The Siren Consolidator Plugin

This plugin provides an Elasticsearch ingest processor which merges documents into a central index from one or more satellite indices. The merging consolidates satellite documents into one unique central document by matching on keys such as driving license number, email, phone number, or any other values known to uniquely identify an entity. Chosen values from different satellite indices can be merged into lists of values in the central document.

For example, a document in the “suspects” index

```
{
  "phone_number": "00-234-56789",
  "fullname": "John Smith",
  "employers": ["Google", "IBM"],
  "gang": "alpha"
}
```

and a document in the “employee” index

```
{
  "phone": "00-234-56789",
  "name_full": "John D Smith",
  "previous_employers": "Apple,
  "employee_id": "E1234"
}
```

could be consolidated, by matching on phone, into one central document in the “people” index:

```
{
  "phone": "00-234-56789",
  "name": ["John Smith", "John D Smith"],
  "employers": ["Apple", "Google", "IBM"],
  "gang": "alpha"
}
```

Install

```
$ bin/elasticsearch-plugin install file:///PATH-TO-SIREN-CONSOLIDATOR/siren-consolidator-7.11.2-1.0.0-proguard-plugin.zip
```

Usage

To merge documents from one satellite index into a central index it is necessary to create an Elasticsearch ingest pipeline containing a single **siren-consolidator** processor, then use that pipeline in an Elasticsearch reindex task with the satellite index as its source index. Each satellite index processed should have its own pipeline and reindex task.

1. Create your central index

Matching of satellite documents to the central index document that they will be merged with is performed by Elasticsearch term queries, so central index fields which are to be used as consolidation keys must be of type keyword; other fields that will be added from the satellite should be given mappings here if the default ES mappings are not desired.

```
PUT people
{
  "mappings": {
    "properties": {
      "phone": {"type": "keyword"}
    }
  }
}
```

2. Create a pipeline containing a consolidator processor

The pipeline below shows how we might merge the *suspects* satellite index into the central *people* index. First we use set processors to modify the field names to match the people field names, then use the **siren-consolidator** processor to perform the merge. The **siren-consolidator** processor has the following parameters:

- **indexField**: the field in the satellite index document which contains the name of the central index it will be merged with. If no such field exists, use a set processor prior to the **siren-consolidator** processor to create one. For example, in the pipeline below, we set the field **centralindex** to “people”. Although this seems more complex than simply specifying the name of the central index directly in a parameter, it allows more flexibility where a conditional set processor could be used to merge documents from one satellite index into different central indices depending on a field value.
- **idFields**: a map defining the fields to be used for matching satellite documents (the map key) to central documents (the map value). Typically, fields would be the same in both satellite and central indices. This would avoid duplicated fields, since fields in the satellite document will be merged into the field with the same name in the matching central document, or created if not found. If the satellite document key field does not match that of the equivalent in the central index, you can use a set processor followed by a remove processor to change the field name. For example, in the pipeline below, we set the field **phone** to the value of the **phone_number** field which is then removed.
- **processorId**: the value given here will appear in the *_siren_consolidated_sources* central index document field. Typically, you would use the name of the satellite index.

A final drop processor should be present to avoid creating a new dest index when reindexing. Since **siren-consolidator** makes its own index requests to the central index, reindexing into dest need not occur - the use of reindex for consolidation is simply to allow the iteration over the source index and applying the processor.

```
PUT _ingest/pipeline/my-consolidator-pipeline
{
  "processors": [
    {
      "set": {
        "field": "centralindex",
        "value": "people"
      }
    },
    {
      "set": {
        "field": "phone",
        "value": "{{{phone_number}}}"
      }
    },
    {
      "remove": {
```

```
        "field": "phone_number"
      }
    },
    {
      "set": {
        "field": "name",
        "value": "{{{fullname}}}"
      }
    },
    {
      "remove": {
         "field": "fullname"
      }
    },
    {
      "siren-consolidator": {
        "indexField": "centralindex",
        "idFields": {
          "phone": "phone"
        },
        "processorId": "suspects"
      }
    },
    {
      "drop": {}
    }
  ]
}
```

3. Reindex

Create an Elasticsearch reindex task with the satellite index as its source index. The siren-consolidator processor performs its own writes to the index specified by the value of indexField so the pipeline should contain a drop processor and dest index will not be written to. The source fields that are made available to the pipeline and which will be merged into the central document can be specified with **_source**.

```
POST _reindex
{
  "source": {
    "index": "suspects",
    "_source": [ "phone_number", "fullname", "employer", "gang"]
  },
  "dest": {
    "index": "this-index-will-not-be-created",
    "pipeline": "my-consolidator-pipeline"
  }
}
```

4. Wait for the final cache flush

The Siren Consolidator adds documents it receives to a cache which is flushed to index documents to the central index once the it reaches *siren.consolidator.max_cache_size*. A separate watcher thread checks for activity every 10s and flushes the cache if no documents have been processed in the last 10s. After reindexing is finished there may be a delay before the processing is reflected in the central index.

5. Central index documents

Central index documents will contain all fields/values from all satellite indices consolidated. Where field names match from multiple satellite indices, unique lists will be present. Additional fields added by the processor: _siren_consolidated_sources (the value given as processorId in the configuration), _siren_consolidated_sourceIds (the _id of the satellite index documents), _siren_consolidated_sourceIds_count (number of satellite documents merged). The example document below is the result of performing two satellite consolidations.

```
{
    "_index" : "people",
    "_type" : "_doc",
    "_id" : "49cf3195-afd0-4d0a-829b-fba75b6f9ade",
    "_score" : 0.0,
    "_source" : {
       "phone": "00-234-56789",
      "name": ["John Smith", "John D Smith"],
```

```
      "employers": ["Apple", "Google", "IBM"],
      "gang": "alpha",
      "index" : "people",
      "_siren_consolidated_sourceIds" : [
        "6ivA2H4BW8CJHXThVAlZ",
        "WivA2H4BW8CJHXThWiH4"
      ],
      "_siren_consolidated_sourceIds_count" : 2,
      "_siren_consolidated_sources" : [
        "suspects",
        "employee"
      ]
    }
}
```

elasticsearch.yml settings

siren.consolidator.max_cache_size: integer with default 10000

siren.consolidator.max_merge_size: integer with default 50; the maximum number of values a central doc will merge into one field list. If this limit is reached a new field/value is added to the central doc: “_siren_consolidation_truncated: true”