Sample Elastic Queries to insert JSON output into Elastic

1. **Kibana Console URL**
   http://localhost:5601/app/dev_tools#/console

2. **Create Index**

```
PUT reporttime_at
```

3. **Insert Data to Index using POST**

```
POST /reporttime_at/_doc
{
        "Timestamp" : "2022-11-13T11:30:00.964",
        "RunID" : "11111",
        "UUID" : "a0db4eb0-2e15-4efc-930c-4cb3c6ade575",
        "reporttime" : {
          "datetime" : "2022-06-13 17:10",
          "browser" : "chrome:100.0.4896.88"
        },
        "testcase" : "Test_Case_1"
}
```

4. **Retrieve Index using GET**

```
GET reporttime_at/_search
```

5. **Retrieve specific data from Index using POST & SQL Query**

```
POST /_sql?format=txt
{
  "query": "SELECT max(RunID) FROM reporttime_at"
}
```

6. **List all indices**

```
GET _cat/indices
```

7. **Delete specific row(s) from Index using match**

```
POST reporttime_at/_delete_by_query
{
  "query": {
    "match": {
      "RunID": 11111
    }
  }
}
```

8. **Delete all data from Index**

```
POST reporttime_at/_delete_by_query
    {
     "query": {
        "match_all": {}
      }
    }
```

9. **Retrieve specific data from Index using GET and match**

```
GET reporttime_at/_search
{
  "query":{
   "match":{
     "RunID": 1
   }
}
```

10. **Delete specific index**

```
DELETE /reporttime_at
```

11. **Update specific data on the Index**

NUMBER

Below json helps to amend existing run id from 1 to 2

```
POST reporttime_at/_update_by_query
{
  "script": {
    "source": "ctx._source.RunID=2",
    "lang": "painless"
  },
  "query": {
    "term": {
      "RunID": "1"
    }
  }
}
```

STRING

```
POST url_poc/_update_by_query
{
  "script": {
    "source": "ctx._source.url='logo.png'",
    "lang": "painless"
  },
  "query": {
    "term": {
      "RunID": 101
    }
  }
}
```

12. **Update default ignore char limits to the fields**

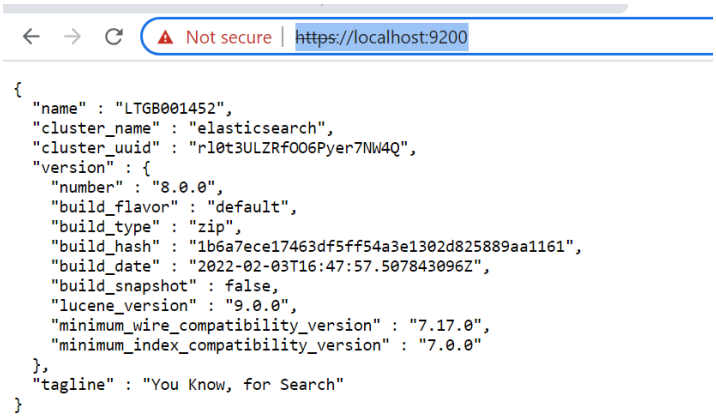
```
PUT violations_at
{
  "mappings": {
    "properties": {
      "message": {
        "type": "keyword",
        "ignore_above": 5000
      },
      "target": {
        "type": "keyword",
        "ignore_above": 5000
      },
      "html": {
        "type": "keyword",
        "ignore_above": 5000
      }
    }
  }
}
```

***Note**: default size of the field is 256. ignore_above has to be set while creating the index itself. The above json is used to create the index 'violations_at' with message, target and html fields. Once index created, we can push the data with new/additional fields if required.*

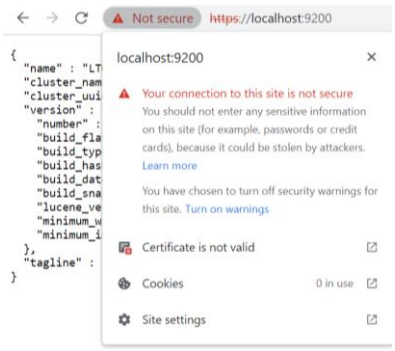
Adding certificate to Certificate Authority for ElasticSearch

It is required to add the certificate into 'cacerts' to enable our solution to write the json output into Elastic Search for accessibility audits.

1. Open elastic search url (e.g.https://localhost:9200/) in browser,



2. Click the Not secure error icon which resides left to the address bar and open 'Certificate is not valid' option



3. Go to 'Details' tab and click 'Export' button
4. Save it in your local drive as '.cer' file (e.g. C:\Users\<username>\ElasticsearchSecurity.cer)
5. Open command prompt
6. Execute the below command from elasticsearch bin directory("<PathToElasticsearch>\jdk\bin") to list down the certificates from sun java:
   keytool -list -keystore "<path to java home>\jre\lib\security\cacerts"

   e.g.
   keytool -list -keystore "C:\Program Files\Java\jdk1.8.0_291\jre\lib\security\cacerts"

   NB: No need to enter password

7. If your certificate is not listed, execute the below command to import your certificate to Certificate Authority (CA):

e.g.

keytool -import -noprompt -trustcacerts -alias ForElastic -file "C:\Users\<username>\ElasticsearchSecurity.cer" -keystore "C:\Program Files\Java\jdk1.8.0_291\jre\lib\security\cacerts" -storepass changeit