



核心设计

Astra Automation

NetApp
December 01, 2023

目录

核心设计	1
REST Web 服务	1
资源和集合	1
HTTP 详细信息	3
URL 格式	5

核心设计

REST Web 服务

表述性状态传输（Representational State Transfer，REST）是一种用于创建分布式 Web 应用程序的模式。在设计 Web 服务 API 时，它会建立一组用于公开基于服务器的资源并管理其状态的主流技术和最佳实践。虽然 REST 为应用程序开发提供了一致的基础，但每个 API 的详细信息可能因特定设计选项而异。在将 Astra Control REST API 用于实时部署之前，您应了解其特征。

资源和状态表示

资源是基于 Web 的系统的基本组件。创建 REST Web 服务应用程序时，早期设计任务包括：

- 识别系统或基于服务器的资源

每个系统都使用和维护资源。资源可以是文件、业务事务、流程或管理实体。在设计基于 REST Web 服务的应用程序时，首先要完成的任务之一是识别资源。

- 资源状态和关联状态操作的定义

资源始终处于数量有限的状态之一。必须明确定义状态以及用于影响状态更改的关联操作。

URI 端点

必须使用定义明确的寻址方案定义和提供每个 REST 资源。资源所在的端点和标识的端点使用统一资源标识符（Uniform Resource Identifier，URI）。URI 提供了一个通用框架，用于为网络中的每个资源创建唯一名称。统一资源定位器（Uniform Resource Locator，URL）是一种用于 Web 服务的 URI 类型，用于标识和访问资源。资源通常以类似于文件目录的分层结构公开。

HTTP 消息

超文本传输协议（HTTP）是 Web 服务客户端和服务器用来交换有关资源的请求和响应消息的协议。在设计 Web 服务应用程序时，HTTP 方法会映射到资源以及相应的状态管理操作。HTTP 为无状态。因此，要将一组相关请求和响应关联为一个事务的一部分，必须将追加信息包含在随请求和响应数据流一起提供的 HTTP 标头中。

JSON 格式化

虽然信息可以通过多种方式在 Web 服务客户端和服务器之间进行结构化和传输，但最受欢迎的选项是 JavaScript 对象表示法（JSON）。JSON 是一种行业标准，用于以纯文本形式表示简单数据结构，并用于传输描述资源的状态信息。Astra Control REST API 使用 JSON 格式化每个 HTTP 请求和响应正文中包含的数据。

资源和集合

通过 Astra Control REST API，可以访问资源实例和资源实例集合。



从概念上讲，REST * 资源 * 类似于使用面向对象的编程（OOP）语言和系统定义的 * 对象 *。有时，这些术语可以互换使用。但一般来说，在外部 REST API 环境中使用时，首选使用 "resource"，而在服务器上存储的相应状态实例数据中使用 "object"。

Astra 资源的属性

Astra Control REST API 符合 RESTful 设计原则。每个 Astra 资源实例都是根据定义明确的资源类型创建的。一组相同类型的资源实例称为 * 集合 *。API 调用会对单个资源或资源集合执行操作。

资源类型

Astra Control REST API 附带的资源类型具有以下特征：

- 每个资源类型均使用模式定义（通常在 JSON 中）
- 每个资源架构都包括资源类型和版本
- 资源类型在全局范围内是唯一的

资源实例

通过 Astra Control REST API 提供的资源实例具有以下特征：

- 资源实例是根据单个资源类型创建的
- 资源类型使用介质类型值来指示
- 实例由由 Astra 服务维护的有状态数据组成
- 每个实例均可通过一个唯一且长期存在的 URL 进行访问
- 如果某个资源实例可以具有多个表示形式，则可以使用不同的介质类型来请求所需的表示形式

资源收集

通过 Astra Control REST API 提供的资源收集具有以下特征：

- 一种资源类型的一组资源实例称为集合
- 资源集合具有一个唯一且长期存在的 URL

实例标识符

创建每个资源实例时，系统都会为其分配一个标识符。此标识符是一个 128 位 UUIDv4 值。分配的 UUIDv4 值是全局唯一且不可变的。发出创建新实例的 API 调用后，将向中的调用方返回具有关联 ID 的 URL Location HTTP 响应的标题。在引用资源实例时，您可以提取此标识符并在后续调用中使用它。



资源标识符是用于收集的主密钥。

Astra 资源的通用结构

每个 Astra Control 资源都使用一个通用结构进行定义。

通用数据

每个 Astra 资源都包含下表所示的键值。

密钥	Description
type	一种全局唯一资源类型，称为 * 资源类型 *。
version	一种称为 * 资源版本 * 的版本标识符。
id	一种全局唯一标识符，称为 * 资源标识符 *。
元数据	一个 JSON 对象，包含各种信息，包括用户和系统标签。

元数据对象

每个 Astra 资源附带的元数据 JSON 对象包含下表所示的键值。

密钥	Description
labels	与资源关联的客户端指定标签的 JSON 数组。
creationTimestamp	JSON 字符串，其中包含指示资源创建时间的时间戳。
modificationTimestamp	JSON 字符串，其中包含 ISO-8601 格式的时间戳，用于指示资源上次更改的时间。
已创建	JSON 字符串，其中包含创建资源的用户 ID 的 UUIDv4 标识符。如果资源是由内部系统组件创建的，并且没有与创建实体关联的 UUID，则使用 * 空 * UUID。

资源状态

选定资源 `state` 用于编排生命周期过渡和控制访问的值。

HTTP 详细信息

Astra Control REST API 使用 HTTP 以及相关参数对资源实例和集合执行操作。下面提供了 HTTP 实施的详细信息。

API 事务和 CRUD 模型

Astra Control REST API 可实施一个事务模式，其中包含定义明确的操作和状态过渡。

请求和响应 API 事务

每次 REST API 调用都是作为对 Astra 服务的 HTTP 请求执行的。每个请求都会向客户端生成关联的响应。此请求响应对可视为 API 事务。

支持 CRUD 操作模式

通过 Astra Control REST API 提供的每个资源实例和集合均可根据 * CRU* 模型进行访问。有四个操作，每个操作都映射到一个 HTTP 方法。这些操作包括：

- 创建
- 读取
- 更新
- 删除

对于某些 Astra 资源，仅支持其中一部分操作。您应查看 ["联机API参考"](#) 有关特定 API 调用的详细信息。

HTTP 方法

下表显示了 API 支持的 HTTP 方法或动词。

方法	CRUD	Description
获取	读取	检索资源实例或集合的对象属性。在与集合结合使用时，此操作被视为 * 列表 * 操作。
发布	创建	根据输入参数创建新的资源实例。长期URL在中返回 Location 响应标头。
PUT	更新	使用提供的 JSON 请求正文更新整个资源实例。系统会保留用户不可修改的密钥值。
删除	删除	删除现有资源实例。

请求和响应标头

下表汇总了与 Astra Control REST API 一起使用的 HTTP 标头。



请参见 ["RFC 7232"](#) 和 ["RFC 7233"](#) 有关详细信息 ...

标题	Type	使用说明
接受	请求	如果此值为"/"或未提供、application/json 在Content-Type响应标头中返回。如果此值设置为 Astra 资源的介质类型，则内容类型标题中将返回相同的介质类型。
Authorization	请求	包含用户 API 密钥的承载令牌。
内容类型	响应	根据返回 Accept 请求标题。
ETAG	响应	随 RFC 7232 中定义的成功附带。该值是整个 JSON 资源的 MD5 值的十六进制表示形式。
如果匹配	请求	一个前提条件请求标头，如第 3.1 节 RFC 7232 中所述实施，并支持 * PUT * 请求。
if-Modified-since	请求	按照第3.4节RFC 7232中所述实施的前提条件请求标头，并支持*Put请求。
如果未修改，则从	请求	按照第3.4节RFC 7232中所述实施的前提条件请求标头，并支持*Put请求。
位置	响应	包含新创建资源的完整 URL 。

查询参数

以下查询参数可用于资源收集。请参见 ["使用收集"](#) 有关详细信息 ...

查询参数	Description
包括	包含读取收集时应返回的字段。

查询参数	Description
筛选器	指示读取收集时要返回的资源必须匹配的字段。
订单	确定读取收集时返回的资源的排序顺序。
limit	限制读取集合时返回的最大资源数。
跳过	设置读取集合时要传递和跳过的资源数量。
count	指示是否应在元数据对象中返回资源总数。

HTTP status codes

下面介绍了 Astra Control REST API 使用的 HTTP 状态代码。



Astra Control REST API 还使用 * HTTP APIs* 标准的问题详细信息。请参见 ["诊断和支持"](#) 有关详细信息 ...

代码	含义	Description
200	确定	表示未创建新资源实例的调用成功。
201	已创建	已成功创建对象，并且位置响应标头包含该对象的唯一标识符。
204.	No Content	尽管未返回任何内容，但请求成功。
400	请求错误	此请求输入无法识别或不适当。
401.	未授权	用户未获得授权，必须进行身份验证。
403.	已禁止	由于授权错误，访问被拒绝。
404	未找到	请求中引用的资源不存在。
409.	冲突	尝试创建对象失败，因为此对象已存在。
500	内部错误	服务器发生一般内部错误。
503.	服务不可用	由于某种原因，此服务尚未准备好处理此请求。

URL 格式

用于通过 REST API 访问资源实例或集合的 URL 的常规结构由多个值组成。此结构反映了底层对象模型和系统设计。

帐户作为 **root**

指向每个 REST 端点的资源路径的根是 Astra 帐户。因此，URL 中的所有路径都以开头 `/account/{account_id}` 其中： `account_id` 是帐户的唯一 UUIDv4 值。内部结构这反映了一种设计，其中所有资源访问都基于特定帐户。

端点资源类别

Astra 资源端点分为三类：

- 核心 (`/core`)

- 受管应用程序 (/k8s)
- 拓扑 (/topology)

请参见 "[Resources](#)" 有关详细信息 ...

类别版本

这三个资源类别中的每一个都有一个全局版本，用于控制所访问资源的版本。根据约定和定义、移动到资源类别的新主要版本(例如 /v1 to /v2)将在API中引入中断更改。

资源实例或集合

根据是否访问资源实例或集合，可以在路径中使用资源类型和标识符的组合。

示例

- 资源路径

根据上述结构、端点的典型路径为： `/accounts/{account_id}/core/v1/users`。

- 完整的 URL

相应端点的完整URL为：

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

版权信息

版权所有 © 2023 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。