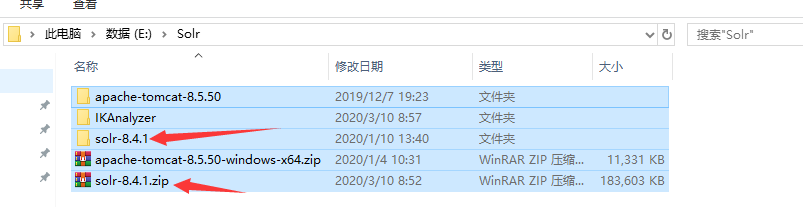
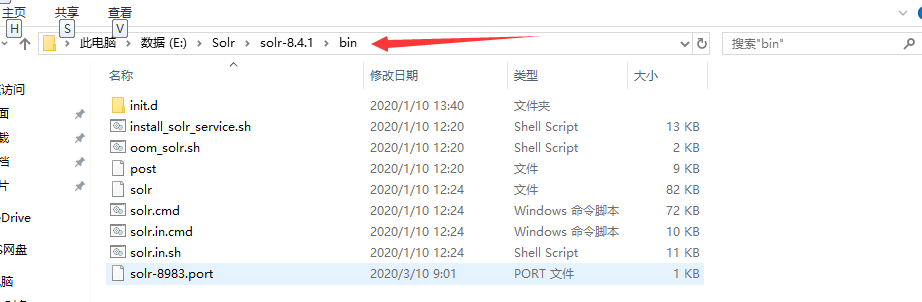
Solr安装以及springboot整合solr(8.4.1版本)

#### 安装Solr（直接启动）

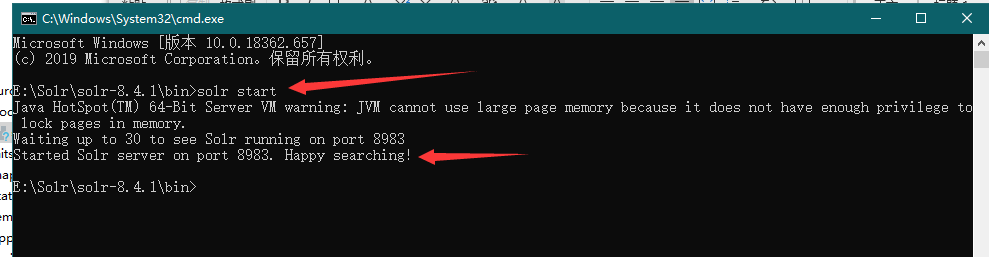
1.下载Solr并解压文件



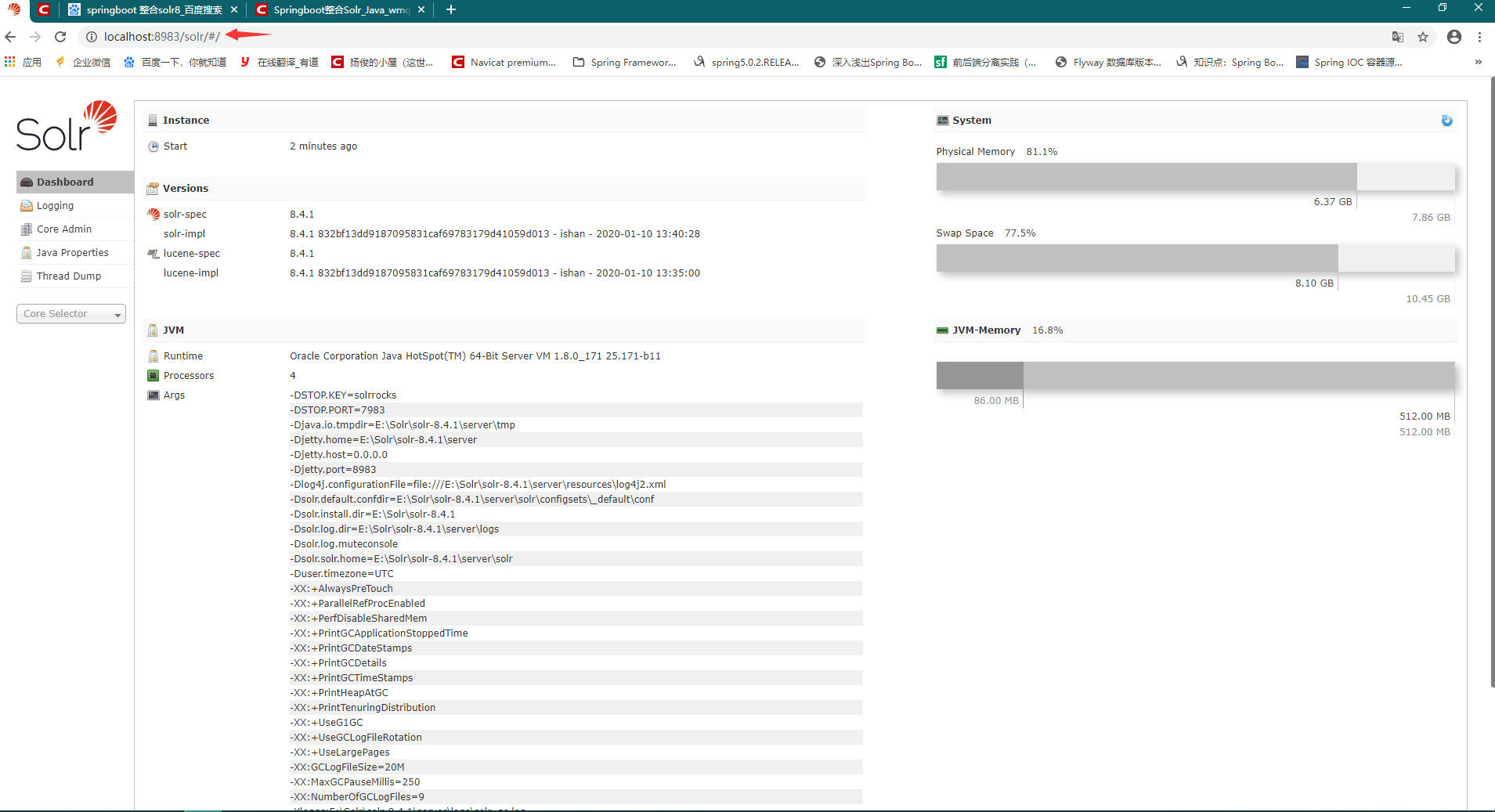
1. 进入Solr的bin目录下 如：E:\Solr\solr-8.4.1\bin

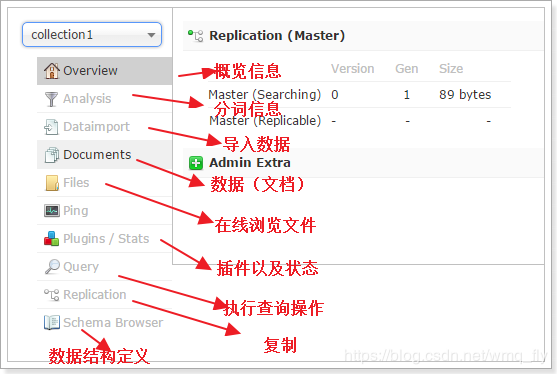
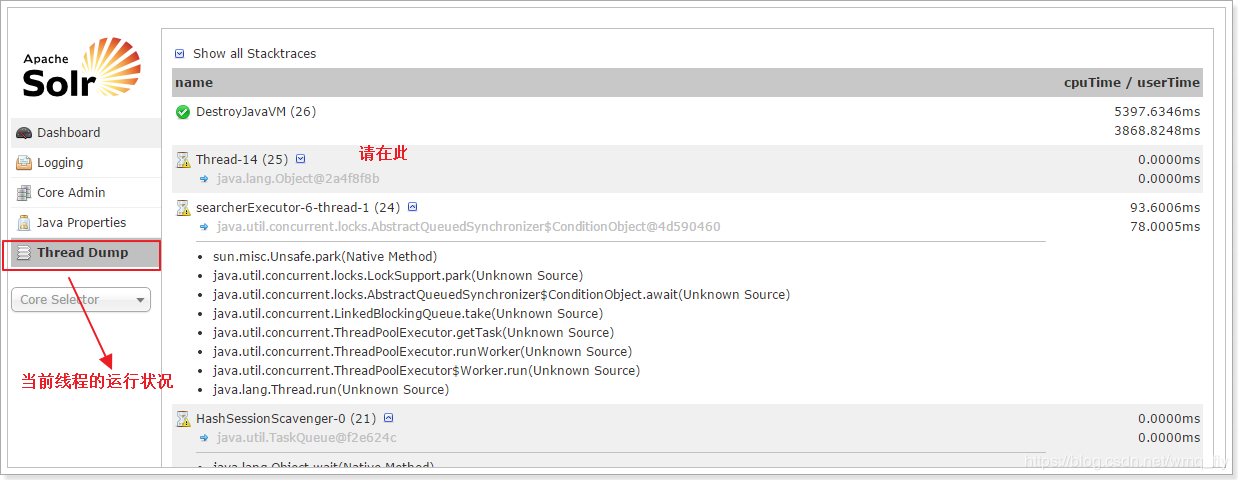
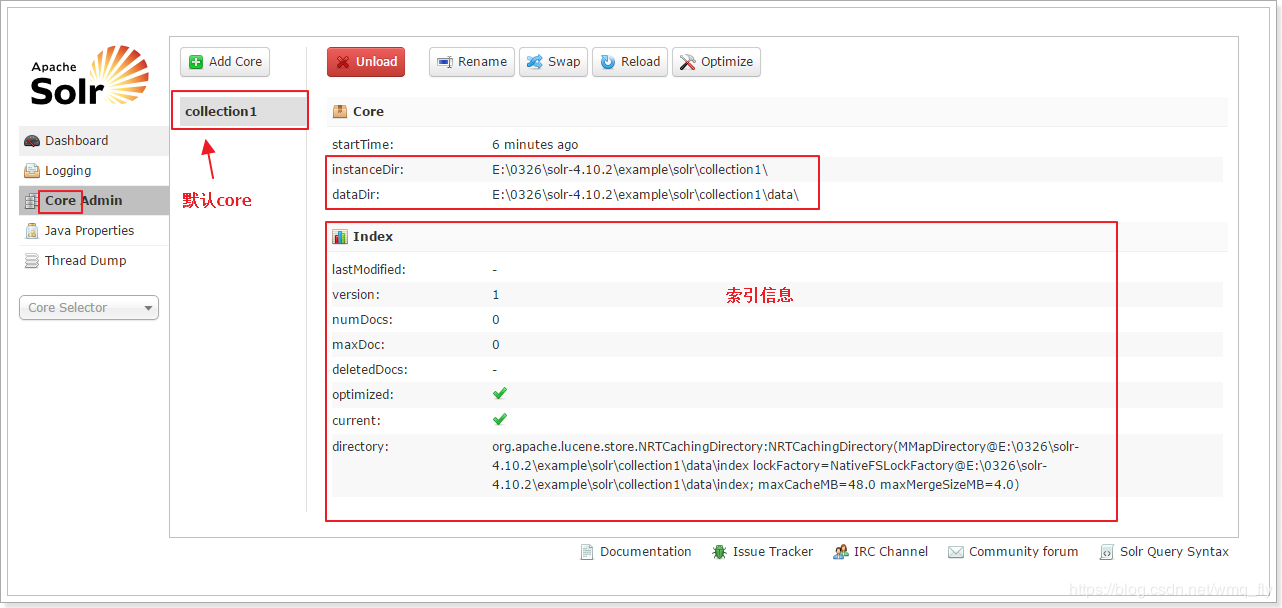
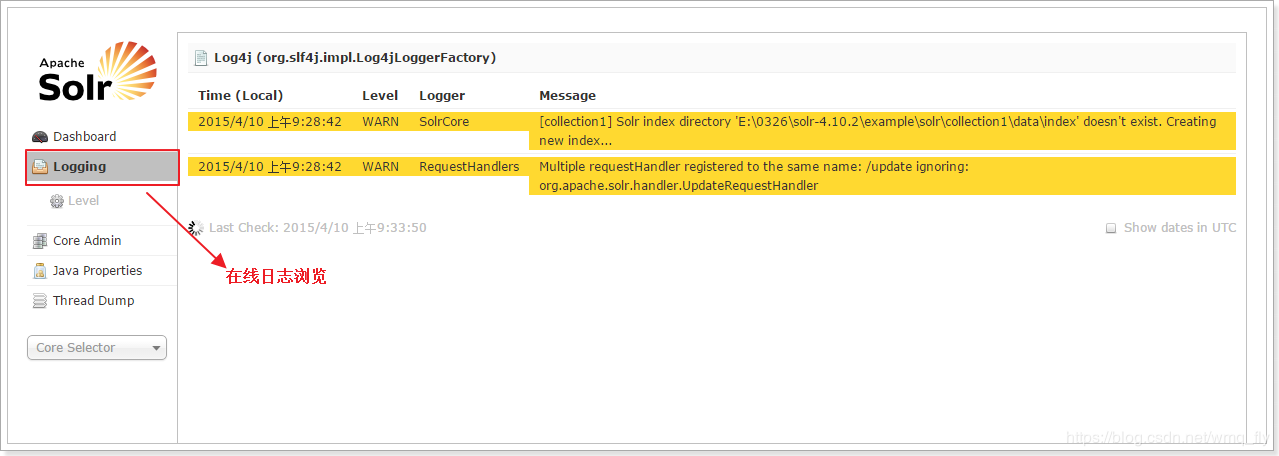
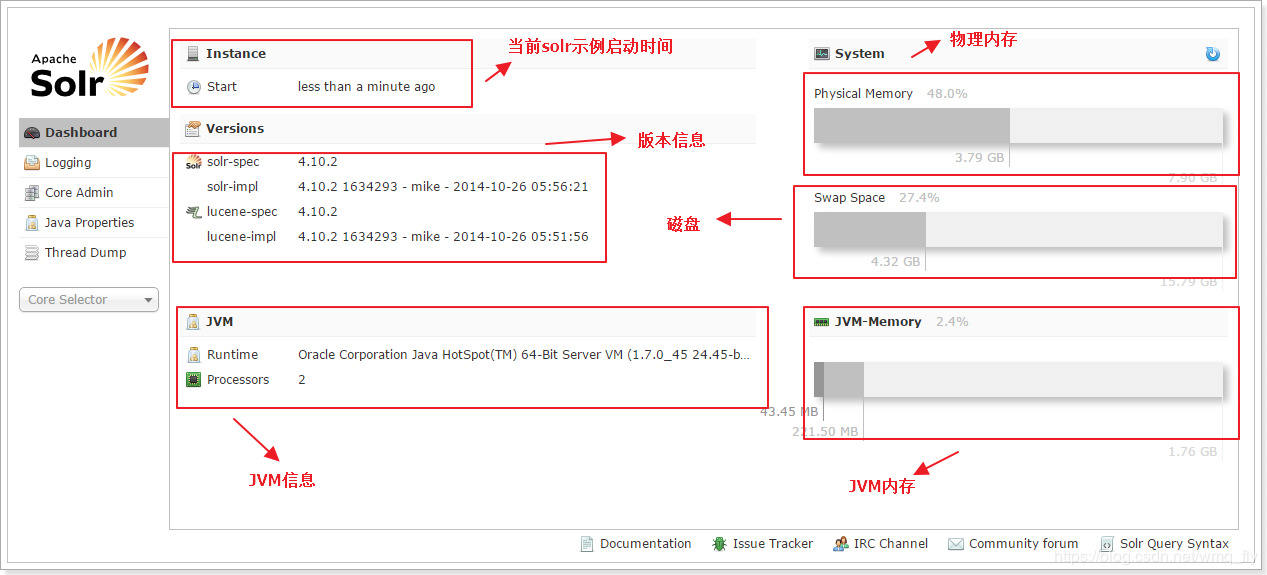


1. 进入cmd窗口，启动Solr,输入指令：solr start 启动Solr，以下为启动成功提示！



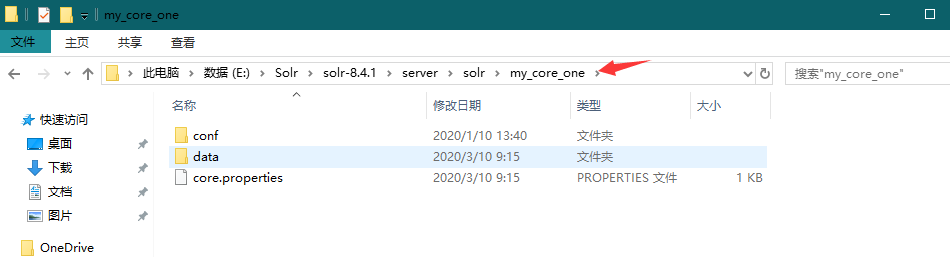
1. 在浏览器中输入<http://localhost:8983/solr> 进行访问solr





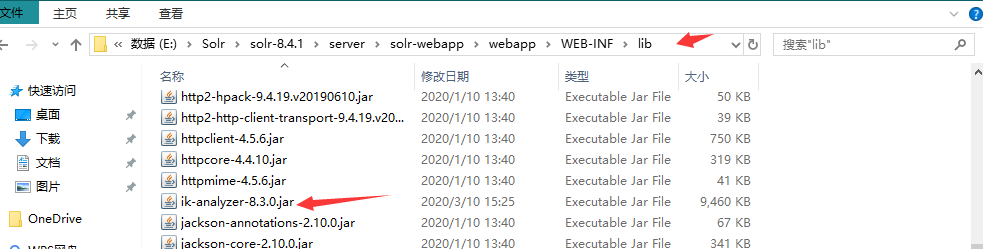
1. 添加core,core就相当于solr的一个项目实例，输入命令solr create -c my\_core\_one 注：my\_core\_one为实列的名称（该指令貌似只支持solr8.4.1版本，8.3.0不行会报错，待确定）

备注：<http://archive.apache.org/dist/lucene/solr/>（下载solr各个历史版本的地址）

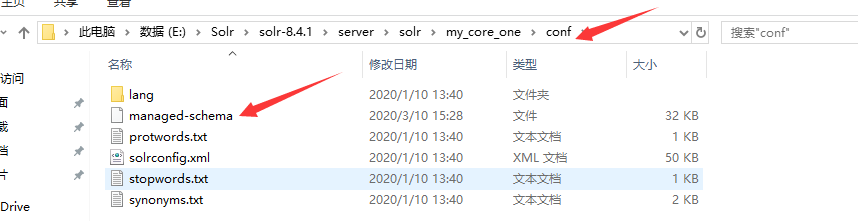


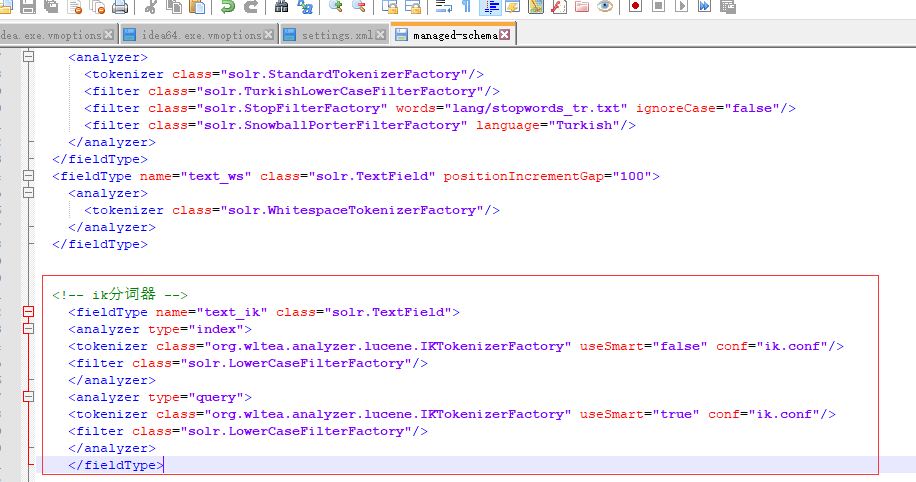
1. 配置中文分词器 IK-Analyzer-Solr8 下载地址：<https://search.maven.org/search?q=com.github.magese>

6.1将下载好的jar包放入solr-8.4.1\server\solr-webapp\webapp\WEB-INF\lib目录中



6.2然后到solr-8.4.1\server\solr\my\_core\_one\conf目录中打开managed-schema文件





<!-- ik分词器 -->

<fieldType name="text\_ik" class="solr.TextField">

<analyzer type="index">

<tokenizer class="org.wltea.analyzer.lucene.IKTokenizerFactory" useSmart="false" conf="ik.conf"/>

<filter class="solr.LowerCaseFilterFactory"/>

</analyzer>

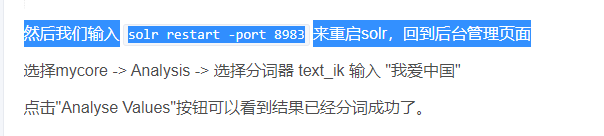
<analyzer type="query">

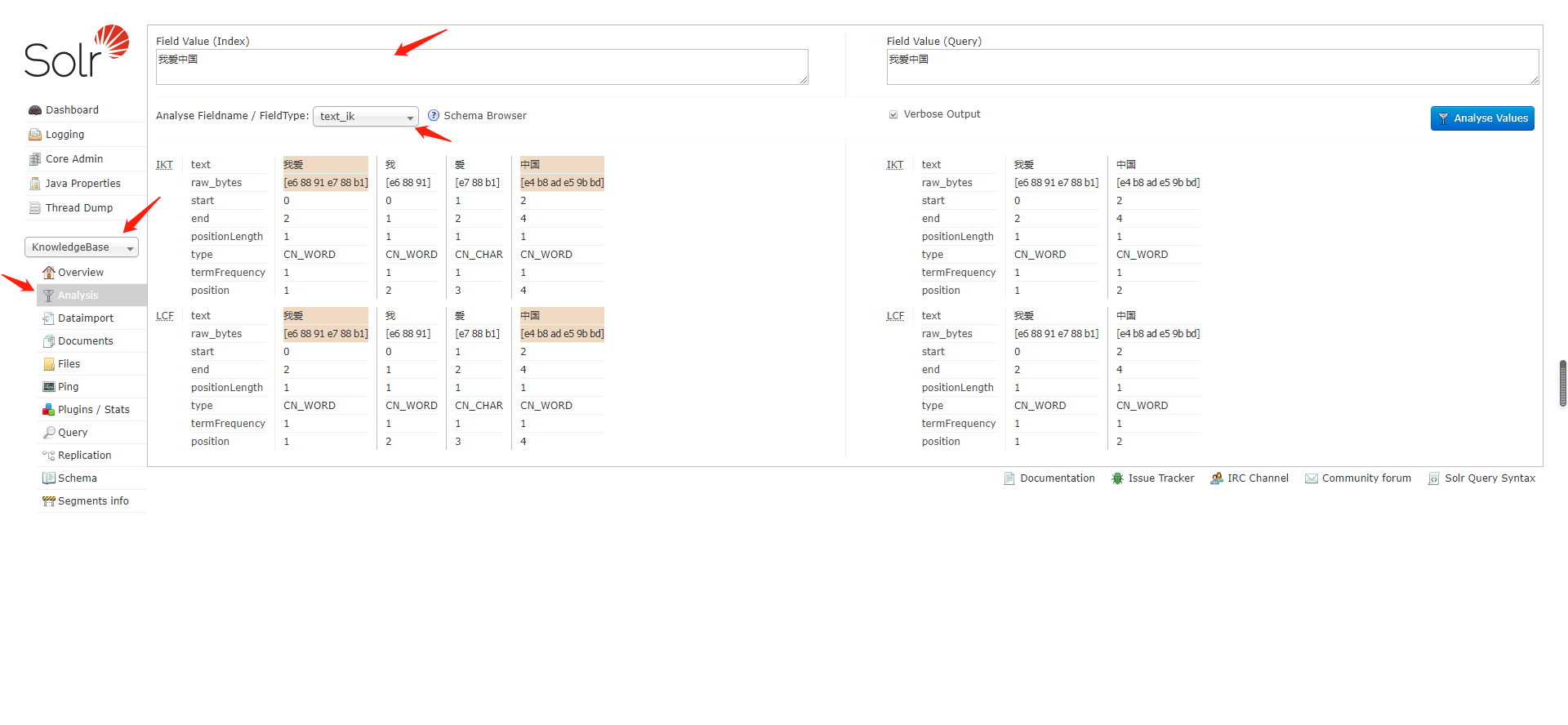
<tokenizer class="org.wltea.analyzer.lucene.IKTokenizerFactory" useSmart="true" conf="ik.conf"/>

<filter class="solr.LowerCaseFilterFactory"/>

</analyzer>

</fieldType>



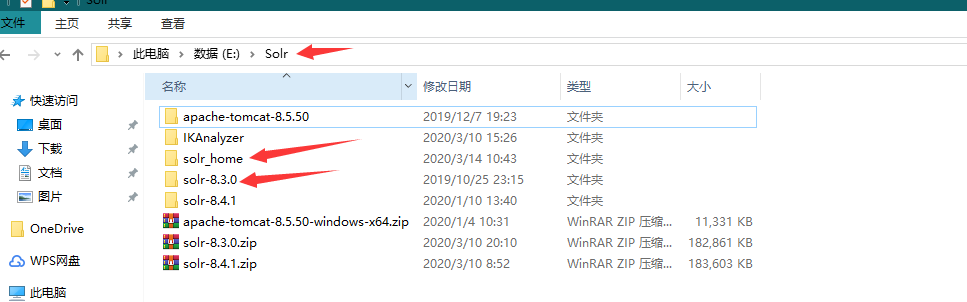




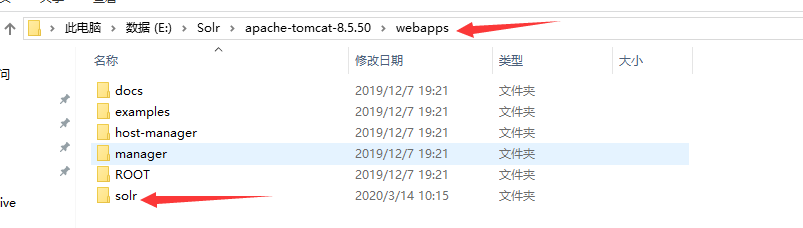


#### Tomcat启动solr配置

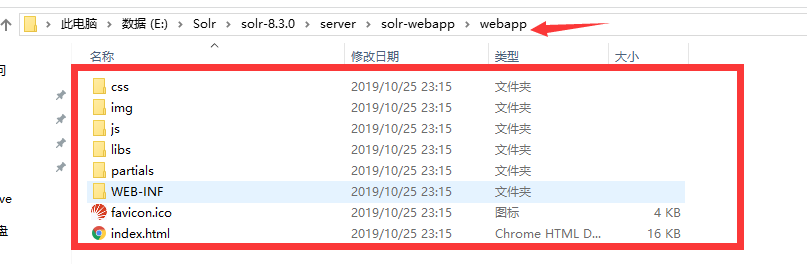
1. 在磁盘中创自己的solr\_home



1. 在部署solr的tomcat的E:\Solr\apache-tomcat-8.5.50\webapps目录下新增solr文件夹

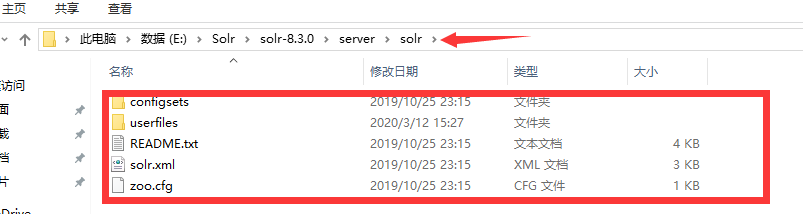


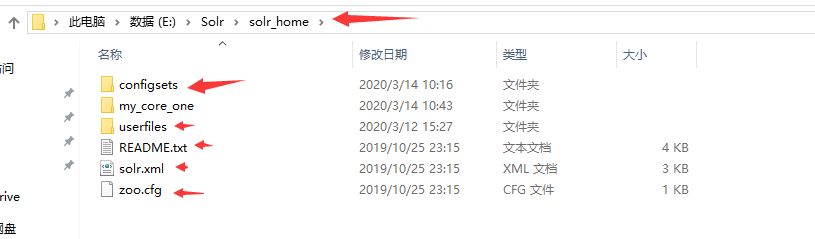
1. 将E:\Solr\solr-8.3.0\server\solr-webapp\webapp文件夹下的所有内容copy到我们的E:\Solr\apache-tomcat-8.5.50\webapps\solr 该目录下



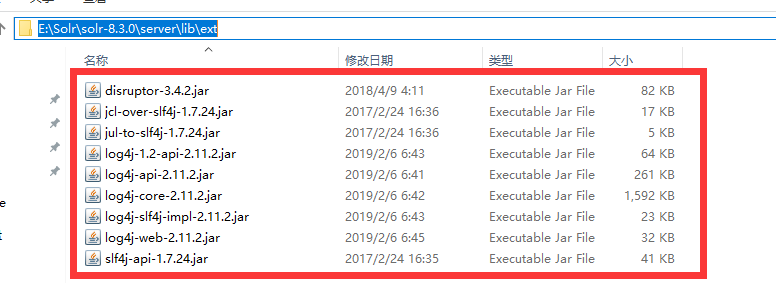


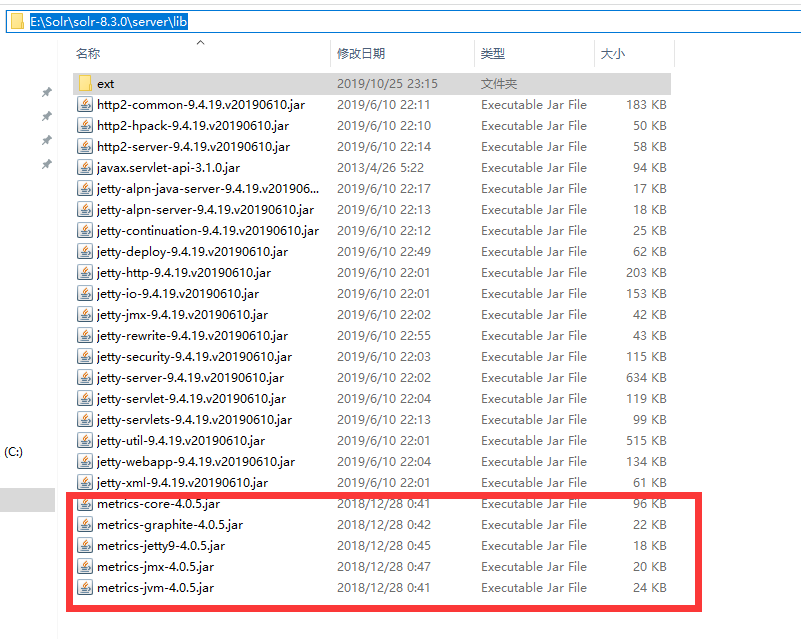
1. 将E:\Solr\solr-8.3.0\server\solr下的全部文件copy到solr\_home 中 该目录将会成为我们的solr data 和core的文件

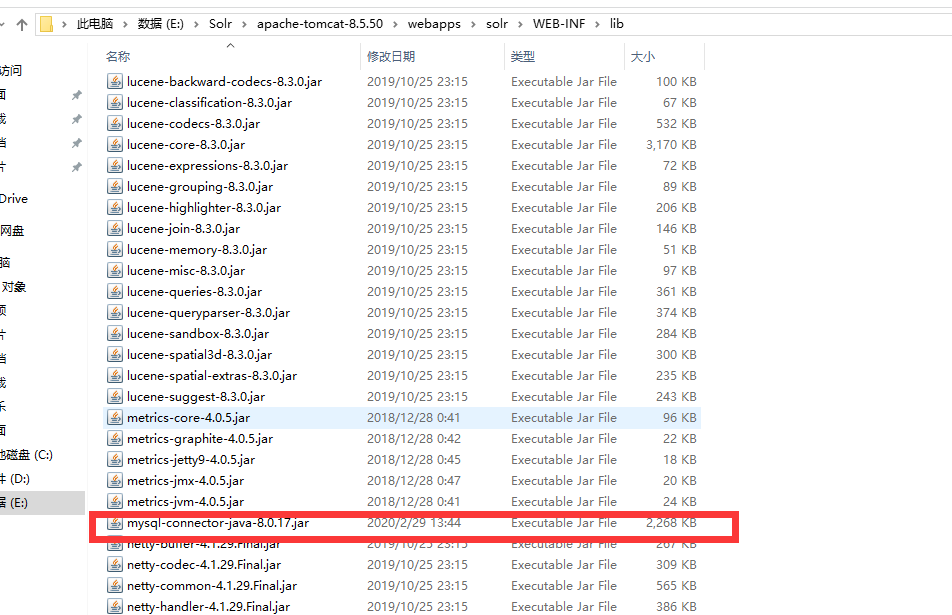




1. 将E:\Solr\solr-8.3.0\server\lib\ext的全部jar和E:\Solr\solr-8.3.0\server\lib 中的metrics-core等五个jar包拷贝到E:\Solr\apache-tomcat-8.5.50\webapps\solr\WEB-INF\lib下  如果不知道E:\testSolr\solr-8.0.0\server\lib 下的额jar包 可以全部拷贝过去就行了。（当然我们还可以在改目录下映入mysql的包 方便以后使用mysql ）







1. 一定要记住配置E:\Solr\apache-tomcat-8.5.50\webapps\solr\WEB-INF 中的文件和在该目录下添加2个配置文件
   1. 第一个配置web.xml文件（放在filter上）

<env-entry>

       <env-entry-name>solr/home</env-entry-name>

       <env-entry-value>E:/testSolr/solr\_home</env-entry-value>

       <env-entry-type>java.lang.String</env-entry-type>

</env-entry>



* 1. 添加 jetty-logging.properties  log4j.properties 2个文件内容

jetty-logging.properties 内容为：

org.eclipse.jetty.util.log.class=org.eclipse.jetty.util.log.Slf4jLog

log4j.properties 中的内容为：

# Logging level

solr.log=logs

#log4j.rootLogger=INFO, file, CONSOLE

log4j.rootLogger=INFO, file

#log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender

#log4j.appender.CONSOLE.layout=org.apache.log4j.EnhancedPatternLayout

#log4j.appender.CONSOLE.layout.ConversionPattern=%-4r %-5p (%t) [%X{collection} %X{shard} %X{replica} %X{core}] %c{1.} %m%n

#- size rotation with log cleanup.

log4j.appender.file=org.apache.log4j.RollingFileAppender

log4j.appender.file.MaxFileSize=4MB

log4j.appender.file.MaxBackupIndex=9

#- File to log to and log format

log4j.appender.file.File=${solr.log}/solr.log

log4j.appender.file.layout=org.apache.log4j.EnhancedPatternLayout

log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss.SSS} %-5p (%t) [%X{collection} %X{shard} %X{replica} %X{core}] %c{1.} %m%n

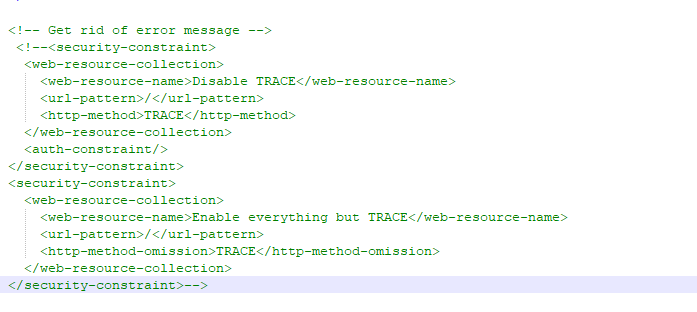
log4j.logger.org.apache.zookeeper=WARN

log4j.logger.org.apache.hadoop=WARN

# set to INFO to enable infostream log messages

log4j.logger.org.apache.solr.update.LoggingInfoStream=OFF

* 1. 最后一步注释掉该目录下中权限认证



1. 启动tomcat即可然后在访问localhost:8080/solr/index.html
2. 启动之后就是我们的具体创建solr的core了（一定要引入mysql那个链接的）

（1）.在E:\testSolr\solr\_home\下新建 自己的core的名字

（2）.将E:\testSolr\solr\_home\configsets\\_default\conf下的所有文件拷贝到E:\testSolr\solr\_home\下（还有一种方式将来再说）

(3).solr的图形页面中添加core即可 名字和刚才新增名字相同

新增前：

新增后：

（3）.这就是我们的意义上的一个库了.  接下来我们可以对呀进行添加字段链接库  可以是从页面中dataimport移入 也可以是从java文件中引入   这里不介绍 dataimport（具体无非就是改solrconfig.xml中引入dataimport  并且将dataimport  中的jar 引入然后从页面操作即可）.

创建完core之后我们可以从页面中引入如下图：

也可以从manager-shema.xml中直接引入

只需要加入我们的字段即可

3.接下来 其实solr的使用 基本应该没什么问题了补充一些solr的删除命令 以及在刚使用时候我们可以直接使用dataimport 和data-config.xml直接 导入数据库的数据

（1）.删除命令 比较常用：客户端

     documents type 选择 XML

    documents 输入下面语句

<delete><query>\*:\*</query></delete>

<commit/>

（2）.fq 和q 这两个我认为没什么说的 就是 字段名：值这种类型的 比如

start  和rows 无非就是分页 对于sort排序 支持desc等

（3）.关于dataimport  和dataconfig.xml有人想用可以来这里看看

首先配置在solrconfig.xml中引入dataconfig.xml

<requestHandler

name="/dataimport"class="org.apache.solr.handler.dataimport.DataImportHandler">

<lst name="defaults">

<str name="config">data-config.xml</str>

</lst>

</requestHandler>

并且 solrconfig.xml中配置   引入相关的dataimport

<lib dir="${solr.install.dir:../../../..}/contrib/dataimporthandler/lib" regex=".\*\.jar" />

<lib dir=".${solr.install.dir:../../../..}/dist/" regex="solr-dataimporthandler-.\*\.jar" />

在该E:\testSolr\solr\_home\testCore  下新增data-config.xml文件  内容如下（配置自己的数据库就行）

<?xml version="1.0" encoding="UTF-8" ?>

<dataConfig>

<dataSource type="JdbcDataSource"

driver="com.mysql.jdbc.Driver"

url="jdbc:mysql://192.168.100.246:3308/dtest1"

user="zkbc"

password="password"/>

<document>

<entity name="user" query="SELECT \* FROM answer\_userinfo ">

<field column="id" name="id"/>

<field column="userId" name="userId"/>

<field column="grade" name="grade"/>

</entity>

</document>

</dataConfig>

配置完毕之后   具体页面操作  重启我们solr的tomcat   页面就是下图了 点击 就可以了就能导入数据了

注：配置中文分词器同上面一样配置即可

#### Tomcat启动solr配置