**Solr全文索引方案**

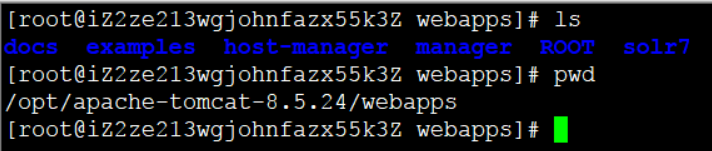
# solr 单服务器部署

1：部署服务前需先下载tomcat和solr

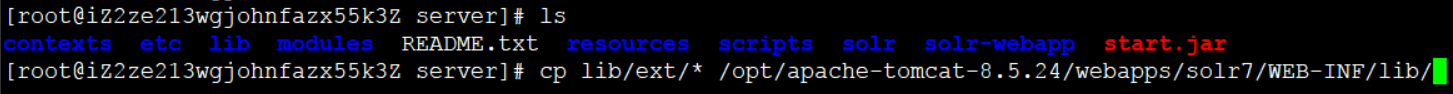
Tomcat下载 https://tomcat.apache.org/download-80.cgi

solr下载 <http://www.apache.org/dyn/closer.lua/lucene/solr/7.1.0>

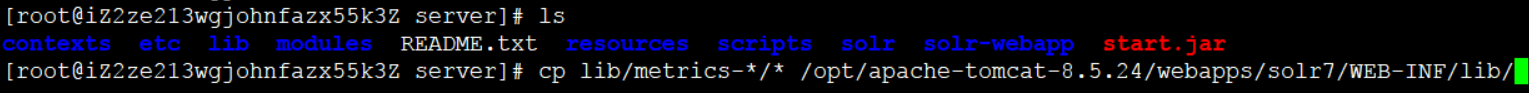
2：复制/usr/local/solr/solr-7.1.0中的server/solr-webapp/webapp下的webapp文件夹复制到tomcat/webapps下,并改个名字,比如改成solr7



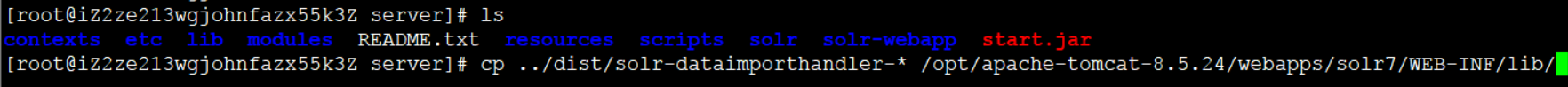
3：复制/opt/solr-7.1.0中的server/lib/ext文件夹中所有jar包到tomcat/webapps/solr7/WEB-INF/lib



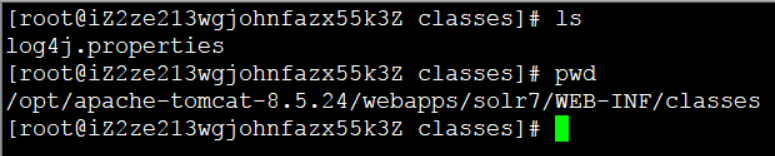
4：复制/usr/local/solr/solr-7.1.0中的server/lib/metrics开头的jar包到tomcat/webapps/solr7/WEB-INF/lib



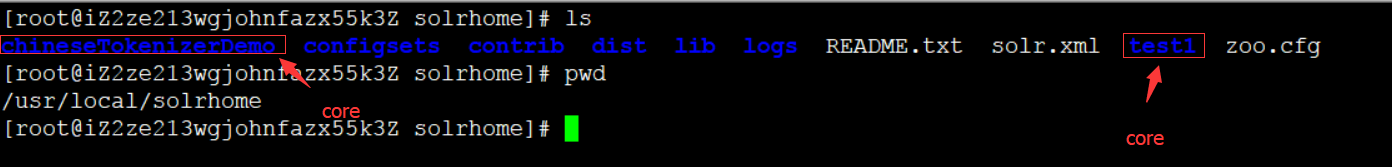
5：复制/usr/local/solr/solr-7.1.0中的dist/solr-dataimporthandler-开头的jar包到tomcat/webapps/solr7/WEB-INF/lib



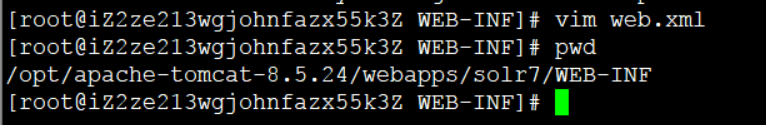
6：在solr7/WEB-INF下新建classes目录并将solr7.1.0文件夹中的server/resources/log4j.properties复制过来



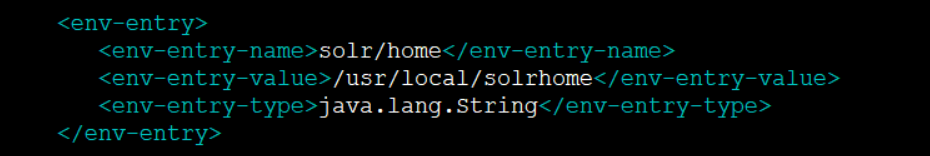
7：在usr/local下新建solrhome文件夹并将solr-7.0.1文件夹下的server/solr文件夹下所有的复制过去



8：编辑web.xml



1. 找到eny-entry标签,去掉注释,并将env-entry-value设置成刚刚新建的solrhome的地址



1. 找到security-constraint,并注释掉,有两个,在文件末尾.

9:启动tomcat,访问<http://IP:PORT/solr7/index.html>,正常访问表示项目启动成功。

参考：

http://blog.csdn.net/cslucifer/article/details/78551941

<http://blog.csdn.net/zjcjava/article/details/56845453>

## solr创建索引以及自动生成索引配置

1：mkdir 你的core名称mycore

2：复制

solr-7.1.0/server/solr/configsets/sample\_techproducts\_configs/conf

文件夹到mycore

3：vim core.properties，编辑 name = your core

4: mkdir data

5:更改solrconfig.xml中的下面这一段配置的路径为自己的solrhome

6：删除params.json

7：vim managed-schema ,在文件末尾加入想要插入的字段类型，type表示字段的数据类型，indexed表示是否自动生成索引。



参考:1:http://item1024.com/2017/09/22/solr/

2:http://www.cnblogs.com/zhangweizhong/p/5593909.html

3：<http://www.cnblogs.com/wander1129/p/6658828.html>

## Solr添加dataimport配置（参考：例子 example-DIH）

### 1）全量导入

1：在solrconfig配置文件中添加如下配置(注意位置一定要放在其他的requestHandle附近，不这样做会导致，页面无法显示)

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

<lst name="defaults">

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

</lst>

</requestHandler>

<lib dir="/usr/local/solrhome/dist/" regex="solr-dataimporthandler-.\*\.jar" />

2：在conf目录添加solr-data-config目录，并添加如下内容，并且修改为自己的配置

<dataConfig>

<dataSource type="JdbcDataSource" driver="com.mysql.jdbc.Driver" url="jdbc:mysql://127.0.0.1:3306/test1?useUnicode=true&amp;characterEncoding=UTF-8" user="test1" password="WOAINIljh223.." />

<document name="course">

<entity name="course" query="select id,name,description from course">

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

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

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

</entity>

</document>

</dataConfig>

将mysql的jar包加入到solr项目的lib目录

### 2）增量导入

**delta-import** : For incremental imports and change detection run the command http://<host>:<port>/solr/dataimport?command=delta-import . It supports the same clean, commit, optimize and debug parameters as full-import command.

增量导入solr-data-import如下所示配置

<dataConfig>

<dataSource type="JdbcDataSource" driver="com.mysql.jdbc.Driver" url="jdbc:mysql://127.0.0.1:3306/test1?useUnicode=true&amp;characterEncoding=UTF-8" user="test1" password="WOAINIljh223.." />

<document name="course">

<entity name="course" pk="id" query="select id,name,description from course"

deltaQuery="select id from course where last\_modified>'${dataimporter.last\_index\_time}'">

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

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

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

<field column="last\_modified" name="last\_modified"/>

</entity>

</document>

</dataConfig>

notice：

1. deltaQuery表示表中的last\_modified 大于 import最后执行的时间。
2. 配置完成后，需要在数据库导入到solr索引库中的表中加入last\_modified,并且在修改，新增和的时候修改last\_modified的时间，这个属性仅仅用于增量导入

3)对于全局变量的理解${dih.last\_index\_time} dih为固定值 表示的是DataImportHandler ${test1.last\_index\_time}的值存放在conf/dataimport.properties中

参考：<http://yiduwangkai.iteye.com/blog/2269602>

https://wiki.apache.org/solr/DataImportHandler

## IK分词器配置

1:将IK分词器 JAR 包拷贝到

solr7/server/solr\_webapp/webapp/WEB-INF/lib/

2:将词典 配置文件拷贝到 solr7/server/solr\_webapp/webapp/WEB-INF/

3: 更改test\_core\conf\managed-schema配置文件

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

<analyzer type="index">

<tokenizer class="org.apache.lucene.analysis.ik.IKTokenizerFactory" useSmart="true"/>

</analyzer>

<analyzer type="query">

<tokenizer class="org.apache.lucene.analysis.ik.IKTokenizerFactory" useSmart="true"/>

</analyzer>

</fieldType>

4：修改需要分词的field

<field name="name" type="text\_ik" indexed="true" stored="true"/>

### 扩展分词

将需要扩展的分词加入到ext.dic中即可，该配置文件在词典配置文件中.

# Solr实现水平搜索，多表之间建立索引

Solr多表之间建立索引需要借助dataimporthandler,首先在数据库中建立两张表，一张course表，一张target表。接下来修改solr-data-import.xml 配置文件，按照如下配置，将数据源等相关信息修改为自己的。

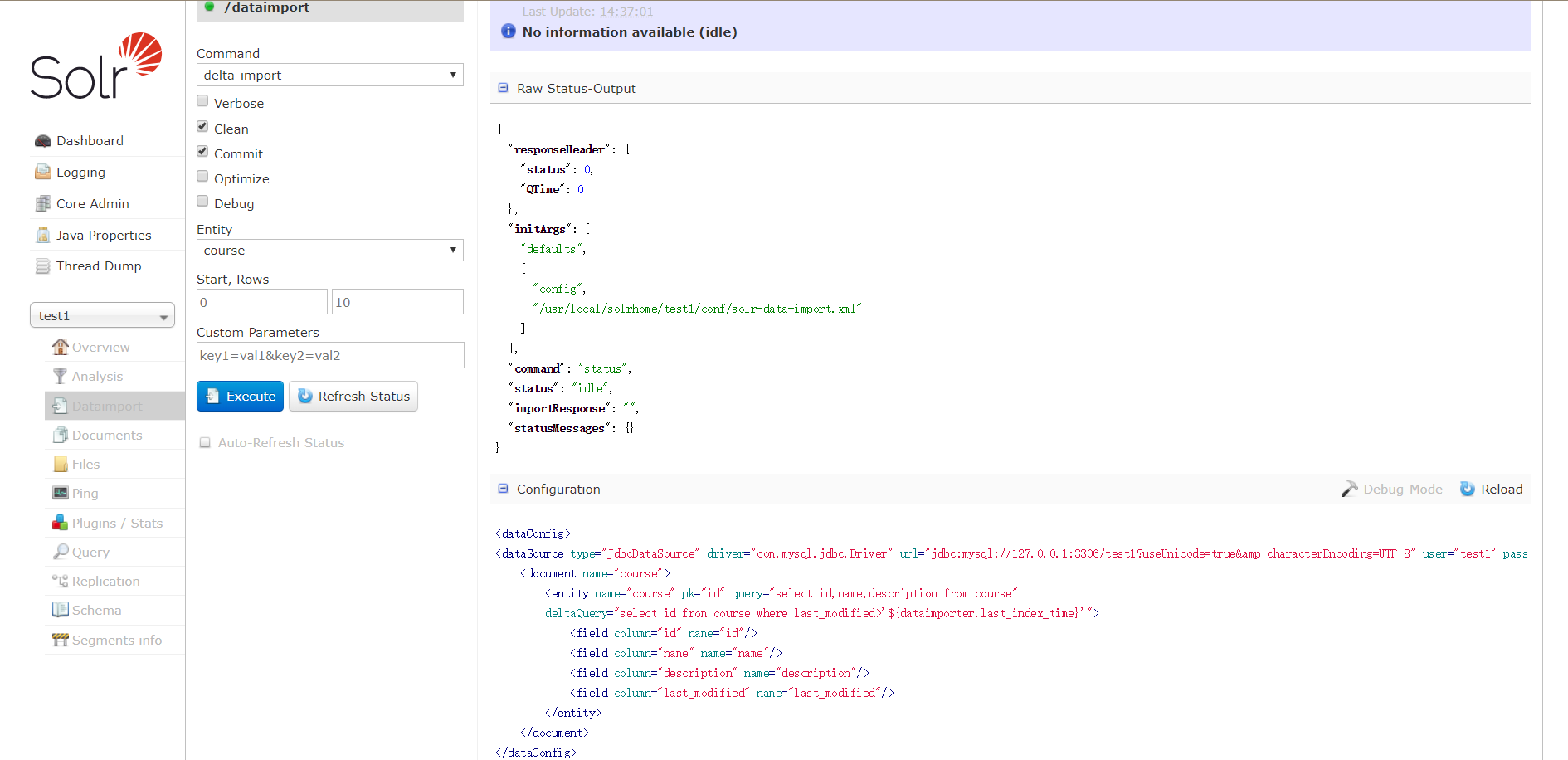


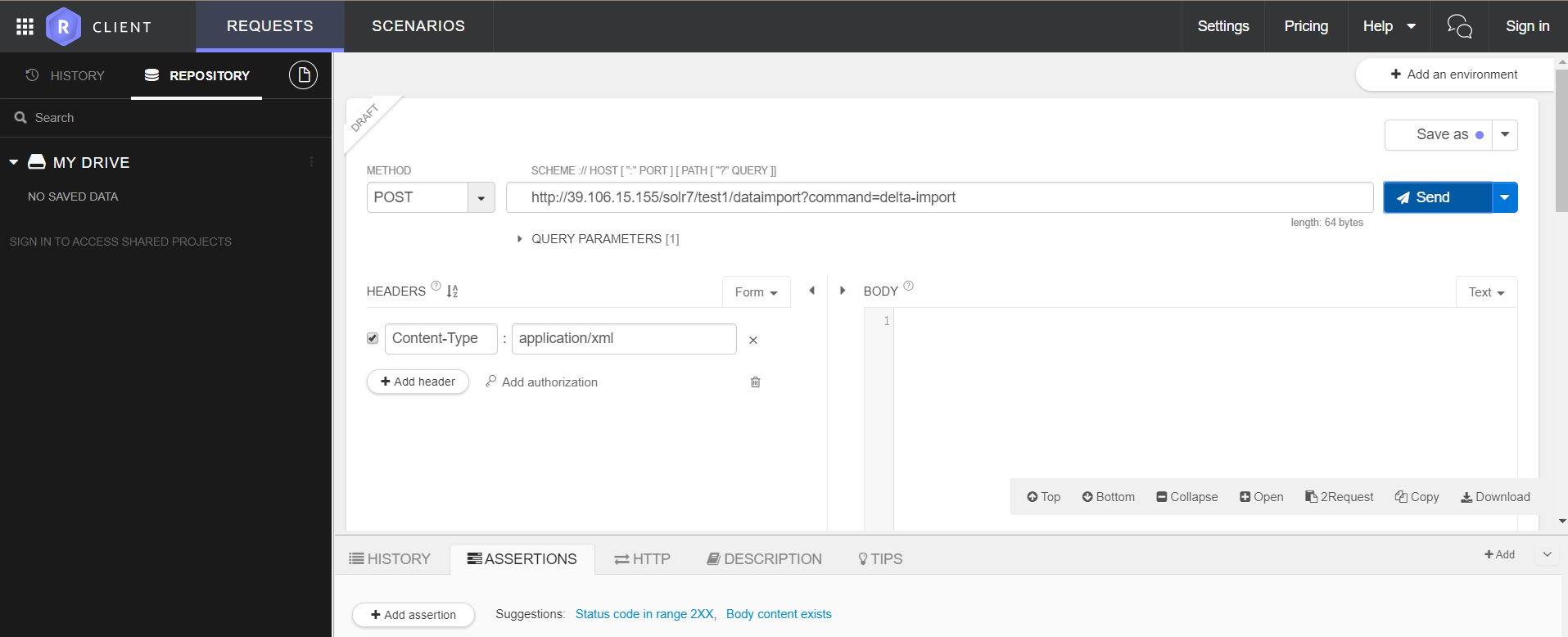
在<document></document>标签中间在添加一个<entity>标签即可,注意如下几点：

1. Entity的name用来区别不同的表,不能和其他的entity相同
2. Pk表示表中的主键
3. Course中的id和target中的id不能重复

一切配置完成后，

第一种方式：访问<http://localhost:8080/solr7/index.html>，选着你的core，点击dataimport，command 选着delta-import ，clean 去掉 entity 选着自己在solr-data-import中定义的实体，点击execute开始导入。

  
第二种方式，采用浏览器的rest工具发送GET请求



第三种方式，通过java代码发送GET请求。

<http://HOST:PORT/solr7/CORE>NAME /dataimport?command=delta-import

# Solr操作

删除索引文档

/opt/solr-7.1.0/bin/post -c chineseTokenizerDemo -url http://localhost/solr7/chineseTokenizerDemo/update -d '<delete><query>\*:\*</query></delete>'

# solr自带工具shell脚本

bin/post -url http://localhost/solr7/new\_core/update -c new\_core ~/innovaee/course.json 可以导入json ,xml 等格式的数据

notice: conf目录下面的params.json参数去掉。

参考url ： <http://lucene.apache.org/solr/guide/7_1/post-tool.html>

# Solr与应用之间集成

Solr与应用集成是运用solrj这个jar包实现的，我自己用solrj封装了一个小型插件solr-plugins，可以调用solr-plugins中的查询方法find,详细使用在文档注释中，接下来完成前端查询界面，把查询参数传入后台，调用solr-plugins中的find方法，返回查询的结果。

# Solr数据存储

1）Solr.xml指定了solr服务实例的配置，更多的配置请参考

<http://lucene.apache.org/solr/guide/7_1/solr-cores-and-solr-xml.html#solr-cores-and-solr-xml>

2）任意solr core

Core.properties 文件为每一个core定义了它的属性，例如他的名字等其他参数，更多的细节请参考

<http://lucene.apache.org/solr/guide/7_1/defining-core-properties.html#defining-core-properties>

3）solrconfig.xml包含了高级操作，比如为数据指定一个备用的地址，详细请参见

<http://lucene.apache.org/solr/guide/7_1/configuring-solrconfig-xml.html#configuring-solrconfig-xml>

4）managed-schema(或者也叫做schame.xml),描述了你要如何配置solr索引，这个文件定义了集合文档的字段field。你既可以定义字段的类型，名称，也可以定义字段是否可以被索引，比如index=true表示字段要自动生成索引,详细参见

<http://lucene.apache.org/solr/guide/7_1/documents-fields-and-schema-design.html#documents-fields-and-schema-design>

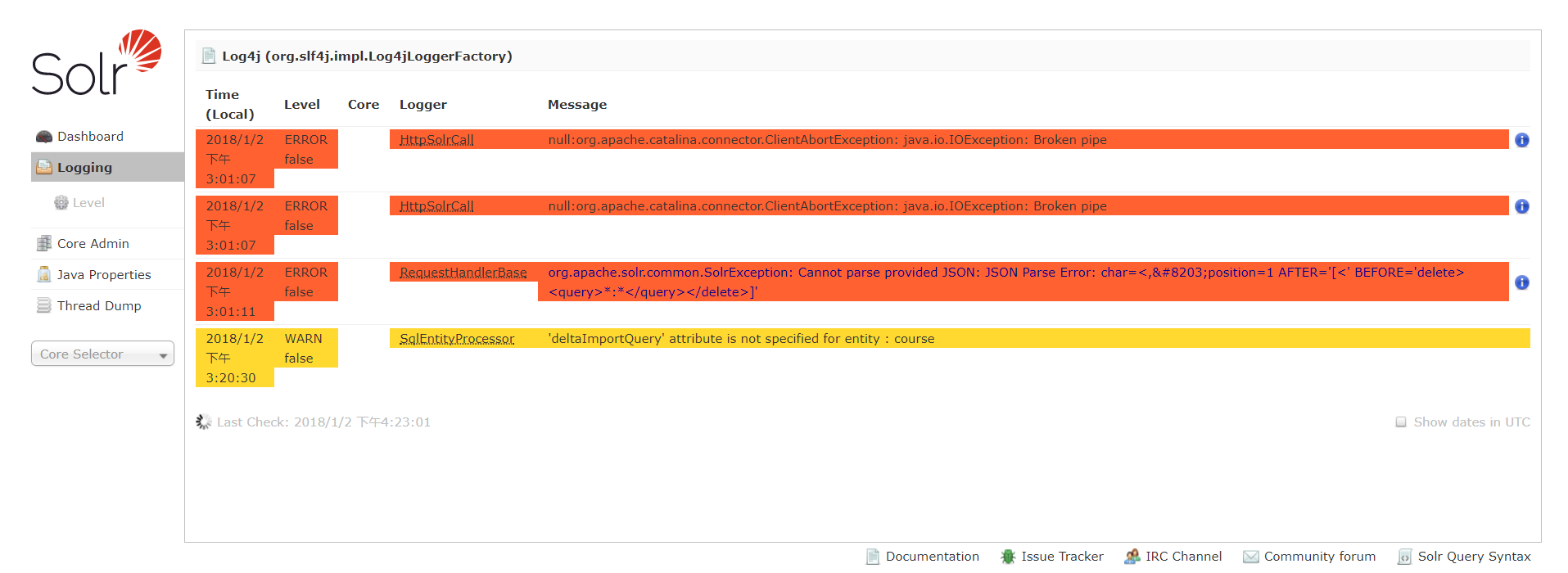
与

<http://lucene.apache.org/solr/guide/7_1/schema-api.html#schema-api>

5） Solr中的索引数据存储在core的data目录下，这个目录包含低级索引文件，(翻译也许不对，我看了下文件，这里的low我觉得应该是想表达，存储的索引文件是二进制的)。

# Solr监控

Solr自身提供了日志功能，可以检查报错，日志可视化界面在log按钮



但是solr目前好像并没有提供监控功能！

# 问和答

1：solr是否要单独部署服务器？

这个问题要看solr中索引的大小，按照我的理解，如何仅仅是把我们平台的内容在solr中生成索引，就没有必要单独部署了！