# 肖芳爱旅行项目总结

1. 项目名称：爱旅行后端系统
2. 开发时长：3周
3. 开发目的：针对爱旅行的人群设计的一款，预订机票酒店，以及了解旅游目的地的一款软件。
4. 开发环境：idea ,tomcat,redis,solr,swagger,jdk1.8,mysql…
5. 完成功能：注册，登录，酒店搜索，支付宝支付

开发详细总结

Solr总结

搭建solr环境

1.下载solr-4.9.1程序安装包 http://archive.apache.org/dist/lucene/solr/ 解压

2.将dist\solr-4.9.1.war文件复制到tomcat的webapps目录下，并命名为solr.war

依赖环境jdk-1.7 tomcat 1.7及以上

3. 复制solr解压包下的example\lib\ext下的所以jar包到tomcat的lib目录下

4.新建solr-home（可以是其他名字）,复制solr-4.9.1\example\solr所有

文件到solr-home（bin目录可删，可不删）

5.开启tomcat 在关闭tomcat(目的解压solr.war,关闭tomcat之后，修改solr.war的后缀solr.war.bak，

随便写后缀，一定先关在改)

6.修改配置文件tomcat\webapps\solr\WEB-INF\web.xml

<env-entry>

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

<env-entry-value>D:/software/solr\_home</env-entry-value>

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

</env-entry>

7.新建core

复制solr-home下的collection1

修改core.properties,name=你自己新建的core的名称

即可

删除core 关闭tomcat删除即可

core导入数据库数据

1.考连接驱动（mysql-connection.........）到tomcat的lib目录

2.配置数据源

新建core

打开core的conf文件solrconfig,xml文件，在

<requestHandler name="/select" class="solr.SearchHandler">

前面加一个dataimport的处理handler

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

<lst name="defaults">

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

</lst>

</requestHandler>

3.配置字段

在conf文件下新建data-config.xml文件

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

<dataConfig>

<dataSource type="JdbcDataSource" driver="com.mysql.jdbc.Driver"

url="jdbc:mysql://127.0.0.1:3306/itripdb"

user="root"

password="baoquan"/>

<document name="common">

<entity name="areaDic" pk="id" query="SELECT id。。。。。">

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

</entity>

</document>

</dataConfig>

自己修改，自己需要的

4.打开core文件conf目录的schema.xml文件

保留-version-这个field

删除其他field,删除dynamicField,保留<uniqueKey>id</uniqueKey>

删除copyField

添加索引字段，这里每个field的name要和data-config.xml里的entity

的filed的name一样，一一对应

例<field name="id" type="string" indexed="true" stored="true"/>

添加

<field name="text" type="text\_general" indexed="true" stored="false" multiValued="true" />

考solr解压包的dist目录下的两个jar包

。。。dataimporthandler......

考到tomcat/webapps/solr的lib目录下

4.执行导入数据

如果报错MA147LL/A，注释solr-home/conf/elevate.xml中的

<query text="ipod">

<doc id="MA147LL/A" /> <!-- put the actual ipod at the top -->

<doc id="IW-02" exclude="true" /> <!-- exclude this cable -->

</query>

5.测试数据导入结果

配置增量更新

1.导入增量更新jar包 apachesolrdataimportscheduler.jar

2.配置增量更新文件

在solr-home文件下新建conf文件夹，命名为dataimport.properties

修改里面的，syncEnabled=1,必须为1，是开启增量更新，不要加空格

syncCores=...同步那个core

时间间隔单位为分钟

3.增加增量更新监听器

复制到tomcat的webapps的solr,web.xml

<listener>

<listener-class>

org.apache.solr.handler.dataimport.sheduler.ApplicationListener

</listener-class>

</listener>

4.编写增量更新sql

在solr-home的core文件的conf目录下的data-config.xml

entity中

添加属性deltaImportQuery

deltaQuery

全文检索

分词器

Word分词器

Ansj分词器

IkAnalyzer分词器 项目中使用的 2012FF\_ui.jar

下载jar包 IK Analyzer 2012FF\_hf1 解压

复制IKAnalyzer.cfg.xml

stopword.dic

IKAnalyzer2012FF\_ui.jar

1.加jar包 2012FF\_ui.jar 考到solrtomcat\webapps\solr\WEB-INF\lib

2.配置过滤词语stopword.dic

solrtomcat\webapps\solr\WEB-INFO新建classes目录加入

IKAnalyzer.cfg.xml

stopword.dic

3.配置分词器

配置schema.xml

在solr-home\core\conf\shema.xml增加

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

<analyzer type="index" isMaxWordLength="false" class="org.wltea.analyzer.lucene.IKAnalyzer" />

<analyzer type="query" isMaxWordLength="true" class="org.wltea.analyzer.lucene.IKAnalyzer" />

</fieldType>

修改字段类型

将要分词的字段改为text\_ik类型

重启tomcat,访问Solr测试分词器

solrj集成步骤

1.创建solr查询接收对象

1.对应schema文件-创建Solr查询的接收对象

2.加Field注解

2.编写客户端程序

创建HttpSolrClient对象

创建SolrQuery对象

创建QueryResponse数据接收对象

接收数据，转化数据

3.测试

Redis总结

Redis在虚拟机上安装步骤

1. 下载redis压缩包
2. 解压redis tar -zxvf redis-3.2.8.tar.gz
3. 切换到redis安装目录 运行make
4. 启动/停止 Redis：

切换到redis安装目录启动：redis-server redis.conf

停止：kill -9 id

注释redis.conf中的bind 127.0.0.1

5开启redis客户端 redis-cli -h 端口号

Redis在本机上安装步骤

1. 下载压缩包，解压 ，切换到解压目录
2. 启动服务命令

redis-server redis.windows.conf

3.系统变量Path添加压缩包路径

4. 设置服务命令

redis-server --service-install redis.windows-service.conf --loglevel verbose

常用的redis服务命令。

卸载服务：redis-server --service-uninstall

开启服务：redis-server --service-start

停止服务：redis-server --service-stop

Redis常用命令

set（set key value）：

1.Redis 是以 key-value 的格式来存储数据的，set 命令就是设置 key 对应

2.的 value 值（string 类型），设置成功，返回 1；失败，返回 0。

3.get（get key）：通过 key 值获取其对应的 value 值，若不存在，则返回null。

4.exists（exists key） ：判断指定的 key 是否存在，存在返回 1，不存在返回 0。

5.del(key)：删除一个指定 key。

6.ttl(key) **查询key的有效期,当 key 不存在时，返回 -2 。 当 key 存在但没有设置剩余生存时间时，返回 -1 。 否则，以秒为单位，返回 key 的剩余生存时间***。*

Redis 的相关配置

1. JedisPoolConfig：Redis 连接池配置

首先进行 spring IoC 配置，在 spring 容器中注册一个 ID 为 jedisPoolConfig 的 bean，

使 用 的 类 是 redis.clients.jedis.JedisPoolConfig

1.maxActive：最大活动数。

若设置 value 为-1，则表示不限制，若 pool 已经分配了 n（设置的 value）个的

jedis 实例，则此时 pool 的状态为 exhausted(耗尽)。

2.maxIdle：最大空闲数。

3.maxWait：表示当 borrow(引入/借调)一个 jedis 实例时，最大的等待时间，如

果超过等待时间，则直接抛出 JedisConnectionException。

4.testOnBorrow：在 borrow 一个 jedis 实例时，是否提前进行 validate 操作；若

为 true，则得到的 jedis 实例均是可用的。

2. jedisPool

在 spring 容 器 中 注 册 一 个 ID 为 jedisPool 的 bean ， 使 用 的 类 是

redis.clients.jedis.JedisPool，再将配置好的redis的连接池配置对象（jedisPoolConfig）

通过 IoC 注入到连接池对象中，连接池对象初始化时引用上面的连接池配置对象，

同时通过配置的 IP 和端口等进行连接的实例化。

3.redisAPI

自己写一个redisAPI并注入到spring容器中，

在实例化这个 bean 的时候要给jedisPool 进行初始

化。

代码实现redisAPI

