Asset Service

Contents

Asset Service Overview

Asset Service Overview

Use the Asset service to create, update, and store asset model data that defines asset properties and relationships between assets and other modeling elements.

The Predix Asset service provides REST APIs to support asset modeling. Application developers use the Asset service to create, update, and store asset model data that defines asset properties as well as relationships between assets and other modeling elements. Data consumers can quickly retrieve vast amounts of asset data, while developers can store asset instance data. For example, an application developer can create an asset model describing the logical component structure of all pumps in an organization, then create instances of the model to represent each pump in an organization. Developers can also create custom modeling objects to meet their unique domain needs. The Asset service consists of an API layer, a query engine, and a graph database. Asset audit history enables you to update asset descriptions and retrieve historical information about your assets.

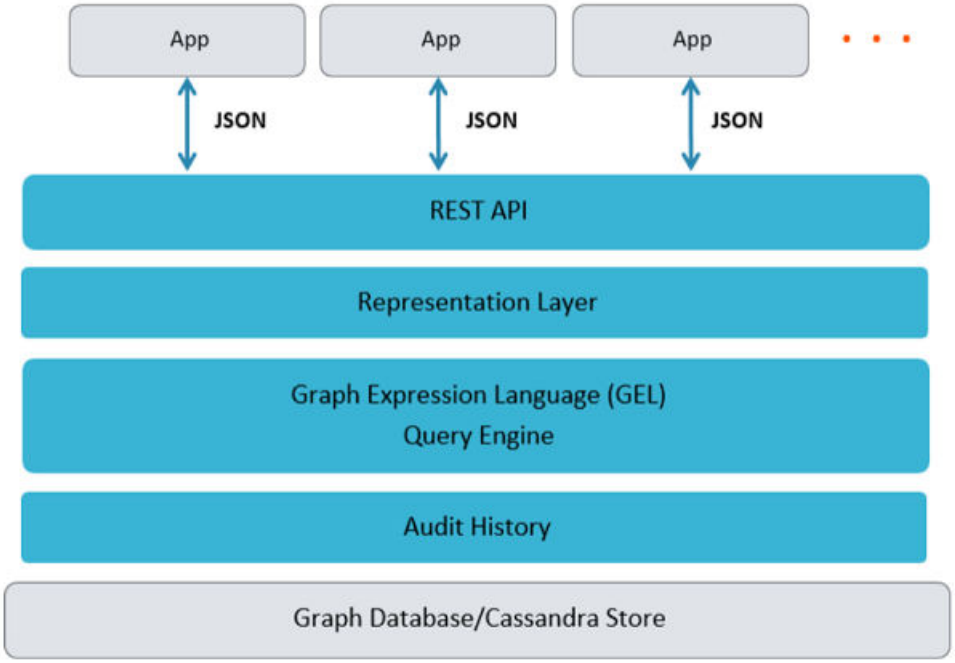


Figure 1: Asset Service Components

The Asset service includes the following components:

- REST API layer
  Client applications can access asset data using Asset service REST API endpoints. These endpoints provide a JSON interface where you can post the data that describes all of your assets. To use these APIs, your application makes HTTPS requests and parses the response. You can use any web-development language to access the APIs.
- Representation layer
  The representation layer translates data from JSON to internal graph representation and back to JSON.
- Query engine

The query engine enables developers to use JSON and Graph Expression Language (GEL) to retrieve data about any object or any object property in the Asset service data store. See Asset Service API Request Methods and Parameters on page 26.

- Audit History service
  The Asset audit history service provides APIs that enable you to retrieve historical information about REST requests in your Asset service repositories.
- Cassandra graph database
  The Asset service stores data in an Apache Cassandra NoSQL graph database. See Graph-based Data Model Basics on page 2.

Additional Information

Exploring Asset Service Guides

Graph-based Data Model Basics

The Asset service is based on a graph data model and it stores data in a graph database.

To understand why we use a graph data model, here is a comparison of relational, hierarchical, and graph data models:

Relational data model	Hierarchical data model	Graph data model
Tables are related by primary key. The relations in a relational database are handled by joins between tables. These can be managed in a permanent declarative fashion by the use of foreign keys, or in a temporary fashion by ad-hoc joins in the queries. A typical query language is Structured Query Language (SQL).	Parent nodes have more intrinsic importance. A typical query language is Extensible Markup Language (XML).	A graph consists of resources related to other resources, with no single resource having any intrinsic importance over another. A typical query language is Resource Description Framework (RDF). In a graph-based data model (RDF graph), the syntax comprises a subject, a predicate, and an object.

We selected a graph database (graph db) model for the Asset Data Service based on the needs of businesses in the era of the Industrial Internet of Things (IIoT). In the IIOT age, developers face many challenges:

- How to store huge numbers of assets and related objects.
- How to define huge numbers of relationships between assets and other objects.
- How to organize and make sense of them.
- How to discover and query relationships among them.

The necessity to define and query relationships is what drives the need for a graph db. Relationships are take priority in a graph db. They are built upon defining relationships between objects and providing performant queries on those relationships. Consider this diagram:

In the pre-IIOT era (shown above the dotted line), customers often had data relevant to an asset stored in typical “alphabet soup” or legacy systems – SCM, EAM, PLM, CRM, and so on. In the IIOT era (below the dotted line), there are many disparate sets of data that are relevant to an asset. For example:

- Streaming time-series data coming from sensors.
- Unstructured data such as audio, video, and documentation.
- Advanced operations data.
- Data from connected devices.
- Weather information.

The list goes on and on.

Many of these data elements may have relationships with one another in addition to linkage with a specific asset or assets. The ecosystem that an asset lives in is getting richer and more complex over time. Understanding and querying on the relationships this data has to your assets is why the graph db and query capability of Asset service is extremely important.

Use Graph Expression Language (GEL) in the `filter` clause of an API request to run a query against Asset service data. See Graph Expression Language (GEL) Syntax on page 27.

Asset Modeling

An asset model represents the information that you are storing about your domain objects, how your objects are organized, and how they are related.

As an application developer, you can use the Asset service APIs to define a consistent asset model and a hierarchical structure for your data. In an Asset service model, each piece of physical equipment is represented by an asset instance. You can organize your domain objects by classification and by any number of custom modeling objects. The Asset service provides an open model.

Asset service domain objects are distinct entities that a customer manages and tracks. All of the elements that make up the asset model, such as Classifications, are domain objects. The Assets entity is also a domain object. Any distinct entity defined by a customer is considered a domain object. Use JSON to describe the domain objects in your asset model. Objects are unordered sets of name-value pairs, and each object has a unique URI. The URI is like a primary key in that it uniquely identifies an asset instance stored in the graph database. The Asset service requires well-formed JSON, which can be validated with available tools such as the validator at http://jsonlint.com/.

The count of distinct domain objects provides a measure of the complexity and richness of an asset model. The count of instances of domain objects indicates the scope of the asset model. For example:

Domain Objects	Domain Object Instances
Assets	1000
Classifications	200
Part Numbers	100
Tags	5000
Templates	100
Number of distinct domain objects: 5	Total number of instances: 6,400

For example, an organization can use a `countries` object to store data about where its pumps are manufactured and a `manufacturers` object to store data about specific pump suppliers. It can use several classifications of `Pumps` to define pump types, and it can assign multiple attributes, such as `Brass` or `Steel`, to each classification. The organization can also associate multiple sensors, such as `Flow` or `Pressure`, to a classification.

The Asset service APIs support assets, classifications, and custom domain modeling objects.

API Category	Description
Assets	Typically, you define your assets in a hierarchical structure composed of a parent asset and one or many peers and children. You can associate an asset with a classification or any number of custom modeling objects. Assets can have any number of customer-defined attributes. An asset can also “stand alone” in the system, not associated with any other modeling elements.
Classifications	Classifications are arranged in a tree structure and provide a means to group similar assets and to track common attributes. A classification can point to multiple assets. Attributes can be assigned at any level in the classification hierarchy.
Custom modeling objects	Custom modeling objects are hierarchies that you can use to provide more information about your assets. For example, you can create separate objects for asset location, manufacturer, and service contract. A location can be associated with multiple assets. Likewise, an individual asset can be associated with multiple locations.

The following domain object examples show hierarchy diagrams and associated JSON code for a simple asset model composed of the following objects: `classifications`, `countries`, `manufacturers`, and `pumps`.

For a more robust data set that you can use to run sample queries, see Asset Model Sample Data on page 59

Data Validation Rules

The JSON schemas that define your domain objects also define your data validation rules. After you upload a new or updated schema, Asset automatically validates any new data ingested against the new schema. Data ingested previously will not be validated automatically against the new or updated schema.

JSON Schema validation requirements:

- The schema must be compliant with JSON schema format.
- The entity type must be `type 'object'`.
- The schema must include a URI property and the URI must be in compliance with Asset default URI rules. See Asset Service API Request Methods and Parameters on page 26
- You may add to a schema previously uploaded, but you cannot change the data type of any schema previously defined.

Best Practices for Data Validation

Because data previously ingested is not validated automatically, you might have data that does not match your new schema. GE recommends that you update your existing data to avoid receiving errors during queries on that data.

Related concepts

Validating Asset Data on page 31

Validate Predix Asset data against a JSON schema upon ingestion.

Related reference

Data Validation Requests on page 44

Example schema updates for Predix Asset domain objects.

Classifications Object Example

The `Pumps` classification has two child classifications: `positiveDisplacementPumps` and `kineticPumps`. The `kineticPumps` classification has one child: `centrifugalPumps`, which in turn has a child: `axialPumps`. The `axialPumps` classification is associated with an attribute: `Brass`.

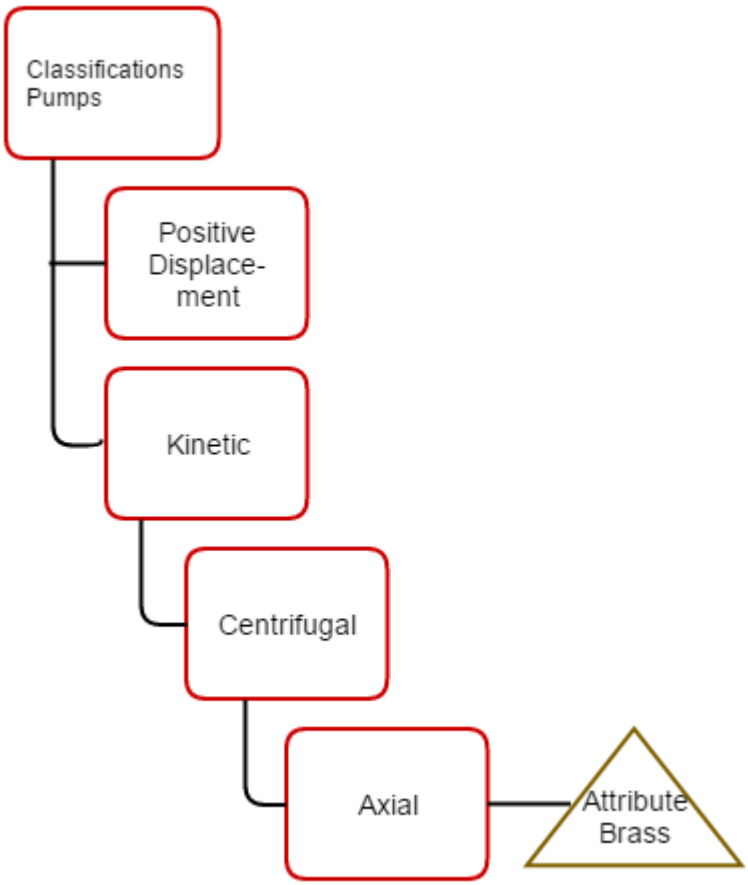


Figure 2: Classification Object Hierarchy

```
[
  {
```

```
    "uri": "/classifications/Pumps",
    "name": "Pumps",
    "description": "Parent Classification of Pumps",
    "obsolete": false,
    "attributes": {
      "Fuel Injector":{
        "type": "string",
        "value": ["Normal"]
      }
    },
    "parameter": {
      "ElectronicThrottlePositionSensor":
      {
        "parameterUri": "/parameter/ElectronicThrottlePositionSensor",
        "parameterId":"ElectronicThrottlePositionSensor1",
        "tsTagId":"/classifications/Pump/
ElectronicThrottlePositionSensor1"
      }
    }
  },
  {
    "uri": "/classifications/positiveDisplacementPumps",
    "name": "Positive Displacement Pumps",
    "description": "positive Displacement Pumps Classification",
    "parent": "/classifications/Pumps",
    "obsolete": false,
    "attributes": {
      "Manifold bolts":{
        "type": "number",
        "value": 8
      },
      "Manifold gasket":{
        "type": "number",
        "value": 1
      }
    }
  },
  {
    "uri": "/classifications/kineticPumps",
    "name": "Kinetic Pumps",
    "description": "Kinetic Pumps Classification",
    "parent": "/classifications/Pumps",
    "obsolete": false,
    "attributes": {
    }
  },
  {
    "uri": "/classifications/centrifugalPumps",
    "name": "Centrifugal Pumps",
    "description": "Classification",
    "parent": "/classifications/kineticPumps",
    "obsolete": false,
    "attributes": {
    }
  },
  {
    "uri": "/classifications/axialPumps",
    "name": "Axial Pumps",
    "description": "Classification",
    "parent": "/classifications/centrifugalPumps",
    "obsolete": false,
```

```
    "attributes": {
      "displayName": {
        "type": "string",
        "value": ["Brass"]
      }
    }
  }

]
```

Countries Object Example

The custom `Countries` modeling object has two children: `USA` and `Japan`.

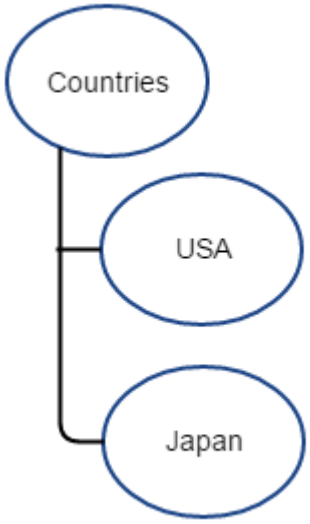


Figure 3: Countries Object

```
    [
  {
    "uri": "/countries",
    "name": "Countries",
    "obsolete": false,
    "attributes": {
    }
  },
  {
    "uri": "/countries/USA",
    "name": "USA",
    "parent": "/countries",
    "obsolete": false,
    "attributes": {
    }
  },
  {
    "uri": "/countries/Japan",
    "name": "Japan",
    "parent": "/countries",
    "obsolete": false,
```

```
    "attributes": {
    }
  }
]
```

Manufacturers Object Example

The custom `manufacturers` modeling object has two children: `JohnBluePumps` and `KachengoPumps`. `KachengoPumps` has two attributes: `Official Manufacturer Name` and `Alias`.

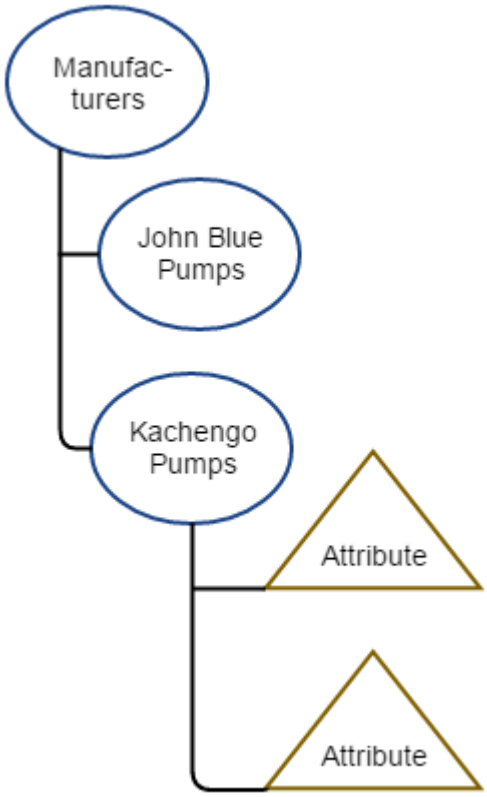


Figure 4: Manufacturers Object

```
[
  {
    "uri": "/manufacturers/JohnBluePumps",
    "name": "John Blue Pumps",
    "parent": "/manufacturers",
    "obsolete": false,
    "attributes": {
    }
  },
  {
    "uri": "/manufacturers/KachengoPumps",
    "name": "Kachengo Pumps",
    "parent": "/manufacturers",
    "obsolete": false,
```

```
    "attributes": {
      "Official Manufacturer Name": {
        "type": "string",
        "value": ["Kachengo Pumps, Limited"]
      },
      "Alias": {
        "type": "string",
        "value": ["Kachengo"]
      }
    }
  }
]
```

Pumps Object Example

There are two pump instances in this example: `Pump1Axial` and `Pump2Positive Displacement`.

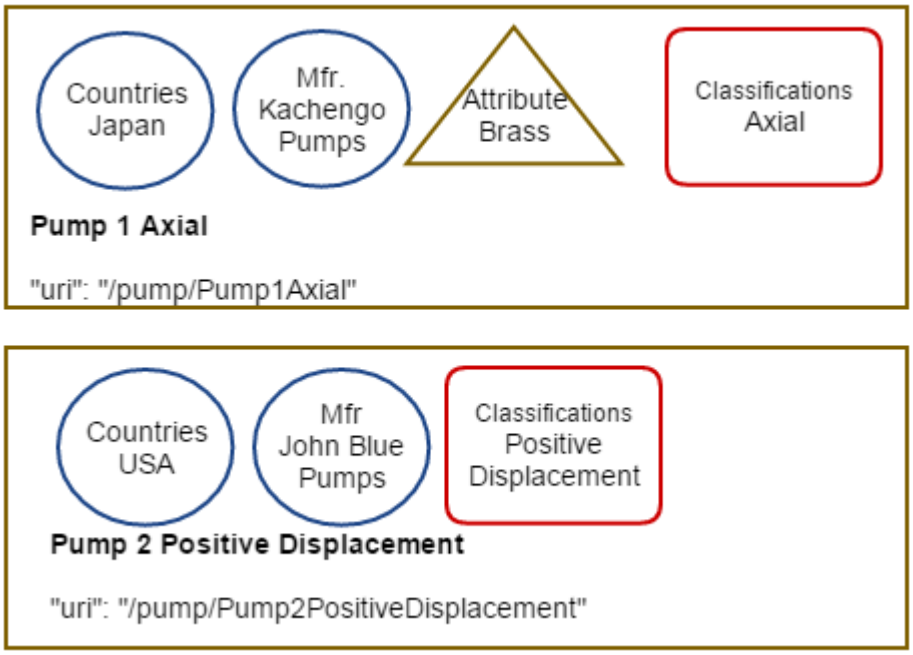


Figure 5: Pump Objects

```
[{
    "uri": "/pumps/Pump1Axial",
    "classification": "/classifications/Axial",
    "manufacturer": "/manufacturers/KachengoPumps",
    "countries": "/countries/Japan",
    "nonserialized": {

    },
    "attributes": {
        "Name": {
            "type": "string",
            "value": ["Pump 1 Axial"]
        },
        "createdOn": {
            "type": "string",
```

```
            "value": ["2015-02-20T00:00:00+0000"]
        },
        "updatedOn": {
            "type": "string",
            "value": ["2015-02-20T00:00:00+0000"]
        },
        "isActive": {
            "type": "boolean",
            "value": [true]
        }
    }
},
{
    "uri": "/pumps/Pump2PositiveDisplacement",
    "classifications": "/classifications/PositiveDisplacement",
    "manufacturers": "/manufacturers/JohnBluePumps",
    "countries": "/countries/USA",
    "nonserialized": {

    },
    "attributes": {
        "Name": {
            "type": "string",
            "value": ["Pump 2 Positive Displacement"]
        },
        "createdOn": {
            "type": "string",
            "value": ["2015-02-20T00:00:00+0000"]
        },
        "updatedOn": {
            "type": "string",
            "value": ["2015-02-20T00:00:00+0000"]
        },
        "isActive": {
            "type": "boolean",
            "value": [true]
        }
    }
}]
```

For a more robust data set that you can use to run sample queries, see Asset Model Sample Data on page 59.

About Asset Audit History

The Asset audit history service provides APIs that enable you to retrieve historical information about REST requests in your Asset service repositories.

Before you begin, you must configure your audit metadata in your request headers. See Asset Service REST Request Headers on page 34.

The audit history APIs provide the following endpoints:

- `/<predix-asset-url>/system/configs`
  The audit history service is disabled by default. Use this endpoint to enable or disable the asset audit history service.
- `/<predix-asset-url>/system/audit`
  Use this endpoint to retrieve the old values and the new values of the requested assets.

- `/<predix-asset-url>/system/audit/changes`
  Use this endpoint to retrieve an old value and a new value record for each element in a collection that has changed.
- `/<predix-asset-url>/system/audit/snapshots`
  Use this endpoint to retrieve details about a single asset at a moment in time.

Get Started with Asset Service

Get Started with Asset Service

Like other Predix platform services, authentication access to Asset service is controlled by the designated trusted issuer and is managed by the User Account and Authentication (UAA) web service. A UAA service instance must be already set up as the trusted issuer before getting started with this Asset service.

For more information about how authentication and authorization is enforced in Predix services, see Understanding Platform Services.

Asset Service Setup

This roadmap lists the high-level tasks for getting started with the Asset service.

Authentication for Asset service is controlled by the designated trusted issuer and is managed by the User Account and Authentication (UAA) web service. You must set up a UAA service instance as the trusted issuer before getting started with the Asset service. For information about authentication and authorization in Predix services, see About Security Services.

You will not perform all tasks if you have already set up UAA services, created a trusted issuer, and an OAuth2 client, as described in Setting Up Platform Services Using Cloud Foundry Commands.

Task Roadmap

#	Task	Information
1	Create a text file to store values that you will need later.	See Creating a Parameters Text File on page 13.
2	Configure your proxy settings if necessary.	Depending on your location and network configuration, you may need to configure your proxy settings to access remote resources. See Defining Proxy Connections to Remote Resources.
3	Set up access to Predix platform Artifactory.	If you need access to Predix platform artifacts, you need to set up access for Artifactory. See Defining Predix Platform Artifactory Access.
4	Deploy your application to Cloud Foundry.	For an example of deploying a Predix Hello World Web application to cloud foundry, see Creating and Deploying a Simple Web App to Cloud Foundry.
5	Create an instance of the trusted issuer.	Create an instance of User Account and Authentication (UAA) service. UAA is the authorization server that each platform service uses for authentication. See Creating a UAA Service Instance on page 13.
6	Create an instance of the Asset service.	See Creating an Asset Service Instance on page 16.

#	Task	Information
7	Create OAuth2 clients to setup access to your service authenticated using UAA.	When you create a UAA instance, an admin client is automatically created for you to access UAA for additional configuration. You can create a new client for your service instance with specific scopes. If an Oauth2 client already exists, you can update the client to add your service instance. See Creating an OAuth2 Client on page 17.
8	Update the Oath2 client to add service specific scopes or authorities.	To enable your application to access a platform service, your JSON Web Token (JWT) must contain the scopes required for a platform service. See Updating the OAuth2 Client for Services on page 20. See Authorities and Scopes Required for Asset Service on page 22.
9	Bind your application to the service instance.	To establish communication between your application and the platform service, you must bind the application to the service. See Binding an Application to an Asset Service Instance on page 22.
10	Add asset model data to your application.	See Adding Asset Model Data to your Asset Service Instance on page 23
11	Start using the Asset service.	See Get Started with Asset Service on page 12.

Creating a Parameters Text File

Create a text file to store values that you will need later.

Procedure

- Copy and paste the following parameters into a text file that you can use to record values as you create the Asset service instance. Once recorded, these values are used in subsequent steps.

```
uaa_instance_issuerId:
uaa_instance_uri:
uaa_admin_account_name:
uaa_admin_client_secret:

predix-hello-world-app-<YourName>:

developer_username:
developer_password:
predix_asset_api_gateway_short_route_url:
```

Creating a UAA Service Instance

You can create multiple instances of the UAA service in your space.

About This Task

As a best practice, first delete any older unused instances before creating a new one.

Procedure

1. Sign into your Predix account at https://www.predix.io.
2. Navigate to **Catalog** > **Services**, then click the **User Account and Authentication** tile.
3. Click **Subscribe** on the required plan.
4. Complete the fields on the **New Service Instance** page.

Field	Description
Org	Select your organization.
Space	Select the space for your application.
Service instance name	Enter a unique name for this UAA service instance.
Service plan	Select a plan.
Admin client secret	Enter a client secret (this is the admin password for this UAA instance). The client secret can be any alphanumeric string. **Note:** Record the client secret in a secure place for later use.
Subdomain	(Optional) Enter a subdomain you might need to use in addition to the domain created for UAA. You must not add special characters in the name of the subdomain. The value of subdomain is case-insensitive.

5. Click **Create Service**.

Results

Your UAA instance is created with the following specifications:

- A client identifier (`admin`).

  Note: An `admin` client is required for bootstrap purposes. You can create additional clients to use with your application.
- A client secret (that you specified while creating the service).

To retrieve additional details of your instance, you can bind an application to your instance.

Using the Command Line to Create a UAA Service Instance

Optional procedure for using the command line instead of the graphical user interface to create a UAA service instance.

About This Task

You can create up to 10 instances of UAA service in your space. If you need additional instances, you must delete an older unused instance and create a new one.

Procedure

1. Use the Cloud Foundry CLI to log into Cloud Foundry.

   ```
   cf login -a <API_Endpoint>
   ```

   Note: If you are a GE employee, you must use the `cf login --sso` command to log into Cloud Foundry. After you enter your SSO, you will receive a one-time passcode URL. Copy this URL and paste it in a browser to retrieve your one-time passcode. Use this code with the `cf` command to complete the CF login process.

   Depending on your Predix.io registration, the value of `<API_Endpoint>` is one of the following:

- Predix US-West
  `https://api.system.aws-usw02-pr.ice.predix.io`
- Predix US-East
  `https://api.system.asv-pr.ice.predix.io`
- Predix Europe
  `https://api.system.aws-eu-central-1-pr.ice.predix.io`

For example,

```
cf login -a https://api.system.aws-usw02-pr.ice.predix.io
```

2. List the services in the Cloud Foundry marketplace by entering the following command.

   ```
   cf marketplace
   ```

   The UAA service, `predix-uaa`, is listed as one of the available services.
3. Create a UAA instance by entering the following command.

   ```
   cf create-service predix-uaa <plan> <my_uaa_instance> -c
   '{"adminClientSecret":"<my_secret>","subdomain":"<my_subdomain>"}'
   ```

   where:

   - `cf` stands for the CLI command, `cloud foundry`
   - `cs` stands for the CLI command `create-service`
   - `<plan>` is the plan associated with a service. For example, you can use the `tiered` plan for the `predix-uaa` service.
   - `-c` option is used to specify following additional parameters.
     - `adminClientSecret` specifies the client secret.
     - `subdomain` specifies a sub-domain you might need to use in addition to the domain created for UAA. This is an optional parameter. You must not add special characters in the name of the sub-domain. The value of sub-domain is case insensitive.

   Note: Cloud Foundry CLI syntax can differ between Windows and Linux operating systems. See the Cloud Foundry help for the appropriate syntax for your operating system. For example, to see help for the `create service` command, run `cf cs`.

Results

Your UAA instance is created with the following specification:

- A client identifier (`admin`).

  Note: An `admin` client is created for bootstrap purposes. You can create additional clients to use with your application.
- A client secret (that you specified while creating the service).

To retrieve additional details of your instance, you can bind an application to your instance.

Example

Create a predix-uaa service instance with client secret as admin and sub-domain as ge-digital:

```
cf cs predix-uaa tiered test-1 -c
'{"adminClientSecret":"admin","subdomain":"ge-digital"}'
```

This is how it appears in VCAP SERVICES when using the `cf env <app_name>` command:

```
"VCAP_SERVICES": {
"predix-uaa": [
   {
    "credentials": {
     "dashboardUrl": "https://uaa-dashboard.run.asv-
pr.ice.predix.io/#/login/04187eb1-
e0cf-4874-8218-9fb77a8b4ed9",
     "issuerId": "https://04187eb1-
e0cf-4874-8218-9fb77a8b4ed9.predix-uaa.run.asv-
pr.ice.predix.io/oauth/token",
     "subdomain": "04187eb1-e0cf-4874-8218-9fb77a8b4ed9",
     "uri": "https://04187eb1-
e0cf-4874-8218-9fb77a8b4ed9.predix-uaa.run.asv-
pr.ice.predix.io",
     "zone": {
      "http-header-name": "X-Identity-Zone-Id",
      "http-header-value": "04187eb1-
e0cf-4874-8218-9fb77a8b4ed9"
     }
    },
    "label": "predix-uaa",
    "name": "testuaa",
    "plan": "Tiered",
    "provider": null,
    "syslog_drain_url": null,
    "tags": [],
    "volume_mounts": []
   }
  ],
```

Creating an Asset Service Instance

Create an Asset service instance to use to create, update, and store asset model data that defines asset properties and relationships between assets and other modeling elements.

Before You begin

An instance of the UAA service has been configured as your trusted issuer. See Task Roadmap: Setting Platform Services.

Procedure

1. Sign into your Predix account at https://www.predix.io.
2. Navigate to **Catalog > Services** tab, and click the **Asset Data Service** tile.
3. Click **Subscribe** on the required plan.
4. On the **New Service Instance** page, enter:

Field	Description
Org	Select your org.
Space	Select the space for your application.

Field	Description
User Account & Authentication (UAA)	Choose an existing UAA instance or create a new instance of UAA. See Creating a UAA Service Instance on page 13.
Service instance name	Specify a unique name for your instance.
Service plan	Select a plan.

5. Click **Create Service**.

Creating an OAuth2 Client

You can create OAuth2 clients with specific permissions for your application to work with Predix Platform services. Often this is the first step after creating an instance of a service.

About This Task

When you create an instance of UAA, the UAA Dashboard is available for configuring that instance of UAA. You can use the Client Management tab in the UAA Dashboard to create the OAuth2 clients.

If you are prefer using the UAA command-line interface (UAAC) instead of UAA Dashboard to create an OAuth2 client, see Using UAAC to Create an OAuth2 Client

Procedure

1. In the Predix.io Console view, select the Space where your services are located.
2. In the Services Instances page, select the UAA instance to configure.
3. Select the **Configure Service Instance** option.
4. In the UAA Dashboard login page, specify your admin client secret and click **Login**.
5. In UAA Dashboard, select the **Client Management** tab.

   The Client Management tab has two views, **Clients** and **Services** . The **Services** view displays the service instances that you have created for your services.

   Note: The service instances displayed in the Services view were created while using the UAA that you are trying to configure. Service instances that you created using other UAA instances are not displayed on this page.
6. Click **Create Client** to open the **Create Client** form.
7. Complete the **Create Client** form.

Field	Description
Client ID	Specify a name for the OAuth2 client you are creating.
Authorized Grant Types	Choose one or more of the following grant types: • **authorization_code** When you use the authorization code grant type, the client directs the resource owner to UAA, which in turn directs the resource owner back to the client with the authorization code. • **client_credentials** When you use the client credentials grant type, the OAuth2 endpoint in UAA accepts the client ID and client secret and provides Access Tokens. • **password** When you use the resource owner password credentials grant type, the OAuth2 endpoint in UAA accepts the username and password and provides Access Tokens. • **refresh_token** The refresh tokens are credentials used to obtain access tokens. You can choose this option to obtain refresh token from UAA. You can then use the refresh token to obtain a new access token from UAA when the current access token becomes invalid or expires, or to obtain additional access tokens with identical or narrower scope. • **implicit** When you use the implicit grant type, UAA directly issues an Access Token to the client without authenticating the client. This reduces the number of round trips required to obtain an access token. For more information on grant types, see RFC 6749.
Client Secret	Specify the password. It is important that you keep a note of this password. If lost, this password cannot be retrieved.
Confirm Client Secret	Reenter the client secret.

Field	Description
Redirect URI	Specify a redirect URI to redirect the client after login or logout (for example, `http://example-app.com/callback`). Use this URI when you start using UAA as the service provider for your external Identity provider. UAA uses the value of Redirect URI for `/oauth/authorize` and `/logout` endpoints. You must specify a Redirect URI value if you use the Authorization Code or Implicit authorization grant type. When you use the Authorization Code grant type, the Redirect URI is your application's endpoint or callback that expects user authorization code. When you use the Implicit grant type, the Redirect URI is the end point where UAA sends the bearer token. Unique Resource Identifier consists of: • Access Protocol, `http` or `https` • Domain or IP address • Access Port such as 80 or 443 • Path If you have a specific URL for your application callback, you can use that to set the Redirect URI value for the related client. For example, `https://your-app-domain.run.aws-usw02-pr.ice.predix.io/path1/path2/callback`. You can specify multiple values for Redirect URI as a list of allowed destinations that UAA server can redirect the users. For example, `https://yourappdomain1.run.aws-usw02-pr.ice.predix.io/path1/path2/callback`, `https://yourappdomain2.run.aws-usw02-pr.ice.predix.io/path1/path2/callback`. If the subdomain of your application is dynamic, you can set the value of Redirect URI using wilcards. For example, `https://*.your-app-domain.run.aws-usw02-pr.ice.predix.io/path1/path2/callback`. **Note:** You must only use '*' for a domain that is exclusive to your application (Such as `your-app-domain` in example above). This prevents the redirect to be routed to an application that you do not own. You cannot use `*` in the top domain and sub domain (such as `predix.io` in the example above).
Scopes	Scopes are permissions associated with an OAuth Client to determine user access to a resource through an application. The user permissions are for authorization grant types `authorization_code`, `password` and `implicit`. By default, the admin client is assigned all required scopes. For a new client, an administrator can select the scopes to be added based on client requirements. For a list of available scopes, see Scopes Authorized by the UAA. To use an OAuth2 client for your Predix Platform service instance, you must update your OAuth2 client to add scopes that are specific to each service after adding the client to the service instance.

Field	Description
Authorities	Authorities are permissions associated with the OAuth Client when an application or API is acting on its own behalf to access a resource with its own credentials, without user involvement. The permissions are for the `client_credentials` authorization grant type. By default, the admin client is assigned all required authorities. For a new client, an administrator can select the authorities to be added based on client requirements. The list of authorities matches the list of scopes. For a list of available UAA scopes, see Scopes Authorized by the UAA. To use an OAuth2 client for your Predix Platform service instance, you must update your OAuth2 client to add authorities that are specific to each service after adding the client to the service instance. **Note:** An admin client is not assigned the default authority to change the user password. To change the user password, you must add the `uaa.admin` authority to your admin client.
Auto Approved Scopes	Specify scopes that can be approved automatically for the client without explicit approval from a resource owner.
Allowed Providers	Specifies the names of the external identity providers, if any. This field is required if you are using external identity providers with UAA as a service provider.
Access Token Validity	Specifies the access-token expiration time in ms.
Refresh Token Validity	Specifies the refresh-token expiration time in ms.

Next Steps

Updating the OAuth2 Client for Services on page 20 for your service specific information.

Updating the OAuth2 Client for Services

To use an OAuth2 client for secure access to your Predix Platform service instance from your application, you must update your OAuth2 client to add additional authorities or scopes that are specific to each service.

About This Task

To enable your application to access a platform service, your JSON Web Token (JWT) must contain the scopes required for a platform service. For example, some of the scope required for Access Control service are `acs.policies.read acs.policies.write`.

The OAuth2 client uses an authorization grant to request an access token. Based on the type of authorization grant that you have used, you must update your OAuth2 client to generate the required JWT. For more information on how the OAuth2 client is created, see Creating OAuth2 client.

If you use the UAA Dashboard to create additional clients, the client is created for the default `client_credentials` grant type. Some required authorities and scopes are automatically added to the client. You must add additional authorities or scopes that are specific to each service.

In addition, the admin client is not assigned the default authority to change the user password. To change the user password, you must add the `uaa.admin` authority to your admin client.

Use the following procedure to update the OAuth2 client.

Procedure

1. In the Console view, select the Space where your services are located.

2. In the Services Instances page, select the UAA instance to configure.
3. Select the **Configure Service Instance** option.
4. In the UAA Dashboard login page, specify your admin client secret and click **Login**.
5. In UAA Dashboard, select the **Client Management** tab.

   The Client Management tab has two views, **Clients** and **Services**. The **Services** view displays the service instances that you have created for your services.

   Note: The service instances displayed in the **Services** view are the instances that you created using the UAA that you are trying to configure. The service instances that you created using some other UAA instance are not displayed on this page.
6. Select the **Switch to Services View** option.
7. In the **Services** view, select the service that you need to update.
8. Choose an existing client or choose the **Create a new client** option. If you chose to create a new client, follow the steps in Creating an OAuth2 Client on page 17.
9. Click **Submit**.
10. Click on the **Switch to Clients View** option.
11. In the **Clients** view, click the edit icon corresponding to the client added in the previous step.
12. Complete the **Edit Client** form.

Field	Description
Authorized Grant Types	Choose one or more of the following grant types: • **authorization_code** When you use the authorization code grant type, the client directs the resource owner to UAA, which in turn directs the resource owner back to the client with the authorization code. • **client_credentials** When you use the client credentials grant type, the OAuth2 endpoint in UAA accepts the client ID and client secret and provides Access Tokens. • **password** When you use the resource owner password credentials grant type, the OAuth2 endpoint in UAA accepts the username and password and provides Access Tokens. • **refresh_token** The refresh tokens are credentials used to obtain access tokens. You can choose this option to obtain refresh token from UAA. You can then use the refresh token to obtain a new access token from UAA when the current access token becomes invalid or expires, or to obtain additional access tokens with identical or narrower scope. • **implicit** When you use the implicit grant type, UAA directly issues an Access Token to the client without authenticating the client. This reduces the number of round trips required to obtain an access token. For more information on grant types, see RFC 6749.
Redirect URI	Specify a redirect URI to redirect the client after login (for example, `http://example-app.com/welcome`). This URI is used when you start using UAA as service provider for your external Identify provider.
Scopes	By default, the client is assigned a few required scopes. For a new client, an administrator can select the scopes to be added based on the selected grant type. If you select the `authorization_code`, `password` and `implicit` grant type, you must update the scopes with service specific scopes. For a complete list of required scopes, see Authorities or Scopes Required for Platform Services. For a list of available UAA scopes, see Scopes Authorized by the UAA.

Field	Description
Authorities	By default, the client is assigned a few required authorities. For a new client, an administrator can select the authorities to be added based on the selected grant type. If you select the `client_credentials` grant type, you must update the authorities with service specific authorities. For a complete list of scopes to be added for each service, see Authorities or Scopes Required for Platform Services. For a list of available UAA authorities, see Scopes Authorized by the UAA.
Auto Approved Scopes	Specify scopes that can be approved automatically for the client without explicit approval from the resource owner.
Allowed Providers	Specify the names of the external identity providers, if any. This field is required if you are using external identity providers with UAA as a service provider.
Access Token Validity	Specifies the access token expiration time in ms.
Refresh Token Validity	Specifies the refresh token expiration time in ms.

Next Steps

You can complete the following additional tasks in UAA Dashboard:

- If you are using authorization grant type as Authorization Code, Implicit, or Resource Owner Password, you can manage users in UAA.
- You can create password policies for user passwords.
- You can set up external identity provider or use UAA as an identity provider. See Managing Identity Providers.

If you have completed your OAuth2 client setup, you can bind your application to your service instance.

Authorities and Scopes Required for Asset Service

To enable applications to access the Asset service, your JSON Web Token (JWT) must contain the following scope: `predix-asset.zones.<service_instance_guid>.user`.

Binding an Application to an Asset Service Instance

You must bind your Asset instance to your application to provision connection details for your Asset service instance in the VCAP_SERVICES environment variable

About This Task

Cloud Foundry runtime uses the VCAP_SERVICES environment variable to communicate with a deployed application about its environment.

Procedure

1. Bind your application to the new Asset service instance.

```
cf bind-service <your_app_name> <asset_instance_name>
```

The <asset_instance_name> instance is bound to your application, and the following message is returned:

```
Binding service <asset_instance_name> to app <your_app_name> in org
cf_org_name / space cf_space_name as cf_username...
OK
TIP: Use 'cf restage' to ensure your env variable changes take effect
```

2. Verify the binding:

```
cf env <your_app_name>
```

Messages that are similar to the following messages are returned:

```
Getting env variables for app your_app_name in org cf_org_name /
space cf_space_name as cf_user_name...
OK
{
 "VCAP_SERVICES": {
  "predix-asset-dev": [
   {
    "credentials": {
     "instanceId": "69c2bd61-4526-4379-9533-a2721fe83ba8",
     "uri": "http://predix-asset-rc.grc-apps.svc.ice.ge.com",
     "zone": {
      "http-header-name": "Predix-Zone-Id",
      "http-header-value": "69c2bd61-4526-4379-9533-a2721fe83ba8",
      "oauth-scope": "predix-asset.zones.69c2bd61-4526-4379-9533-
a2721fe83ba8.user"
     }
    },
    "label": "predix-asset",
    "name": "testAsset",
    "plan": "<your_plan_name>",
    "tags": []
   }
  ]
 }
}
```

Adding Asset Model Data to your Asset Service Instance

You can copy the Asset Model Sample Data to add a collection of assets to your Asset service instance.

About This Task

Client applications can access asset data using Asset service REST API endpoints. These endpoints provide a JSON interface where you can post the data that describes all of your assets. To use these APIs, your application makes HTTPS requests and parses the response. You can use any web-development language to access the APIs.

For more information about this API, see the API Documentation.

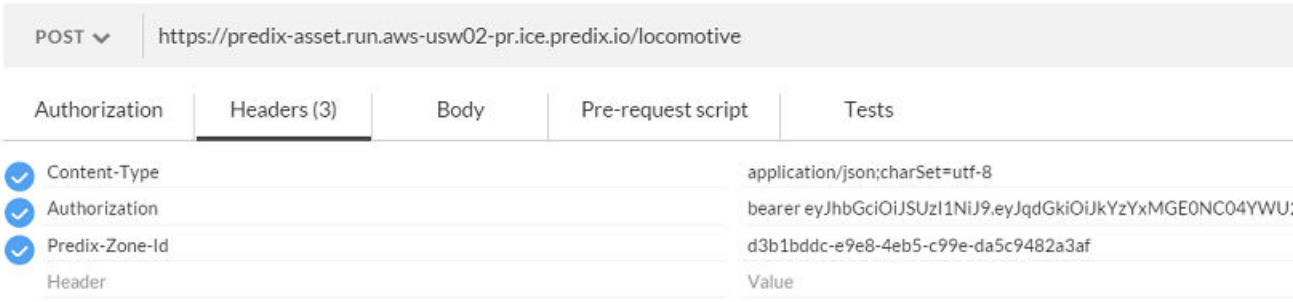
To add a collection of assets to your Asset service instance:

Procedure

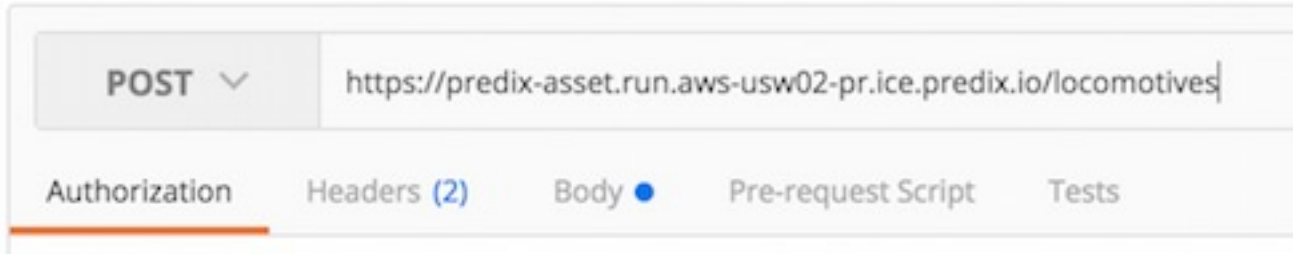
1. Copy the sample files in Asset Model Sample Data on page 59.
2. Retrieve Asset instance details from the VCAP_SERVICES environment variable:
   - Asset instance `uri`
   - The ID required for making any REST calls, `instanceId`.
3. In a REST client:

   a) Choose the `POST` method.
   b) Enter the URI of your Asset service instance, from the VCAP_SERVICES environment variable.
   c) Add the following HTTP headers:

   - `Content-Type`
     Enter the value.`application/json`.
   - `Authorization`
     Enter the token you receive from UAA in the form of `Bearer <token from trusted issuer>`. See Binding an Application to the UAA Instance.
   - `Predix-Zone-Id`
     Enter the ID required for making any REST calls, using the value from `instanceId` from the VCAP_SERVICES environment variable.



   d) Enter well-formed JSON in the Body of your request.
   To use sample data, copy the JSON objects from Asset Model Sample Data on page 59.




```
[
    {
        "uri": "/locomotives/1",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "001",
        "emission_tier": "0+",
        "fleet": "/fleets/up-1",
        "manufacturer": "/manufacturers/GE",
```

```
      "engine": "/engines/v12-1",
      "installedOn": "01/12/2005",
      "dateIso": "2005-12-01T13:15:31Z",
      "hqLatLng": {
          "lat": 33.914605,
          "lng": -117.253374
      }
   },
   }
      "uri": "/locomotives/84",
      "type": "Diesel-electric",
      "model": "3GS21B",
      "serial_no": "0084",
      "emission_tier": "0+",
      "fleet": "/fleets/cn-4",
      "manufacturer": "/manufacturers/national-railway-equipment",
      "engine": "/engines/QSK19-26",
      "hqLatLng": {
          "lat": 49.296307,
          "lng": -118.827314
      }
    }

    ]
```

4. Use the Asset service API endpoints to run the Sample Asset Service API Requests on page 40.

Using Asset Service

Using Asset Service

Use the Asset service to create, update, and store asset model data that defines asset properties and relationships between assets and other modeling elements.

Asset Service API Request Methods and Parameters

The Asset service supports `POST`, `GET`, `PUT`, `PATCH`, and `DELETE` request methods. The Asset service supports the `fields`, `filter`, and `pageSize` request parameters.

Client applications can access asset data using Asset service REST API endpoints. These endpoints provide a JSON interface where you can post the data that describes all of your assets. To use these APIs, your application makes HTTPS requests and parses the response. You can use any web-development language to access the APIs.

Requests timeout automatically after 120 seconds.

Request Methods

Method	Description
`POST`	Creates or updates a set of resources. You can `POST` to a collection, but not to a specific resource. A `POST` request cannot exceed 50 MB. If you plan to use Audit History, you need to add Audit History metadata to your request headers.
`GET`	Retrieves resources. Use `GET /assets` to return all domain objects.
`PUT`	Creates or updates a single resource. You can use this method for an individual resource, but not for a collection.
`PATCH`	Update values of a resource property. You can add, remove, modify, and replace values of a resource record. You can use this method for an individual resource, but not for a collection. Use this method if you are concerned about network efficiency.
`DELETE`	Removes or deletes a resource. You can use this method for an individual resource, but not for a collection. **Note:** Excessive `DELETE` requests un-stabilize Predix-Asset service. Use them only occasionally when necessary to delete an object.

Request Parameters

Request Parameters	Description
`fields`	Retrieves selected fields of a large object or a collection of large objects. For example, to retrieve a few attributes from an asset with many attributes, indicate the selected attributes in the `fields` clause. See Partial Response Requests on page 41.
`filter`	Use Graph Expression Language (GEL) in the `filter` clause of a GET request to filter the data that appears in results. See Graph Expression Language (GEL) Syntax on page 27.
`pageSize`	Defines the number of entities to be returned per page (default is 100). Maximum `pageSize` is 1000. See Pagination Requests on page 40.

The following sample request uses `filter` and the `fields` request parameters:

```
http://<your-asset_instance-uri>/locomotives?filter=type=Diesel-electric&fields=uri,model
```

The request syntax is described in the following table:

Request item	Description
`http://<your-asset_instance-uri>/`	URI of your Asset service instance, from the VCAP_SERVICES environment variable. A typical URI looks similar to this: `http://predix-asset-rc.grc-apps.svc.ice.ge.com`.
`/locomotives`	Target endpoint for your request.
`?`	Separates the endpoint from the request parameters.
`filter=type=Diesel-electric`	Filter clause containing a GEL query requesting all `locomotives` of the type `Diesel-electric`.
`&`	Separates request parameters.
`fields=uri,model`	Clause requesting that only `uri` and `model` are returned.

See also Sample Asset Service API Requests on page 40.

Graph Expression Language (GEL) Syntax

Understand the Graph Expression Language (GEL) that is used to query and update Predix Asset repositories.

Use Graph Expression Language (GEL) in the `filter` clause of a API request to run a query against Asset service data. You can use any combination of operators or multiple operators of the same type.

The `filter` clause in a GEL query allows you to filter the data that appears in results. This section lists several operators and expressions with examples, but you can use any combination of operators or multiple operators of the same type. Filters are not allowed in `POST`, `PUT`, `PATCH`, or `DELETE` requests.

GEL Symbols and Operators

Use the following syntax when constructing GEL queries.

Symbol	Operator	Interpretation
()		Use parentheses to override the default left-to-right order of operations. Evaluation then takes place from left to right within the parentheses first. The single result is then used as a single term for evaluation outside the parentheses.
:	AND	Read the colon character ":" as AND. The ampersand "&" is a reserved character and has special meaning in URLs. See EQUALS (=) combined with AND (:).
|	OR	Read the pipe "|"as OR. See EQUALS (=) combined with OR (|).
..	RANGE	Read two periods (..) as a range between two numbers, two dates, or two strings. See RANGE (..) Operator Requests on page 48

Using the Dot Character in Keys and Values

If your JSON Key value includes a dot character, you need to escape the Dot (.) in a query to pass the information after the Dot in the query to the server. Otherwise, your query will be truncated at the Dot.

```
uri: "/assets/1",
model.id: "76865"
```

To query this information: `filter=model\.id=76865`

If a Dot appears in your data value, do not escape the Dot.

```
filter=hqLatLng.lat=47.655492
```

Symbol	Operator	Interpretation
.	Dot	The dot (.) is used to flatten keys of internal JSON objects. Example: `value1.value2` If a key contains a dot, there are 2 cases you must consider. Case 1: The key has the dot in the data posted by the user. In this case the user needs to escape it with a backslash at the time of querying so that it is not interpreted as a control character in the GEL syntax. Example JSON: `{uri: /assets/1, a.b : c}` The query for "c" would look like this: `filter=a\.b=c` Case 2: The key has a dot which is formed when internal objects are flattened. In this case, do not escape the dot character. The key has the dot in the data posted by the user. In this case the user needs to escape it with a backslash so that it is not interpreted as a control character in the GEL syntax. On the other hand, if the value contains the dot character, do not escape it. Example JSON: `{uri : /assets/1, a:{b:c}}` To query for "c" in this case, do not escape the dot. `filter=a.b=c` .

RELATE Operations

Use forward and backward relate requests to query relationships of objects stored in your asset model. There are two ways to traverse relationships:

- A forward-relate request uses the “>” operator and traverses in the direction of the relationship.
- A backward-relate request uses the “<” operator and traverses in the opposite direction of the relationship.

See RELATE (”>” and “<”) Operators Requests on page 55.

Symbol	Operator	Interpretation
>	FORWARD-RELATE	Read the greater than character “>” as forward-relate. It selects values stored in the currently selected objects based on predicate (property key). The new selection is the set of values where the object.predicate == value is included in the current selection.
>	FORWARD-RELATE	Read the greater than character “>” as forward-relate. It selects values stored in the currently selected objects based on predicate (property key). The new selection is the set of values where the object.predicate == value is included in the current selection.
> [token]	FORWARD-RELATE with transitive closure	Read the letter “t” followed by a number enclosed in square brackets [t*n*] as a transitive closure token. For example, the following query retrieves all parent assets of `piston` asset, moving up five levels in the asset hierarchy: `/asset?filter=name=piston>parent[t5]`
<	BACKWARD-RELATE	Read the less than character “<” as backward-relate. If the predicate is the property key in another REST resource, and the value in the URI of a REST resource in the current selection, then that object is selected. For example, the following query finds all locomotives with a given manufacturer: `<asset-app-url>/locomotive?filter=(name=General Electric Transportation)<manufacturer` See RELATE (”>” and “<”) Operators Requests on page 55.
<[token]	BACKWARD-RELATE with transitive closure	Read the letter “t” followed by a number enclosed in square brackets [t*n*] as a transitive closure token. For example, the following query retrieves all child assets of the `GE-Aircraft` asset, moving five levels down in the asset hierarchy: `/asset?filter=name=GE-Aircraft<parent[t5]`

Using Special Characters in GEL Query Keys and Values

If your query keys and values include special characters that conflict with the GEL operators, escape the special character with a backslash to prevent a processing conflict.

For example, for a forward relate (>) in a key, precede the > with a backslash (\>) before you submit the query.

Character Limitations for Attribute Keys and Values

Attribute keys and attribute values are limited to 65,000 characters.

Entity	Behavior
Attribute values	If an attribute value contains more than 65,000 characters, the JSON object will be stored in the Asset repository, but searching on a 65,000-character value is not supported.

Attribute keys	An attribute key cannot contain more than 65,000 characters. The JSON will be rejected at the time of the `PUT` or `POST` request.
	The following error message will be returned:
	`PA_ENTITY_KEY_EXCEEDS_CHAR_LIMIT`
	`"Entity keys may not contain more than 65000 characters"`

Creating GEL Queries

Follow this railroad diagram to create your GEL queries. Start on the left edge and follow the tracks to the right edge. Oval shapes represent literals. Rounded rectangles represent rules and descriptions.

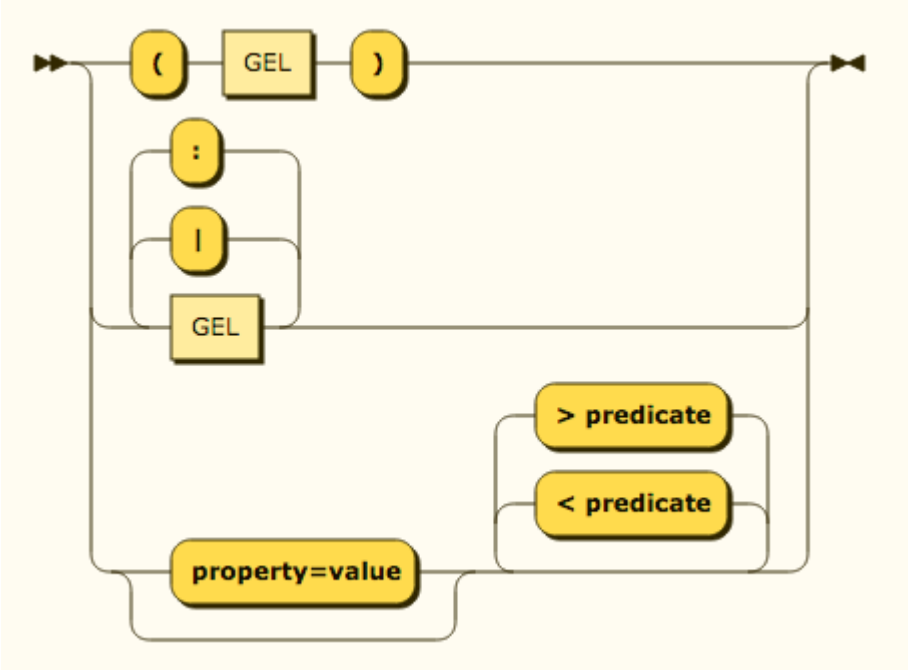


Figure 6: GEL Query Railroad Diagram

Unbinding an Asset Service Instance

Use the `cf unbind-service` command to unbind and asset instance from an application.

Procedure

- Unbind an Asset service instance:
  where
  - `<appName>` is the name of your application.
  - `<service-instance-name>` is `my_asset_instance`.

The service instance is unbound from the application.

Deleting an Asset Service Instance

Use the `cf delete-service` command to delete an asset service instance. Deleting an Asset service instance also deletes any data that is in the Asset service instance data repositories.

Procedure

- Delete an Asset service instance:

```
cf delete-service predix-asset <my_asset_instance>
```

Validating Asset Data

Validate Predix Asset data against a JSON schema upon ingestion.

Data ingested into a Predix Asset service is automatically validated against the schema that defines the domain objects.

Supported REST Methods

- `GET` to display the current schema that defines your domain objects.
- `DELETE` to delete a current schema that defines your domain objects; you cannot modify properties, but you can delete the current schema and replace it with a new one.
- `PUT` to upload a new schema; you can add properties to an existing schema or replace an entire schema, but you cannot modify any existing properties in the current schema.

Note: JSON Version 4 is required. Any other version will return an error code.

Syntax

```
/system/schema/entities/{entity-type}
```

Endpoint

```
/system/schema/entities/assets
```

Troubleshooting Asset Data Validation

Error messages are returned showing the error code, customized message describing the error, and the suggested resolution, as shown in this example:

```
"code": "PA_SCHEMA_DATA_VALIDATION_FAILURE",
 "message": "Provided data has failed schema validation(s)",
"suggestion": "Please ensure it conforms with defined
     schema",
```

Error Condition	Response Status Code	Event Message and Resolution
The uploaded data does not match the schema that describes the domain object.	422	Message: Provided data has failed schema validation(s)", Suggestion: Ensure that the data you provide is compliant with the schema definition.
The schema could not be uploaded.	422	Messages: Provided schema has failed JSON schema validation(s). Suggestion: Please ensure the schema conforms to JSON schema, Version 4.
An attempt to remove a property from an existing schema failed.	422	Message: Provided schema ca not remove existing schema property. Suggestion: Property <custom field> can not be removed. You can add new properties to an existing schema, but you cannot remove an existing property. You must delete the existing schema and upload a new one to change properties.
An attempt to modify an existing schema property failed	422	Message: Provided schema cannot change existing schema property type. Suggestion: Property <property-name> cannot change from type {1} to {2}. You cannot change a property type in an existing schema. You must delete the schema and upload a new one.
An attempt to upload a schema has failed because the schema contained an incorrect property type.	422	Message: Provided schema has failed JSON schema validations(s). Suggestion: The schema must contain: {properties:{uri:{type:'string'}}}. Any other property type causes the schema validation to fail.
Other error notifications.	422	Message: A custom message will be sent indicating details of the error Suggestion: .

Using Asset Audit History

About Asset Audit History

The Asset audit history service provides APIs that enable you to retrieve historical information about REST requests in your Asset service repositories.

Before you begin, you must configure your audit metadata in your request headers. See Asset Service REST Request Headers on page 34.

The audit history APIs provide the following endpoints:

- `/<predix-asset-url>/system/configs`
  The audit history service is disabled by default. Use this endpoint to enable or disable the asset audit history service.
- `/<predix-asset-url>/system/audit`
  Use this endpoint to retrieve the old values and the new values of the requested assets.
- `/<predix-asset-url>/system/audit/changes`
  Use this endpoint to retrieve an old value and a new value record for each element in a collection that has changed.
- `/<predix-asset-url>/system/audit/snapshots`
  Use this endpoint to retrieve details about a single asset at a moment in time.

Enabling or Disabling Asset Audit History

Make sure that audit history is disabled before you begin a bulk upload of data. Enable audit history only when you want to track asset record changes.

Procedure

- To check the state of audit history, submit this GET request:

  ```
  GET <predix-asset-url>/system/configs
  ```

  The JSON returned indicates either `auditEnabled true` or `auditEnabled false`.
- To enable or disable audit history explicitly, submit a POST request with `"enableAudit" : true` or `"enableAudit" : false` in the JSON body:

  Submit a POST request:

  ```
  POST <predix-asset-url>/system/configs
  ```

  To enable audit history, include this JSON array in the body:

  ```
  [
  {
      "enableAudit" : true
  }
  ]
  ```

To disable audit history, include this JSON array in the body:

```
[
{
    "enableAudit" : false
}
]
```

Asset Service REST Request Headers

All REST requests must contain the following headers for multi-tenant support (provided by Predix UAA Service). Whenever the application makes a call to the service, these headers identify which tenant the application belongs to.

UAA Authorization

- Header name: "`Authorization`"
- Header value: "Bearer <*token obtained from trusted issuer*>". For more information, see Connecting Your Application to a Platform Service Instance.

Predix Zone

- Header name: The `zone-http-header-name` from the VCAP_SERVICES environment variable for your Asset service instance. For example, `Predix-Zone-Id`.
- Header value: The `zone-http-header-value` from the VCAP_SERVICES environment variable for your Asset service instance. For example, `ba99ab7b-172f-4fe4-8ce9-45afa8a4ab0c`.

Audit History

- Header name:
  `x-audit-metadata`
- Example header value:

```
{"userId":"abc","reason"."xxx","evenTimestamp":"2016-03-25T06:10:3
6.661+05:30".eventId"."new equip2","masterEventId":new
asset","otherInfo":"blue"}
```

See Audit History Metadata Header on page 34

Audit History Metadata Header

If you add the `x-audit-metadata` header to an Asset request, you can later query the audit history of the values included in the header. The results of the request include system-generated and user-defined metadata.

System-Generated Metadata Attributes

Predix Asset service provides the following system-generated attributes:

Attribute	Attribute Description
`timestamp`	Indicates the date and time the request was submitted.
`action`	Indicates the operation (CRUD) used in the request.
`elementOldValue` `elementNewValue`	Indicates the value of the element before the update and the new value after the change.

These attributes are optional and are tagged as metadata and indexed so they will return meaningful results if used in queries.

User-defined Metadata Attributes

You can define your own audit history metadata and use it in your headers:

Attribute	Attribute Description
`userId`	Identifies the `userID` as you have defined it. For example, you might define `userID` as the person who updated the record, or track the person who performed the physical work.
`eventTimestamp`	Tracks the date and time a submitted change to an asset actually occurs. For example, a maintenance event might be entered into the system at a given date and time, but the actual maintenance might have been performed before the data was updated. The `eventTimestamp` allows users to track when the actual change was made.
`reason`	Indicates the purpose of the request by a text string or a predefined set of codes defined for the business.
`eventId`	Customers can create separate change events that track metadata about a change. If a separate event is part of the asset model, then `eventId` can be used to link the change on the asset to the details found in the event. Alternately, customers could use this as a descriptor of the type of change that has taken place, such as “Scheduled Maintenance” or “Annual Inspection”.
`masterEventId`	The `masterEventId` can be used to tie smaller changes into a larger overarching change. For example, an engine overhaul master event would include many smaller, discrete events such as change the oil, replace valve stem seals, replace cylinders, and so on. Each of these individual changes would be recorded as part of the larger engine overhaul event.
`otherInfo`	Other info to be determined by the user.

Example Audit History API Requests

Example API requests for Predix Asset Audit History feature.

/system/audit Requests

Get Audit History for Assets for a Specific Collection

This example uses the `identifier` parameter to retrieve the history for assets with `/locomotives*` in the `uri`.

```
<asset-app-url>/system/audit?filter=identifier=/locomotives*
```

Returns


```
[
  {
    "id": 789,
    "versionId": -1,
    "userId": "",
    "timestamp": "2016-04-13T10:16:42.016-07:00[America/Los_Angeles]",
    "reason": "",
    "action": "CREATE",
    "identifier": "/locomotives/89",
    "eventId": "",
    "masterEventId": "",
    "otherInfo": "",
    "oldValue": "{}",
    "newValue": "{\n  \"uri\": \"/locomotives/89\",\n  \"type\":
\"Diesel-electric\",\n  \"model\": \"3GS21B-89\",\n  \"serial_no\":
\"0086\",\n  \"emission_tier\": \"0+\",\n  \"fleet\": \"/fleets/cn-4\",
\n  \"manufacturer\": \"/manufacturers/national-railway-equipment\",
\n  \"engine\": \"/engines/QSK19-28\",\n  \"hqLatLng\": {\n    \"lat
\": 46.244201,\n    \"lng\": -73.666786\n  },\n  \"parent\": \"/
locomotives/88\",\n  \"installedOn\": \"05/12/2015\",\n  \"dateIso\":
\"2015-12-05T13:15:31Z\"\n}"
  },
  {
    "id": 788,
    "versionId": -1,
    "userId": "",
    "timestamp": "2016-04-13T10:16:42.016-07:00[America/Los_Angeles]",
    "reason": "",
    "action": "CREATE",
    "identifier": "/locomotives/88",
    "eventId": "",
    "masterEventId": "",
    "otherInfo": "",
    "oldValue": "{}",
    "newValue": "{\n  \"uri\": \"/locomotives/88\",\n  \"type\":
\"Diesel-electric\",\n  \"model\": \"3GS21B-88\",\n  \"serial_no\":
\"0085\",\n  \"emission_tier\": \"0+\",\n  \"fleet\": \"/fleets/cn-4\",
\n  \"manufacturer\": \"/manufacturers/national-railway-equipment\",
\n  \"engine\": \"/engines/QSK19-27\",\n  \"hqLatLng\": {\n    \"lat
```

```
\": 49.640363,\n    \"lng\": -126.242998\n  },\n  \"parent\": \"/
locomotives/87\",\n  \"dateIso\": \"2015-11-05T13:15:30Z\"\n}"
  },
  {
    "id": 787,
    "versionId": -1,
    "userId": "",
    "timestamp": "2016-04-13T10:16:42.015-07:00[America/Los_Angeles]",
    "reason": "",
    "action": "CREATE",
    "identifier": "/locomotives/87",
    "eventId": "",
    "masterEventId": "",
    "otherInfo": "",
    "oldValue": "{}",
    "newValue": "{\n  \"uri\": \"/locomotives/87\",\n  \"type\":
\"Diesel-electric\",\n  \"model\": \"3GS21B-87\",\n  \"serial_no\":
\"0087\",\n  \"emission_tier\": \"0+\",\n  \"fleet\": \"/fleets/cn-4\",
\n  \"manufacturer\": \"/manufacturers/national-railway-equipment\",
\n  \"engine\": \"/engines/QSK19-26\",\n  \"hqLatLng\": {\n    \"lat
\": 49.296307,\n    \"lng\": -118.827314\n  },\n  \"parent\": \"/
locomotives/86\",\n  \"installedOn\": \"05/12/2014\",\n  \"dateIso\":
\"2014-12-05T13:15:31Z\"\n}"
  },
  {
    "id": 786,
    "versionId": -1,
    "userId": "",
    "timestamp": "2016-04-13T10:16:42.015-07:00[America/Los_Angeles]",
    "reason": "",
    "action": "CREATE",
    "identifier": "/locomotives/86",
    "eventId": "",
    "masterEventId": "",
    "otherInfo": "",
    "oldValue": "{}",
    "newValue": "{\n  \"uri\": \"/locomotives/86\",\n  \"type\":
\"Diesel-electric\",\n  \"model\": \"3GS21B\",\n  \"serial_no\":
\"0086\",\n  \"emission_tier\": \"0+\",\n  \"fleet\": \"/fleets/cn-4\",
\n  \"manufacturer\": \"/manufacturers/national-railway-equipment\",
\n  \"engine\": \"/engines/QSK19-28\",\n  \"hqLatLng\": {\n    \"lat
\": 46.244201,\n    \"lng\": -73.666786\n  }\n}"
  },

    {
    ... SNIP
]
```

Get Audit History for a Specific Collection Before a Timestamp

This example uses the `before` parameter to retrieve the history for asset records matching `/planets/venus` with changes before a specific timestamp.

```
GET <asset-app-url>/system/audit?filter=identifier=/planets/
venus:before=2016-08-10T13\:09\:58.747-07\:00
```

Response

```
[
  {
     "id": 608306,
     "versionId": -1,
     "timestamp": "2016-08-10T13:09:58.747-07:00[America/Los_Angeles]",
     "action": "UPDATE",
     "identifier": "/planets/venus",
     "eventTimestamp": "2016-08-10T13:09:58.747-07:00[America/
Los_Angeles]",
     "oldValue": "{\n  \"uri\": \"/planets/venus\",\n  \"color\":
\"gold\",\n  \"radius\": \"3,760 mi\"\n}",
     "newValue": "{\n  \"uri\": \"/planets/venus\",\n  \"color\":
\"gold\",\n  \"radius\": \"3,760 mi\",\n  \"change\": 0\n}"
   },
   {
     "id": 608303,
     "versionId": -1,
     "timestamp": "2016-08-10T13:08:09.119-07:00[America/Los_Angeles]",
     "action": "CREATE",
     "identifier": "/planets/venus",
     "eventTimestamp": "2016-08-10T13:08:09.119-07:00[America/
Los_Angeles]",
     "oldValue": "{}",
     "newValue": "{\n  \"uri\": \"/planets/venus\",\n  \"color\":
\"gold\",\n  \"radius\": \"3,760 mi\"\n}"
   }\]
```

Response

/system/audit/changes Requests

GET a Specific Number of Changed Records

This example uses the `changes` and `pagesize` parameters to retrieve the detailed changes for three asset records.

```
<asset-app-url>/system/audit/changes?pageSize=3
```

Returns

```
[
  {
    "id": 3293,
    "versionId": -1,
    "userId": "",
    "timestamp": "2016-04-13T10:18:43.296-07:00[America/Los_Angeles]",
    "reason": "",
    "action": "CREATE",
    "identifier": "/engines/QSK19-28",
    "eventId": "",
    "masterEventId": "",
```

```
    "otherInfo": "",
    "elementAction": "added",
    "elementUri": "/engines/QSK19-28.manufacturer",
    "elementOldValue": "",
    "elementNewValue": "/manufacturers/cummins"
  },
  {
    "id": 3292,
    "versionId": -1,
    "userId": "",
    "timestamp": "2016-04-13T10:18:43.296-07:00[America/Los_Angeles]",
    "reason": "",
    "action": "CREATE",
    "identifier": "/engines/QSK19-28",
    "eventId": "",
    "masterEventId": "",
    "otherInfo": "",
    "elementAction": "added",
    "elementUri": "/engines/QSK19-28.RPM",
    "elementOldValue": "",
    "elementNewValue": "800"
  },
  {
    "id": 3291,
    "versionId": -1,
    "userId": "",
    "timestamp": "2016-04-13T10:18:43.296-07:00[America/Los_Angeles]",
    "reason": "",
    "action": "CREATE",
    "identifier": "/engines/QSK19-28",
    "eventId": "",
    "masterEventId": "",
    "otherInfo": "",
    "elementAction": "added",
    "elementUri": "/engines/QSK19-28.bore",
    "elementOldValue": "",
    "elementNewValue": "159"
  }
]
```

Sample Asset Service API Requests

Sample Asset Service API Requests

Use the Asset service APIs to retrieve and manipulate resources like assets, classifications, and custom domain objects.

To run the sample requests, use the Asset Model Sample Data on page 59 within a `POST` method. See Adding Asset Model Data to your Asset Service Instance on page 23.

The samples on this page abbreviate the base URL for the host that processes Asset service API requests. You need to include the full URI of your Asset service instance, from the VCAP_SERVICES environment variable, in your requests.

Pagination Requests

Use the `pageSize` parameter to indicate the number of entities to be returned per page. The default value is 100. The maximum `pageSize` is 1000.

For example, this request returns four locomotive objects:

`<asset-app-url>/locomotive?pageSize=4`

The response headers contain a link to the next page of locomotive objects. Copy and paste the link in a `GET` request to display the next page.

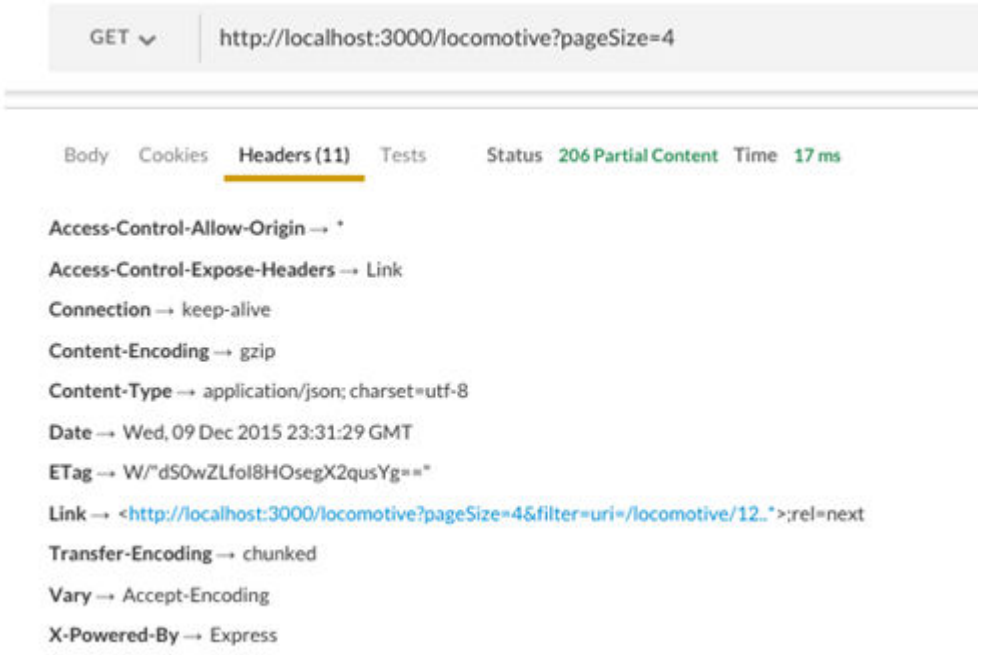


Figure 7: Link in Response Headers

Partial Response Requests

Example API requests and responses for Predix Asset service.

You can submit an API request in which you specify the fields that will be included in the API response. Use the `field` clause in a request method to retrieve selected fields of a large object or a collection of large objects. For example, to retrieve a few attributes from an asset with many attributes, indicate your selected fields in the `field` clause.

Get all fields of an object

`<asset-app-url>/locomotives/1`

Returns

```
[
    {
        "uri": "/locomotives/1",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "001",
        "emission_tier": "0+",
        "fleet": "/fleets/up-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-1",
        "installedOn": "01/12/2005",
        "dateIso": "2005-12-01T13:15:31Z",
        "hqLatLng": {
            "lat": 33.914605,
            "lng": -117.253374
        }
    },
    ]
```

Get selected fields of an object

<asset-app-url>/locomotives/1?fields=uri,type,manufacturer

Returns

```
[
    {
        "uri": "/locomotives/1",
        "type": "Diesel-electric",
        "manufacturer": "/manufacturers/GE"
    }
]
```

Get selected fields of a collection of objects

<asset-app-url>/locomotives?fields=uri,type,manufacturer

Returns

Note: The actual results are much larger than this example response.

```
[
    {
        "uri": "/locomotives/1",
        "type": "Diesel-electric",
        "manufacturer": "/manufacturers/GE"
    },
    {
        "uri": "/locomotives/10",
        "type": "Diesel-electric",
        "manufacturer": "/manufacturers/electro-motive-diesel"
    },
    {
        "uri": "/locomotives/11",
        "type": "Diesel-electric",
        "manufacturer": "/manufacturers/GE"
    },
    {
        "uri": "/locomotives/12",
        "type": "Diesel-electric",
        "manufacturer": "/manufacturers/electro-motive-diesel"
    },
    ... SNIP
```

EQUALS (=) Operator Requests

The EQUALS operator returns any objects that match the specified value.

EQUALS

```
<asset-app-url>/locomotives?filter=serial_no=0084
```

Returns

```
[
    {
        "uri": "/locomotives/84",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0084",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-4",
        "manufacturer": "/manufacturers/national-railway-equipment",
        "engine": "/engines/QSK19-26",
        "hqLatLng": {
            "lat": 49.296307,
            "lng": -118.827314
        }
    }
]
```

EQUALS (=) combined with AND (:)

Read the colon character ":" as AND. The ampersand "&" is a reserved character in URLs.

`<asset-app-url>/locomotives?filter=model=SD70ACe:fleet=/fleets/up-5`

Returns


```
[
    {
        "uri": "/locomotives/15",
        "type": "Diesel-electric",
        "model": "SD70ACe",
        "serial_no": "0015",
        "emission_tier": "0+",
        "fleet": "/fleets/up-5",
        "manufacturer": "/manufacturers/electro-motive-diesel",
        "engine": "/engines/v16-2-8",
        "hqLatLng": {
            "lat": 45.314087,
            "lng": -90.969283
        }
    }
]
```


EQUALS (=) combined with OR (|)

Read the pipe | as OR.

`<asset-app-url>/locomotives?filter=engine=/engines/v16-2-5|fleet=/fleets/csx-1`

Returns


```
[
    {
        "uri": "/locomotives/10",
        "type": "Diesel-electric",
        "model": "SD70ACe",
        "serial_no": "0010",
        "emission_tier": "0+",
        "fleet": "/fleets/up-4",
        "manufacturer": "/manufacturers/electro-motive-diesel",
        "engine": "/engines/v16-2-5",
        "hqLatLng": {
            "lat": 47.941049,
            "lng": -100.126484
        }
    },
    {
        "uri": "/locomotives/37",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "0037",
        "emission_tier": "0+",
        "fleet": "/fleets/csx-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-15",
        "hqLatLng": {
            "lat": 43.488604,
            "lng": -116.913932
        }
    },
    {
```

```
        "uri": "/locomotives/38",
        "type": "Diesel-electric",
        "model": "GL23TCe",
        "serial_no": "0038",
        "emission_tier": "0+",
        "fleet": "/fleets/csx-1",
        "manufacturer": "/manufacturers/electro-motive-diesel",
        "engine": "/engines/v16-2-9",
        "hqLatLng": {
            "lat": 40.26462,
            "lng": -74.496063
        }
    },
    {
        "uri": "/locomotives/39",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "0039",
        "emission_tier": "0+",
        "fleet": "/fleets/csx-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-16",
        "hqLatLng": {
            "lat": 43.650927,
            "lng": -74.570743
        }
    }
]
```

Data Validation Requests

Example schema updates for Predix Asset domain objects.

Use these examples as guidelines to define your own schema updates. Data uploaded to your domain is validated against the schema you have in place at the time of the upload. If you add properties to an existing schema or replace an existing schema, you should update your data to comply with the new schema; Queries can fail

Successful Schema PUT Request

`PUT` Request:`<asset-url>/system/schema/entities/assets`

Request Body:

```
{
    "type" :       "object",
    "properties" : {  "uri" : { "type" :"string" }, "name" :
{ "type" : "string" }},
    "required" :   [ "uri", "name"]
    }
```

PUT Request to Add New Properties to an Existing Schema

Existing Schema:

```
{
```

```
   "type": "object",
   "properties":{
   "uri": {
    "type": "string"
   },
   "name": {
    "type": "string"
   },
  "required":[
   "uri",
   "name"
  ]
}
```

Use a `PUT` request to add properties to the schema. This example shows adding a `"description"` property to the existing schema.

```
{
   "type": "object",
   "properties":{
   "uri": {
    "type": "string"
   },
   "name": {
    "type": "string"
   },
   "description": {
   "type": "string"
   }
  },
  "required":[
   "uri",
   "name"
  ]
}
```

You cannot remove or change a property from an existing schema. You must delete the existing schema and upload a new one. This attempt to remove a property definition returns an error.

```
{
   "type": "object",
   "properties":{
   "uri": {
    "type": "string"
   },
   "name": {
    "type": "string"
   },
  "required":[
   "uri",
   "name"
  ]
}
```

Response:

```
[
  {
   "code":"PA_SCHEMA_VALIDATION_PROP_REMOVED",
   "message":"Provided schema cannot remove existing schema property",
   "suggestion":"Property 'description' cannot be removed",
```

Example Audit History API Requests

Example API requests for Predix Asset Audit History feature.

PATCH Requests

Predix Asset API PATCH request method description and examples. Update properties of a resource using the PATCH request method.

Use the `PATCH` method to replace, add, modify, or copy values in an existing asset record. Predix Asset Supports the entire RFC that defines the `PATCH` API method. For more details about `PATCH`, refer to the RFC5789 at https://tools.ietf.org/html/rfc5789

Example Asset Model

```
[
{
  uri: '/assets/patchJson',
  A1: 'A',
  A2: 'B',
  A3: 'C'
}
]
```

PATCH Request to Replace an Existing Value in the Asset Record

PATCH request: `/assets/patchJson` with body: `[{“op": “replace=", “path": “/A1", value: “X”}]`

Result:

```
[
  {
    uri: '/assets/patchJson',
    A1:  'X',
    A2: 'B',
    A3: 'C'
  }
]
```

PATCH Request to Remove a Value from an Asset Record

PATCH request: `/assets/patchJson` with body:

```
[{"op": "remove", "path": "/A1"}]
```

Result:

```
[
  {
   uri:'assets/patchjson',
   A2 'B',
   A3: 'C'
  }
]
```

PATCH Request to Add a Value to an Asset Record

PATCH request:

```
/assets/patchjson
[
  {
    "op": "add",
    "path":"/A4",
    "value": "D"
  }
]
```

Result:

```
[
  {
    uri: '/assets/patchjson',
    A1: 'A',
    A2: 'B',
    A3: 'C',
    A4: 'D'
   }
]
```

PATCH Request to Copy A Value

PATCH Request:

```
/asset/patchjson
[
  {
   "op":"copy",
   "from":"/A1",
   "path":"/A2"
   }
]
```

Result:

```
[
  {
    uri: '/assets/patchjson',
```

```
    A1: "A",
    A2: "C",
    A3: "C"
   }
]
```

PATCH Request to Move a Value

PATCH Request:

```
/asset/patchjson
[
  {
   "op":"move",
   "from": "/A1",
   "path": "/A2"
   }
]
```

Result:

```
[
  {
    uri: '/assets/patchjson',
    A2: 'A',
    A3: 'C'
   }
]
```

RANGE (..) Operator Requests

Read two periods (..) as a range between two numbers, two dates, or two strings.

RANGE (..) between Two Numbers

<asset-app-url>/locomotives?filter=serial_no=001..004

Returns

```
[
    {
        "uri": "/locomotives/1",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "001",
        "emission_tier": "0+",
        "fleet": "/fleets/up-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-1",
        "installedOn": "01/12/2005",
        "dateIso": "2005-12-01T13:15:31Z",
        "hqLatLng": {
            "lat": 33.914605,
            "lng": -117.253374
        }
```

```
    },
    {
        "uri": "/locomotives/2",
        "type": "Diesel-electric",
        "model": "SD70ACe",
        "serial_no": "002",
        "emission_tier": "0+",
        "fleet": "/fleets/up-1",
        "manufacturer": "/manufacturers/electro-motive-diesel",
        "engine": "/engines/v16-2-1",
        "hqLatLng": {
            "lat": 47.655492,
            "lng": -117.427025
        }
    },
    {
        "uri": "/locomotives/3",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "003",
        "emission_tier": "0+",
        "fleet": "/fleets/up-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-2",
        "installedOn": "02/12/2005",
        "dateIso": "2005-12-02T13:15:31Z",
        "hqLatLng": {
            "lat": 46.860395,
            "lng": -109.473494
        }
    },
    {
        "uri": "/locomotives/4",
        "type": "Diesel-electric",
        "model": "AD40SPe",
        "serial_no": "004",
        "emission_tier": "0+",
        "fleet": "/fleets/up-2",
        "manufacturer": "/manufacturers/electro-motive-diesel",
        "engine": "/engines/v16-2-2",
        "hqLatLng": {
            "lat": 45.784314,
            "lng": -108.500856
        }
    }
]
```

RANGE (..) between Two Strings

`<asset-app-url>/locomotive?filter=model=ES43AC..ES44AC`

Returns

```
[
    {
        "uri": "/locomotives/1",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "001",
        "emission_tier": "0+",
```

```
        "fleet": "/fleets/up-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-1",
        "installedOn": "01/12/2005",
        "dateIso": "2005-12-01T13:15:31Z",
        "hqLatLng": {
            "lat": 33.914605,
            "lng": -117.253374
        }
    },
    {
        "uri": "/locomotives/11",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "0011",
        "emission_tier": "0+",
        "fleet": "/fleets/up-4",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-6",
        "hqLatLng": {
            "lat": 32.7086,
            "lng": -108.190375
        }
    },
    {
        "uri": "/locomotives/14",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "0014",
        "emission_tier": "0+",
        "fleet": "/fleets/up-5",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-7",
        "hqLatLng": {
            "lat": 35.931077,
            "lng": -79.643374
        }
    },
    {
        "uri": "/locomotives/3",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "003",
        "emission_tier": "0+",
        "fleet": "/fleets/up-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-2",
        "installedOn": "02/12/2005",
        "dateIso": "2005-12-02T13:15:31Z",
        "hqLatLng": {
            "lat": 46.860395,
            "lng": -109.473494
        }
    },
    {
        "uri": "/locomotives/37",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "0037",
        "emission_tier": "0+",
        "fleet": "/fleets/csx-1",
```

```
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-15",
        "hqLatLng": {
            "lat": 43.488604,
            "lng": -116.913932
        }
    },
    {
        "uri": "/locomotives/39",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "0039",
        "emission_tier": "0+",
        "fleet": "/fleets/csx-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-16",
        "hqLatLng": {
            "lat": 43.650927,
            "lng": -74.570743
        }
    },
    {
        "uri": "/locomotives/41",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "0041",
        "emission_tier": "0+",
        "fleet": "/fleets/csx-2",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-17",
        "hqLatLng": {
            "lat": 45.40759,
            "lng": -103.471649
        }
    },
    {
        "uri": "/locomotives/43",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "0043",
        "emission_tier": "0+",
        "fleet": "/fleets/csx-2",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-18",
        "hqLatLng": {
            "lat": 39.173252,
            "lng": -90.552123
        }
    },
    {
        "uri": "/locomotives/45",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "0045",
        "emission_tier": "0+",
        "fleet": "/fleets/csx-3",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-19",
        "hqLatLng": {
            "lat": 41.394707,
            "lng": -119.079633
```

```
        }
    },
    {
        "uri": "/locomotives/5",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "005",
        "emission_tier": "0+",
        "fleet": "/fleets/up-2",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-3",
        "installedOn": "10/12/2005",
        "dateIso": "2005-12-10T13:15:31Z",
        "hqLatLng": {
            "lat": 35.022757,
            "lng": -83.009365
        }
    },
    {
        "uri": "/locomotives/7",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "007",
        "emission_tier": "0+",
        "fleet": "/fleets/up-3",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-4",
        "installedOn": "12/12/2005",
        "dateIso": "2005-12-12T13:15:31Z",
        "hqLatLng": {
            "lat": 36.739438,
            "lng": -79.66268
        }
    },
    {
        "uri": "/locomotives/9",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "009",
        "emission_tier": "0+",
        "fleet": "/fleets/up-3",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-5",
        "hqLatLng": {
            "lat": 36.669408,
            "lng": -115.617391
        }
    }
]
```


About Date RANGE (..) Queries

If you select a range between dates that are stored in any other format except ISO, the dates are compared as strings. Compare the following two query result sets. When a date is not in ISO format, the dates are compared as strings, and the results can return dates outside of the intended date range.

Date as String RANGE query

`<asset-app-url>/locomotive?filter=installedOn=01/12/2005..10/12/2005`

Returns


```
[
    {
        "uri": "/locomotives/1",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "001",
        "emission_tier": "0+",
        "fleet": "/fleets/up-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-1",
        "installedOn": "01/12/2005",
        "dateIso": "2005-12-01T13:15:31Z",
        "hqLatLng": {
            "lat": 33.914605,
            "lng": -117.253374
        }
    },
    {
        "uri": "/locomotives/3",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "003",
        "emission_tier": "0+",
        "fleet": "/fleets/up-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-2",
        "installedOn": "02/12/2005",
        "dateIso": "2005-12-02T13:15:31Z",
        "hqLatLng": {
            "lat": 46.860395,
            "lng": -109.473494
        }
    },
    {
        "uri": "/locomotives/5",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "005",
        "emission_tier": "0+",
        "fleet": "/fleets/up-2",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-3",
        "installedOn": "10/12/2005",
        "dateIso": "2005-12-10T13:15:31Z",
        "hqLatLng": {
            "lat": 35.022757,
            "lng": -83.009365
        }
    },
    {
        "uri": "/locomotives/87",
        "type": "Diesel-electric",
        "model": "3GS21B-87",
        "serial_no": "0087",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-4",
        "manufacturer": "/manufacturers/national-railway-equipment",
        "engine": "/engines/QSK19-26",
        "hqLatLng": {
```

```
            "lat": 49.296307,
            "lng": -118.827314
        },
        "parent": "/locomotives/86",
        "installedOn": "05/12/2014",
        "dateIso": "2014-12-05T13:15:31Z"
    },
    {
        "uri": "/locomotives/89",
        "type": "Diesel-electric",
        "model": "3GS21B-89",
        "serial_no": "0086",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-4",
        "manufacturer": "/manufacturers/national-railway-equipment",
        "engine": "/engines/QSK19-28",
        "hqLatLng": {
            "lat": 46.244201,
            "lng": -73.666786
        },
        "parent": "/locomotives/88",
        "installedOn": "05/12/2015",
        "dateIso": "2015-12-05T13:15:31Z"
    }
]
```

Date in ISO date format RANGE Query

Use the backslash "\" to escape colons in an ISO date.`<asset-app-url>locomotive?filter=dateIso=2005-12-01T13\:15\:31Z..2005-12-10T13\:15\:31Z`

Returns

```
[
    {
        "uri": "/locomotives/1",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "001",
        "emission_tier": "0+",
        "fleet": "/fleets/up-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-1",
        "installedOn": "01/12/2005",
        "dateIso": "2005-12-01T13:15:31Z",
        "hqLatLng": {
            "lat": 33.914605,
            "lng": -117.253374
        }
    },
    {
        "uri": "/locomotives/3",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "003",
        "emission_tier": "0+",
        "fleet": "/fleets/up-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-2",
        "installedOn": "02/12/2005",
```

```
        "dateIso": "2005-12-02T13:15:31Z",
        "hqLatLng": {
            "lat": 46.860395,
            "lng": -109.473494
        }
    },
    {
        "uri": "/locomotives/5",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "005",
        "emission_tier": "0+",
        "fleet": "/fleets/up-2",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-3",
        "installedOn": "10/12/2005",
        "dateIso": "2005-12-10T13:15:31Z",
        "hqLatLng": {
            "lat": 35.022757,
            "lng": -83.009365
        }
    }
]
```

RELATE (”>” and “<”) Operators Requests

Use forward and backward requests to query relationships of objects stored in your asset model. There are two ways to traverse relationships:

- A forward-relate request uses the “>” operator and traverses in the direction of the relationship.
- A backward-relate request uses the “<” operator and traverses in the opposite direction of the relationship.

RELATE (>)

Retrieves all engines linked to locomotives of type `Diesel-electric`.

`<asset-app-url>/engines?filter=type=Diesel-electric>engine`

Returns

Note: The actual results are much larger than this example response.

```
[
    {
        "uri": "/engines/645",
        "type": "7FDL",
        "horsepower": "1500",
        "stroke": "190",
        "bore": "230",
        "RPM": "900",
        "manufacturer": "/manufacturers/electro-motive-diesel"
    },
    {
        "uri": "/engines/QSK19-1",
        "type": "Diesel",
        "horsepower": "2100",
        "stroke": "159",
```

```
        "bore": "159",
        "RPM": "800",
        "manufacturer": "/manufacturers/cummins"
    },
    {
        "uri": "/engines/QSK19-10",
        "type": "Diesel",
        "horsepower": "2100",
        "stroke": "159",
        "bore": "159",
        "RPM": "800",
        "manufacturer": "/manufacturers/cummins"
    },
    {
        "uri": "/engines/QSK19-11",
        "type": "Diesel",
        "horsepower": "2100",
        "stroke": "159",
        "bore": "159",
        "RPM": "800",
        "manufacturer": "/manufacturers/cummins"
    },
    {
    ... SNIP
]
```

BACKWARDS RELATE (<)

Retrieves all locomotives with a given manufacturer.

```
<asset-app-url>/locomotives?filter=(name=General Electric
Transportation)<manufacturers
```

Returns

Note: The actual results are much larger than this example response.

```
[
    {
        "uri": "/locomotives/1",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "001",
        "emission_tier": "0+",
        "fleet": "/fleets/up-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-1",
        "installedOn": "01/12/2005",
        "dateIso": "2005-12-01T13:15:31Z",
        "hqLatLng": {
            "lat": 33.914605,
            "lng": -117.253374
        }
    },
    {
        "uri": "/locomotives/11",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "0011",
        "emission_tier": "0+",
```

```
        "fleet": "/fleets/up-4",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-6",
        "hqLatLng": {
            "lat": 32.7086,
            "lng": -108.190375
        }
    },
    {
        "uri": "/locomotives/14",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "0014",
        "emission_tier": "0+",
        "fleet": "/fleets/up-5",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-7",
        "hqLatLng": {
            "lat": 35.931077,
            "lng": -79.643374
        }
    },
    {
        "uri": "/locomotives/16",
        "type": "Diesel-electric",
        "model": "C40-8W ",
        "serial_no": "0016",
        "emission_tier": "0+",
        "fleet": "/fleets/bnsf-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v16-1",
        "hqLatLng": {
            "lat": 47.781463,
            "lng": -116.940009
    ... SNIP

]
```

Wild Card Search with Regular Expression

This query returns any objects that have an expression matching a pattern of strings in the set.

`<asset-app-url>/locomotives?filter=type=/.*ese.*/`

Returns

Note: The actual results are much larger than this example response.

```
[
    {
        "uri": "/locomotives/1",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "001",
        "emission_tier": "0+",
        "fleet": "/fleets/up-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-1",
```

```
        "installedOn": "01/12/2005",
        "dateIso": "2005-12-01T13:15:31Z",
        "hqLatLng": {
            "lat": 33.914605,
            "lng": -117.253374
        }
    },
    {
        "uri": "/locomotives/10",
        "type": "Diesel-electric",
        "model": "SD70ACe",
        "serial_no": "0010",
        "emission_tier": "0+",
        "fleet": "/fleets/up-4",
        "manufacturer": "/manufacturers/electro-motive-diesel",
        "engine": "/engines/v16-2-5",
        "hqLatLng": {
            "lat": 47.941049,
            "lng": -100.126484
        }
    },
    {
        "uri": "/locomotives/11",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "0011",
        "emission_tier": "0+",
        "fleet": "/fleets/up-4",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-6",
        "hqLatLng": {
            "lat": 32.7086,
            "lng": -108.190375
        }
    },
         ...SNIP
```

Asset Model Sample Data

Asset Model Sample Data

To run the Sample Asset Service API Requests on page 40, use the asset model sample JSON data within a `POST` method.

See the related JSON samples:

- Locomotives Sample JSON on page 76
- Engines Sample JSON on page 62
- Manufacturers Sample JSON on page 62
- Fleets Sample JSON on page 59

Locomotive Domain Object Model

A transportation company has hundreds of locomotive assets, and each locomotive might have thousands of parts. By constructing an asset model, you can perform the following tasks:

- Manage the parts inventory.
- Quickly locate exact replacements to minimize equipment failures.
- Track asset updates, such as engine replacements

The company can use the Asset service to create a JSON model of its domain objects.

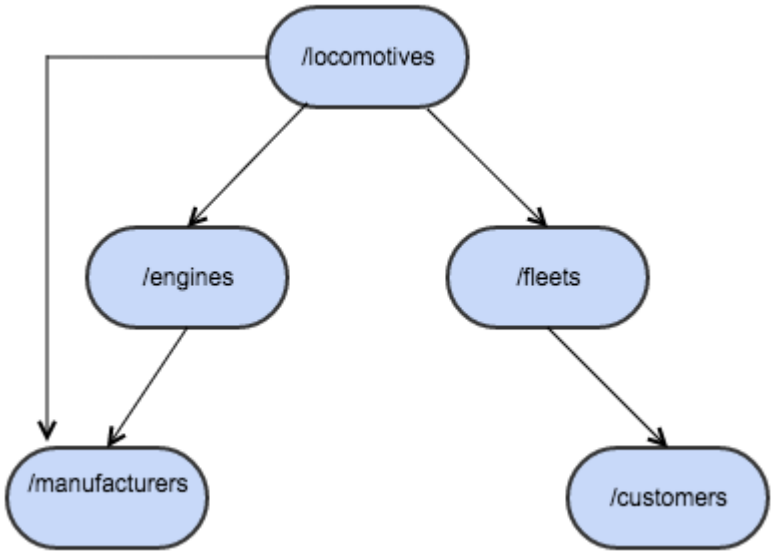


Figure 8: Locomotives Domain Objects

Fleets Sample JSON

`POST` the following sample data if you want to try out the Sample Asset Service API Requests on page 40.

```
[
  {
```

```
    "uri": "/fleets/up-1",
    "name": "Union Pacific Fleet 1",
    "customer": "/customers/union-pacific"
  },
  {
    "uri": "/fleets/up-2",
    "name": "Union Pacific Fleet 2",
    "customer": "/customers/union-pacific"
  },
  {
    "uri": "/fleets/up-3",
    "name": "Union Pacific Fleet 3",
    "customer": "/customers/union-pacific"
  },
  {
    "uri": "/fleets/up-4",
    "name": "Union Pacific Fleet 4",
    "customer": "/customers/union-pacific"
  }
, {
  "uri": "/fleets/up-5",
  "name": "Union Pacific Fleet 5",
  "customer": "/customers/union-pacific"
  },

  {
  "uri": "/fleets/bnsf-1",
  "name": "Burlington Northern Fleet 1",
  "customer": "/customers/burlington-northern-santa-fe"
  },
  {
    "uri": "/fleets/bnsf-2",
    "name": "Burlington Northern Fleet 2",
    "customer": "/customers/burlington-northern-santa-fe"
  },
  {
  "uri": "/fleets/bnsf-3",
  "name": "Burlington Northern Fleet 3",
  "customer": "/customers/burlington-northern-santa-fe"
  },
  {
    "uri": "/fleets/bnsf-4",
    "name": "Burlington Northern Fleet 4",
    "customer": "/customers/burlington-northern-santa-fe"
  },
  {
  "uri": "/fleets/bnsf-5",
  "name": "Burlington Northern Fleet 5",
  "customer": "/customers/burlington-northern-santa-fe"
  },
  {
    "uri": "/fleets/csx-1",
    "name": "CSX Fleet 1",
    "customer": "/customers/csx"
  },
  {
    "uri": "/fleets/csx-2",
    "name": "CSX Fleet 2",
    "customer": "/customers/csx"
  },
  {
```

```
    "uri": "/fleets/csx-3",
    "name": "CSX Fleet 3",
    "customer": "/customers/csx"
  },
  {
    "uri": "/fleets/ns-1",
    "name": "Norfolk Southern Fleet 1",
    "customer": "/customers/norfolk-southern"
  },
  {
    "uri": "/fleets/ns-2",
    "name": "Norfolk Southern Fleet 2",
    "customer": "/customers/norfolk-southern"
  },

  {
    "uri": "/fleets/cn-1",
    "name": "Canadian National Fleet 1",
    "customer": "/customers/canadian-national"
  },
  {
    "uri": "/fleets/cn-2",
    "name": "Canadian National Fleet 2",
    "customer": "/customers/canadian-national"
  },
  {
    "uri": "/fleets/cn-3",
    "name": "Canadian National Fleet 3",
    "customer": "/customers/canadian-national"
  },
  {
    "uri": "/fleets/cn-4",
    "name": "Canadian National Fleet 4",
    "customer": "/customers/canadian-national"
  },
  {
    "uri": "/fleets/cn-5",
    "name": "Canadian National Fleet 5",
    "customer": "/customers/canadian-national"
  },
  {
    "uri": "/fleets/cn-6",
    "name": "Canadian National Fleet 6",
    "customer": "/customers/canadian-national"
  },
  {
    "uri": "/fleets/cn-7",
    "name": "Canadian National Fleet 7",
    "customer": "/customers/canadian-national"
  },
  {
    "uri": "/fleets/cn-8",
    "name": "Canadian National Fleet 8",
    "customer": "/customers/canadian-national"
  }
]
```

Manufacturers Sample JSON

`POST` the following sample data if you want to try out the Sample Asset Service API Requests on page 40.

```
[
  { "uri": "/manufacturers/GE",
    "name": "General Electric Transportation",
    "year_founded": "1892" ,
    "hqLatLng":{"lat": 41.881138, "lng": -87.640666 }
  },

  {"uri": "/manufacturers/electro-motive-diesel",
    "name": "Electro-Motive Diesel",
    "year_founded": "1922",
    "hqLatLng": {"lat": 41.798091, "lng": -87.849247}
  },

  {"uri": "/manufacturers/national-railway-equipment",
    "name": "National Railway Equipment",
    "year_founded": "1984",
    "hqLatLng": {"lat": 41.515181 , "lng":-90.411707 }
  },

  {"uri": "/manufacturers/cummins",
    "name": "Cummins",
    "year_founded": "1919",
    "hqLatLng": {"lat":39.204000, "lng": -85.922910}
  }
]
```

Engines Sample JSON

`POST` the following sample data if you want to try out the Sample Asset Service API Requests on page 40.

```
[
  {
    "uri": "/engines/v12-1",
    "type": "7FDL",
    "horsepower": "4400",
    "stroke": "230",
    "bore": "220",
    "RPM": "2400",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v12-2",
    "type": "7FDL",
    "horsepower": "4400",
    "stroke": "230",
    "bore": "220",
    "RPM": "2400",
    "manufacturer": "/manufacturers/GE"
```

```
  },
  {
    "uri": "/engines/v12-3",
    "type": "7FDL",
    "horsepower": "4400",
    "stroke": "230",
    "bore": "220",
    "RPM": "2400",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v12-4",
    "type": "7FDL",
    "horsepower": "4400",
    "stroke": "230",
    "bore": "220",
    "RPM": "2400",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v12-5",
    "type": "7FDL",
    "horsepower": "4400",
    "stroke": "230",
    "bore": "220",
    "RPM": "2400",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v12-6",
    "type": "7FDL",
    "horsepower": "4400",
    "stroke": "230",
    "bore": "220",
    "RPM": "2400",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v12-7",
    "type": "7FDL",
    "horsepower": "4400",
    "stroke": "230",
    "bore": "220",
    "RPM": "2400",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v12-8",
    "type": "7FDL",
    "horsepower": "4400",
    "stroke": "230",
    "bore": "220",
    "RPM": "2400",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v12-9",
    "type": "7FDL",
    "horsepower": "4400",
    "stroke": "230",
    "bore": "220",
```

```
    "RPM": "2400",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v12-10",
    "type": "7FDL",
    "horsepower": "4400",
    "stroke": "230",
    "bore": "220",
    "RPM": "2400",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v12-11",
    "type": "7FDL",
    "horsepower": "4400",
    "stroke": "230",
    "bore": "220",
    "RPM": "2400",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v12-12",
    "type": "7FDL",
    "horsepower": "4400",
    "stroke": "230",
    "bore": "220",
    "RPM": "2400",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v12-13",
    "type": "7FDL",
    "horsepower": "4400",
    "stroke": "230",
    "bore": "220",
    "RPM": "2400",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v12-14",
    "type": "7FDL",
    "horsepower": "4400",
    "stroke": "230",
    "bore": "220",
    "RPM": "2400",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v12-15",
    "type": "7FDL",
    "horsepower": "4400",
    "stroke": "230",
    "bore": "220",
    "RPM": "2400",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v12-16",
    "type": "7FDL",
    "horsepower": "4400",
```

```
    "stroke": "230",
    "bore": "220",
    "RPM": "2400",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v12-17",
    "type": "7FDL",
    "horsepower": "4400",
    "stroke": "230",
    "bore": "220",
    "RPM": "2400",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v12-18",
    "type": "7FDL",
    "horsepower": "4400",
    "stroke": "230",
    "bore": "220",
    "RPM": "2400",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v12-19",
    "type": "7FDL",
    "horsepower": "4400",
    "stroke": "230",
    "bore": "220",
    "RPM": "2400",
    "manufacturer": "/manufacturers/GE"
  },

  {
    "uri": "/engines/v16-1",
    "type": "7FD",
    "horsepower": "4400",
    "stroke": "210",
    "bore": "220",
    "RPM": "900",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v16-2",
    "type": "7FD",
    "horsepower": "4400",
    "stroke": "210",
    "bore": "220",
    "RPM": "900",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v16-3",
    "type": "7FD",
    "horsepower": "4400",
    "stroke": "210",
    "bore": "220",
    "RPM": "900",
    "manufacturer": "/manufacturers/GE"
  },
  {
```

```
    "uri": "/engines/v16-4",
    "type": "7FD",
    "horsepower": "4400",
    "stroke": "210",
    "bore": "220",
    "RPM": "900",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v16-5",
    "type": "7FD",
    "horsepower": "4400",
    "stroke": "210",
    "bore": "220",
    "RPM": "900",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v16-6",
    "type": "7FD",
    "horsepower": "4400",
    "stroke": "210",
    "bore": "220",
    "RPM": "900",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v16-7",
    "type": "7FD",
    "horsepower": "4400",
    "stroke": "210",
    "bore": "220",
    "RPM": "900",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v16-8",
    "type": "7FD",
    "horsepower": "4400",
    "stroke": "210",
    "bore": "220",
    "RPM": "900",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v16-9",
    "type": "7FD",
    "horsepower": "4400",
    "stroke": "210",
    "bore": "220",
    "RPM": "900",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v16-10",
    "type": "7FD",
    "horsepower": "4400",
    "stroke": "210",
    "bore": "220",
    "RPM": "900",
    "manufacturer": "/manufacturers/GE"
```

```
  },
  {
    "uri": "/engines/v16-11",
    "type": "7FD",
    "horsepower": "4400",
    "stroke": "210",
    "bore": "220",
    "RPM": "900",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v16-12",
    "type": "7FD",
    "horsepower": "4400",
    "stroke": "210",
    "bore": "220",
    "RPM": "900",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v16-13",
    "type": "7FD",
    "horsepower": "4400",
    "stroke": "210",
    "bore": "220",
    "RPM": "900",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v16-14",
    "type": "7FD",
    "horsepower": "4400",
    "stroke": "210",
    "bore": "220",
    "RPM": "900",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v16-15",
    "type": "7FD",
    "horsepower": "4400",
    "stroke": "210",
    "bore": "220",
    "RPM": "900",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v16-16",
    "type": "7FD",
    "horsepower": "4400",
    "stroke": "210",
    "bore": "220",
    "RPM": "900",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v16-17",
    "type": "7FD",
    "horsepower": "4400",
    "stroke": "210",
    "bore": "220",
```

```
    "RPM": "900",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v16-18",
    "type": "7FD",
    "horsepower": "4400",
    "stroke": "210",
    "bore": "220",
    "RPM": "900",
    "manufacturer": "/manufacturers/GE"
  },
  {
    "uri": "/engines/v16-19",
    "type": "7FD",
    "horsepower": "4400",
    "stroke": "210",
    "bore": "220",
    "RPM": "900",
    "manufacturer": "/manufacturers/GE"
  },

  {
    "uri": "/engines/v16-2-1",
    "type": "710",
    "horsepower": "4300",
    "stroke": "250",
    "bore": "200",
    "RPM": "215",
    "manufacturer": "/manufacturers/electro-motive-
diesel"
  },
  {
    "uri": "/engines/v16-2-2",
    "type": "710",
    "horsepower": "4300",
    "stroke": "250",
    "bore": "200",
    "RPM": "215",
    "manufacturer": "/manufacturers/electro-motive-
diesel"
  },
  {
    "uri": "/engines/v16-2-3",
    "type": "710",
    "horsepower": "4300",
    "stroke": "250",
    "bore": "200",
    "RPM": "215",
    "manufacturer": "/manufacturers/electro-motive-
diesel"
  },
  {
    "uri": "/engines/v16-2-4",
    "type": "710",
    "horsepower": "4300",
    "stroke": "250",
    "bore": "200",
    "RPM": "215",
    "manufacturer": "/manufacturers/electro-motive-
diesel"
```

```
  },
  {
    "uri": "/engines/v16-2-5",
    "type": "710",
    "horsepower": "4300",
    "stroke": "250",
    "bore": "200",
    "RPM": "215",
    "manufacturer": "/manufacturers/electro-motive-
diesel"
  },
  {
    "uri": "/engines/v16-2-6",
    "type": "710",
    "horsepower": "4300",
    "stroke": "250",
    "bore": "200",
    "RPM": "215",
    "manufacturer": "/manufacturers/electro-motive-
diesel"
  },
  {
    "uri": "/engines/v16-2-7",
    "type": "710",
    "horsepower": "4300",
    "stroke": "250",
    "bore": "200",
    "RPM": "215",
    "manufacturer": "/manufacturers/electro-motive-
diesel"
  },
  {
    "uri": "/engines/v16-2-8",
    "type": "710",
    "horsepower": "4300",
    "stroke": "250",
    "bore": "200",
    "RPM": "215",
    "manufacturer": "/manufacturers/electro-motive-
diesel"
  },
  {
    "uri": "/engines/v16-2-9",
    "type": "710",
    "horsepower": "4300",
    "stroke": "250",
    "bore": "200",
    "RPM": "215",
    "manufacturer": "/manufacturers/electro-motive-
diesel"
  },
  {
    "uri": "/engines/v16-2-10",
    "type": "710",
    "horsepower": "4300",
    "stroke": "250",
    "bore": "200",
    "RPM": "215",
    "manufacturer": "/manufacturers/electro-motive-
diesel"
  },
```

```
  {
    "uri": "/engines/v16-2-11",
    "type": "710",
    "horsepower": "4300",
    "stroke": "250",
    "bore": "200",
    "RPM": "215",
    "manufacturer": "/manufacturers/electro-motive-
diesel"
  },
  {
    "uri": "/engines/v16-2-12",
    "type": "710",
    "horsepower": "4300",
    "stroke": "250",
    "bore": "200",
    "RPM": "215",
    "manufacturer": "/manufacturers/electro-motive-
diesel"
  },
  {
    "uri": "/engines/v16-2-13",
    "type": "710",
    "horsepower": "4300",
    "stroke": "250",
    "bore": "200",
    "RPM": "215",
    "manufacturer": "/manufacturers/electro-motive-
diesel"
  },

  {
    "uri": "/engines/645",
    "type": "7FDL",
    "horsepower": "1500",
    "stroke": "190",
    "bore": "230",
    "RPM": "900",
    "manufacturer": "/manufacturers/electro-motive-
diesel"
  },

  {
    "uri": "/engines/QSK19-1",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },
  {
    "uri": "/engines/QSK19-2",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },
  {
```

```
    "uri": "/engines/QSK19-3",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },
  {
    "uri": "/engines/QSK19-4",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },
  {
    "uri": "/engines/QSK19-5",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },
  {
    "uri": "/engines/QSK19-6",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },
  {
    "uri": "/engines/QSK19-7",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },
  {
    "uri": "/engines/QSK19-8",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },
  {
    "uri": "/engines/QSK19-9",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
```

```
  },
  {
    "uri": "/engines/QSK19-10",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },
  {
    "uri": "/engines/QSK19-11",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },
  {
    "uri": "/engines/QSK19-12",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },
  {
    "uri": "/engines/QSK19-13",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },
  {
    "uri": "/engines/QSK19-14",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },
  {
    "uri": "/engines/QSK19-15",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },
  {
    "uri": "/engines/QSK19-16",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
```

```
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },
  {
    "uri": "/engines/QSK19-17",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },{
  "uri": "/engines/QSK19-18",
  "type": "Diesel",
  "horsepower": "2100",
  "stroke": "159",
  "bore": "159",
  "RPM": "800",
  "manufacturer": "/manufacturers/cummins"
},{
  "uri": "/engines/QSK19-19",
  "type": "Diesel",
  "horsepower": "2100",
  "stroke": "159",
  "bore": "159",
  "RPM": "800",
  "manufacturer": "/manufacturers/cummins"
},
  {
    "uri": "/engines/QSK19-20",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },
  {
    "uri": "/engines/QSK19-21",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },
  {
    "uri": "/engines/QSK19-22",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },
  {
    "uri": "/engines/QSK19-23",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
```

```
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },
  {
    "uri": "/engines/QSK19-24",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },
  {
    "uri": "/engines/QSK19-25",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },
  {
    "uri": "/engines/QSK19-26",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },
  {
    "uri": "/engines/QSK19-27",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },
  {
    "uri": "/engines/QSK19-28",
    "type": "Diesel",
    "horsepower": "2100",
    "stroke": "159",
    "bore": "159",
    "RPM": "800",
    "manufacturer": "/manufacturers/cummins"
  },


  {
    "uri": "/engines/v20-1",
    "type": "7FDL",
    "horsepower": "5000",
    "stroke": "230",
    "bore": "230",
    "RPM": "650",
    "manufacturer": "/manufacturers/national-railway-
equipment"
  },
  {
```

```
    "uri": "/engines/v20-2",
    "type": "7FDL",
    "horsepower": "5000",
    "stroke": "230",
    "bore": "230",
    "RPM": "650",
    "manufacturer": "/manufacturers/national-railway-
equipment"
  },
  {
    "uri": "/engines/v20-3",
    "type": "7FDL",
    "horsepower": "5000",
    "stroke": "230",
    "bore": "230",
    "RPM": "650",
    "manufacturer": "/manufacturers/national-railway-
equipment"
  },
  {
    "uri": "/engines/v20-4",
    "type": "7FDL",
    "horsepower": "5000",
    "stroke": "230",
    "bore": "230",
    "RPM": "650",
    "manufacturer": "/manufacturers/national-railway-
equipment"
  },
  {
    "uri": "/engines/v20-5",
    "type": "7FDL",
    "horsepower": "5000",
    "stroke": "230",
    "bore": "230",
    "RPM": "650",
    "manufacturer": "/manufacturers/national-railway-
equipment"
  },
  {
    "uri": "/engines/v20-6",
    "type": "7FDL",
    "horsepower": "5000",
    "stroke": "230",
    "bore": "230",
    "RPM": "650",
    "manufacturer": "/manufacturers/national-railway-
equipment"
  },
  {
    "uri": "/engines/v20-7",
    "type": "7FDL",
    "horsepower": "5000",
    "stroke": "230",
    "bore": "230",
    "RPM": "650",
    "manufacturer": "/manufacturers/national-railway-
equipment"
```

```
  }
]
```

Locomotives Sample JSON

`POST` the following sample data if you want to try out the Sample Asset Service API Requests on page 40.

```
[
    {
        "uri": "/locomotives/1",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "001",
        "emission_tier": "0+",
        "fleet": "/fleets/up-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-1",
        "installedOn": "01/12/2005",
        "dateIso": "2005-12-01T13:15:31Z",
        "hqLatLng": {
            "lat": 33.914605,
            "lng": -117.253374
        }
    },
    {
        "uri": "/locomotives/2",
        "type": "Diesel-electric",
        "model": "SD70ACe",
        "serial_no": "002",
        "emission_tier": "0+",
        "fleet": "/fleets/up-1",
        "manufacturer": "/manufacturers/electro-motive-
diesel",
        "engine": "/engines/v16-2-1",
        "hqLatLng": {
            "lat": 47.655492,
            "lng": -117.427025
        }
    },
    {
        "uri": "/locomotives/3",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "003",
        "emission_tier": "0+",
        "fleet": "/fleets/up-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-2",
        "installedOn" : "02/12/2005",
        "dateIso": "2005-12-02T13:15:31Z",
        "hqLatLng": {
            "lat": 46.860395,
            "lng": -109.473494
        }
    },
    {
```

```
        "uri": "/locomotives/4",
        "type": "Diesel-electric",
        "model": "AD40SPe",
        "serial_no": "004",
        "emission_tier": "0+",
        "fleet": "/fleets/up-2",
        "manufacturer": "/manufacturers/electro-motive-
diesel",
        "engine": "/engines/v16-2-2",
        "hqLatLng": {
            "lat": 45.784314,
            "lng": -108.500856
        }
    },
    {
        "uri": "/locomotives/5",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "005",
        "emission_tier": "0+",
        "fleet": "/fleets/up-2",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-3",
        "installedOn": "10/12/2005",
        "dateIso": "2005-12-10T13:15:31Z",
        "hqLatLng": {
            "lat": 35.022757,
            "lng": -83.009365
        }
    },
    {
        "uri": "/locomotives/6",
        "type": "Diesel-electric",
        "model": "SD70ACe",
        "serial_no": "006",
        "emission_tier": "0+",
        "fleet": "/fleets/up-2",
        "manufacturer": "/manufacturers/electro-motive-
diesel",
        "engine": "/engines/v16-2-3",
        "installedOn": "11/12/2005",
        "dateIso": "2005-12-11T13:15:31Z",
        "hqLatLng": {
            "lat": 33.120118,
            "lng": -81.563755
        }
    },
    {
        "uri": "/locomotives/7",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "007",
        "emission_tier": "0+",
        "fleet": "/fleets/up-3",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-4",
        "installedOn": "12/12/2005",
        "dateIso": "2005-12-12T13:15:31Z",
        "hqLatLng": {
            "lat": 36.739438,
            "lng": -79.66268
```

```
        }
    },
    {
        "uri": "/locomotives/8",
        "type": "Diesel-electric",
        "model": "GL23TCe",
        "serial_no": "008",
        "emission_tier": "0+",
        "fleet": "/fleets/up-3",
        "manufacturer": "/manufacturers/electro-motive-
diesel",
        "engine": "/engines/v16-2-4",
        "hqLatLng": {
            "lat": 41.876813,
            "lng": -113.717141
        }
    },
    {
        "uri": "/locomotives/9",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "009",
        "emission_tier": "0+",
        "fleet": "/fleets/up-3",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-5",
        "hqLatLng": {
            "lat": 36.669408,
            "lng": -115.617391
        }
    },
    {
        "uri": "/locomotives/10",
        "type": "Diesel-electric",
        "model": "SD70ACe",
        "serial_no": "0010",
        "emission_tier": "0+",
        "fleet": "/fleets/up-4",
        "manufacturer": "/manufacturers/electro-motive-
diesel",
        "engine": "/engines/v16-2-5",
        "hqLatLng": {
            "lat": 47.941049,
            "lng": -100.126484
        }
    },
    {
        "uri": "/locomotives/11",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "0011",
        "emission_tier": "0+",
        "fleet": "/fleets/up-4",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-6",
        "hqLatLng": {
            "lat": 32.7086,
            "lng": -108.190375
        }
    },
    {
```

```
        "uri": "/locomotives/12",
        "type": "Diesel-electric",
        "model": "SD70ACe",
        "serial_no": "0012",
        "emission_tier": "0+",
        "fleet": "/fleets/up-4",
        "manufacturer": "/manufacturers/electro-motive-
diesel",
        "engine": "/engines/v16-2-6",
        "hqLatLng": {
            "lat": 34.099394,
            "lng": -90.261741
        }
    },
    {
        "uri": "/locomotives/13",
        "type": "Diesel-electric",
        "model": "GL23TCe",
        "serial_no": "0013",
        "emission_tier": "0+",
        "fleet": "/fleets/up-5",
        "manufacturer": "/manufacturers/electro-motive-
diesel",
        "engine": "/engines/v16-2-7",
        "hqLatLng": {
            "lat": 36.050366,
            "lng": -91.340363
        }
    },
    {
        "uri": "/locomotives/14",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "0014",
        "emission_tier": "0+",
        "fleet": "/fleets/up-5",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-7",
        "hqLatLng": {
            "lat": 35.931077,
            "lng": -79.643374
        }
    },
    {
        "uri": "/locomotives/15",
        "type": "Diesel-electric",
        "model": "SD70ACe",
        "serial_no": "0015",
        "emission_tier": "0+",
        "fleet": "/fleets/up-5",
        "manufacturer": "/manufacturers/electro-motive-
diesel",
        "engine": "/engines/v16-2-8",
        "hqLatLng": {
            "lat": 45.314087,
            "lng": -90.969283
        }
    },
    {
        "uri": "/locomotives/16",
        "type": "Diesel-electric",
```

```
        "model": "C40-8W ",
        "serial_no": "0016",
        "emission_tier": "0+",
        "fleet": "/fleets/bnsf-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v16-1",
        "hqLatLng": {
            "lat": 47.781463,
            "lng": -116.940009
        }
    },
    {
        "uri": "/locomotives/17",
        "type": "",
        "model": "B40-8",
        "serial_no": "0017",
        "emission_tier": "0+",
        "fleet": "/fleets/bnsf-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-8",
        "hqLatLng": {
            "lat": 32.863603,
            "lng": -97.327141
        }
    },
    {
        "uri": "/locomotives/18",
        "type": "Diesel-electric",
        "model": "MP15AC",
        "serial_no": "0018",
        "emission_tier": "0+",
        "fleet": "/fleets/bnsf-1",
        "manufacturer": "/manufacturers/electro-motive-
diesel",
        "engine": "/engines/645",
        "hqLatLng": {
            "lat": 42.090136,
            "lng": -102.875767
        }
    },
    {
        "uri": "/locomotives/19",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0019",
        "emission_tier": "0+",
        "fleet": "/fleets/bnsf-2",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-1",
        "hqLatLng": {
            "lat": 32.147267,
            "lng": -83.87744
        }
    },
    {
        "uri": "/locomotives/20",
        "type": "Diesel-electric",
        "model": "ES44DC",
        "serial_no": "0020",
        "emission_tier": "0+",
```

```
        "fleet": "/fleets/bnsf-2",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-9",
        "hqLatLng": {
            "lat": 38.191536,
            "lng": -85.472672
        }
    },
    {
        "uri": "/locomotives/21",
        "type": "Diesel-electric",
        "model": "SD80MAC",
        "serial_no": "0021",
        "emission_tier": "0+",
        "fleet": "/fleets/bnsf-2",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/v20-1",
        "hqLatLng": {
            "lat": 43.49188,
            "lng": -122.944105
        }
    },
    {
        "uri": "/locomotives/22",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0022",
        "emission_tier": "0+",
        "fleet": "/fleets/bnsf-2",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-2",
        "hqLatLng": {
            "lat": 45.96556,
            "lng": -116.989515
        }
    },
    {
        "uri": "/locomotives/23",
        "type": "Diesel-electric",
        "model": "ES44DC",
        "serial_no": "0023",
        "emission_tier": "0+",
        "fleet": "/fleets/bnsf-2",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-10",
        "hqLatLng": {
            "lat": 41.788715,
            "lng": -114.601854
        }
    },
    {
        "uri": "/locomotives/24",
        "type": "Diesel-electric",
        "model": "SD80MAC",
        "serial_no": "0024",
        "emission_tier": "0+",
        "fleet": "/fleets/bnsf-2",
        "manufacturer": "/manufacturers/national-railway-
equipment",
```

```
        "engine": "/engines/v20-2",
        "hqLatLng": {
            "lat": 38.097843,
            "lng": -118.161825
        }
    },
    {
        "uri": "/locomotives/25",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0025",
        "emission_tier": "0+",
        "fleet": "/fleets/bnsf-2",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-3",
        "hqLatLng": {
            "lat": 38.12204,
            "lng": -106.260561
        }
    },
    {
        "uri": "/locomotives/26",
        "type": "Diesel-electric",
        "model": "ES44DC",
        "serial_no": "0026",
        "emission_tier": "0+",
        "fleet": "/fleets/bnsf-2",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-11",
        "hqLatLng": {
            "lat": 36.628722,
            "lng": -103.908348
        }
    },
    {
        "uri": "/locomotives/27",
        "type": "Diesel-electric",
        "model": "SD80MAC",
        "serial_no": "0027",
        "emission_tier": "0+",
        "fleet": "/fleets/bnsf-3",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/v20-3",
        "hqLatLng": {
            "lat": 33.544224,
            "lng": -104.506977
        }
    },
    {
        "uri": "/locomotives/28",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0028",
        "emission_tier": "0+",
        "fleet": "/fleets/bnsf-3",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-4",
        "hqLatLng": {
```

```
            "lat": 36.87125,
            "lng": -102.557858
        }
    },
    {
        "uri": "/locomotives/29",
        "type": "Diesel-electric",
        "model": "ES44DC",
        "serial_no": "0029",
        "emission_tier": "0+",
        "fleet": "/fleets/bnsf-3",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-12",
        "hqLatLng": {
            "lat": 34.708707,
            "lng": -97.179306
        }
    },
    {
        "uri": "/locomotives/30",
        "type": "Diesel-electric",
        "model": "SD80MAC",
        "serial_no": "0030",
        "emission_tier": "0+",
        "fleet": "/fleets/bnsf-3",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/v20-4",
        "hqLatLng": {
            "lat": 41.103915,
            "lng": -78.286601
        }
    },
    {
        "uri": "/locomotives/31",
        "type": "Diesel-electric",
        "model": "SD80MAC",
        "serial_no": "0031",
        "emission_tier": "0+",
        "fleet": "/fleets/bnsf-4",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/v20-5",
        "hqLatLng": {
            "lat": 31.351436,
            "lng": -105.258803
        }
    },
    {
        "uri": "/locomotives/32",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0032",
        "emission_tier": "0+",
        "fleet": "/fleets/bnsf-4",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-5",
        "hqLatLng": {
            "lat": 30.062855,
            "lng": -102.111381
```

```
        }
    },
    {
        "uri": "/locomotives/33",
        "type": "Diesel-electric",
        "model": "ES44DC",
        "serial_no": "0033",
        "emission_tier": "0+",
        "fleet": "/fleets/bnsf-4",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-13",
        "hqLatLng": {
            "lat": 26.634179,
            "lng": -98.514328
        }
    },
    {
        "uri": "/locomotives/34",
        "type": "Diesel-electric",
        "model": "SD80MAC",
        "serial_no": "0034",
        "emission_tier": "0+",
        "fleet": "/fleets/bnsf-4",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/v20-6",
        "hqLatLng": {
            "lat": 29.93051,
            "lng": -95.969945
        }
    },
    {
        "uri": "/locomotives/35",
        "type": "Diesel-electric",
        "model": "ES44DC",
        "serial_no": "0035",
        "emission_tier": "0+",
        "fleet": "/fleets/bnsf-5",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-14",
        "hqLatLng": {
            "lat": 44.940593,
            "lng": -110.825888
        }
    },
    {
        "uri": "/locomotives/36",
        "type": "Diesel-electric",
        "model": "SD80MAC",
        "serial_no": "0036",
        "emission_tier": "0+",
        "fleet": "/fleets/bnsf-5",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/v20-7",
        "hqLatLng": {
            "lat": 41.307371,
            "lng": -104.574386
        }
    },
    {
```

```
        "uri": "/locomotives/37",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "0037",
        "emission_tier": "0+",
        "fleet": "/fleets/csx-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-15",
        "hqLatLng": {
            "lat": 43.488604,
            "lng": -116.913932
        }
    },
    {
        "uri": "/locomotives/38",
        "type": "Diesel-electric",
        "model": "GL23TCe",
        "serial_no": "0038",
        "emission_tier": "0+",
        "fleet": "/fleets/csx-1",
        "manufacturer": "/manufacturers/electro-motive-
diesel",
        "engine": "/engines/v16-2-9",
        "hqLatLng": {
            "lat": 40.26462,
            "lng": -74.496063
        }
    },
    {
        "uri": "/locomotives/39",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "0039",
        "emission_tier": "0+",
        "fleet": "/fleets/csx-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-16",
        "hqLatLng": {
            "lat": 43.650927,
            "lng": -74.570743
        }
    },
    {
        "uri": "/locomotives/40",
        "type": "Diesel-electric",
        "model": "SD70ACe",
        "serial_no": "0040",
        "emission_tier": "0+",
        "fleet": "/fleets/csx-2",
        "manufacturer": "/manufacturers/electro-motive-
diesel",
        "engine": "/engines/v16-2-10",
        "hqLatLng": {
            "lat": 41.394707,
            "lng": -83.905661
        }
    },
    {
        "uri": "/locomotives/41",
        "type": "Diesel-electric",
        "model": "ES44AC",
```

```
        "serial_no": "0041",
        "emission_tier": "0+",
        "fleet": "/fleets/csx-2",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-17",
        "hqLatLng": {
            "lat": 45.40759,
            "lng": -103.471649
        }
    },
    {
        "uri": "/locomotives/42",
        "type": "Diesel-electric",
        "model": "SD70ACe",
        "serial_no": "0042",
        "emission_tier": "0+",
        "fleet": "/fleets/csx-2",
        "manufacturer": "/manufacturers/electro-motive-
diesel",
        "engine": "/engines/v16-2-11",
        "hqLatLng": {
            "lat": 42.284877,
            "lng": -99.513644
        }
    },
    {
        "uri": "/locomotives/43",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "0043",
        "emission_tier": "0+",
        "fleet": "/fleets/csx-2",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-18",
        "hqLatLng": {
            "lat": 39.173252,
            "lng": -90.552123
        }
    },
    {
        "uri": "/locomotives/44",
        "type": "Diesel-electric",
        "model": "SD70ACe",
        "serial_no": "0044",
        "emission_tier": "0+",
        "fleet": "/fleets/csx-2",
        "manufacturer": "/manufacturers/electro-motive-
diesel",
        "engine": "/engines/v16-2-12",
        "hqLatLng": {
            "lat": 40.775543,
            "lng": -123.336355
        }
    },
    {
        "uri": "/locomotives/45",
        "type": "Diesel-electric",
        "model": "ES44AC",
        "serial_no": "0045",
        "emission_tier": "0+",
        "fleet": "/fleets/csx-3",
```

```
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v12-19",
        "hqLatLng": {
            "lat": 41.394707,
            "lng": -119.079633
        }
    },
    {
        "uri": "/locomotives/46",
        "type": "Diesel-electric",
        "model": "GL23TCe",
        "serial_no": "0046",
        "emission_tier": "0+",
        "fleet": "/fleets/csx-3",
        "manufacturer": "/manufacturers/electro-motive-
diesel",
        "engine": "/engines/v16-2-13",
        "hqLatLng": {
            "lat": 39.921845,
            "lng": -111.985095
        }
    },
    {
        "uri": "/locomotives/47",
        "type": "Diesel-electric",
        "model": "C40-8W ",
        "serial_no": "0047",
        "emission_tier": "0+",
        "fleet": "/fleets/ns-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v16-2",
        "hqLatLng": {
            "lat": 41.674215,
            "lng": -110.118111
        }
    },
    {
        "uri": "/locomotives/48",
        "type": "Diesel-electric",
        "model": "C40-8W ",
        "serial_no": "0048",
        "emission_tier": "0+",
        "fleet": "/fleets/ns-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v16-3",
        "hqLatLng": {
            "lat": 43.829065,
            "lng": -108.038521
        }
    },
    {
        "uri": "/locomotives/49",
        "type": "Diesel-electric",
        "model": "C40-8W ",
        "serial_no": "0049",
        "emission_tier": "0+",
        "fleet": "/fleets/ns-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v16-4",
        "hqLatLng": {
            "lat": 48.730575,
```

```
            "lng": -121.358846
        }
    },
    {
        "uri": "/locomotives/50",
        "type": "Diesel-electric",
        "model": "C40-8W ",
        "serial_no": "0050",
        "emission_tier": "0+",
        "fleet": "/fleets/ns-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v16-5",
        "hqLatLng": {
            "lat": 47.988616,
            "lng": -117.610012
        }
    },
    {
        "uri": "/locomotives/51",
        "type": "Diesel-electric",
        "model": "C40-8W ",
        "serial_no": "0051",
        "emission_tier": "0+",
        "fleet": "/fleets/ns-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v16-6",
        "hqLatLng": {
            "lat": 46.141609,
            "lng": -120.321935
        }
    },
    {
        "uri": "/locomotives/52",
        "type": "Diesel-electric",
        "model": "C40-8W ",
        "serial_no": "0052",
        "emission_tier": "0+",
        "fleet": "/fleets/ns-1",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v16-7",
        "hqLatLng": {
            "lat": 44.629206,
            "lng": -120.960034
        }
    },
    {
        "uri": "/locomotives/53",
        "type": "Diesel-electric",
        "model": "C40-8W ",
        "serial_no": "0053",
        "emission_tier": "0+",
        "fleet": "/fleets/ns-2",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v16-8",
        "hqLatLng": {
            "lat": 48.572493,
            "lng": -114.57904
        }
    },
    {
        "uri": "/locomotives/54",
```

```
        "type": "Diesel-electric",
        "model": "C40-8W ",
        "serial_no": "0054",
        "emission_tier": "0+",
        "fleet": "/fleets/ns-2",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v16-9",
        "hqLatLng": {
            "lat": 48.677936,
            "lng": -109.873057
        }
    },
    {
        "uri": "/locomotives/55",
        "type": "Diesel-electric",
        "model": "C40-8W ",
        "serial_no": "0055",
        "emission_tier": "0+",
        "fleet": "/fleets/ns-2",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v16-10",
        "hqLatLng": {
            "lat": 48.572493,
            "lng": -104.927787
        }
    },
    {
        "uri": "/locomotives/56",
        "type": "Diesel-electric",
        "model": "C40-8W ",
        "serial_no": "0056",
        "emission_tier": "0+",
        "fleet": "/fleets/ns-2",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v16-11",
        "hqLatLng": {
            "lat": 46.746176,
            "lng": -104.608737
        }
    },
    {
        "uri": "/locomotives/57",
        "type": "Diesel-electric",
        "model": "C40-8W ",
        "serial_no": "0057",
        "emission_tier": "0+",
        "fleet": "/fleets/ns-2",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v16-12",
        "hqLatLng": {
            "lat": 41.721683,
            "lng": -109.952819
        }
    },
    {
        "uri": "/locomotives/58",
        "type": "Diesel-electric",
        "model": "C40-8W ",
        "serial_no": "0058",
        "emission_tier": "0+",
        "fleet": "/fleets/ns-2",
```

```
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v16-13",
        "hqLatLng": {
            "lat": 43.771496,
            "lng": -107.240897
        }
    },
    {
        "uri": "/locomotives/59",
        "type": "Diesel-electric",
        "model": "C40-8W ",
        "serial_no": "0059",
        "emission_tier": "0+",
        "fleet": "/fleets/ns-2",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v16-14",
        "hqLatLng": {
            "lat": 43.424917,
            "lng": -93.681285
        }
    },
    {
        "uri": "/locomotives/60",
        "type": "Diesel-electric",
        "model": "C40-8W ",
        "serial_no": "0060",
        "emission_tier": "0+",
        "fleet": "/fleets/ns-2",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v16-15",
        "hqLatLng": {
            "lat": 42.018661,
            "lng": -95.994395
        }
    },
    {
        "uri": "/locomotives/61",
        "type": "Diesel-electric",
        "model": "C40-8W ",
        "serial_no": "0061",
        "emission_tier": "0+",
        "fleet": "/fleets/ns-2",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v16-16",
        "hqLatLng": {
            "lat": 33.191119,
            "lng": -100.700378
        }
    },
    {
        "uri": "/locomotives/62",
        "type": "Diesel-electric",
        "model": "C40-8W ",
        "serial_no": "0062",
        "emission_tier": "0+",
        "fleet": "/fleets/ns-2",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v16-17",
        "hqLatLng": {
            "lat": 39.481317,
            "lng": -80.440722
```

```
        }
    },
    {
        "uri": "/locomotives/63",
        "type": "Diesel-electric",
        "model": "C40-8W ",
        "serial_no": "0063",
        "emission_tier": "0+",
        "fleet": "/fleets/ns-2",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v16-18",
        "hqLatLng": {
            "lat": 42.211153,
            "lng": -71.617224
        }
    },
    {
        "uri": "/locomotives/64",
        "type": "Diesel-electric",
        "model": "C40-8W ",
        "serial_no": "0064",
        "emission_tier": "0+",
        "fleet": "/fleets/ns-2",
        "manufacturer": "/manufacturers/GE",
        "engine": "/engines/v16-19",
        "hqLatLng": {
            "lat": 44.171027,
            "lng": -69.992906
        }
    },
    {
        "uri": "/locomotives/65",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0065",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-1",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-6",
        "hqLatLng": {
            "lat": 50.750961,
            "lng": -111.035917
        }
    },
    {
        "uri": "/locomotives/66",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0066",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-1",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-7",
        "hqLatLng": {
            "lat": 54.165178,
            "lng": -125.631473
        }
    },
    {
```

```
        "uri": "/locomotives/67",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0067",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-1",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-8",
        "hqLatLng": {
            "lat": 54.517653,
            "lng": -111.462679
        }
    },
    {
        "uri": "/locomotives/68",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0068",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-1",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-9",
        "hqLatLng": {
            "lat": 57.466212,
            "lng": -113.51957
        }
    },
    {
        "uri": "/locomotives/69",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0069",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-2",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-10",
        "hqLatLng": {
            "lat": 58.648156,
            "lng": -105.758545
        }
    },
    {
        "uri": "/locomotives/70",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0070",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-2",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-11",
        "hqLatLng": {
            "lat": 52.479733,
            "lng": -107.757597
        }
    },
    {
        "uri": "/locomotives/71",
```

```
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0071",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-2",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-12",
        "hqLatLng": {
            "lat": 50.259228,
            "lng": -103.14772
        }
    },
    {
        "uri": "/locomotives/72",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0072",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-2",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-13",
        "hqLatLng": {
            "lat": 49.939207,
            "lng": -108.136696
        }
    },
    {
        "uri": "/locomotives/73",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0073",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-2",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-14",
        "hqLatLng": {
            "lat": 55.252082,
            "lng": -99.996572
        }
    },
    {
        "uri": "/locomotives/74",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0074",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-2",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-15",
        "hqLatLng": {
            "lat": 52.213168,
            "lng": -95.47359
        }
    },
    {
        "uri": "/locomotives/75",
        "type": "Diesel-electric",
```

```
        "model": "3GS21B",
        "serial_no": "0075",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-3",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-16",
        "hqLatLng": {
            "lat": 44.264612,
            "lng": -78.588661
        }
    },
    {
        "uri": "/locomotives/76",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0076",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-3",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-17",
        "hqLatLng": {
            "lat": 44.088886,
            "lng": -65.04682
        }
    },
    {
        "uri": "/locomotives/77",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0077",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-3",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-18",
        "hqLatLng": {
            "lat": 46.513021,
            "lng": -67.070163
        }
    },
    {
        "uri": "/locomotives/78",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0078",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-3",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-19",
        "hqLatLng": {
            "lat": 48.387737,
            "lng": -72.315866
        }
    },
    {
        "uri": "/locomotives/79",
        "type": "Diesel-electric",
        "model": "3GS21B",
```

```
        "serial_no": "0079",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-3",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-20",
        "hqLatLng": {
            "lat": 43.12077,
            "lng": -81.576942
        }
    },
    {
        "uri": "/locomotives/80",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0080",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-4",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-21",
        "hqLatLng": {
            "lat": 49.281775,
            "lng": -117.735624
        }
    },
    {
        "uri": "/locomotives/81",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0081",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-4",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-22",
        "hqLatLng": {
            "lat": 49.267239,
            "lng": -119.584813
        }
    },
    {
        "uri": "/locomotives/82",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0082",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-4",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-23",
        "hqLatLng": {
            "lat": 49.209053,
            "lng": -120.320033
        }
    },
    {
        "uri": "/locomotives/83",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0083",
```

```
        "emission_tier": "0+",
        "fleet": "/fleets/cn-4",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-25",
        "hqLatLng": {
            "lat": 49.179934,
            "lng": -120.854738
        }
    },
    {
        "uri": "/locomotives/84",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0084",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-4",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-26",
        "hqLatLng": {
            "lat": 49.296307,
            "lng": -118.827314
        }
    },
    {
        "uri": "/locomotives/85",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0085",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-4",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-27",
        "hqLatLng": {
            "lat": 49.640363,
            "lng": -126.242998
        }
    },
    {
        "uri": "/locomotives/86",
        "type": "Diesel-electric",
        "model": "3GS21B",
        "serial_no": "0086",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-4",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-28",
        "hqLatLng": {
            "lat": 46.244201,
            "lng": -73.666786
        }
    },
    {
        "uri": "/locomotives/87",
        "type": "Diesel-electric",
        "model": "3GS21B-87",
        "serial_no": "0087",
        "emission_tier": "0+",
```

```
        "fleet": "/fleets/cn-4",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-26",
        "hqLatLng": {
            "lat": 49.296307,
            "lng": -118.827314
        },
        "parent": "/locomotives/86",
        "installedOn": "05/12/2014",
        "dateIso": "2014-12-05T13:15:31Z"

    },
    {
        "uri": "/locomotives/88",
        "type": "Diesel-electric",
        "model": "3GS21B-88",
        "serial_no": "0085",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-4",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-27",
        "hqLatLng": {
            "lat": 49.640363,
            "lng": -126.242998
        },
        "parent": "/locomotives/87",
        "dateIso": "2015-11-05T13:15:30Z"

    },
    {
        "uri": "/locomotives/89",
        "type": "Diesel-electric",
        "model": "3GS21B-89",
        "serial_no": "0086",
        "emission_tier": "0+",
        "fleet": "/fleets/cn-4",
        "manufacturer": "/manufacturers/national-railway-
equipment",
        "engine": "/engines/QSK19-28",
        "hqLatLng": {
            "lat": 46.244201,
            "lng": -73.666786
        },
        "parent": "/locomotives/88",
        "installedOn": "05/12/2015",
       "dateIso": "2015-12-05T13:15:31Z"
    }
]
```

Troubleshooting Asset Service

Troubleshooting Asset Service

Standard HTTP error codes used by Predix Asset service.

Predix Asset services returns standard HTTP error codes. Custom messages are delivered to describe the context for error codes generated by the various Asset features. The following are general issues you may experience when using the Asset service and tips for solving them.

400 Bad Request

Causes

- Invalid query syntax or operators.
- Headers are not defined properly.
- `pageSize` is higher than 1000.
- Request to a URL that contains more than three forward slashes is not allowed.

Solution

Verify that your syntax is correct. See Graph Expression Language (GEL) Syntax on page 27.

Verify that your headers are correct. See Adding Asset Model Data to your Asset Service Instance on page 23.

401 Unauthorized

Cause

Authentication is required and has failed or has not yet been provided.

Solution

- Make sure that your token is not expired.
- Make sure the correct Predix-Zone-ID is in the header.
- Make sure that UAA instance URL was used during service creation.
- Make sure the appropriate scope has been granted to the UAA client.

403 Forbidden

Cause

The user does not have the proper scopes or access for the zone.

Solution

Verify that the token being passed in has the correct scope. This can be verified when the access token is obtained form the `scopes` field of the token response.

405 Method is not supported

Cause

You can `POST`, `PUT`, `PATCH`, and `DELETE` to a collection, but not to a specific resource.

Solution

Verify that you are using the correct request method.

422 Unprocessable

Causes

The request contains semantic errors, such as:

- Invalid JSON.
- URI does not match the URL.
- Invalid characters within the resource ID.
- A request with a `filter` clause that requests a URI is not valid.
  For example, `<asset-app-url>?filter=uri=/asset/asset0001` is invalid, but `<asset-app-url>/asset/asset0001` is valid.

Solution

Verify that your syntax is correct. See Graph Expression Language (GEL) Syntax on page 27.

Asset Service Release Notes

Asset Service

Q3 2016 Release

Data Validation

The JSON schemas that define your domain objects also define your data validation rules. After you upload a new or updated schema, Asset service automatically validates any new data ingested against the new schema. Data ingested previously is not validated automatically against the new or updated schema.

See Validating Asset Data on page 31.

Displaying Time-Based Audit History Data

You can query a particular asset or domain object to see an historical snapshot of the configuration and properties of that asset for a specific point in time. A `GET` request returns the last recorded state of the asset in the Audit History database prior to the timestamp indicated in the query. History is retained for deleted assets. This feature is useful for configuration management solutions that track the changes made to an asset over its life cycle.

See Using Audit History.

Q2 2016 Release

Audit History

The Asset audit history APIs enable you to update and retrieve historical information about assets stored in your Asset repositories. The audit history infrastructure captures and indexes CREATE, UPDATE, and DELETE requests in the audit trail. Asset audit history is disabled by default.

See Using Audit History.

Scripting Engine to Add Business Logic to Asset Queries (Service Deprecated Q4 2017)

The Predix Asset scripting engine is a service that allows users to bind their custom business logic to the Predix Asset REST API. Developers create scripts containing business logic which enable non-technical business users to define business rules and interact with their Asset repositories. Client applications that bind to the Predix Asset service have access to the scripting engine and may send their requests to either the scripting engine or directly to Predix Asset.

Q4 2017

The Scripting Engine was removed from the Predix Asset.