# 1 JSON模板

# uem\_template.json

{

"template" : "uem\_json\_\*",

"order[[1]](#footnote-1)" : 0,

"settings" : {

"index.number\_of\_replicas[[2]](#footnote-2)" : "1",

"index.number\_of\_shards[[3]](#footnote-3)" : "5",

"index.refresh\_interval[[4]](#footnote-4)" : "10s"

},

"mappings[[5]](#footnote-5)" : {

"doc[[6]](#footnote-6)" : {

"properties[[7]](#footnote-7)" : {

"@timestamp" : {

"format" : "strict\_date\_optional\_time || epoch\_millis",

"type" : "date[[8]](#footnote-8)",

~~"analyzer" : "standard[[9]](#footnote-9)"~~

},

"IP" : { "type" : "keyword[[10]](#footnote-10)" },

"MAC" : { "type" : "keyword" },

"STAFFID" : { "type" : "keyword" },

"STAFFTELNO" : { "type" : "keyword" },

"ORGAID" : { "type" : "keyword" },

"Curr\_Time\_of\_Term" : { "type" : "keyword" },

"Elem\_Outer\_Html" : { "type" : "keyword" },

"Oper\_Type" : { "type" : "keyword" },

"New\_Value" : { "type" : "keyword" },

"Location" : { "type" : "keyword" },

"XTag" : { "type" : "keyword" },

"ERROR" : { "type" : "keyword" },

"Cause" : { "type" : "keyword" },

"ERR\_DESC" : { "type" : "keyword" },

"ERR\_DETAIL" : { "type" : "keyword" },

"DETAIL" : { "type" : "keyword" },

"SOLUTION" : { "type" : "keyword" },

"ERRCODE" : { "type" : "keyword" },

"STACK\_INFO" : { "type" : "keyword" },

"Sys\_Prompt" : { "type" : "keyword" },

"Confirm" : { "type" : "keyword" },

"Open" : { "type" : "keyword" },

"Modal" : { "type" : "keyword" },

"Modeless" : { "type" : "keyword" },

"Popup\_Tab" : { "type" : "keyword" },

"Shutdown\_Tab" : { "type" : "keyword" }

}

}

}

}

# 2 启动配置

# uem.conf

input {

tcp {

port => 9514[[11]](#footnote-11)

type => syslog

codec => "json"[[12]](#footnote-12)

}

udp {

port => 9514

type => syslog

codec => "json"

}

}

output {

elasticsearch {

hosts => ["http://127.0.0.1:9200"]

index => "uem\_json\_%{+YYYY.MM.dd}"

template => "/home/elk/logstash-6.4.2/config/uem\_template.json"[[13]](#footnote-13)

template\_name => "uem\_json"[[14]](#footnote-14)

template\_overwrite => true

}

}

1. Order默认值为0，order值越大在merge的时候优先级越高。 [↑](#footnote-ref-1)
2. ES默认为一个索引创建5个主分片，并分别为其创建一个副本分片。也就是说每个索引都由5个主分片成本，而每个主分片都相应的有一个copy。 [↑](#footnote-ref-2)
3. 因为ES是个分布式的搜索引擎，所以索引通常都会分解成不同部分，而这些分布在不同节点的数据就是分片。ES自动管理和组织分片，并在必要的时候对分片数据进行再平衡分配，所以用户基本上不用担心分片的处理细节，一个分片默认最大文档数量是20亿。 [↑](#footnote-ref-3)
4. 索引刷新的时间间隔，每10s创建一个新的段，可以理解为lucene的索引文件，增加这个值可以允许更大的segment写入，减少以后的segment合并压力。在初始化索引时，可以禁用 refresh 和 replicas 数量，

   如果需要一次加载较大的数据量进index里面时，可以先禁用refresh，把index.refresh\_interval设置成为 -1 ，把 index.number\_of\_replicas设置成 0。暂时把多个shard副本关闭（即如果当前index发生损坏便用丢失数据），但是这样做可以大大加快索引速度。当初始化索引完成，可以将 index.refresh\_interval 和 index.number\_of\_replicas 设置回原来的值。 [↑](#footnote-ref-4)
5. 文档的field定义，相当于数据库的字段 [↑](#footnote-ref-5)
6. 固定写法，必须要有doc键值 [↑](#footnote-ref-6)
7. 固定写法，必须要有properties键值 [↑](#footnote-ref-7)
8. 数据类型，有简单数据类型：string、date、long、double、integer、boolean以及ip等。层级结构类型：JSON型的object，嵌套类型（都是JSON）。特殊结构类型：geo\_point、geo\_shape以及completion。 [↑](#footnote-ref-8)
9. ES的分析器，6.0以后此配置失效，默认为standard，还有其他一些分析器如：simple、whitespace、stop、keyword、pattern、language、fingerprint。 [↑](#footnote-ref-9)
10. 从6.4版本开始，不再支持string类型，而主要使用keyword类型 [↑](#footnote-ref-10)
11. Linux下1024以内的端口bind需要root权限 [↑](#footnote-ref-11)
12. 使用json解码才能正确识别json报文中的key [↑](#footnote-ref-12)
13. 此处配置JSON解析模板路径 [↑](#footnote-ref-13)
14. 可以自由定义 [↑](#footnote-ref-14)