das vom Webservice-Client und -Server zum Austausch von Anforderungs- und Antwortmeldungen zu den Ressourcen verwendet wird. Im Rahmen der Entwicklung einer Web-Services-Anwendung werden HTTP-Methoden den Ressourcen und entsprechenden Statusmanagement-Aktionen zugeordnet. HTTP ist statusfrei. Um im Rahmen einer Transaktion eine Reihe verwandter Anforderungen und Antworten zuzuordnen, müssen daher zusätzliche Informationen in die HTTP-Header enthalten sein, die mit den Anforderungs- und Antwortdatenströmen verwendet werden.

JSON-Formatierung

Während Informationen auf verschiedene Weise zwischen einem Web-Services-Client und Server strukturiert und übertragen werden können, ist die beliebteste Option JavaScript Object Notation (JSON). JSON ist ein Branchenstandard für die Darstellung einfacher Datenstrukturen im Klartext und wird zur Übertragung von Zustandsdaten zur Beschreibung der Ressourcen verwendet. Die Astra Control REST API verwendet JSON, um die Daten zu formatieren, die im Körper von jeder HTTP-Anfrage und Antwort.

Ressourcen und Sammlungen

DIE Rest-API von Astra Control bietet Zugriff auf Ressourceninstanzen und Ressourcensammlungen.

Konzeptionell ist eine REST **Ressource** ähnlich wie ein **Objekt** wie mit den objektorientierten Programmiersprachen und -Systemen definiert. Manchmal werden diese Begriffe synonym verwendet. Aber im Allgemeinen wird „Ressource“ bevorzugt, wenn sie im Kontext der externen REST API verwendet wird, während „Objekt“ für die entsprechenden zustandsorientierte Instanz Daten verwendet wird, die auf dem Server gespeichert sind.

Eigenschaften der Astra-Ressourcen

Die Astra Control REST API entspricht den Prinzipien des RESTful Designs. Jede Astra-Ressourceninstanz wird auf Basis eines klar definierten Ressourcentyps erstellt. Eine Reihe von Ressourceninstanzen desselben Typs wird als **Sammlung** bezeichnet. Die API-Aufrufe wirken sich auf einzelne Ressourcen oder Ressourcensammlungen aus.

Ressourcentypen

Die in der Astra Control REST API enthaltenen Ressourcentypen weisen folgende Merkmale auf:

- Jeder Ressourcentyp wird mit einem Schema definiert (in der Regel in JSON)
- Jedes Ressourcenschema enthält den Ressourcentyp und die -Version
- Ressourcentypen sind global eindeutig

Ressourceninstanzen

Die über die Astra Control REST-API verfügbaren Ressourceinstanzen weisen folgende Merkmale auf:

- Ressourceninstanzen werden auf Basis eines einzelnen Ressourcentyps erstellt
- Der Ressourcentyp wird mit dem Wert Medientyp angezeigt
- Instanzen bestehen aus statusorientierten Daten, die vom Astra-Service gewartet werden
- Auf jede Instanz kann über eine eindeutige und langlebige URL zugegriffen werden
- In Fällen, in denen eine Ressourceninstanz mehr als eine Darstellung haben kann, können verschiedene Medientypen verwendet werden, um die gewünschte Darstellung anzufordern

Ressourcensammlungen

Die Ressourcensammlungen, die über die ASTRA Control REST-API verfügbar sind, weisen folgende Merkmale auf:

- Der Satz von Ressourceninstanzen eines einzelnen Ressourcentyps wird als Sammlung bezeichnet
- Ressourcensammlungen haben eine einzigartige und langlebige URL

Instanz-IDs

Jeder Ressourceninstanz wird bei der Erstellung eine Kennung zugewiesen. Diese Kennung ist ein 128-Bit UUIDv4-Wert. Die zugewiesenen UUUIDv4-Werte sind global eindeutig und unveränderbar. Nachdem ein API-Aufruf ausgegeben wurde, der eine neue Instanz erstellt, wird eine URL mit der zugehörigen id an den Anrufer in A zurückgegeben `Location` Kopfzeile der HTTP-Antwort. Sie können die Kennung extrahieren und bei nachfolgenden Anrufen verwenden, wenn Sie sich auf die Ressourceninstanz beziehen.

Die Ressourcen-ID ist der primäre Schlüssel, der für Sammlungen verwendet wird.

Gemeinsame Struktur für Astra-Ressourcen

Jede Astra Control-Ressource ist mit einer gemeinsamen Struktur definiert.

Einheitliche Daten

Jede Astra-Ressource enthält die in der folgenden Tabelle aufgeführten Schlüsselwerte.

Taste	Beschreibung
Typ	Ein global eindeutiger Ressourcentyp, der als **Ressourcentyp** bezeichnet wird.
Version	Eine Version-ID, die als **Resource-Version** bezeichnet wird.
id	Ein global eindeutiger Bezeichner, der als **Resource Identifier** bezeichnet wird.
Metadaten	Ein JSON-Objekt mit verschiedenen Informationen, einschließlich Benutzer- und Systemetiketten.

Metadatenobjekt

Das JSON-Metadatenobjekt, das in jeder Astra-Ressource enthalten ist, enthält die in der folgenden Tabelle aufgeführten Schlüsselwerte.

Taste	Beschreibung
Etiketten	JSON-Array mit Client-angegebenen Beschriftungen, die der Ressource zugeordnet sind.
CreationZeitstempel	JSON-Zeichenfolge mit einem Zeitstempel, der angibt, wann die Ressource erstellt wurde.
Änderungszeitstempel	JSON-Zeichenfolge mit einem ISO-8601-formatierten Zeitstempel, der angibt, wann die Ressource zuletzt geändert wurde.
Erstellt von	JSON-Zeichenfolge mit der UUUIDv4-Kennung der Benutzer-id, die die Ressource erstellt hat. Wenn die Ressource von einer internen Systemkomponente erstellt wurde und der Erstellungseinheit keine UUID zugeordnet ist, wird die **Null** UUID verwendet.

Ressourcenstatus

Ausgewählte Ressourcen A `state` Wert, der zur Orchestrierung von Lifecycle-Übergängen und zur Steuerung des Zugriffs eingesetzt wird.

HTTP-Details

Die Astra Control REST-API verwendet HTTP und zugehörige Parameter, um auf die Ressourceinstanzen und -Sammlungen zu reagieren. Einzelheiten zur HTTP-Implementierung finden Sie unten.

API-Transaktionen und das CRUD-Modell

Die Astra Control REST API implementiert ein transaktionsorientiertes Modell mit klar definierten Abläufen und Zustandsübergängen.

API-Transaktion bei Anfrage und Reaktion

Jeder REST-API-Aufruf erfolgt als HTTP-Anfrage an den Astra-Service. Jede Anforderung generiert eine entsprechende Antwort zurück an den Client. Dieses Request-Response-Paar kann als API-Transaktion betrachtet werden.

Unterstützung für CRUD-Betriebsmodell

Auf Grundlage des **CRUD**-Modells kann auf alle über die Astra Control REST API verfügbaren Ressourcen und Sammlungen zugegriffen werden. Es gibt vier Vorgänge, von denen jede einer einzigen HTTP-Methode zugeordnet wird. Dazu gehören:

- Erstellen
- Lesen
- Aktualisierung
- Löschen

Bei einigen der Astra-Ressourcen wird nur ein Teil dieser Vorgänge unterstützt. Sie sollten die überprüfen "Online-API-Referenz" Weitere Informationen zu einem bestimmten API-Aufruf.

HTTP-Methoden

Die von der API unterstützten HTTP-Methoden oder Verben werden in der folgenden Tabelle dargestellt.

Methode	CRUD	Beschreibung
GET	Lesen	Ruft Objekteigenschaften für eine Ressourceninstanz oder -Sammlung ab. Dies wird als **list**-Operation bei Verwendung mit einer Sammlung betrachtet.
POST	Erstellen	Erstellt eine neue Ressourceninstanz basierend auf den Eingabeparametern. Die langfristige URL wird in A zurückgegeben `Location` Kopfzeile der Antwort.
PUT	Aktualisierung	Aktualisiert eine gesamte Ressourceninstanz mit dem mitgelieferten JSON Request Body. Wichtige Werte, die nicht vom Benutzer änderbar sind, bleiben erhalten.
Löschen	Löschen	Löscht eine vorhandene Ressourceninstanz.

Header für Anfragen und Antworten

Die folgende Tabelle fasst die HTTP-Header zusammen, die mit der Astra Control REST API verwendet werden.

Siehe "RFC 7232" Und "RFC 7233" Finden Sie weitere Informationen.

Kopfzeile	Typ	Nutzungshinweise
Akzeptieren	Anfrage	Wenn der Wert „*/*“ ist oder nicht angegeben wird, `application/json` Wird in der Kopfzeile der Inhaltstyp-Antwort zurückgegeben. Wenn der Wert auf den Astra Resource Media Type gesetzt ist, wird derselbe Medientyp in der Kopfzeile des Inhaltstyps zurückgegeben.
Autorisierung	Anfrage	Träger-Token mit dem API-Schlüssel für den Benutzer.
Inhaltstyp	Antwort	Wird basierend auf dem zurückgegeben `Accept` Kopfzeile der Anfrage.
Etag	Antwort	Im Lieferumfang eines erfolgreichen RFC 7232-Standards enthalten. Der Wert ist eine hexadezimale Darstellung des MD5-Werts für die gesamte JSON-Ressource.
If-Match	Anfrage	Ein Precondition Request Header, wie in Abschnitt 3.1 RFC 7232 beschrieben und unterstützt **PUT** Anforderungen.
Wenn-Geändert-Seit	Anfrage	Ein Anforderungsheader, der gemäß Abschnitt 3.4 RFC 7232 implementiert wurde und die Unterstützung für **PUT** -Anforderungen bietet.
Wenn-Unmodified-Since	Anfrage	Ein Anforderungsheader, der gemäß Abschnitt 3.4 RFC 7232 implementiert wurde und die Unterstützung für **PUT** -Anforderungen bietet.
Standort	Antwort	Enthält die vollständige URL der neu erstellten Ressource.

Abfrageparameter

Die folgenden Abfrageparameter stehen zur Verwendung mit Ressourcensammlungen zur Verfügung. Siehe "Arbeit mit Sammlungen" Finden Sie weitere Informationen.

Abfrageparameter	Beschreibung
Einschließlich	Enthält die Felder, die beim Lesen einer Sammlung zurückgegeben werden sollen.
Filtern	Gibt die Felder an, die für die Rückgabe einer Ressource beim Lesen einer Sammlung übereinstimmen müssen.
Orderby	Bestimmt die Reihenfolge der beim Lesen einer Sammlung zurückgegebenen Ressourcen.
Grenze	Begrenzt die maximale Anzahl an Ressourcen, die beim Lesen einer Sammlung zurückgegeben werden.
überspringen	Legt fest, wie viele Ressourcen beim Lesen einer Sammlung weitergehen und überspringen sollen.
Zählen	Gibt an, ob die Gesamtzahl der Ressourcen im Metadatenobjekt zurückgegeben werden soll.

HTTP-Statuscodes

Im Folgenden werden die HTTP-Statuscodes beschrieben, die von der REST-API von Astra Control verwendet werden.

Die Astra Control REST API nutzt auch den **Problemdetails für HTTP APIs** Standard. Siehe "Diagnose und Support" Finden Sie weitere Informationen.

Codieren	Bedeutung	Beschreibung
200	OK	Zeigt Erfolg für Anrufe an, die keine neue Ressourceninstanz erstellen.
201	Erstellt	Ein Objekt wurde erfolgreich erstellt, und die Kopfzeile für die Standortantwort enthält die eindeutige Kennung für das Objekt.
204	Kein Inhalt	Die Anfrage war erfolgreich, obwohl kein Inhalt zurückgegeben wurde.
400	Schlechte Anfrage	Die Eingabe der Anfrage ist nicht erkannt oder nicht angemessen.
401	Nicht Autorisiert	Der Benutzer ist nicht autorisiert und muss authentieren.
403	Verboten	Der Zugriff wird aufgrund eines Autorisierungsfehlers verweigert.
404	Nicht gefunden	Die Ressource, auf die in diesem Antrag verwiesen wird, ist nicht vorhanden.
409	Konflikt	Der Versuch, ein Objekt zu erstellen, ist fehlgeschlagen, weil das Objekt bereits vorhanden ist.
500	Interner Fehler	Ein allgemeiner interner Fehler ist auf dem Server aufgetreten.
503	Service nicht verfügbar	Der Dienst ist aus irgendeinem Grund nicht bereit, die Anfrage zu bearbeiten.

URL-Format

Die allgemeine Struktur der URL, die für den Zugriff auf eine Ressourceninstanz oder -Sammlung über DIE REST-API verwendet wird, besteht aus mehreren Werten. Diese Struktur spiegelt das zugrunde liegende Objektmodell und das Systemdesign wider.

Konto als Root

Die Wurzel des Ressourcenpfads zu jedem REST-Endpunkt ist das Astra-Konto. Daher beginnen alle Pfade in der URL mit `/account/{account_id}` Wo `account_id` Ist der eindeutige UUUIDv4-Wert für das Konto. Interne Struktur Dies ist ein Design, in dem der gesamte Ressourcenzugriff auf einem bestimmten Konto basiert.

Kategorie der Endpoint-Ressourcen

Die Astra-Ressourcenendpunkte lassen sich in drei verschiedene Kategorien einteilen:

- Kern (`/core`)
- Gemanagte Applikation (`/k8s`)
- Topologie (`/topology`)

Siehe "Ressourcen" Finden Sie weitere Informationen.

Kategorienversion

Jede der drei Ressourcenkategorien verfügt über eine globale Version, die die Version der Ressourcen steuert, auf die zugegriffen wird. Nach Konventionen und Definition zu einer neuen Hauptversion einer Ressourcenkategorie wechseln (z. B. von) `/v1` Bis `/v2`) Wird Bruchänderungen in der API einführen.

Ressourceinstanz oder -Sammlung

Eine Kombination von Ressourcentypen und Identifikatoren kann im Pfad verwendet werden, basierend darauf, ob auf eine Ressourceninstanz oder -Sammlung zugegriffen wird.

Beispiel

- Ressourcenpfad

  Basierend auf der oben dargestellten Struktur ist ein typischer Pfad zu einem Endpunkt: `/accounts/{account_id}/core/v1/users`.

- Vollständige URL

  Die vollständige URL für den entsprechenden Endpunkt lautet: `https://astra.netapp.io/accounts/{account_id}/core/v1/users`.