Quick Reference

Version 2.7.5

Quick Reference

Chapter 1. Dynamic Domain Methods

1.1. countHits

1.1.1. Purpose

Returns the number of hits for a specified search query.

Only domains that are [root-mapped|guide:classMapping] have this method injected.

1.1.2. Examples

```
def res = MyDomain.countHits("${params.query}")
def res = MyDomain.countHits {
    queryString("${params.query}")
}
```

Description

`search` signature:

```
def countHits(Closure query, Map params)
def countHits(Closure query)
def countHits(Map params, Closure query)
def countHits(String query, Map params)
def countHits(String query)
```

Parameters

- `query` - The search query. You may use one or more terms to look for, or use the Lucene syntax for advanced searching.
- `params` - A list of additional parameters to customize the searching process

Returned value

Returns an `Integer` representing the number of hits for the query.

1.2. index

1.2.1. Purpose

Index specific domain instances to ElasticSearch.

Internally, the plugin uses the Bulk API of ElasticSearch to perform the index requests.

Only domains that are [root-mapped|guide:classMapping] have this method injected.

1.2.2. Examples

```
// Index all instances of the MyDomain domain class
MyDomain.index()

// Index a specific domain instance
MyDomain md = new MyDomain(value:'that')
md.id = 1
md.index()

// Index a collection of domain instances
def ds = [new MyDomain(value:'that'), new MyDomain(name:'that2'), new
MyDomain(value:'thatagain')]
ds.eachWithIndex { val, i->  ds[i].id = i }
MyDomain.index(ds)
```

Description

`index` signatures:

```
// Index a specific domain instance
def index()
// Index ALL instances of a domain class
static index()
// Index a Collection of domain instances
static index(Collection<Domain> domains)
// Same with an ellipsis
static index(Domain... domain)
```

Parameters

- `Collection<Domain>` domains - A `Collection` of domain instances to index.
- `Domain…` domain - Same as `Collection<Domain>`, but with an ellipsis.

1.3. search

1.3.1. Purpose

Search through an index for the specified search query.

The returned result only contains hits for the corresponding domain.

Only domains that are [root-mapped|guide:classMapping] have this method injected.

1.3.2. Examples

```
def highlighter = {
  field 'message', 0, 0
  preTags '<strong>'
  postTags '</strong>'
}

def res = MyDomain.search("${params.query}")
def res = MyDomain.search("${params.query}", [from:0, size:30,
highlighter:highlighter])
def res = MyDomain.search(){
    queryString("${params.query}")
}
def res = MyDomain.search(from:0, size:30) {
    queryString("${params.query}")
}
```

Description

`search` signature:

```
def search(Closure query, Map params)
def search(Map params, Closure query)
def search(String query, Map params)
```

Parameters

- `query` - The search query. You may use one or more terms to look for, or use the Lucene syntax for advanced searching.
- `params` - A list of additional parameters to customize the searching process
  - `from` and `size` - From (hit) and the size (number of hits) to return.
  - `sort` - Sort based on different fields including ElasticSearch's internal ones (like `_score`)
  - `highlighter` - A `Closure` containing the highlighting settings.

Returned value

Return an object of `ElasticSearchResult` containing:

- a `total` entry, representing the total number of hits found
- a `searchResults` entry, containing the hits
- a `scores` entry, containing the hits' scores.

- a `highlight` entry if the `highlighter` parameter was set.

1.4. unindex

1.4.1. Purpose

Remove specific domain instances from the ElasticSearch indices.

Internally, the plugin uses the Bulk API of ElasticSearch to perform the delete requests.

Only domains that are [root-mapped|guide:classMapping] have this method injected.

1.4.2. Examples

```
// Unindex all instances of the MyDomain domain class
MyDomain.unindex()

// Unindex a specific domain instance
MyDomain md = MyDomain.get(1)
md.unindex()

// Index a collection of domain instances
def ds = MyDomain.findAllByValue('that')
MyDomain.unindex(ds)
```

Description

`unindex` signatures:

```
// Unindex a specific domain instance
def unindex()
// Unindex ALL instances of a domain class
static unindex()
// Unindex a Collection of domain instances
static unindex(Collection<Domain> domains)
// Same with an ellipsis
static unindex(Domain... domain)
```

Parameters

- `Collection<Domain>` domains - A `Collection` of domain instances to unindex.
- `Domain…` domain - Same as `Collection<Domain>`, but with an ellipsis.

Chapter 2. ElasticSearchAdminService

2.1. deleteIndex

2.1.1. Purpose

Delete one or more existing indices, whether they correspond to a searchable class or not.

2.1.2. Examples

```
elasticSearchAdminService.deleteIndex()
```

Description

`deleteIndex` signatures:

```
void deleteIndex()   // Delete all indices
void deleteIndex(String... indices)
void deleteIndex(Class... searchableClasses)
```

Parameters

Without parameters, **deletes all indices**.

- `indices` - A `Collection` of indices to delete.
- `searchableClasses` - A `Collection` of searchable classes which indices are to be deleted.

2.2. refresh

2.2.1. Purpose

Explicitly refresh one or more index, making all operations performed since the last refresh available for search.

2.2.2. Examples

```
elasticSearchService.index(domain)
// Some code...
// ...
elasticSearchService.index(domain2)
// Some code...
// ...
elasticSearchService.index(domain3)

// Some code...
// ...
elasticSearchAdminService.refresh() // Ensure that the 3 previous index requests
                                    // have been made searchable by ES
```

Description

`refresh` signatures:

```
public void refresh()   // Refresh all indices
public void refresh(String... indices)
public void refresh(Class... searchableClasses)
```

Parameters

Without parameters, refreshes all indices.

- `indices` - A `Collection` of indices to refresh.
- `searchableClasses` - A `Collection` of searchable classes which indices are to be refreshed.

Chapter 3. ElasticSearchService

3.1. countHits

3.1.1. Purpose

Returns the number of hits for a specified search query.

3.1.2. Examples

```
def res = elasticSearchService.countHits("${params.query}")
def res = elasticSearchService.countHits("${params.query}", [indices:"tweet"])
def res = elasticSearchService.countHits {
    queryString("${params.query}")
}
def res = elasticSearchService.countHits(indices:"tweet") {
    queryString("${params.query}")
}
```

Description

`countHits` signatures:

```
def countHits(Closure query, Map params)
def countHits(Closure query)
def countHits(Map params, Closure query)
def countHits(String query, Map params)
def countHits(String query)
```

Parameters

- `query` - The search query.
  - As a `String`, the query is parsed by the Lucene query parser for advanced searching.
  - Can also be a `Closure`, using the Groovy Query DSL of the ElasticSearch groovy client.
- `params` - A list of additional parameters to customize the searching process
  - `indices` - Limit the search only to the specified indices (may be a `String`, or `Collection` of `String`)
  - `types` - Limit the search only to the specified types (domains) (may be a `String`, or `Collection` of `String`).

Returned value

Returns an `Integer` representing the number of hits for the query.

3.2. index

3.2.1. Purpose

Index domain instances to ElasticSearch. Internally, the plugin uses the Bulk API of ElasticSearch to perform the index requests.

3.2.2. Examples

```
// Index all searchable instances
elasticSearchService.index()

// Index a specific domain instance
MyDomain md = new MyDomain(value:'that')
md.save()
elasticSearchService.index(md)

// Index a collection of domain instances
def ds = [new MyDomain(value:'that'), new MyOtherDomain(name:'this'), new
MyDomain(value:'thatagain')]
ds*.save()
elasticSearchService.index(ds)

// Index all instances of the specified domain class
elasticSearchService.index(MyDomain)
elasticSearchService.index(class:MyDomain)
elasticSearchService.index(MyDomain, MyOtherDomain)
elasticSearchService.index([MyDomain, MyOtherDomain])
```

Description

`index` signatures:

```
def index()
def index(Map params)
def index(Collection<Domain> domains)
def index(Domain... domain)
def index(Collection<Class> classes)
def index(Class... class)
```

Parameters

- `Map` params - A map containing the `Class` (or `Collection` of `Class`) of the domain. Will index ALL persisted instances of the specified class(es).
- `Collection<Domain>` domains - A `Collection` of domain instances to index.
- `Domain…` domain - Same as `Collection<Domain>`, but with an ellipsis.

- `Collection<Class>` classes - A `Collection` of `Class` of the domain to index. Will index ALL persisted instances of the specified classes.
- `Class…` classes - Same as `Collection<Class>`, but with an ellipsis.

3.3. search

3.3.1. Purpose

Search through indices for the specified search query. The returned result may contain different type of domain.

3.3.2. Examples

```
def highlighter = {
  field 'message', 0, 0
  preTags '<strong>'
  postTags '</strong>'
}

def res = elasticSearchService.search("${params.query}")
def res = elasticSearchService.search("${params.query}", [from:0, size:30,
highlighter:highlighter])
def res = elasticSearchService.search(){
    queryString("${params.query}")
}
def res = elasticSearchService.search(from:0, size:30, indices:"tweet") {
    queryString("${params.query}")
}

def sortBuilder = SortBuilders.fieldSort("name").order(SortOrder.ASC)

def res = elasticSearchService.search([sort: sortBuilder], { match("name": "foo") })

def res = elasticSearchService.search(null as Closure, {
            range {
                "price"(gte: 1.99, lte: 2.3)
            }
        })

def res = elasticSearchService.search({
    wildcard("name": "*st")
}, [:])

def hasParentQuery = QueryBuilders.hasParentQuery("store",
QueryBuilders.matchQuery("owner", "Horst"))
def result = elasticSearchService.search(hasParentQuery)
```

Description

`search` signatures:

```
def search(Map params, Closure query, filter = null)
def search(Map params, QueryBuilder query, filter = null)
def search(String query, Map params = [:])
def search(Closure query, Map params)
def search(Closure query, filter = null, Map params = [:])
def search(QueryBuilder query, filter = null, Map params = [:])
```

Parameters

- `query` - The search query.
  - As a `String`, the query is parsed by the Lucene query parser for advanced searching.
  - Can also be a `Closure`, using the Groovy Query DSL of the ElasticSearch groovy client.
  - Can also be a `QueryBuilder`.
- `filter` - The search filter
  - A `Closure`, using the Groovy Query DSL of the ElasticSearch groovy client.
  - can also be a `FilterBuilder`.
- `params` - A list of additional parameters to customize the searching process
  - `from` and `size` - From (hit) and the size (number of hits) to return.
  - `sort` - Sort based on different fields including ElasticSearch's internal ones (like `_score`)
    - As a `String`, sort by the field with this name (e.g. `sort: 'name'` means 'sort by `name')
    - Can also be a `SortBuilder`
  - `order` - Sort order ("ASC" or "DESC", case insensitive). Default: "ASC"
  - `indices` - Limit the search only to the specified indices (may be a `String`, or `Collection` of `String`)
  - `types` - Limit the search only to the specified types (domains) (may be a `String`, or `Collection` of `String`).
  - `highlight` - A `Closure` containing the highlighting settings.

Returned value

Return a `Map` containing:

- a `total` entry, representing the total number of hits found.
- a `searchResults` entry, containing the hits.
- a `scores` entry, containing the hits' scores.
- a `highlight` entry if the `highlighter` parameter was set.
- a `sort` entry if the `sort` parameter was set. Contains all `sortValue`s of the search. Maps the id of

a search hit to the sort values

- Example: `[id: [12.34]]`

3.4. ElasticSearch Builders

ElasticSearch provides many builders (e.g. QueryBuilders) and builder factories (ending with **Builders**, e.g. QueryBuilders). The factories provide methods for creating the concrete builders - e.g. QueryBuilders.matchQuery() produces a MatchQueryBuilder, which can be used to create a match query.

Here is a list of builder factories:

- org.elasticsearch.index.mapper.MapperBuilders
- org.elasticsearch.index.query.QueryBuilders
- org.elasticsearch.index.query.FilterBuilders
- org.elasticsearch.search.sort.SortBuilders
- org.elasticsearch.search.facet.FacetBuilders
- org.elasticsearch.cluster.routing.RoutingBuilders

3.5. JSON vs. closure syntax comparison for queries and filters

Since there seems to be no obvious rule for the conversion of queries and filters between the JSON syntax and the closure syntax, I just provide some examples.

3.5.1. Range filter

JSON

```
{
    "range" : {
        "price" : {
            "gte" : 1.99,
            "lte" : 2.3
        }
    }
}
```

Closure

```
{
    range {
        "price"(gte: 1.99, lte: 2.3)
    }
}
```

3.5.2. Wildcard Query

JSON

```
{
    "wildcard" : {
        "name" : "*st"
    }
}
```

Closure

```
{
    wildcard("name": "*st")
}
```

3.5.3. Bool Query

JSON

```
{
    "bool" : {
        "must" : {
            "term" : { "firstname" : "kimchy" }
        }
    }
}
```

Closure

```
{
  bool {
      must {
          term(firstname: "kimchy")
      }
  }
}
```

3.5.4. Geo_reference filter

JSON

```
{
    "filtered" : {
        "filter" : {
            "geo_distance" : {
                "distance" : "200km",
                "pin.location" : {
                    "lat" : 40,
                    "lon" : -70
                }
            }
        }
    }
}
```

Closure

```
{
    geo_distance(
        'distance': '200km',
        'pin.location': [lat: 40, lon: -70]
    )
}
```

3.6. unindex

3.6.1. Purpose

Remove domain instances from the ElasticSearch indices. Internally, the plugin uses the Bulk API of ElasticSearch to perform the delete requests.

3.6.2. Examples

```
// Unindex all searchable instances
elasticSearchService.unindex()

// Unindex a specific domain instance
MyDomain md = new MyDomain(value:'that')
md.save()
elasticSearchService.unindex(md)

// Unindex a collection of domain instances
def ds = [new MyDomain(value:'that'), new MyOtherDomain(name:'this'), new
MyDomain(value:'thatagain')]
ds*.save()
elasticSearchService.unindex(ds)

// Unindex all instances of the specified domain class
elasticSearchService.unindex(MyDomain)
elasticSearchService.unindex(class:MyDomain)
elasticSearchService.unindex(MyDomain, MyOtherDomain)
elasticSearchService.unindex([MyDomain, MyOtherDomain])
```

Description

`unindex` signatures:

```
def unindex()
def unindex(Map params)
def unindex(Collection<Domain> domains)
def unindex(Domain... domain)
def unindex(Collection<Class> classes)
def unindex(Class... class)
```

Parameters

- `Map` params - A map containing the `Class` (or `Collection` of `Class`) of the domain. Will unindex ALL persisted instances of the specified class(es).
- `Collection<Domain>` domains - A `Collection` of domain instances to unindex.
- `Domain…` domain - Same as `Collection<Domain>`, but with an ellipsis.
- `Collection<Class>` classes - A `Collection` of `Class` of the domain to unindex. Will unindex ALL persisted instances of the specified classes.
- `Class…` classes - Same as `Collection<Class>`, but with an ellipsis.