



# コア設計

## Astra Automation

NetApp  
August 11, 2025

# 目次

コア設計	1
REST Web サービス	1
リソースと状態の表示	1
URI エンドポイント	1
HTTP メッセージ	1
JSON 形式	1
リソースとコレクション	2
Astra リソースの属性	2
Astra のリソースに共通の構造	3
HTTP の詳細	3
API トランザクションと CRUD モデル	4
HTTP メソッド	4
要求と応答のヘッダー	4
クエリパラメータ	5
HTTP ステータスコード	5
URL 形式	6

# コア設計

## REST Web サービス

Representational State Transfer (REST) は、分散 Web アプリケーションの作成に使用される形式です。Web サービス API の設計に適用されることで、サーバベースのリソースを公開し、その状態を管理するための一連の主流テクノロジーとベストプラクティスが確立されます。REST はアプリケーション開発のための一貫した基盤を提供しますが、各 API の詳細は設計内容に応じて異なる場合があります。ライブ環境で使用する前に、Astra Control REST API の特性を理解しておく必要があります。

### リソースと状態の表示

リソースは、Web ベースシステムの基本コンポーネントです。REST Web サービスアプリケーションを作成する場合、設計の早い段階で次の作業を行います。

- システムまたはサーバベースのリソースの識別

すべてのシステムは、リソースを使用および管理します。リソースには、ファイル、ビジネストランザクション、プロセス、または管理エンティティがあります。REST Web サービスに基づいてアプリケーションを設計する際に行う最初の作業の 1 つは、リソースを識別することです。

- リソースの状態および関連する状態操作の定義

リソースの状態の数は有限で、リソースは必ずそのいずれかの状態にあります。状態、および状態の変化に影響する関連操作を明確に定義する必要があります。

### URI エンドポイント

すべての REST リソースは、明確に定義されたアドレス指定方式を使用して定義および使用可能にする必要があります。リソースが置かれているエンドポイントは、Uniform Resource Identifier (URI) で識別されます。URI は、ネットワーク内の各リソースに一意的な名前を作成するための一般的なフレームワークです。Uniform Resource Locator (URL) は、リソースを識別してアクセスするために Web サービスで使用される URI の一種です。リソースは通常、ファイルディレクトリに似た階層構造で公開されます。

### HTTP メッセージ

Hypertext Transfer Protocol (HTTP) は、Web サービスのクライアントとサーバがリソースに関する要求と応答のメッセージを交換する際に使用するプロトコルです。Web サービスアプリケーションの設計の一環として、HTTP メソッドはリソースおよび対応する状態管理アクションにマッピングされます。HTTP はステートレスです。したがって、関連する一連の要求と応答を 1 つのトランザクションの一部として関連付けるには、要求と応答のデータフローで伝送される HTTP ヘッダーに追加情報を含める必要があります。

### JSON 形式

Web サービスのクライアントとサーバの間で情報を構造化して転送する方法は複数ありますが、最も広く使用されているのは JavaScript Object Notation (JSON) です。JSON は、単純なデータ構造をプレーンテキストで表すための業界標準であり、リソースについての状態情報の転送に使用されます。Astra Control REST

API では、JSON を使用して、各 HTTP 要求と応答の本文で伝送されるデータをフォーマットします。

## リソースとコレクション

Astra Control REST API を使用すると、リソースインスタンスとリソースインスタンスのコレクションにアクセスできます。



概念的には REST \* リソース \* は、オブジェクト指向プログラミング（OOP）言語およびシステムで定義される \* オブジェクト \* に似ています。これらの用語が同じ意味で使用されることもあります。ただし一般には、外部 REST API のコンテキストで使用する場合は「リソース」が推奨され、サーバに格納される対応するステートフルインスタンスデータには「オブジェクト」が使用されます。

### Astra リソースの属性

Astra Control REST API は、RESTful 設計の原則に準拠しています。各アストラリソースインスタンスは、明確に定義されたリソースタイプに基づいて作成されます。同じタイプのリソースインスタンスのセットを \* 集合 \* と呼びます。API 呼び出しは、個々のリソースまたはリソースの集合に対して機能します。

#### リソースタイプ

Astra Control REST API に含まれるリソースタイプには、次の特徴があります。

- すべてのリソースタイプはスキーマを使用して定義されます（通常は JSON 形式）。
- すべてのリソーススキーマには、リソースタイプとバージョンが含まれています
- リソースタイプはグローバルに一意です

#### リソースインスタンス

Astra Control REST API から使用できるリソースインスタンスには、次のような特徴があります。

- リソースインスタンスは、単一のリソースタイプに基づいて作成されます
- リソースタイプは、 [メディアタイプ] の値を使用して示されます
- インスタンスは、アストラサービスによって管理されるステートフルデータで構成されます
- 各インスタンスには、一意で長く存続する URL を使用してアクセスできます
- リソースインスタンスが複数のリプレゼンテーションを持つことができる場合は、異なるメディアタイプを使用して必要なリプレゼンテーションを要求できます

#### リソースコレクション

Astra Control REST API で使用できるリソース収集には、次のような特徴があります。

- 単一のリソースタイプの一連のリソースインスタンスをコレクションと呼びます
- 一連のリソースには、一意で一時的な URL があります

#### インスタンス ID

すべてのリソースインスタンスには、作成時に識別子が割り当てられます。この識別子は 128 ビットの UUIDv4 値です。割り当てられた UUIDv4 の値は、グローバルに一意で変更できません。新しいインスタンスを作成する API 呼び出しを発行すると、関連付けられた ID を含む URL が呼び出し元に返されます Location

HTTP応答のヘッダー。リソースインスタンスを以降の呼び出しで参照する際には、この識別子を抽出して使用できます。



リソース識別子は、コレクションで使用される主キーです。

## Astra のリソースに共通の構造

すべてのアストラ制御リソースは、共通の構造を使用して定義されます。

### 共通のデータ

すべてのアストラリソースには、次の表に示すキーと値が含まれています。

キーを押します	説明
を入力します	グローバルに一意的リソースタイプ。 * リソースタイプ * と呼ばれます。
バージョン	* リソースバージョン * と呼ばれるバージョン識別子。
ID	グローバル一意識別子。 * リソース識別子 * と呼ばれます。
メタデータ	ユーザラベルやシステムラベルなどのさまざまな情報を含む JSON オブジェクト。

### メタデータオブジェクト

各アストラリソースに含まれるメタデータ JSON オブジェクトには、次の表に示すキーと値が含まれていません。

キーを押します	説明
ラベル	リソースに関連付けられているクライアント指定のラベルの JSON 配列。
作成タイムスタンプ	リソースが作成されたときのタイムスタンプを含む JSON 文字列。
modificationTimestamp	リソースの最終変更日時を示す ISO-8601 形式のタイムスタンプを含む JSON 文字列。
作成者	リソースを作成したユーザ ID の UUIDv4 識別子を含む JSON 文字列。リソースが内部システムコンポーネントによって作成され、作成するエンティティに関連付けられている UUID がない場合は、「 * null * UUID 」が使用されます。

### リソースの状態

選択したリソースA state ライフサイクル移行のオーケストレーションとアクセスの制御に使用する値。

## HTTP の詳細

Astra Control REST APIは、HTTPおよび関連パラメータを使用して、リソースのインスタンスおよびコレクションに対して動作します。HTTP 実装の詳細を以下に示します。

## API トランザクションと CRUD モデル

Astra Control REST API は、明確に定義された操作と状態の移行を伴うトランザクションモデルを実装します。

### 要求と応答の API トランザクション

すべての REST API 呼び出しは、Astra サービスへの HTTP 要求として実行されます。要求ごとに、関連付けられた応答がクライアントに生成されます。この要求と応答のペアで API トランザクションを使用できます。

### CRUD 操作モデルのサポート

Astra Control REST API で使用できる各リソースインスタンスとコレクションには、\*CRUD\* モデルに基づいてアクセスします。4 つの操作があり、それぞれが単一の HTTP メソッドにマッピングされます。処理は次のとおりです。

- 作成
- 読み取り
- 更新
- 削除

一部の Astra リソースでは、これらの操作の一部のみがサポートされています。を確認しておきます ["オンラインのAPIリファレンス"](#) 特定の API 呼び出しに関する詳細については、を参照してください。

## HTTP メソッド

次の表に、API でサポートされる HTTP メソッドまたは動詞を示します。

メソッド	CRUD	説明
取得	読み取り	リソースインスタンスまたはコレクションのオブジェクトプロパティを取得します。これは、コレクションとともに使用される場合、* list * 演算と見なされます。
投稿 (Post)	作成	入力パラメータに基づいて新しいリソースインスタンスを作成します。長期URLは、で返されます Location 応答ヘッダー。
PUT	更新	指定した JSON 要求の本文でリソースインスタンス全体を更新します。ユーザが変更できないキー値は保持されます。
削除	削除	既存のリソースインスタンスを削除します。

## 要求と応答のヘッダー

次の表に、Astra Control REST API で使用される HTTP ヘッダーを示します。



を参照してください ["RFC 7232"](#) および ["RFC 7233"](#) を参照してください。

ヘッダー	を入力します	使用上の注意
同意します	リクエスト	値が「/」または指定されていない場合、application/json Content-Type 応答ヘッダーで返されます。値が Astra リソースのメディアタイプに設定されている場合、同じメディアタイプが Content-Type ヘッダーに返されます。
承認	リクエスト	ユーザーの API キーを使用した Bearer トークン。
コンテンツタイプ	応答	に基づいて返されます Accept 要求ヘッダー。
ETag	応答	RFC 7232 で定義されたとおりに、成功したに含まれます。値は、JSON リソース全体の MD5 値を 16 進数で表したものです。
if-match	リクエスト	セクション 3.1 「RFC 7232」で説明されているように、*PUT* 要求をサポートする前提条件の要求ヘッダーが実装されています。
If-Modified-Since の略	リクエスト	セクション 3.4 「RFC 7232」に記載されている前提条件の要求ヘッダーと、*PUT* 要求のサポート。
If - unmodified- since	リクエスト	セクション 3.4 「RFC 7232」に記載されている前提条件の要求ヘッダーと、*PUT* 要求のサポート。
場所	応答	新しく作成されたリソースの完全な URL が含まれます。

## クエリパラメータ

リソースコレクションでは、次のクエリパラメータを使用できます。を参照してください "[コレクションを操作する](#)" を参照してください。

クエリパラメータ	説明
含める	コレクションを読み取る時に返すフィールドが含まれています。
フィルタ	コレクションの読み取り時にリソースが返されるために一致しなければならないフィールドを示します。
OrderBy	コレクションの読み取り時に返されるリソースのソート順序を指定します。
制限 (Limit)	コレクションの読み取り時に返されるリソースの最大数を制限します。
スキップします	コレクションの読み取り時に、パスオーバーおよびスキップするリソースの数を設定します。
カウント	メタデータオブジェクトで返されるリソースの総数。

## HTTP ステータスコード

Astra Control REST API で使用される HTTP ステータスコードを次に示します。



Astra Control REST API は、HTTP API \* 標準の \* 問題の詳細も使用します。を参照してください "[診断とサポート](#)" を参照してください。

コード	意味	説明
200	わかりました	新しいリソースインスタンスを作成しない呼び出しが成功したことを示します。
201	作成済み	オブジェクトが作成され、場所の応答ヘッダーにオブジェクトの一意の識別子が含まれている。
204	コンテンツがありません	要求は成功しましたが、コンテンツが返されませんでした。
400	無効な要求です	要求の入力が認識されないか不適切です。
401	権限がありません	ユーザは認証されていないため、認証が必要です。
403です	禁止されている	認証エラーによりアクセスが拒否されました。
404です	が見つかりません	要求で参照されているリソースが存在しません。
409だ	競合しています	オブジェクトがすでに存在するため、オブジェクトの作成に失敗しました。
500ドル	内部エラー	サーバで一般的な内部エラーが発生しました。
503	サービスを利用できません	サービスは何らかの理由で要求を処理する準備ができていません。

## URL 形式

REST API を使用したリソースインスタンスまたはコレクションへのアクセスに使用される URL の一般的な構造は、いくつかの値で構成されます。この構造は、基礎となるオブジェクトモデルとシステム設計を反映しています。

**root** としてアカウントを作成します

すべての REST エンドポイントへのリソースパスのルートは Astra アカウントです。そのため、URL内のすべてのパスはで始まります `/account/{account_id}` ここで、`account_id` は、アカウントの一意のUUIDv4値です。内部構造：すべてのリソースアクセスが特定のアカウントに基づいている設計を反映します。

エンドポイントリソースカテゴリ

Astra リソースエンドポイントは、次の 3 つのカテゴリに分類されます。

- コア (`/core`)
- マネージドアプリケーション (`/k8s`)
- トポロジ (`/topology`)

を参照してください ["リソース"](#) を参照してください。

カテゴリバージョン

3 つのリソースカテゴリのそれぞれに、アクセスされるリソースのバージョンを制御するグローバルバージョンがあります。規則および定義により、リソースカテゴリの新しいメジャーバージョン（例えば、から）に移動します `/v1` 終了： `/v2`) を使用すると、APIの変更が破棄されます。

リソースインスタンスまたはコレクション

リソースインスタンスまたはコレクションにアクセスするかどうかに基づいて、パスでリソースタイプと識別子の組み合わせを使用できます。

例

- リソースパス

上記の構造に基づいて、エンドポイントへの一般的なパスは次のとおりです。

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

- 完全な URL

対応するエンドポイントの完全なURLは次のとおりです。

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

## 著作権に関する情報

Copyright © 2025 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および/または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用权を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用权については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

## 商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。