Migrating Resources and Transformations Between Environments

Last Modified on 08/30/2021 1:06 pm EDT

If you are developing in multiple environments or accounts, you might need to move your common resources and transformations from one account or environment to another.

Migrate Common Resources

Migrating common resources is a two step process where you first get the common resource definition from one account or environment, and then post it to another.

> ℹ **Note:** Sub-objects appear differently depending on how you created them, as common resources or as My Objects in Connector Mapper. If you created them with Connector Mapper, they are identified by a custom `type` and are stored as separate objects with the same name as the `type` value. For example, a sub-object called `address` will appear in the definition as:
>
> ```
> {
> "type":"address"
> "path":"address"
> }
> ```
>
> Sub-objects created as common resources are shown in dot notation, e.g., `"path": "Address.street"`.

To migrate common resources:

1. In the source account or environment, call `GET /accounts/objects/{objectName}/definitions`, replacing `{objectName}` with the name of the common resource.

   > ℹ **Note:** Use `GET /accounts/objects/definitions` to get the list of all common resources.

   The JSON response looks like this:

```
{
  "fields":[
    {
      "type":"string",
      "path":"birthdate"
    },
    {
      "type":"string",
      "path":"FirstName"
    },
    {
      "type":"string",
      "path":"id"
    },
    {
      "type":"string",
      "path":"LastName"
    }
  ],
  "level":"account"
}
```

2. Optional. If you created your common resource in an earlier version of the software, it might include a sub-object that it is stored separately. Run the call again using the name of the sub-object for `{objectName}`.

3. In your target account or environment, make a `POST /accounts/objects/definitions` API call, replacing `objectName` with the name of the common resource, and the `fields` object with the `fields` object from the previous step.

```
{
  "":{
    "fields":[
      {
        "type":"string",
        "path":"birthdate"
      },
      {
        "type":"string",
        "path":"FirstName"
      },
      {
        "type":"string",
        "path":"id"
      },
      {
        "type":"string",
        "path":"LastName"
      }
    ]
  }
}
```

Migrate Transformations

Migrating transformations is a two step process where you first get the transformations definition from one account or environment, and then post it to another.

To migrate transformations:

1. In the source account or environment, call `GET /accounts/elements/{keyOrId}/transformations/{objectName}`, replacing `keyOrId` with the connector key and `{objectName}` with the name of the common resource.

   The JSON response looks like this:

   ```
   {
     "level":"account",
     "objectName":"myContacts_API",
     "vendorName":"Contact",
     "startDate":"2017-04-24 21:05:05.51129",
     "fields":[
       {
         "type":"string",
         "path":"birthdate",
         "vendorPath":"Birthdate",
         "level":"account"
       },
       {
         "type":"string",
         "path":"FirstName",
         "vendorPath":"FirstName",
         "level":"account"
       },
       {
         "type":"string",
         "path":"id",
         "vendorPath":"Id",
         "level":"account"
       },
       {
         "type":"string",
         "path":"LastName",
         "vendorPath":"LastName",
         "level":"account"
       }
     ],
     "configuration":[
       {
         "type":"passThrough",
         "properties":{
           "fromVendor":false,
           "toVendor":false
         }
       },
       {
         "type":"inherit"
       }
     ],
     "isLegacy":false
   }
   ```

2. In your target account or environment, make a `POST /accounts/elements/{keyOrId}/transformations/{objectName}` API call, replacing `keyOrId` with the connector key and `objectName` with the name of the common resource. Include the JSON payload from the previous step.