Columns

You can obtain all the details you need for the dataset columns in a specified catalog.

Request

A query for columns must include the following:

- ▲ The SELECT statement must start with one or more column parameters (comma separated).
- ▲ After the parameters you must provide the following data source: `FROM $system.DBSCHEMA_COLUMNS`
- ▲ The name of the published catalog (AtScale project) for which you need column information must be specified in the `Properties > PropertyList > Catalog` element. For details on obtaining a catalog name, see Catalogs.

You can request the following column parameters:

- ▲ CATALOG_NAME
- ▲ COLUMN_NAME
- ▲ CONNECTION_ID
- ▲ DATASET_NAME
- ▲ DATA_TYPE
- ▲ EXPRESSION

For details about the parameter types, see the `xsd:schema` element in the sample response provided below.

Response

The response contains a `<row>` element for each column in the specified catalog. Each of these elements contains the values of the requested parameters.

Sample Request

The example here shows how to make a request with the `curl` tool:

- ▲ The requested data are the name (COLUMN_NAME) and data type (DATA_TYPE) of the column.
- ▲ The request is for the columns in the Sales Insights catalog.
- ▲ The token is obtained in advance.
- ▲ The query is for the default organization.
- ▲ The address of the AtScale system is: http://example.com:10502/xmla/default

The XML part of the request is as follows:

```xml
<?xml version="1.0"?>
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <Execute xmlns="urn:schemas-microsoft-com:xml-analysis">
      <Command>
        <Statement>SELECT COLUMN_NAME, DATA_TYPE FROM $system.DBSCHEMA_COLUMNS</Statement>
      </Command>
      <Properties>
        <PropertyList>
          <Catalog>Sales Insights</Catalog>
        </PropertyList>
      </Properties>
    </Execute>
  </Body>
</Envelope>
```

Here is the full form of the curl request:

```
curl -X POST \
-H "Authorization:Bearer $token" -H 'Content-Type: application/xml' \
-d '<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">\
<Body><Execute xmlns="urn:schemas-microsoft-com:xml-analysis">\
  <Command>\
    <Statement>SELECT COLUMN_NAME, DATA_TYPE FROM $system.DBSCHEMA_COLUMNS</Statement>\
  </Command>\
  <Properties><PropertyList>\
    <Catalog>Sales Insights</Catalog>\
  </PropertyList></Properties>\
</Execute></Body></Envelope>' \
http://example.com:10502/xmla/default
```

Sample Response

The response for the sample request above contains `<row>` elements for each column found. As requested, each of these elements contains only the name and data type of the column:

```xml
<row>
  <COLUMN_NAME>productsubcategorykey</COLUMN_NAME>
  <DATA_TYPE>3</DATA_TYPE>
</row>
<row>
  <COLUMN_NAME>productsubcategoryname</COLUMN_NAME>
  <DATA_TYPE>130</DATA_TYPE>
</row>
```

Here is the full response:

```xml
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <as:queryId xmlns:as="http://xsd.atscale.com/soap_v1" soap:mustUnderstand="0">3d64d843-fd56-4f95-9331-635b2f04e5e4</as:queryId>
  </soap:Header>
  <soap:Body>
    <ExecuteResponse xmlns="urn:schemas-microsoft-com:xml-analysis">
      <return>
        <root xmlns="urn:schemas-microsoft-com:xml-analysis:rowset"
xmlns:msxmla="http://schemas.microsoft.com/analysisservices/2003/xmla"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <xsd:schema xmlns:sql="urn:schemas-microsoft-com:xml-sql" targetNamespace="urn:schemas-microsoft-com:xml-analysis:rowset" elementFormDefault="qualified">
            <xsd:element name="root">
              <xsd:complexType>
                <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                  <xsd:element name="row" type="row"/>
                </xsd:sequence>
              </xsd:complexType>
            </xsd:element>
            <xsd:simpleType name="uuid">
              <xsd:restriction base="xsd:string">
                <xsd:pattern value="[0-9a-zA-Z]{8}-[0-9a-zA-Z]{4}-[0-9a-zA-Z]{4}-[0-9a-zA-Z]{4}-[0-9a-zA-Z]{12}"/>
              </xsd:restriction>
            </xsd:simpleType>
            <xsd:complexType name="xmlDocument">
              <xsd:sequence>
                <xsd:any/>
              </xsd:sequence>
            </xsd:complexType>
            <xsd:complexType name="row">
              <xsd:sequence>
                <xsd:element sql:field="COLUMN_NAME" name="COLUMN_NAME" type="xsd:string" minOccurs="0"/>
                <xsd:element sql:field="DATA_TYPE" name="DATA_TYPE" type="xsd:int" minOccurs="0"/>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:schema>
          <row>
            <COLUMN_NAME>productsubcategorykey</COLUMN_NAME>
            <DATA_TYPE>3</DATA_TYPE>
          </row>
          <row>
            <COLUMN_NAME>productsubcategoryname</COLUMN_NAME>
            <DATA_TYPE>130</DATA_TYPE>
          </row>
        </root>
      </return>
    </ExecuteResponse>
  </soap:Body>
</soap:Envelope>
```

More Information

- ▲ Request reference - how to make requests to the Data Catalog API.
- ▲ Response reference - how to receive and process responses from the Data Catalog API.