

DataStores

DataStores are a great way of importing large datasets into AuditComply. The use/function of these will be discussed in a different article, but consider them similar to Scores/Categories as pre-set response choices.

Assuming you have a valid token, use this token to create/update a datastore at this endpoint.

<https://api.auditcomply.com/datastoresinfo/>

Use the following example JSON to create a new DataStore with the name "Films & Ratings".

```
{
  "name": "Films & Ratings",
  "format": {
    "column_1":{
      "value": "Film Name",
      "type": "TEXT"
    },
    "column_2":{
      "value": "Film Rating",
      "type": "NUMBER"
    }
  },
  "fields":[{
    "value1": "The Matrix",
    "value2": "99"
  },{
    "value1": "The Departed",
```

```
    "value2": "95"
  }]
}
```

The DataStore endpoint requires 3 key elements, a *name*, a *format* and the *fields*. Each are broken out below.

DataStore Name

```
{ "name": "Films & Ratings" }
```

When initially pushing data to AuditComply, chosen an appropriate name, as this will be used to reflect what choices an Admin user will have when selecting DataStores in the Template Builder. When updating a DataStore, use the same name to ensure your new data versions correctly. If you modify the name of your DataStore in your next API push, it will create a brand new element available to Admins on the AuditComply Dashboard.

DataStore Format

```
{ "format": {
  "column_1":{
    "value": "Column Name",
    "type": "Column Format"
  }
}
```

The format will dictate what type of data you are importing. The **value** column describes the name of the column, and the **type** describes the value format.

Ensure the **value** in the column is unique, to give the Admin user a clear understanding of what data is present. The **types** available are:

TEXT, NUMBER, BOOLEAN

DataStore Fields

This is the core of the DataStore JSON - this is the raw data that will be imported into the AuditComply system, allowing for choices to be searched/selected by an end Auditor user.

Depending on how many columns you have defined, this should reflect how many elements are in your JSON object, for example as I have defined two fields in the data above, I should reflect two values in the object:

```
{ "value1": "The Matrix", "value2": "99" }
```

Versioning

To differentiate between different DataStores, as previously mentioned - the **name** is the key. Keeping the name the same will version the DataStore, modifying it will create a new DataStore at Version 1.

To ensure a smooth transition between versions, we require that each sync contain the complete set of data you will need to display. For example, if you have 10 DataStore fields in version 1, but only require 9 fields in version 2 - push all 9 fields when pushing for the second sync.

The format fields are required between each sync, and can not be modified after an initial version has been created. If these need modified, create a new DataStore element (ie, one with a different name), and manually archive the previous DataStore on the AuditComply Dashboard.