# 安装IK分词器

在学习ES概念的时候，我们知道ES创建倒排索引表的时候，需要对文档内容进行分词，而且我们去ES搜索的时候，需要对用户输入的内容也去做分词！

那么分词就要对中文或者英文的语义进行分析，按照词语含义进行拆分！这种拆分其实是比较复杂的，需要有特殊的算法去支持，在ES中，默认分词的规则对于中文的支持是比较差的，我们可以通过测试来看一下：

分词器：

ES在创建倒排索引的时候需要对文档分词；在搜索的时候，需要对用户输入内容分词，但默认的分词规则对中文处理并不友好，我们在Kibana的DevTools中测试：

**POST /\_analyze**

**{**

**“analyzer”: “standard”,**

**“text”: “hello，你好世界”**

**}**

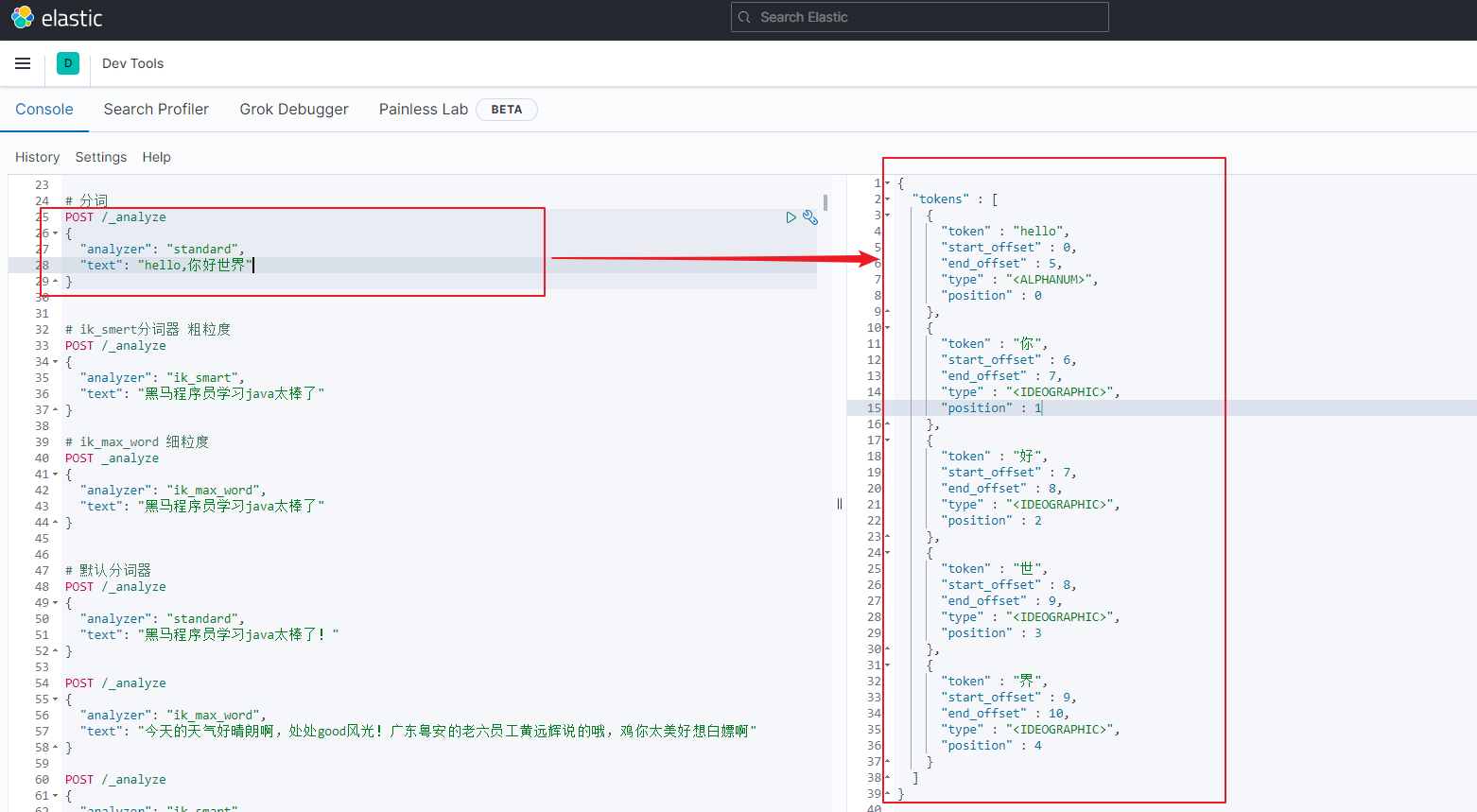
**POST是请求方式POST；**

**\_/analyze代表要做分词分析；**

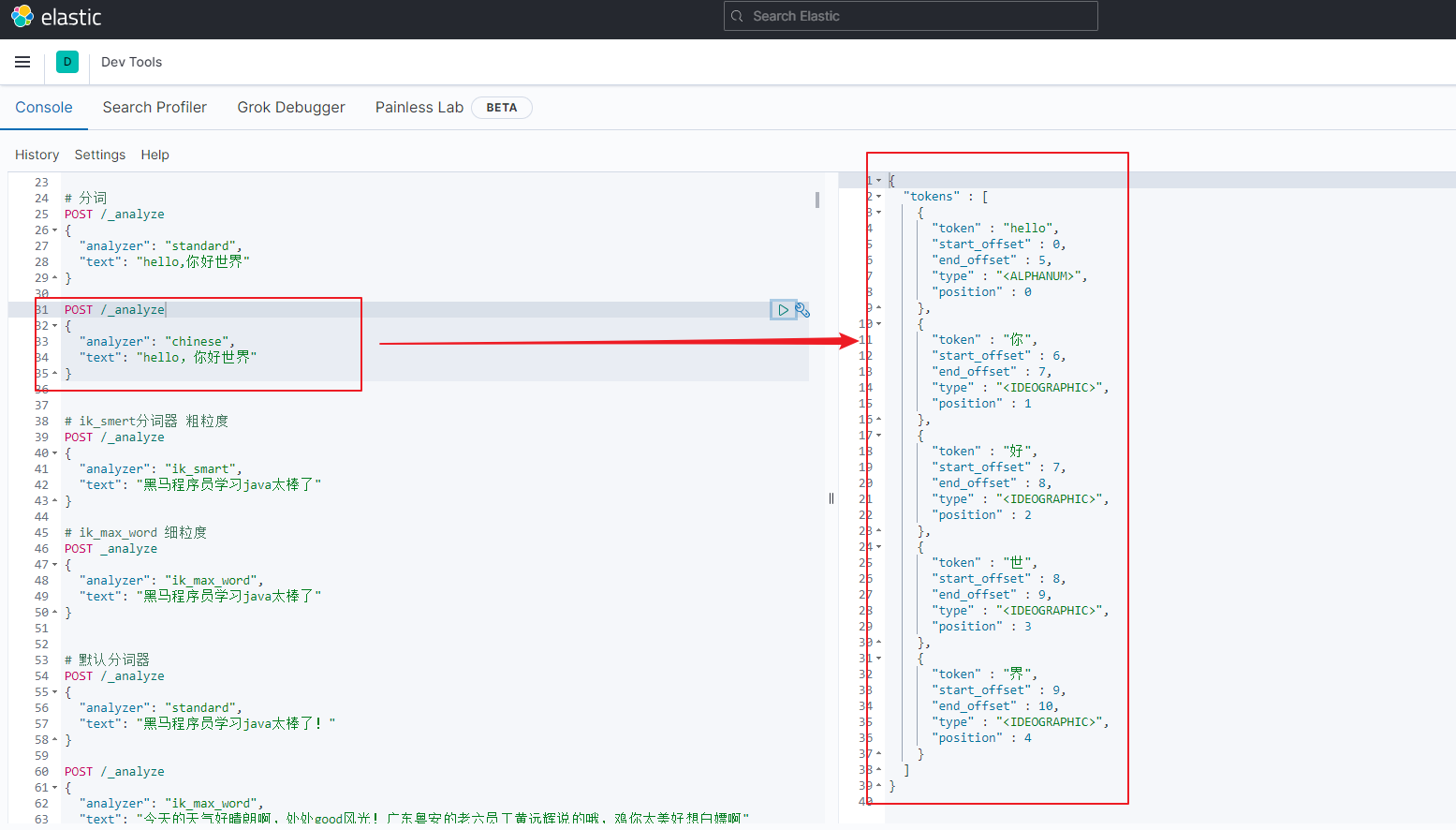
**POST /\_analyze下面就是json格式的DSL语句；**

**“analyzer”: “standard”：analyzer表示分词器，standard表示使用ES中默认的标准分词器；**

**“text”: “hello，你好世界”：表示要分词的文本，文本有中文有英文哦**



可以看到，使用默认分词器分词时，英文还是不错的，可以准确分词，但是中文却是逐字分词，你会问是不是分词器的问题啊？分词器换成chinese试一试：



可以发现，分词器换成chinese，对中文依旧是一字一分，没办法去理解中文含义按内容分！这就是ES中提供的分词器的问题了，包括标准分词器也是如此！

所以，我们要想去分词中文，就不能使用默认分词器，一般中文分词器会使用IK分词器

## IK分词器：

专门给ES来使用的，IK分词器内部有两种模式，一种时ik\_smart，一种时ik\_max\_word

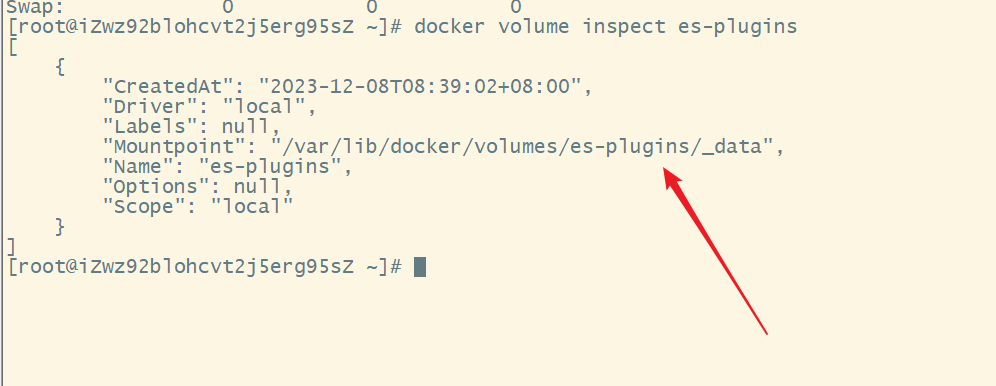
安装IK分词器有两种方式，一种是在线安装，一种是离线安装，在线安装很慢，推荐使用离线安装！离线安装需要把IK分词器插件放到ES的plugins目录位置！

## 离线安装IK分词器：

### 1：查看数据卷目录

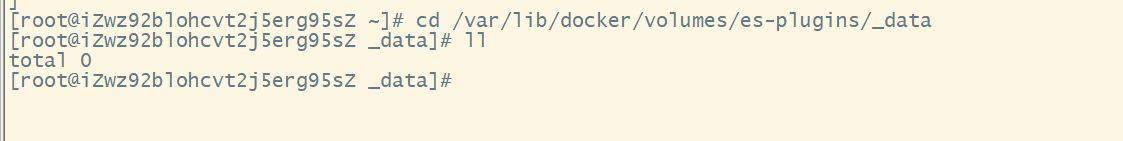
安装插件需要知道elasticsearch的plugins目录位置，因为我们创建ES容器时用了数据卷挂载，因此需要查看elasticsearch的数据卷目录，通过以下目录查看：

**docker volume inspect es-plugins**

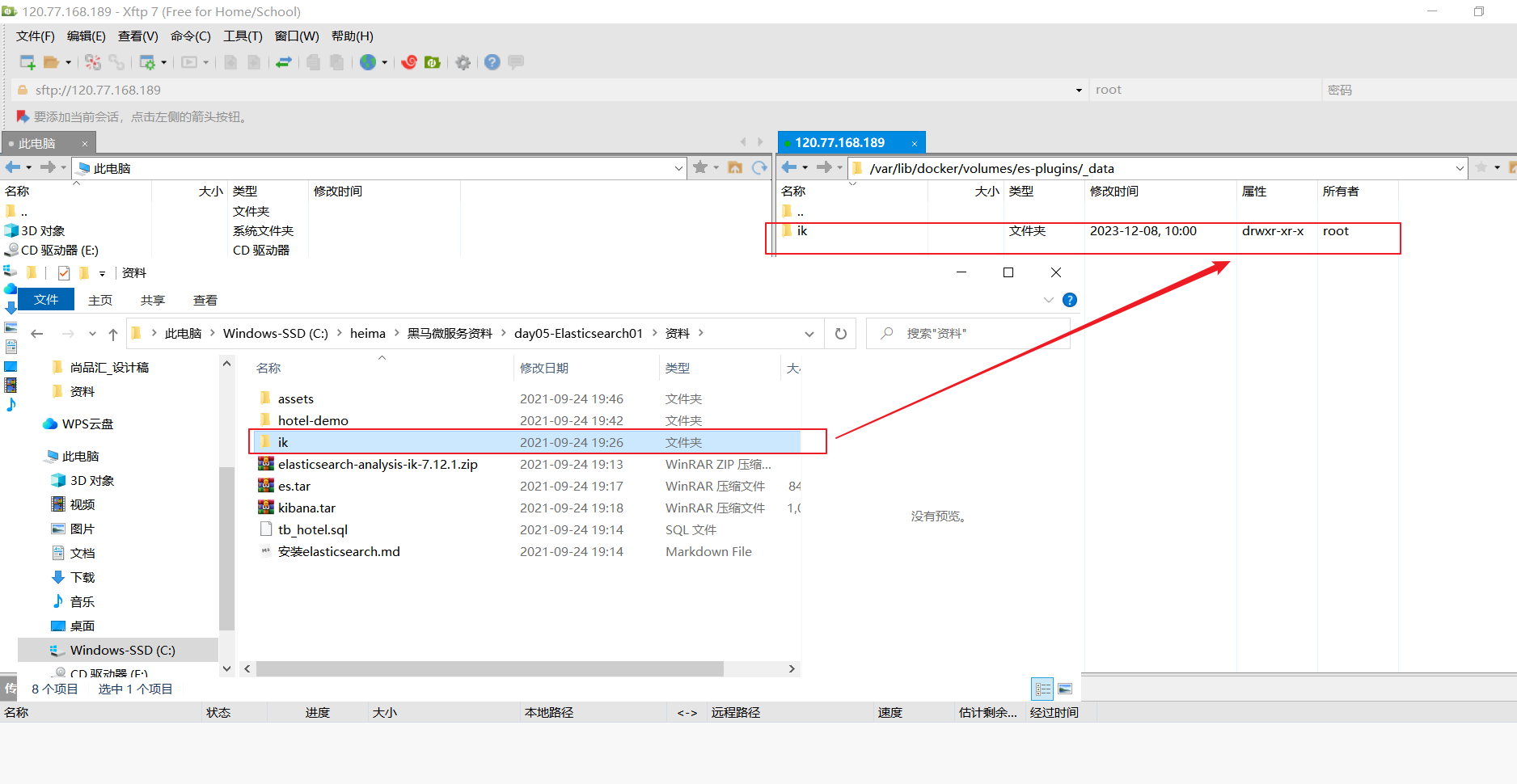


### 2：进入数据卷es-plugins在宿主机挂载目录：

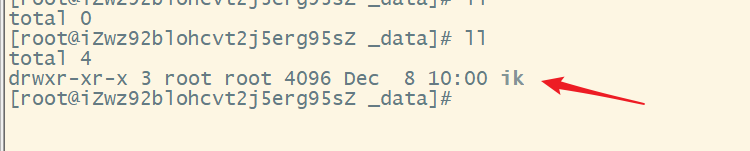
cd /var/lib/docker/volumes/es-plugins/\_data



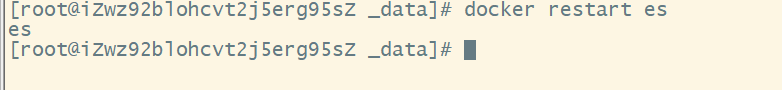
### 3：上传ik分词器插件到es-plugins在宿主机挂载目录



### 4：查看上传的的IK分词器插件



### 5：重启es容器：docker restart es



## 测试IK分词器

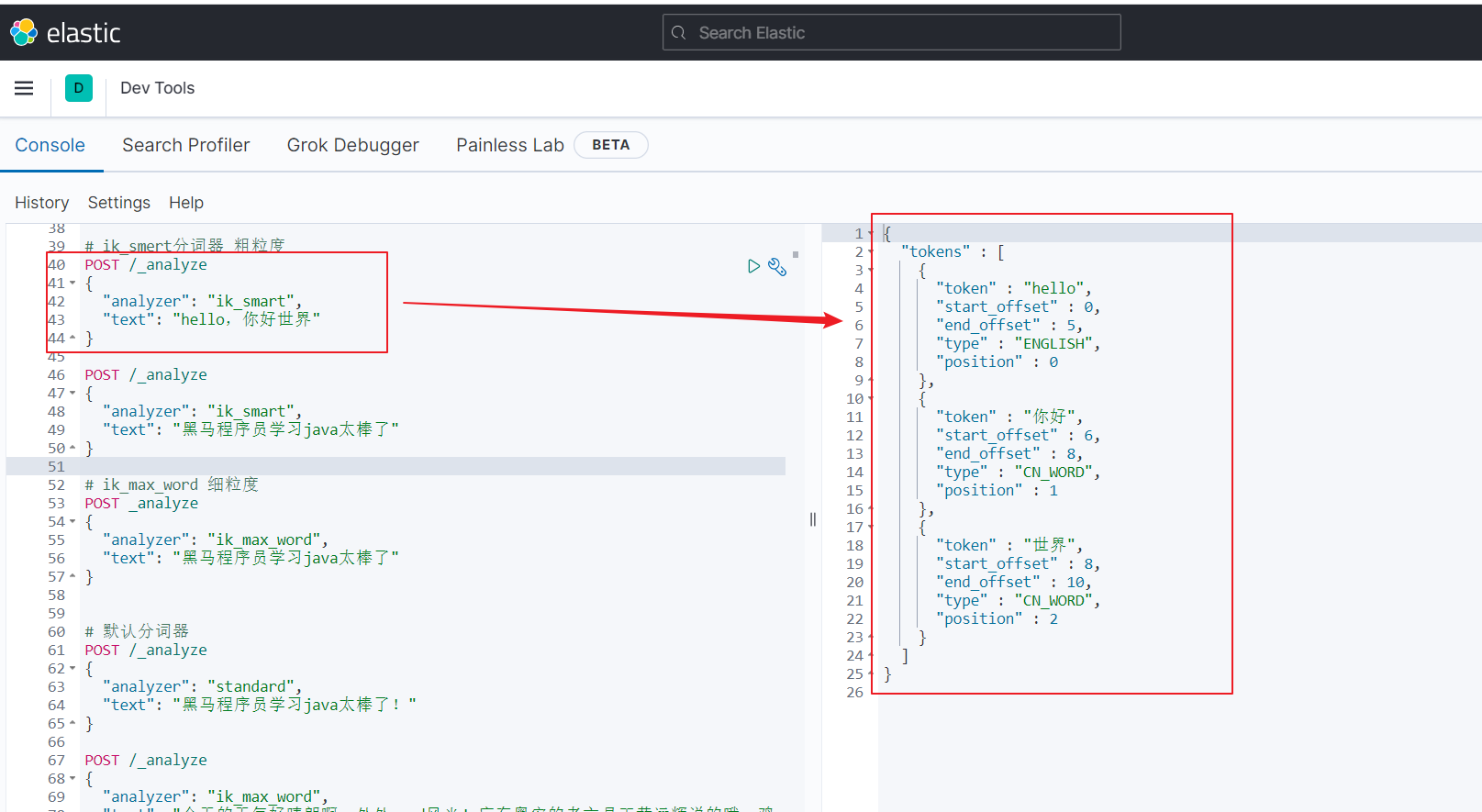
### ik\_samrt和ik\_max\_word

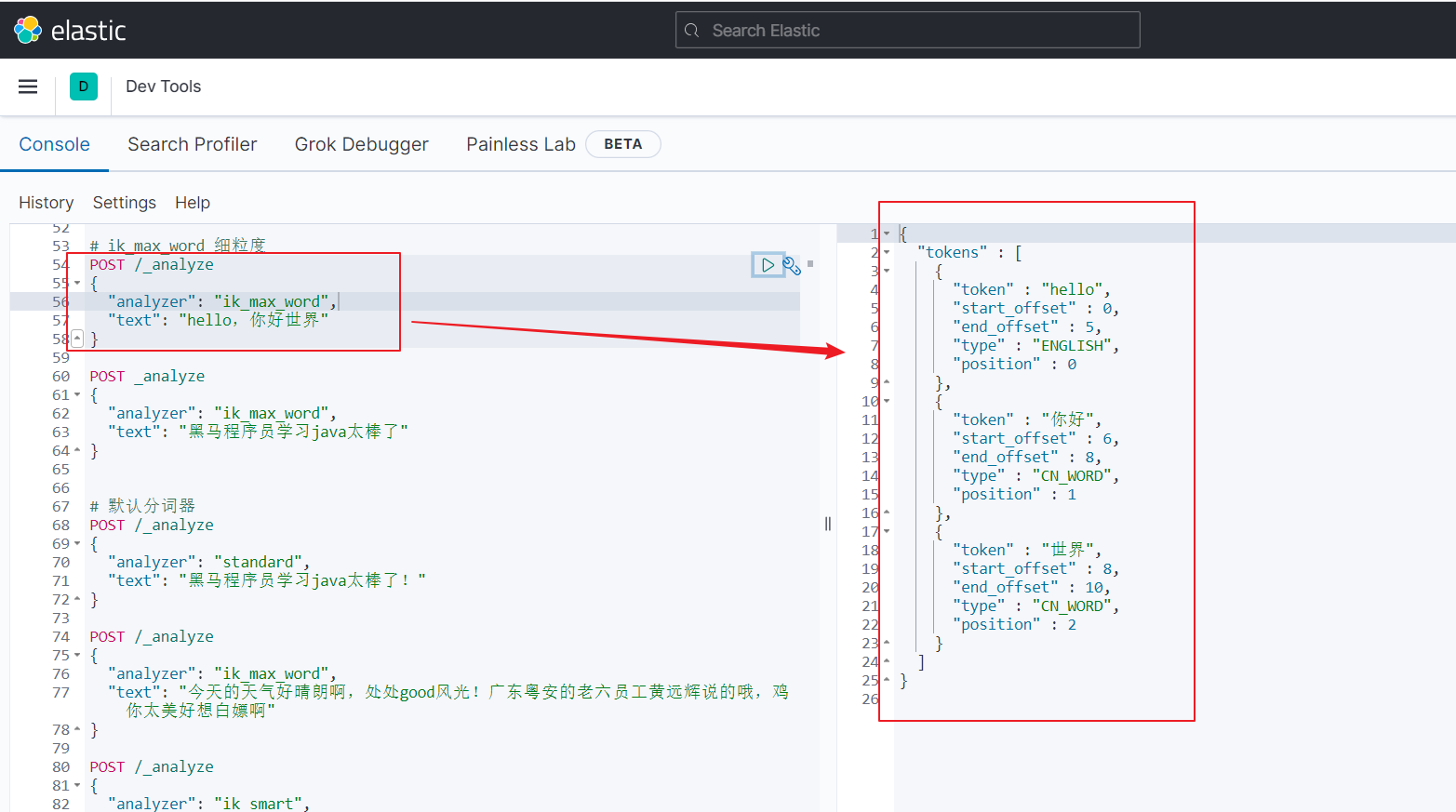
刚才提到过，IK分词器有两种模式，一种时ik\_smart，一种时ik\_max\_word，ik\_smart是最少切分，ik\_max\_word是最细切分.

总而言之。Ik\_max\_word是按照最细粒度切分，ik\_samrt是粗粒度切分。

ik\_smart会从字符最多开始，往字符越来越少去看，比如看五个字是不是一个词，如果不是再看四个字，四个还不是再看三个，如果三个字的程序员刚好是一个词，就不往下切分了，就不再往下看程序是不是词了，所以ik\_samrt分词的粒度是比较粗的，词分得会比较少！

而ik\_maxword则相反，还会继续往下分，词分得就比较多！





### 那这种分词多和少带来的后果是什么？

你想啊，ik\_max\_word分出来的词比较多，用户搜索的时候，无论输入“程序员”，“程序”，“员”这是三个词的任何一个，都能搜到索引的文档，但是如果是ik\_smart，你分出来的只有“程序员”这一个词，所以这个时候如果用户搜“程序”，这篇文档就不会被搜到，搜索的概率就比较低，但是ik\_smart的好处是分词少，占用的内存空间也就少，将来内存就可以缓存更多的数据，插叙效率就高一点！

因此你需要在内存占用、查询效率、被搜索到的概率这之间做出选择！