



💡 java logback . 6.2 .

:

- (LogBack) ,
- LogBack SocketAppend
- Json

LogStash

logstash.conf

```
# [Beats input plugin]
# listen on port 5044 for incoming Beats connections
input {
  tcp {
    port => 5044
    codec => json_lines
  }
}

filter {
  json {
    source => "message"
  }
}

# [Elasticsearch output plugin]
# index into Elasticsearch
output {
  elasticsearch {
    hosts => "localhost:9200"
    manage_template => false
    index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
  }
  stdout { codec => rubydebug }
}
```

5044 tcp , json .

Lib

```

    <!-- LogBack for LogStash -->
    <dependency>
      <groupId>net.logstash.logback</groupId>
      <artifactId>logstash-logback-encoder</artifactId>
      <version>5.1</version>
    </dependency>
    <dependency>
      <groupId>ch.qos.logback</groupId>
      <artifactId>logback-classic</artifactId>
    </dependency>
    <dependency>
      <groupId>ch.qos.logback</groupId>
      <artifactId>logback-core</artifactId>
    </dependency>

    <dependency>
      <groupId>ch.qos.logback</groupId>
      <artifactId>logback-access</artifactId>
    </dependency>

```

logback logstash-logback-encoder .

LogBack.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <appender name="STASH" class="net.logstash.logback.appender.LogstashTcpSocketAppender">
    <keepAliveDuration>5 minutes</keepAliveDuration>
    <reconnectionDelay>10 second</reconnectionDelay>
    <waitStrategyType>sleeping</waitStrategyType>
    <writeBufferSize>16384</writeBufferSize>
    <destination>180.70.98.37:5044</destination>
    <encoder class="net.logstash.logback.encoder.LogstashEncoder">
      <version>6</version>
    </encoder>
  </appender>

  <logger name="org.springframework.data" level="error" />

  <root level="info">
    <appender-ref ref="STASH"/>
  </root>
</configuration>

```

DB

LogStash Socket .

LogsStash .

Log Class

```

import java.io.Serializable;
import com.fasterxml.jackson.databind.ObjectMapper;

public class InFlowLog implements Serializable {
    // . ?
    // RDB Table , SQL
    // ... SQL .

    private String category1;

    private String category2;

    private String category3;

    private String category4;

    private String region;

    private String goodscode;

    private String goodsname;

    private String search;

    private String router;

    private String action;

    private long ctime;

    private String userip;

    private int age;

    private String sex;

    private String ordertip;

    private String refgoodscode;

    @Override
    public String toString() {
        ObjectMapper mapper = new ObjectMapper();
        String jsonString = "";
        try {
            jsonString = mapper.writeValueAsString(this);
        } catch (IOException e) {
            jsonString = "jsonerror";
        }
        return jsonString;
    }
}

```

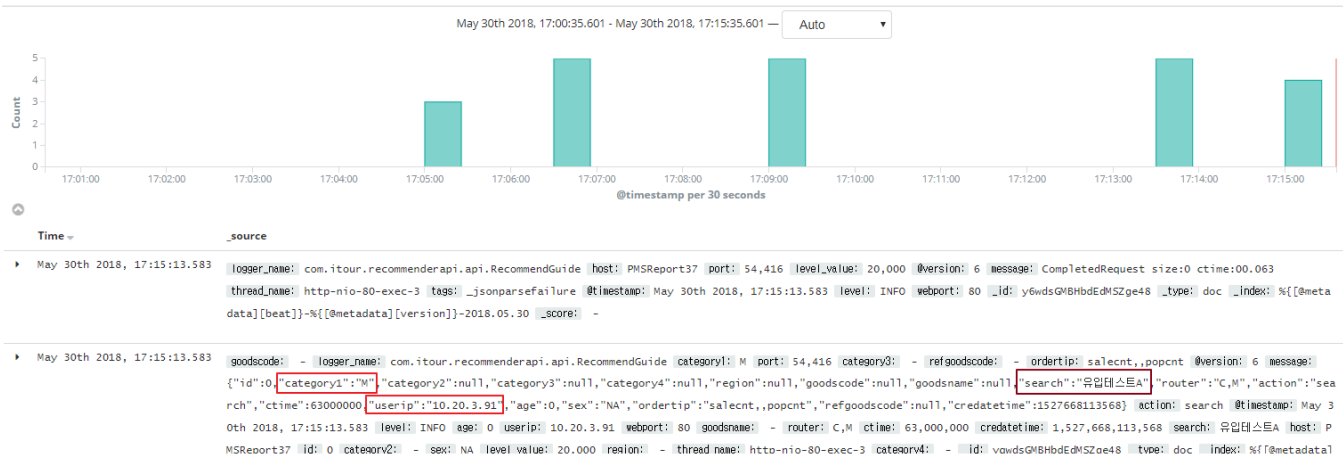
, toString JsonString .

Log Class

```

InFlowLog addLog = new InFlowLog();
logger.info( "{}",addLog.toString() ); // ....

```



Class , .

Management / Kibana

Index Patterns Saved Objects Advanced Settings

+ Create Index Pattern

★ ★

★ *

, Refresh , .

RDB DBA ,

, . NOSQL .

GEOPINT

```

public class Location implements Serializable {

    private Double lat;

    private Double lon;

    public Double getLat() {
        return lat;
    }

    public void setLat(Double lat) {
        this.lat = lat;
    }

    public Double getLon() {
        return lon;
    }

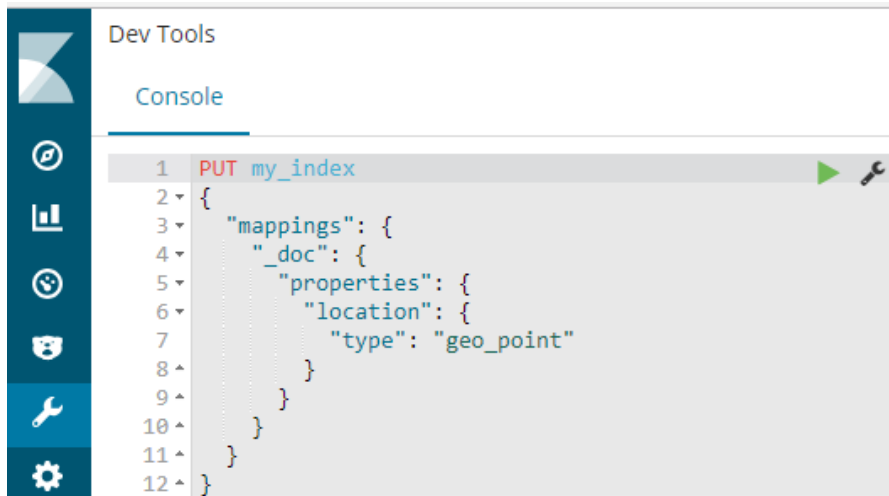
    public void setLon(Double lon) {
        this.lon = lon;
    }

}

public class InFlowLog implements Serializable {
    private Location location;
}

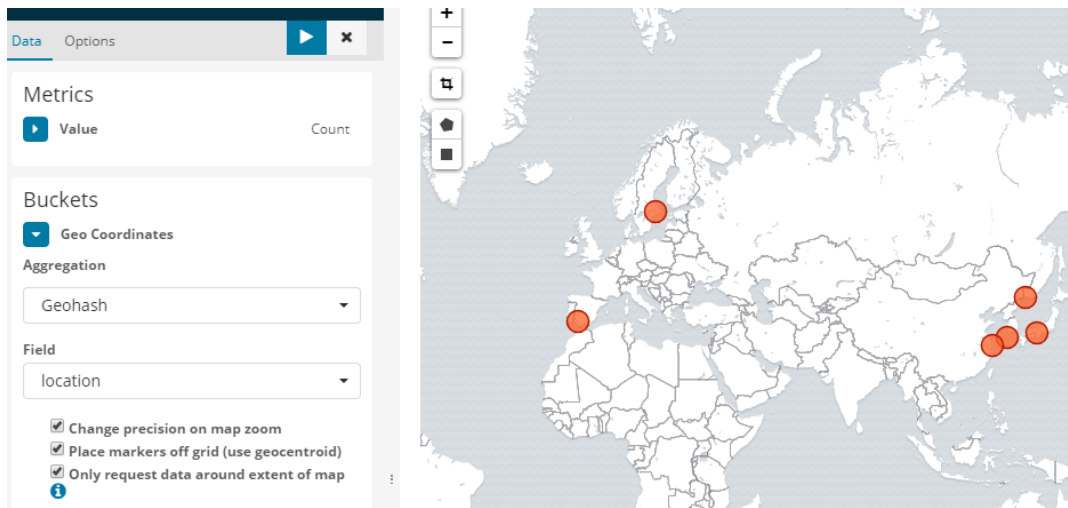
```

geopoint



```
put logstash-idxname
{
  "mappings":{
    "doc":{
      "properties": {
        "location": {
          "type": "geo_point"
        }
      }
    }
  }
}
```

- .



Links:

- <https://github.com/logstash/logstash-logback-encoder>
- <https://layer4.fr/blog/2016/10/25/efficient-logging-spring-boot-logback-logstash/>
-