Programme en Java qui permet de créer :  
  
- Une configuration pour l’outil logstash qui permet d’insérer les données dans l’index elasticsearch.  
  
- un fichier Json qui permet de créer un mapping D’Elasticsearch qui permet de créer des types de documents dans un index

exemple :  
  
On a des logs sous la forme :  
id,timestamp  
1,1466587092061

10,1466587092061

100,1466587108108

1000,1466587133210

10000,1466587298268

100000,1466587299770

10001,1466587092061

10002,1466587092061

10003,1466587092061

10004,1466587092061

10005,1466587092061

10006,1466587092061

10007,1466587092061

10008,1466587092061

10009,1466587092061

1001,1466587133210

Pour traiter ce document avec Logstash, il faut un fichier de configuration qui ressemble à ça :  
  
input {

file{

path =>"**@path**"

type => "**@type**"

start\_position => "beginning"

}

}

filter{

if [message] =~ /^"COL1","COL2"/ {

drop { }

}

csv {

columns => [**@columns**]

separator => "**@separator**"

source => message

remove\_field => ["message","host","path","@version"]

}

}

output {

elasticsearch{

action => "index"

hosts => "localhost"

index => "logvf"

workers => 1

}

stdout{

codec => rubydebug

}

}

Le but du programme java d’alimenter ce type de configuration avec les éléments nécessaires à une bonne lecture et insertion du fichier ‘exemple’ dans le type qui lui correspond dans l’index Elasticsearch.  
  
Dans le cas cité ci-dessus, les **“élements”** seront remplacés, via le programme java, par des inputs (champs, types de chaque champs, separateur et le chemin des fichiers csv respectant cette architecture) permettant de lire et d’insérer les documents csv concernés dans l’index et le type qui leur sont approriés.

Pour ce qui est du Mapping, un mapping elasticsearch contient : les champs (*properties)*, leurs types( [string](https://www.elastic.co/guide/en/elasticsearch/reference/current/string.html), [date](https://www.elastic.co/guide/en/elasticsearch/reference/current/date.html), [long](https://www.elastic.co/guide/en/elasticsearch/reference/current/number.html), [double](https://www.elastic.co/guide/en/elasticsearch/reference/current/number.html), [boolean](https://www.elastic.co/guide/en/elasticsearch/reference/current/boolean.html) or [ip](https://www.elastic.co/guide/en/elasticsearch/reference/current/ip.html) [Pour les strings il faut préciser s’il faut indexer en full text alias “analyzed” ou bien stocker comme mots clés “not\_analyzed”et pour date il faut préciser le format/ )  
  
Pour notre exemple, le mapping qui sera produit par notre programme ressemblera à ça :  
  
{  
 "mappings": {  
  
 "properties": {   
 "id": { "type": "string" },  
 "timestamp": { "type": "date" },  
  
 }  
}  
}  
  
Les fonctions qui produiront le mapping utiliseront aussi des éléments des inputs mais uniquement ceux nécessaires à la création du mappin comme par exemple l’index et les champs et les types de champs.