

Elastic Search(EL) 개발환경 교체 [php -> Spring]

작성일: 20.05.11

작성자: 도원진

1. Spring 환경설정

[pom.xml 수정]

java 버전 및 컴파일 버전 변경

```
<properties>
  <java-version>1.8</java-version>
  <org.springframework-version>5.0.7.RELEASE</org.springframework-version>
</properties>
```

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>2.5.1</version>
  <configuration>
    <source>1.8</source>
    <target>1.8</target>
    <compilerArgument>-Xlint:all</compilerArgument>
    <showWarnings>true</showWarnings>
    <showDeprecation>true</showDeprecation>
  </configuration>
</plugin>
```

maven 설치

- EL 관련 라이브러리
- API response 를 위한 JSON 변환 라이브러리

```
<dependency>
  <groupId>org.elasticsearch.client</groupId>
  <artifactId>elasticsearch-rest-high-level-client</artifactId>
  <version>6.6.1</version>
</dependency>

<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <version>1.18.0</version>
  <scope>provided</scope>
</dependency>
```

<!-- <https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind> -->

```

<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-databind</artifactId>
  <version>2.9.6</version>
</dependency>

<!--
https://mvnrepository.com/artifact/com.fasterxml.jackson.dataformat/jackson-
dataformat-xml -->
<dependency>
  <groupId>com.fasterxml.jackson.dataformat</groupId>
  <artifactId>jackson-dataformat-xml</artifactId>
  <version>2.9.6</version>
</dependency>

<!-- https://mvnrepository.com/artifact/com.google.code.gson/gson -->
<dependency>
  <groupId>com.google.code.gson</groupId>
  <artifactId>gson</artifactId>
  <version>2.8.2</version>
</dependency>

```

2. 기본 DSL Query (php , spring 비교)

2.1 must DSL

```

GET /pageview-2020-03-*/_search
{
  "from":0,
  "size":2,
  "query":{
    "bool":{
      "must":[
        {
          "terms":{
            "pfno":[
              "12845"
            ],
            "boost":1.0
          }
        }
      ],
      "adjust_pure_negative":true,
      "boost":1.0
    }
  },
  "_source":{
    "includes":[
      "dpuid",
      "vt",
      "pfno"
    ],
    "excludes":[]
  }
}

```

```
}
}
}
```

Kibana에서 확인한 Response

http://172.16.100.70:5601/app/kibana#/dev_tools/console

```
{
  "took" : 3,
  "timed_out" : false,
  "_shards" : {
    "total" : 15,
    "successful" : 15,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 84,
      "relation" : "eq"
    },
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "pageview-2020-03-19",
        "_type" : "_doc",
        "_id" : "sv6j8nAB212ZgcqwfYvx",
        "_score" : 1.0,
        "_source" : {
          "dpuid" : "rRExHwYxyoXobzeBHVf3GlE0jfmnfuXGmBe37cAG",
          "pfno" : "12845",
          "vt" : "2020-03-19T11:52:44.000Z"
        }
      },
      {
        "_index" : "pageview-2020-03-19",
        "_type" : "_doc",
        "_id" : "mJej8nABxX_lJrFHjRej",
        "_score" : 1.0,
        "_source" : {
          "dpuid" : "rRExHwYxyoXobzeBHVf3GlE0jfmnfuXGmBe37cAG",
          "pfno" : "12845",
          "vt" : "2020-03-19T11:52:47.000Z"
        }
      }
    ]
  }
}
```

```

<?php
require_once __DIR__ . '/../../vendor/autoload.php';
require_once __DIR__ . '/../inc/config.php";          //EL 연동 설정정보가 있는
php파일
$params = [
    'index' => 'pageview-*',
    'body' => [
        'from' =>0,
        'size' =>2,
        "_source" => ["dpuid", "vt", "pfno", "du"],
        'query' => [
            'bool' => [
                'must' => [
                    'terms' => [
                        'pfno' => ["12845", "23884"]
                    ]
                ]
            ]
        ]
    ]
];
$searchResponse = $es->search($params);          //DSL query를 통해 EL에 요청후
결과 받기
echo('<pre>'. json_encode($params). '</pre>'); //JSON형식으로 변환
echo('<pre>'. print_r($searchResponse, true). '</pre>');

```

[SearchResponse 출력결과]

위 소스 두 번째 echo에 대한 결과

```

//EL에 요청후 결과
Array
(
    [took] => 5
    [timed_out] =>
    [_shards] => Array
        (
            .....(생략)
        )
    [hits] => Array          //첫 번째 hits
        (
            [total] => Array
                (
                    ....(생략)
                )
            [max_score] => 1
            [hits] => Array          //두 번째 hits
                (
                    [0] => Array
                        (
                            [_index] => pageview-2020-03-19
                            [_type] => _doc
                            [_id] => Sv6j8nAB212ZgcqwFYvx
                            [_score] => 1
                            [_source] => Array

```

```

        (
            [dpuid] =>
rRExHwYxyoXobzeBHVf3GlE0jfmnfuXGmBe37cAG
            [du] =>
http://demo.logger.co.kr/item/list.php?
            [pfno] => 12845
            [vt] => 2020-03-19T11:52:44.000Z
        )
    )

    [1] => Array
    (
        .....(위와 동일)
    )

)

)
)

```

2.1.2 java 소스

[EL에서 제공하는 SearchSourceBuilder를 통해 DSL Query를 생성]

```

//ElasticService.java
SearchSourceBuilder sourceBuilder = new SearchSourceBuilder();
sourceBuilder
    .from(0)
    .size(2)
    .fetchSource(includeFields, null)
    .query(QueryBuilders.boolQuery()
        .must(QueryBuilders.termsQuery("pfno", "12845"))
    );
System.out.println("sourceBuilder: \n" + sourceBuilder.toString());
searchRequest.source(sourceBuilder);

// query and fetch
try {
    searchResponse = this.client.search(searchRequest,
RequestOptions.DEFAULT);
} catch (Exception e) {
    e.printStackTrace();
}
System.out.println("searchResponse: \n" + searchResponse);

```

[SearchResponse 출력결과]

```

{
  "took":8,
  "timed_out":false,
  "_shards":{

```

```

        "total":51,
        "successful":51,
        "skipped":0,
        "failed":0
    },
    "hits":{
        "total":0,
        "max_score":1.0,
        "hits":[
            {
                "_index":"pageview-2020-03-19",
                "_type":"_doc",
                "_id":"Sv6j8nAB212ZgcqwfYvx",
                "_score":1.0,
                "_source":{
                    "dpuid":"rRExHwYxyoXobzeBHVf3G1E0jfmnfuxGmBe37cAG",
                    "pfno":"12845",
                    "vt":"2020-03-19T11:52:44.000Z"
                }
            },
            {
                ....(위와 동일)
            }
        ]
    }
}

```

2.2 filter DSL

```

GET /pageview-2020-03-*/_search
{
    "from":0,
    "size":2,
    "query":{
        "bool":{
            "must":[
                {
                    "terms":{
                        "pfno":[
                            "12845"
                        ],
                        "boost":1.0
                    }
                }
            ],
            "filter":[
                {
                    "range":{
                        "vt":{
                            "from":"2020-03-01T00:00:00+0900",
                            "to":"2020-04-29T23:59:59+0900",
                            "include_lower":true,
                            "include_upper":true,

```

```

        "boost":1.0
      }
    }
  },
  ],
  "adjust_pure_negative":true,
  "boost":1.0
}
},
"_source":{
  "includes":[
    "dpuid",
    "vt",
    "pfno"
  ],
  "excludes":[]
}
},
"aggregations":{
  "by_date":{
    "date_histogram":{
      "field":"vt",
      "format":"yyyy-MM-dd",
      "time_zone":"Asia/Seoul",
      "interval":"1d",
      "offset":0,
      "order":{
        "_key":"asc"
      },
      "keyed":false,
      "min_doc_count":0
    }
  }
}
}
}

```

Kibana에서 확인한 Response

```

{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 15,
    "successful" : 15,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 84,
      "relation" : "eq"
    },
    "max_score" : 1.0,
    "hits" : [
      {

```

```

    "_index" : "pageview-2020-03-19",
    "_type" : "_doc",
    "_id" : "Sv6j8nAB212ZgcqwfYvx",
    "_score" : 1.0,
    "_source" : {
      "dpuid" : "rRExHwYxyoXobzeBHVf3GlE0jfmnfuXGmBe37cAG",
      "pfno" : "12845",
      "vt" : "2020-03-19T11:52:44.000Z"
    }
  },
  {
    "_index" : "pageview-2020-03-19",
    "_type" : "_doc",
    "_id" : "mJej8nABxX_lJrFHjRej",
    "_score" : 1.0,
    "_source" : {
      "dpuid" : "rRExHwYxyoXobzeBHVf3GlE0jfmnfuXGmBe37cAG",
      "pfno" : "12845",
      "vt" : "2020-03-19T11:52:47.000Z"
    }
  }
]
}
}

```

2.2.1 php소스

```

<?php

require_once __DIR__ . '/../../vendor/autoload.php';
require_once __DIR__ . '/../inc/config.php";
$params = [
  'index' => 'pageview-*',
  'body' => [
    'from' => 0,
    'size' => 2,
    "_source" => ["dpuid", "vt", "pfno", "du"],
    'query' => [
      'bool' => [
        'must' => [
          'terms' => [
            'pfno' => ["12845", "23884"]
          ]
        ],
        'filter' => [
          'range' => [
            'vt' => [
              'gte' => "2020-03-01T00:00:00+0900",
              'lte' => "2020-04-29T23:59:59+0900"
            ]
          ]
        ]
      ]
    ]
  ]
]

```



```
]
];
```

[SearchResponse 출력결과]

```
Array
(
    .....(생략)

    [hits] => Array
        (
            [total] => Array
                (
                    [value] => 377
                    [relation] => eq
                )

            [max_score] => 1
            [hits] => Array
                (
                    [0] => Array
                        (
                            [_index] => pageview-2020-03-19
                            [_type] => _doc
                            [_id] => Sv6j8nAB212ZgcqwfYvx
                            [_score] => 1
                            [_source] => Array
                                (
                                    [dpuid] =>
rRExHWYxyoXobzeBHVf3GlE0jfmnfuXGmBe37cAG
                                    [du] =>
http://demo.logger.co.kr/item/list.php?
                                    [pfno] => 12845
                                    [vt] => 2020-03-19T11:52:44.000Z
                                )
                        )
                    [1] => Array
                        (
                            ....(위와 동일)
                        )
                )
            )
        )
    )
```

2.2.2 java소스

ElasticService.java

```
//ElasticService.java
sourceBuilder
    .from(0)
    .size(10)
    .fetchSource(includeFields, null)
    .query(QueryBuilders.boolQuery()
        .must(QueryBuilders.termsQuery("pfno", "12845"))
        .filter(QueryBuilders
            //filter추가
            .rangeQuery("vt")
            .gte("2020-03-01T00:00:00+0900")
            .lte("2020-04-29T23:59:59+0900"))
    );
```

[SearchResponse 출력결과]

```
{
  "took":3,
  "timed_out":false,
  "_shards":{
    "total":51,
    "successful":51,
    "skipped":0,
    "failed":0
  },
  "hits":{
    "total":0,
    "max_score":1.0,
    "hits":[
      {
        "_index":"pageview-2020-03-19",
        "_type":"_doc",
        "_id":"sv6j8nAB212ZgcqwfYvx",
        "_score":1.0,
        "_source":{
          "dpuid":"rRExHwYxyoXobzeBHVf3G1E0jfmnfuXGmBe37cAG",
          "pfno":"12845",
          "vt":"2020-03-19T11:52:44.000Z"
        }
      },
      {
        ....(위와 동일)
      }
    ]
  }
}
```

2.3 aggregation DSL

filter DSL에서 aggregations 추가

```
GET /pageview-2020-03-*/_search
{
  "from":0,
```

```

"size":2,
  "query":{
    "bool":{
      "must":{
        "terms":{
          "pfno":[
            "12845",
            "23884"
          ]
        }
      },
      "filter":{
        "range":{
          "vt":{
            "gte":"2020-03-01T00:00:00+0900",
            "lte":"2020-04-30T23:59:59+0900"
          }
        }
      }
    }
  },
  "aggs":{
    "by_date":{
      "date_histogram":{
        "field":"vt",
        "fixed_interval":"1d",
        "format":"yyyy-MM-dd",
        "time_zone":"Asia/Seoul",
        "min_doc_count":0
      },
      "aggs":{
        "pageviews":{
          "value_count":{
            "field":"_id"
          }
        },
        "visits":{
          "cardinality":{
            "field":"sid"
          }
        },
        "visitors":{
          "cardinality":{
            "field":"uid"
          }
        },
        "ips":{
          "cardinality":{
            "field":"locations.ip"
          }
        }
      }
    }
  }
}

```

Kibana에서 확인한 Response

```
{
  "took" : 10,
  "timed_out" : false,
  "_shards" : {
    "total" : 15,
    "successful" : 15,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 92,
      "relation" : "eq"
    },
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "pageview-2020-03-19",
        "_type" : "_doc",
        "_id" : "Sv6j8nAB212ZgcqwfYvx",
        "_score" : 1.0,
        "_source" : {
          "du_dq" : [
            {
              "value" : "A",
              "param" : "bigCat"
            },
            {
              "value" : "G",
              "param" : "midCat"
            },
            {
              "value" : "K",
              "param" : "smallCat"
            }
          ]
        },
        "dpuid" : "rREXHwYxyoXobzeBHVf3G1E0jfmnfuXGmBe37cAG",
        "trex" : 5,
        "vt" : "2020-03-19T11:52:44.000Z",
        "pfno" : "12845",
        "du" : "http://demo.logger.co.kr/item/list.php?
bigCat=A&midCat=G&smallCat=K",
        "du_dp" : {
          "d1" : "item",
          "d0" : "http://demo.logger.co.kr",
          "full" : "http://demo.logger.co.kr/item/list.php",
          "d2" : "list.php"
        },
        "dr_dp" : {
          "d1" : "item",
          "d0" : "http://demo.logger.co.kr",
          "full" : "http://demo.logger.co.kr/item/view.php",
          "d2" : "view.php"
        }
      },

```

```

"cp" : [
  "상품 A 카테고리 > 제조사_G > 브랜드_K"
],
"pi" : "PLV",
"dr_dq" : [
  {
    "value" : "AFK001",
    "param" : "itemCode"
  }
],
"sid" : "01bcd15f0f96e5e1c0d865e2f1119ab5",
"uid" : "8a207da187810e03118b833c83fb45c6",
"dr" : "http://demo.logger.co.kr/item/view.php?itemCode=AFK001",
"@timestamp" : "2020-03-19T11:52:44.722Z",
"locations" : {
  "ip" : "122.99.192.187"
},
"@version" : "1",
"environments" : {
  "os" : "Macintosh",
  "ce" : true,
  "js" : true,
  "cd" : 24,
  "ss" : [
    1920,
    1080
  ],
  "tz" : -540,
  "user_agent" : "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.122 Safari/537.36",
  "lg" : "ko",
  "an" : "Chrome 80.",
  "je" : false
}
},
{
  "_index" : "pageview-2020-03-19",
  "_type" : "_doc",
  "_id" : "mJej8nABxx_lJrFHjRej",
  "_score" : 1.0,
  "_source" : {
    "du_dq" : [
      {
        "value" : "A",
        "param" : "bigCat"
      },
      {
        "value" : "F",
        "param" : "midCat"
      },
      {
        "value" : "L",
        "param" : "smallCat"
      }
    ],
    "dpuid" : "rRExHwYxyoXobzeBHVf3G1E0jfmnfuXGmBe37cAG",
    "trex" : 7,

```

```

"vt" : "2020-03-19T11:52:47.000Z",
"pfno" : "12845",
"du" : "http://demo.logger.co.kr/item/list.php?
bigCat=A&midCat=F&smallCat=L",
"du_dp" : {
  "d1" : "item",
  "d0" : "http://demo.logger.co.kr",
  "full" : "http://demo.logger.co.kr/item/list.php",
  "d2" : "list.php"
},
"dr_dp" : {
  "d1" : "item",
  "d0" : "http://demo.logger.co.kr",
  "full" : "http://demo.logger.co.kr/item/view.php",
  "d2" : "view.php"
},
"cp" : [
  "상품 A 카테고리 > 제조사_F > 브랜드_L"
],
"pi" : "PLV",
"dr_dq" : [
  {
    "value" : "AGK000",
    "param" : "itemCode"
  }
],
"sid" : "01bcd15f0f96e5e1c0d865e2f1119ab5",
"uid" : "8a207da187810e03118b833c83fb45c6",
"dr" : "http://demo.logger.co.kr/item/view.php?itemCode=AGK000",
"@timestamp" : "2020-03-19T11:52:48.723Z",
"locations" : {
  "ip" : "122.99.192.187"
},
"@version" : "1",
"environments" : {
  "os" : "Macintosh",
  "ce" : true,
  "js" : true,
  "cd" : 24,
  "ss" : [
    1920,
    1080
  ],
  "tz" : -540,
  "user_agent" : "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_3)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.122 Safari/537.36",
  "lg" : "ko",
  "an" : "Chrome 80.",
  "je" : false
}
}
}
]
},
"aggregations" : {
  "by_date" : {
    "buckets" : [
      {

```

```

        "key_as_string" : "2020-03-19",
        "key" : 1584543600000,
        "doc_count" : 12,
        "visitors" : {
            "value" : 1
        },
        "visits" : {
            "value" : 1
        },
        "pageviews" : {
            "value" : 12
        },
        "ips" : {
            "value" : 1
        }
    },
    {
        .....
    },
    {
        "key_as_string" : "2020-03-21",
        "key" : 1584716400000,
        "doc_count" : 0,
        "visitors" : {
            "value" : 0
        },
        "visits" : {
            "value" : 0
        },
        "pageviews" : {
            "value" : 0
        },
        "ips" : {
            "value" : 0
        }
    }
]
}
}
}
}

```

php 코드

```

<?php

require_once __DIR__ . '/../../vendor/autoload.php';
require_once __DIR__ . '/../inc/config.php';

$params = [
    'index' => 'pageview-*',
    'body' => [
        'from' => 0,
        'size' => 2,
        'query' => [

```

```

        'bool' => [
            'must' => [
                'terms' => [
                    'pfno' => ["12845", "23884"]
                ]
            ],
            'filter' => [
                'range' => [
                    'vt' => [
                        'gte' => "2020-04-25T00:00:00+0900",
                        'lte' => "2020-04-29T23:59:59+0900"
                    ]
                ]
            ]
        ],
        'aggs' => [
            'by_date' => [
                'date_histogram' => [
                    'field' => 'vt',
                    'fixed_interval' => "1d",
                    'format' => "yyyy-MM-dd",
                    'time_zone' => 'Asia/Seoul',
                    'min_doc_count' => 0
                ],
                'aggs' => [
                    'pageviews' => [ 'value_count' => [ 'field' => '_id' ] ],
                    'visits' => [ 'cardinality' => [ 'field' => 'sid' ] ],
                    'visitors' => [ 'cardinality' => [ 'field' => 'uid' ] ],
                    'ips' => [ 'cardinality' => [ 'field' => 'locations.ip' ] ]
                ]
            ]
        ]
    ],
    ];

    echo('<pre>'. json_encode($params). '</pre>');
    $results = $es->search($params);
    echo('<pre>'. print_r($results, true). '</pre>');

```

Search결과

```

Array
(
    [took] => 11
    [timed_out] =>
    [_shards] => Array
        (
            [total] => 48
            [successful] => 48
            [skipped] => 0
            [failed] => 0
        )
)

```



```

[hits] => Array
(
    .....
    .....
)

[aggregations] => Array
(
    [by_date] => Array
    (
        [buckets] => Array
        (
            [0] => Array
            (
                [key_as_string] => 2020-04-27
                [key] => 1587913200000
                [doc_count] => 27
                [visitors] => Array
                (
                    [value] => 3
                )
                [visits] => Array
                (
                    [value] => 3
                )
                [pageviews] => Array
                (
                    [value] => 27
                )
                [ips] => Array
                (
                    [value] => 2
                )
            )
            [1] => Array
            ....(위와 동일)
        )
    )
)
)
)
)
)

```

2.3.2 java소스

ElasticService.java

```

//ElasticService.java
sourceBuilder
    .from(0)
    .size(2)
    .fetchSource(includeFields, null)
    .query(QueryBuilders.boolQuery()
        .must(QueryBuilders.termsQuery("pfno", "12845"))
        .filter(QueryBuilders
            .rangeQuery("vt")
            .gte("2020-03-01T00:00:00+0900")

```

```

        .lte("2020-04-29T23:59:59+0900"))
    ).aggregation(AggregationBuilders
        .dateHistogram("by_date")
        .field("vt")
        .dateHistogramInterval(DateHistogramInterval.DAY)
        .format("yyyy-MM-dd")
        .timeZone(DateTimeZone.forID("Asia/Seoul"))
        .minDocCount(0)
    );

```

SearchResponse

```

{
  "took":10,
  "timed_out":false,
  "_shards":{
    "total":51,
    "successful":51,
    "skipped":0,
    "failed":0
  },
  "hits":{
    ....(생략)
  },
  "aggregations":{
    "date_histogram#by_date":{
      "buckets":[
        {
          "key_as_string":"2020-03-19",
          "key":1584543600000,
          "doc_count":12
        },
        {
          "key_as_string":"2020-03-20",
          "key":1584630000000,
          "doc_count":0
        },
        ....(위와 동일)
        {
          "key_as_string":"2020-04-28",
          "key":1587999600000,
          "doc_count":2
        }
      ]
    }
  }
}

```

3. 필요한 값 추출, 데이터셋에 담기

[{key : value} 형식인 SearchResponse객체에서 value를 얻기 위한 과정]

연관배열인 php와 달리 java에서는 별도의 메소드를 통해 value를 얻어야 한다.

Search response

```
{
  "took":7,
  "timed_out":false,
  "_shards":{
    (생략)
  },
  "hits":{
    "total":0,
    "max_score":1.0,
    "hits":[
      {
        "_index":"pageview-2020-03-19",
        "_type":"_doc",
        "_id":"Sv6j8nAB212ZgcqwfYvx",
        "_score":1.0,
        "_source":{
          "dpuid":"rRExHwYxyoXobzeBHVf3GlE0jfmnfuXGmBe37cAG",
          "pfno":"12845",
          "vt":"2020-03-19T11:52:44.000Z"
        }
      },
      {
        ....(위와 동일)
      }
    ]
  },
  "aggregations":{
    "sterms#by_campaign":{
      "doc_count_error_upper_bound":0,
      "sum_other_doc_count":0,
      "buckets":[
        {
          "key":"4444",
          "doc_count":44,
          "sum#duration":{
            "value":1975.0
          },
          "cardinality#visitors":{
            "value":1
          },
          "cardinality#visits":{
            "value":4
          },
          "sum#revenue":{
            "value":7600.0
          },
          "value_count#pageviews":{
            "value":44
          },
          "sum#buy":{
            "value":3.0
          },
          "sum#cartin":{
```

```

        "value":0.0
    },
    "cardinality#solditem":{
        "value":6
    },
    "sum#register":{
        "value":1.0
    }
},
{
    ....
},
{
    ....
},
]
}
}
}

```

3.1. Hit의 Source

- getHits() 를 통해 Hit 정보를 얻는다.
- 여러개의 Hits를 for 반복문을 모든 Hit의 Value값을 얻는다.
- Source의 구조는 getSourceAsMap()을 통해 Map 타입의 값으로 얻는다. 이후 `System.out.println` 으로 확인한다.

예) `Map<String, Object> sourceAsMap = hit.getSourceAsMap();`

sourceAsMap을 출력한 결과

```
{dpuid=rRExHWYxyoXobzeBHVf3GlE0jfmnfuxGmBe37cAG, pfno=12845, vt=2020-03-19T11:52:44.000Z}
```

ElasticService.java

```

//ElasticService.java
....(생략)
System.out.println("searchResponse: \n" + searchResponse);

SearchHits hits = searchResponse.getHits();
SearchHit[] searchHits = hits.getHits();
for (SearchHit hit : searchHits) {
    String index = hit.getIndex();
    String id = hit.getId();
    float score = hit.getScore();
    Map<String, Object> sourceAsMap = hit.getSourceAsMap();
    String dpuid = (String) sourceAsMap.get("dpuid");
    String pfno1 = (String) sourceAsMap.get("pfno");
    String vt = (String) sourceAsMap.get("vt");
    System.out.println(vt + " " + pfno + " " + dpuid);
}

```

3.2. aggregation

위 Hits를 가져올 때와 동일함.

ElasticService.java

```
//ElasticService.java
....(생략)
Aggregations resultAggs = searchResponse.getAggregations();
ParsedDateHistogram timeHist = resultAggs.get("by_date");//Date일 경우
//ParsedStringTerms timeHist = resultAggs.get("by_dimension");//Dimension
//경우

List<ParsedDateHistogram.ParsedBucket> timeBuckets
    = (List<ParsedDateHistogram.ParsedBucket>) timeHist.getBuckets();
//List<ParsedStringTerms.ParsedBucket> timeBuckets
//    = (List<ParsedStringTerms.ParsedBucket>) timeHist.getBuckets();

for(ParsedDateHistogram.ParsedBucket timeBucket : timeBuckets) {
    //Cardinality에 대한 방식
    Cardinality visitorsCD =
timeBucket.getAggregations().get("visitors");
    long visitors = visitorsCD.getValue();

    //value_count에 대한 방식
    ParsedValueCount pageviewValueCount =
timeBucket.getAggregations().get("pageviews");
    long pageview = pageviewValueCount.getValue();

    //KeyAsString에 대한 방식
    String keyAsString = timeBucket.getKeyAsString();

    //sum에 대한 방식
    long buy= new Double((double) (
        ((Sum) timeBucket
            .getAggregations()
            .get("buy"))
            .getValue()))
        .longValue();
}
```

3.3. JSON result

(Search response 에서 필요한 값을 선택적으로 데이터 셋에 담기)

ElasticService.java

```
//ElasticService.java
System.out.println("searchResponse: \n" + searchResponse);
```

```

// "data" value
Map<String, List<Object>>> elem
    = new HashMap<String, List<Object>>>();
elem.put("labels", new ArrayList<Object>());
elem.put("pageviews", new ArrayList<Object>());
elem.put("visits", new ArrayList<Object>());
elem.put("visitors", new ArrayList<Object>());
elem.put("ips", new ArrayList<Object>());

// JSON 데이터 셋
Map<String, Map<String, List<Object>>>> jsonObj
    = new HashMap<String, Map<String, List<Object>>>>();
jsonObj.put("data", elem);

Aggregations resultAggs = searchResponse.getAggregations();
ParsedDateHistogram timeHist = resultAggs.get("by_date");
// loop by data list
List<ParsedDateHistogram.ParsedBucket> timeBuckets
    = (List<ParsedDateHistogram.ParsedBucket>) timeHist.getBuckets();

for(ParsedDateHistogram.ParsedBucket timeBucket : timeBuckets) {
    Cardinality visitorsCD =
timeBucket.getAggregations().get("visitors");
    Cardinality visitsCD = timeBucket.getAggregations().get("visits");
    Cardinality ipsCD = timeBucket.getAggregations().get("ips");
    ParsedValueCount pageviewValueCount =
timeBucket.getAggregations().get("pageviews");

    long visitors = visitorsCD.getValue();
    long visits = visitsCD.getValue();
    String keyAsString = timeBucket.getKeyAsString();
    long pageview = pageviewValueCount.getValue();
    long ips = ipsCD.getValue();

    // 필요한 값들 선택적으로 담기
    elem.get("labels").add(keyAsString);
    elem.get("pageviews").add(pageview);
    elem.get("visits").add(visits);
    elem.get("visitors").add(visitors);
    elem.get("ips").add(ips);
}
System.out.println(jsonObj);

return jsonObj;

```

API의 결과물인 JSON 객체는 Php와 Java 모두 동일

```

{
  "data": {
    "labels": [
      "2020-03-19",
      "2020-03-20",
      ...
      "2020-04-27",

```

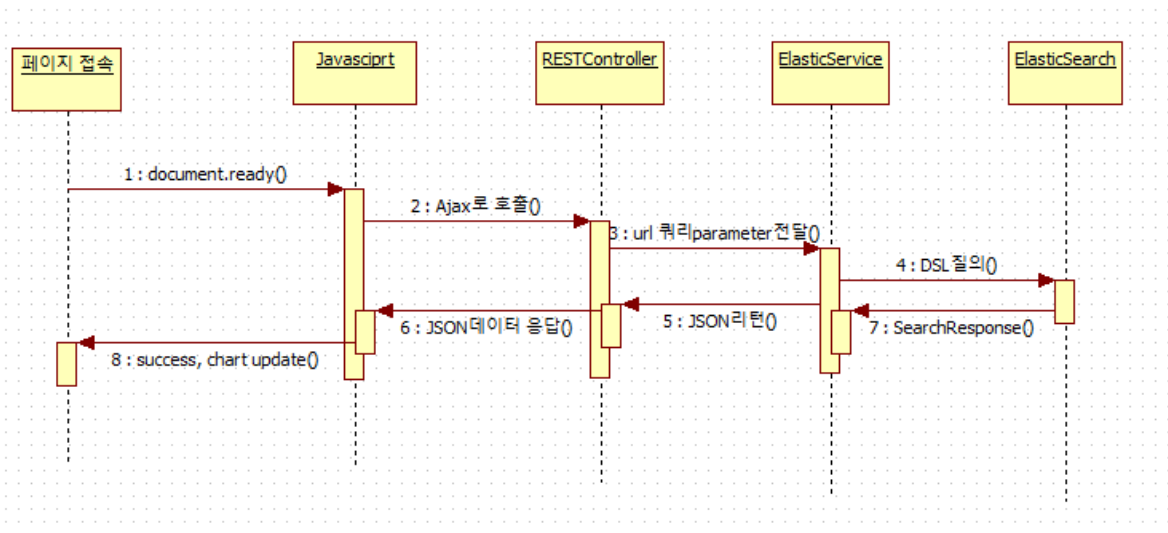
```

    "2020-04-28"
  ],
  "buy": [
    0,
    0,
    0,
    ...
    0,
    1,
    0
  ],
  "revenue": [
    0,
    0,
    0,
    ...
    5400,
    0,
    0,
    4000,
    0
  ],
  "cartin": [
    0,
    0,
    0,
    ...
    0,
    0
  ],
  "register": [
    0,
    0,
    0,
    ...
    1,
    0
  ],
  "solditem": [
    0,
    0,
    ...
    3,
    0
  ]
]
}

```

4. Chart 그리기

4.1. API호출 과정

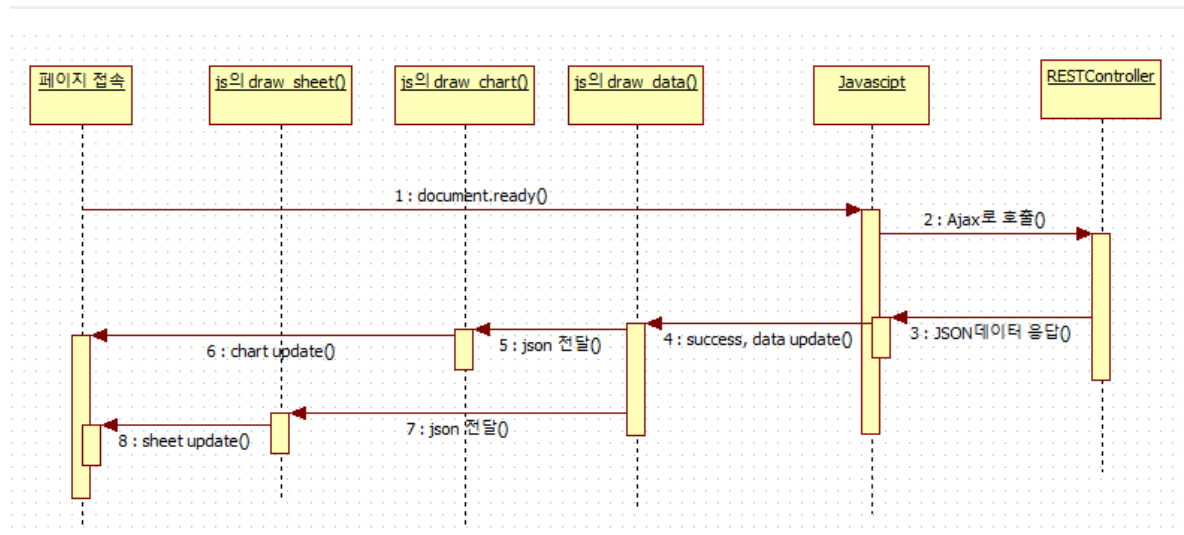


1. \$(document).ready 시 Ajax를 통해 API 호출
2. RestController: URL 쿼리파라미터 바인딩 및 ElasticService클래스의 메소드 호출
3. ElasticSearch 질의 및 응답
4. ElasticService: JSON 응답데이터 가공
5. RESTController: 클라이언트로 응답

- (참고) php코드에서 작성된 js 로직은 Java의 Jsp로 변경할 때, 수정할 필요가 없음.

js의 draw_chart() 함수에서 사용되는 _label 과 _data에 유의하여 Ajax를 통해 받아오는 데이터가 js로직에서 잘 사용될 수 있도록 API의 response의 key값을 일치시켜준다.

4.2. JavaScript 함수 호출과정 상세



page.jsp

```

$(document).ready(get_draw_data());

function get_draw_data() {
    <jsp:useBean id="toDay" class="java.util.Date" />
    //API호출
    $.ajax({
        url: "/apis/basic",
        data: { pfno: [ 12845, 23884 ] , ps: "2020-03-01", pe: "<fmt:formatDate
value="${toDay}" pattern="yyyy-MM-dd"/>", dimension: "page" },
        method: "GET",
  
```



```

        dataType: "json",
        success: function(returned) {
            draw_chart(returned.data);
            draw_sheet(returned.data);
        }
    })
}

```

4.2.1 도넛 차트그리기

page.jsp

```

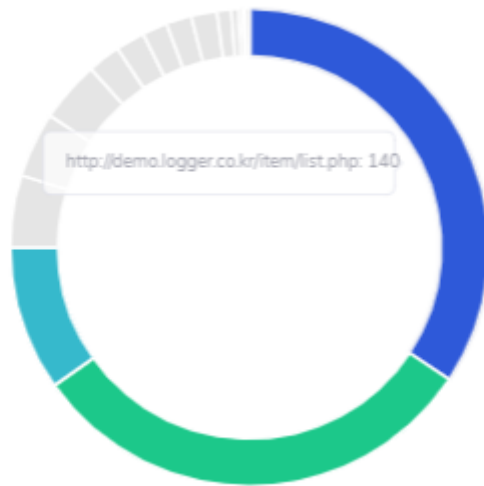
/**차트그리기**
function draw_chart(data) {

    var _labels = [], _data = [];
    for(var i=0; i<data.length; i++) {
        _labels[i] = data[i].dimension;
        _data[i] = data[i].pageviews;
    }

    var ctx = document.getElementById("canvasBasic");
    var pagesChart = new Chart(ctx, {
        type: 'doughnut',
        data: {
            labels: _labels, //필요한 값 : 측정대상이 되는 url
            datasets: [{
                data: _data, //필요한 값 : pageViews 조회수
                backgroundColor: ['#4e73df', '#1cc88a', '#36b9cc'],
                hoverBackgroundColor: ['#2e59d9', '#17a673', '#2c9faf'],
                hoverBorderColor: "rgba(234, 236, 244, 1)",
            }],
        },
        options: {
            maintainAspectRatio: false,
            tooltips: {
                backgroundColor: "rgb(255,255,255)",
                bodyFontColor: "#858796",
                borderColor: '#dddfeb',
                borderWidth: 1,
                xPadding: 15,
                yPadding: 15,
                displayColors: false,
                caretPadding: 10,
            },
            legend: {
                display: false
            },
            cutoutPercentage: 80,
        },
    });
}

```

[차트그림]



4.2.2 시계열 차트 그리기

page.jsp

```
function draw_chart(data) {  
    _labels = [];  
    _data = [];  
    arrIdx = 0;  
  
    if(typeof(data.labels) != 'undefined') {  
        _labels = data.labels;  
    }  
  
    if(typeof(data.pageviews) != 'undefined') {  
        _data_ = linesPreset[arrIdx];  
        _data_.label = "Pageviews";  
        _data_.data = data.pageviews;  
        _data[arrIdx++] = _data_;  
    }  
  
    if(typeof(data.visits) != 'undefined') {  
        _data_ = linesPreset[arrIdx];  
        _data_.label = "Visits";  
        _data_.data = data.visits;  
        _data[arrIdx++] = _data_;  
    }  
  
    if(typeof(data.visitors) != 'undefined') {  
        _data_ = linesPreset[arrIdx];  
        _data_.label = "Visitors";  
        _data_.data = data.visitors;  
        _data[arrIdx++] = _data_;  
    }  
  
    if(typeof(data.ips) != 'undefined') {  
        _data_ = linesPreset[arrIdx];  
        _data_.label = "IP addresses";
```

```

    _data_.data = data.ips;
    _data[arrIdx++] = _data_;
}

var ctx = document.getElementById("canvasTraffic");
var pageviewChart = new Chart(ctx, {
  type: 'line',
  data: {
    labels: _labels,
    datasets: _data,
  },
  options: {
    maintainAspectRatio: false,
    layout: {
      padding: {
        left: 10,
        right: 25,
        top: 25,
        bottom: 0
      }
    },
    scales: {
      xAxes: [{
        time: {
          unit: 'date'
        },
        gridLines: {
          display: false,
          drawBorder: false
        },
        ticks: {
          maxTicksLimit: 7
        }
      }],
      yAxes: [{
        ticks: {
          maxTicksLimit: 5,
          padding: 10,
          // Include a dollar sign in the ticks
          callback: function(value, index, values) {
            return number_format(value);
          }
        },
        gridLines: {
          color: "rgb(234, 236, 244)",
          zeroLineColor: "rgb(234, 236, 244)",
          drawBorder: false,
          borderDash: [2],
          zeroLineBorderDash: [2]
        }
      }],
    },
    legend: {
      display: false
    },
    tooltips: {
      backgroundColor: "rgb(255,255,255)",
      bodyFontColor: "#858796",

```

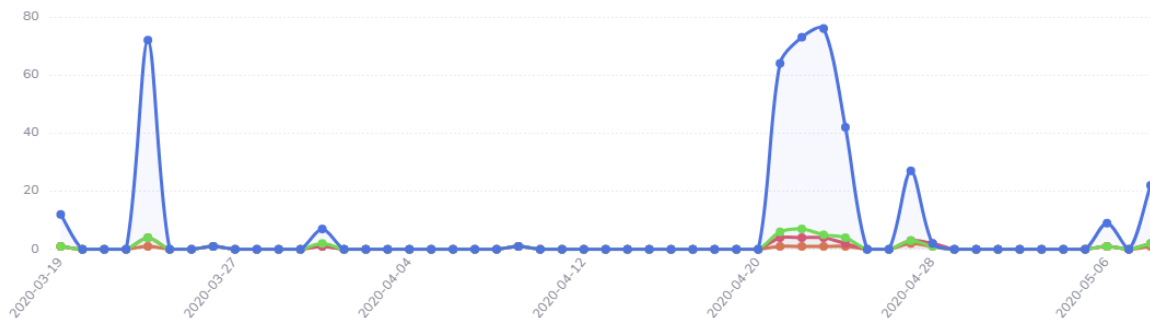
```

    titleMarginBottom: 10,
    titleFontColor: '#6e707e',
    titleFontSize: 14,
    borderColor: '#dddfef',
    borderWidth: 1,
    xPadding: 15,
    yPadding: 15,
    displayColors: false,
    intersect: false,
    mode: 'index',
    caretPadding: 10,
    callbacks: {
      label: function(tooltipItem, chart) {
        var datasetLabel = chart.datasets[tooltipItem.datasetIndex].label
        || '';

        return datasetLabel + ': ' + number_format(tooltipItem.yLabel);
      }
    }
  }
});
}

```

[차트그림]



4.2.3 시트 그리기

page.jsp

```

/**시트 그리기**
function draw_sheet(data) {
    var tbody = $('#tableBasic');

    for(var i=0; i<data.length; i++) {
        $('#tableBasic tr:last').after(
            '<tr>'
            + '<td>' + data[i].dimension + '</td>'
            + '<td>' + data[i].pageviews + '</td>'
            + '<td>' + data[i].visits + '</td>'
            + '<td>' + data[i].visitors + '</td>'
            + '<td>' + data[i].duration + '</td>'
            + '</tr>');
    }
}

```

[시트 그림]

Page	Pageviews	Visits	Visitors	Duration
http://demo.logger.co.kr/item/list.php	140	30	15	2310
http://demo.logger.co.kr/item/view.php	126	29	15	3504
http://demo.logger.co.kr/cart/cart.php	40	18	10	104
http://demo.logger.co.kr/	20	18	11	41
http://demo.logger.co.kr/order/order.php	18	16	8	67
http://demo.logger.co.kr/order/orderSubmit.php	17	16	8	1239
http://bizspring.co.kr/website/home/	9	4	2	71
http://demo.logger.co.kr/register/register.php	8	7	6	17
http://demo.logger.co.kr/login/login.php	7	5	5	23