

Contentful

Intershop APP

The Intershop app lets editors easily add product(s) or category/categories directly from their Intershop account and reference them in Contentful.

Editors can preview selected product(s) and category/categories via the Intershop app.

This enables them to use the latest products and categories retail sites and experiences.

Features:

- Browse product(s) or category/categories on Intershop in Contentful
- Directly link product(s) or category/categories on Intershop to Contentful entries
- Exclude some products from category selection
- Preview the selected product(s) or category/categories

Requirements:

- An Intershop account

How to configure

API

The API Base is the foundation of the endpoint from which the products and categories calls will be made.

The Image Base is the endpoint from which the images will be retrieved. This field can be left blank in case the returned images are valid links.

API endpoint:

https://pwa-ish-demo.test.intershop.com/INTERSHOP/rest/WFS/inSPIRED-inTRONICS_Business-Site/rest/products

Image endpoint:

https://demo.icm.intershop.de/INTERSHOP/static/WFS/inSPIRED-inTRONICS_Business-Site/rest/inSPIRED/en_US/M/M_201807181_front.jpg

Channels and applications

Channels and/or applications can be added which'll be made available within the connector through a separate menu.

When either of these fields contains a value then the API Base must include an indicator of the position of the channel/application value.

Channels

The channels that can be used within the application

inSPIRED-inTRONICS_Business-Site inSPIRED-inTRONICS_Business-Site2

Applications

The applications that can be used within the application

rest rest2

Based on these options there'll be 4 variants of the API Base available within the application

- **.../INTERSHOP/REST/WFS/inSPIRED-inTRONICS_Business-Site/rest**
- **.../INTERSHOP/REST/WFS/inSPIRED-inTRONICS_Business-Site/rest2**
- **.../INTERSHOP/REST/WFS/inSPIRED-inTRONICS_Business-Site2/rest**
- **.../INTERSHOP/REST/WFS/inSPIRED-inTRONICS_Business-Site2/rest2**

Mappers

There are 3 mappers to configure within the connector:

- **Product Mapper:** Handles the /products response mapping
- **Product Category Mapper:** Handles the defaultCategory mapping within each product
- **Category Mapper:** Handles the /categories response mapping

Through these mappers it's possible to alter which data represents our internal properties

These mappers operate through filters that are able to retrieve a specific value, an object with a particular key-value pair or all values from an array of object with a particular key

Single value retrieval

This filter requires that the key of the value is provided. It can be chained to with an object retrieval and a single value retrieval filter, this chain is indicated using `.`

Structure

`key` or `.keyFromPreviousFilterResult`

E.g.

`manufacturer.value` retrieves the `manufacturer value` and then selects the `value` within the `manufacturer` with the key `value`

```
{
  title: "Freecom 56073", // Internal name property
  manufacturer: {
    value: "Freecom", // Internal brand property
  },
  sku: "11168791",
}
```

The title and manufacturer of the above product can be mapped to the connector name and brand properties using the following filters:

Name (required)

The name of the product

title

Product Mapper: Name = Freecom 56073

Brand (required)

The brandname of the product

manufacturer.value

Object retrieval within an array

This filter must be preceded by a single value retrieval filter that links to an array. Then the key-value pair which we want to match is surrounded in {} so only the object with that key-value pair is retrieved.

Structure:

```
{ key : expectedValue }
```

E.g.

`attributes{name: manufacturer}.value` retrieves the value with key `attributes`, looks for an attribute whose `name` equals `manufacturer` and finally selects the `value` of that object

```
{
  title: "Freecom 56073",
  attributes: [
    {
      name: "manufacturer",
      value: "Freecom" // Internal brand property
    },
    {
      name: "sku",
      value: "11168791" // Internal sku property
    }
  ]
}
```

The manufacturer and sku of the above product can be mapped to the connector brand and sku properties using the following filters:

Brand (required)

The brandname of the product

```
attributes{name: manufacturer}.value
```

Skus (required)

The sku value of the product

```
attributes{name: sku}.value
```

Product Mapper: Sku = 11168791

Values with keys to object

This filter goes through all objects within an array and retrieves only the values whose key equals the one provided

Structure

```
[ key ]
```

E.g.

`categoryPath[id]` retrieves the value with key `categoryPath` and then goes through the objects and returns the values with key `id`

```
{
  title: "Freecom 56073",
  attributes: [
    ...
    defaultCategory: {
      value: { // Internal defaultCategory property
        name: "Storage Servers",
        categoryPath: [ // Contains the ids for internal categoryPath property
          {
            id: "servers"
          },
          {
            id: "servers-data-storage"
          },
          {
            id: "servers-storage-servers"
          }
        ]
      }
    }
  ]
}
```

The categoryPath ids can be mapped to the internal categoryPath property using the following filters:

Default Category (required)

The category metadata under which the product is listed

```
attributes{name: defaultCategory}.value
```

Product Mapper: DefaultCategory =

```
{
  name: "Storage Servers",
  categoryPath: [ // Contains the ids for internal categoryPath property
    {
      id: "servers"
    },
    {
      id: "servers-data-storage"
    },
    {
      id: "servers-storage-servers"
    }
  ]
}
```

CategoryPath (required)

An array containing the ids of all categories in the category path

```
categoryPath[id]
```

Product Category Mapper: CategoryPath =
["servers", "servers-data-storage", "servers-storage-servers"]

Untitled