In this blog we will see how to search a custom product attribute that contains comma separated value by building a custom elastic search tokenizer and analyzer.

The default elastic search tokenizer that magento implements is space separated, i.e if you have a string "Love Magento" we can search by the word "Love" and "Magento" because by default all the strings are tokenized using space.

Consider an requirement where we have multiple barcodes for a given product and these barcodes are stored in the product attribute "barcode" and searching any of these barcode values should return the respective product.

We have two solutions for this

Method	GraphQl Query	Pros	Cons
Using Space Separated The barcodes are stored with Space Separated Ex 1001 1002 1003	`{ products(search: "1002") { total_count items { sku barcode } } }`	No Customization Required	Possibility of returning Duplicate data
Using Comma Separated The barcodes are separated with comma Ex 1001,1002,1003	`{ products(filter: {barcode:{match:"1002"}}) { total_count items { sku barcode } } }`	Customization Required	100% accuracy

Method 1 : Using Space Separated

By default magento supports text search i.e All the words within the text field and tokenized and we can search for any word which will return the correct data

For ex if we have this data

barcode = "1001 1002 1003"

we can search by either 1001 or 1002 or 1003.

The advantage of this approach is that we don't need to do any kind of customization, we just need to update the barcodes with space separated

The problem with this is that in case if we have any value like 1001 in some other product name or any other searchable product attribute those product will also get returned which we really don't want.

Method 2 : Using Comma Separated

To use comma separated search for a field, we first need to define the new tokenizer and analyzer settings for the elastic search index. Lets see in detail below.

Create new tokenizer and analyzer for comma separated

The tokenizer and analyzer setting for the elastic search index in defined in this file

\Magento\Elasticsearch\Model\Adapter\Index\Builder

We need to updated these setting to support our new tokenizer so lets create a plugin for that

Create a new plugin in app/etc/di.xml

```
<type name="Magento\Elasticsearch\Model\Adapter\Index\Builder">
   <plugin name="add_comma_tokenizer"
type="You\YourModule\Plugin\Magento\Elasticsearch\Model\Adapter\Index\
BuilderPlugin"/>
</type>
```

Now create the analyzer and tokenizer in that plugin class

You\YourModule\Plugin\Magento\Elasticsearch\Model\Adapter\Index\BuilderPlugin.php

```
class BuilderPlugin
{
    /**
     * @param Builder $subject
     * @param array $settings
     * @return array
     * @SuppressWarnings(PHPMD.UnusedFormalParameter)
     */
    public function afterBuild(
        Builder $subject,
        array $settings
    ) {
        /**
         * Define the tokenizer
         */
        $settings['analysis']['tokenizer'] =
array_merge($settings['analysis']['tokenizer'] ?? [], [
            "comma_tokenizer" => [
                "type" => "pattern",
                "pattern" => ",",
            ]
        ]);

        /**
         * Set the analyzer
         */
        $settings['analysis']['analyzer'] =
array_merge($settings['analysis']['analyzer'], [
            'comma_separated' => [
                'type' => 'custom',
                'tokenizer' => 'comma_tokenizer',
                'filter' => array_merge(
                    ['lowercase', 'keyword_repeat']
                ),
            ]
        ]);
        return $settings;
```

```
    }
```

Assign the new analyzer to the field barcode

Now that we have created and analyzer its time to assign this analyzer to the field "barcode".

The field specific analyzers are defined in this class

\Magento\Elasticsearch\Model\Adapter\FieldMapper\Product\FieldProvider\StaticField::getField

So lets create a plugin to update the barcode field

app/etc/di.xml

```
<type
name="Magento\Elasticsearch\Model\Adapter\FieldMapper\Product\FieldProvider\StaticField">
        <plugin name="set_tokenizer"
type="You\YourModule\Plugin\Magento\Elasticsearch\Model\Adapter\FieldMapper\Product\FieldProvider\StaticFieldPlugin" />
    </type>
```

You\YourModule\Plugin\Magento\Elasticsearch\Model\Adapter\FieldMapper\Product\FieldProvider\StaticFieldPlugin.php

```
class StaticFieldPlugin
{
    /**
     * @param FieldProviderInterface $subject
     * @param $fieldMapping
     * @return array
     * @SuppressWarnings(PHPMD.UnusedFormalParameter)
     */
    public function afterGetField(
```

```
        FieldProviderInterface $subject,
        $fieldMapping
    ) {
        // Set the custom analyzer for barcode
        if(isset($fieldMapping['aims_barcode'])) {
            $fieldMapping['barcode']['analyzer'] = 'comma_separated';
        }
        return $fieldMapping;
    }
}
```

Do a Catalog Search Full Reindex

A full reindex is required as we are changing the setting of the elastic search index.

```
php bin/magento indexer:reindex catalogsearch_fulltext
```

Validate in Elastic Search

Lets see how the field mapping for our attribute "barcode" is set in elastic search.

```
curl --location --request GET
'http://localhost:9200/magento2_product_1/_mapping/field/barcode'
```

```
{
    "magento2_product_1_v45": {
        "mappings": {
            "aims_barcode": {
                "full_name": "barcode",
                "mapping": {
                    "aims_barcode": {
                        "type": "text",
                        "fields": {
                            "keyword": {
```

```
                                "type": "keyword"
                            }
                        },
                        "copy_to": [
                            "_search"
                        ],
                        "analyzer": "comma_separated"
                    }
                }
            }
        }
    }
}
```

if you notice the analyzer is set to "comma_separated" so any data that you enter in to this field with comma will be tokenized.