初识solr:

<https://www.cnblogs.com/elvinle/p/8099482.html>

# **[solr(一) : 整合 tomcat](https://www.cnblogs.com/elvinle/p/8124776.html)**

<https://www.cnblogs.com/elvinle/p/8124776.html>

# **[solr(二) : 整合ik-analyzer](https://www.cnblogs.com/elvinle/p/8127875.html)**

<https://www.cnblogs.com/elvinle/p/8127875.html>

# **[solr(三) : 导入数据库表数据](https://www.cnblogs.com/elvinle/p/8137951.html)**

<https://www.cnblogs.com/elvinle/p/8137951.html>

# **[solr(四) : springboot 整合 solr](https://www.cnblogs.com/elvinle/p/8149256.html)**

<https://www.cnblogs.com/elvinle/p/8149256.html>

# **Solr配置同义词和拼音**

<https://dscq.iteye.com/blog/2383399>

# **Solr 配置一对多数据导入**

<https://blog.csdn.net/LDR1109/article/details/98955892>

# **[zookeeper集群](https://www.cnblogs.com/elvinle/p/8453743.html)**

<https://www.cnblogs.com/elvinle/p/8453743.html>

# **[solr(六): 集群](https://www.cnblogs.com/elvinle/p/8453771.html)**

<https://www.cnblogs.com/elvinle/p/8453771.html>

<tomcat 中的bin中的 catalina.bat 配置信息最后加上zookeeper 的配置地址>

set JAVA\_OPTS=%JAVA\_OPTS% -DzkHost=127.0.0.1:2181,127.0.0.1:2182,127.0.0.1:2183

zkcli.bat -zkhost 127.0.0.1:2181 127.0.0.1:2182 127.0.0.1:2183 -cmd upconfig -confdir /tool5m/solrtest/solrcloud2/solrhome1/core1/conf -confname solrConf

**[windows环境下solrcloud的配置安装(单机模拟多服务器)](https://www.cnblogs.com/ryanwangblog/p/3931484.html)**

**<https://www.cnblogs.com/ryanwangblog/p/3931484.html>**

Solr查询字段

<https://www.cnblogs.com/zhangweizhong/p/5056884.html>

# [Spring Boot 中使用 SolrCloud](https://segmentfault.com/a/1190000011619699)

<https://segmentfault.com/a/1190000011619699>

*<!-- solr -->*<**dependency**>  
 <**groupId**>org.springframework.data</**groupId**>  
 <**artifactId**>spring-data-solr</**artifactId**>  
 <**version**>2.1.1.RELEASE</**version**>  
</**dependency**>

*//slor test*

@Autowired  
**private** SolrClient **client**;

@RequestMapping(value=**"ha"**,method = RequestMethod.***POST***)  
**public** List<Users> userlist(@RequestBody Users users) **throws** IOException, SolrServerException {  
 List<Users> userslist=**new** ArrayList<>();  
 SolrQuery params = **new** SolrQuery();  
 **if**(users.getIntroduce()!=**""**){  
 params.set(**"q"**, users.getIntroduce());  
 }**else**{  
 params.set(**"q"**, **"\*:\*"**);  
 }  
 params.set(**"df"**, **"solr\_introduce"**);  
 params.set(**"fl"**, **"id,solr\_name,solr\_introduce"**);  
 params.addHighlightField(**"solr\_introduce"**); *// 高亮字段  
 //高亮  
 //打开开关* params.setHighlight(**true**);  
 *//设置前缀* params.setHighlightSimplePre(**"<span style='color:red'>"**);  
 *//设置后缀* params.setHighlightSimplePost(**"</span>"**);  
 QueryResponse queryResponse = **client**.query(**"core1"**,params);  
 SolrDocumentList results = queryResponse.getResults();  
 Map<String, Map<String, List<String>>> highlight = queryResponse.getHighlighting();  
 **int** i=0;  
 **for** (SolrDocument result : results) {  
 Users user=**new** Users();  
 String highname=**""**;  
 Map<String, List<String>> map = highlight.get(result.get(**"id"**));  
 List<String> list = map.get(**"solr\_introduce"**);  
 **if**(list==**null**){  
 highname=(String)result.get(**"solr\_introduce"**);  
 }**else**{  
 highname=list.get(i);  
 }  
 user.setId((String)result.get(**"id"**));  
 user.setName((String)result.get(**"solr\_name"**));  
 user.setIntroduce(highname);  
 userslist.add(user);  
 i++;  
 }  
 **return** userslist;  
}  
  
@RequestMapping(value=**"queryWbxUserById"**,method = RequestMethod.***GET***)  
Users queryWbxUserById(@RequestParam(value=**"id"**) String id) **throws** IOException, SolrServerException {  
 SolrDocument document = **client**.getById(**"core1"**, id);  
 Users user=**new** Users();  
 user.setId((String)document.get(**"id"**));  
 user.setName((String)document.get(**"solr\_name"**));  
 user.setIntroduce((String)document.get(**"solr\_introduce"**));  
 **return** user;  
}  
@RequestMapping(value=**"updateWbxUser"**,method = RequestMethod.***POST***)  
**void** updateWbxUser(@RequestBody Users user){  
 **try** {  
  
 SolrInputDocument doc = **new** SolrInputDocument();  
 doc.setField(**"id"**, user.getId());  
 doc.setField(**"solr\_name"**, user.getName());  
 */\* 如果spring.data.solr.host 里面配置到 core了, 那么这里就不需要传 core1 这个参数  
 \* 下面都是一样的  
 \*/* **client**.add(**"core1"**, doc);  
 *//client.commit();* **client**.commit(**"core1"**);  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
  
}  
@RequestMapping(value=**"saveWbxUser"**,method = RequestMethod.***POST***)  
**void** saveWbxUser(@RequestBody Users user){  
 **try** {  
 String uuid = UUID.*randomUUID*().toString().replaceAll(**"-"**, **""**);  
 SolrInputDocument doc = **new** SolrInputDocument();  
 doc.setField(**"id"**, uuid);  
 doc.setField(**"solr\_introduce"**, user.getIntroduce());  
 doc.setField(**"solr\_name"**, user.getName());  
 */\* 如果spring.data.solr.host 里面配置到 core了, 那么这里就不需要传 core1 这个参数  
 \* 下面都是一样的  
 \*/* **client**.add(**"core1"**, doc);  
 *//client.commit();* **client**.commit(**"core1"**);  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
}  
@RequestMapping(value=**"deleteWbxUser"**,method = RequestMethod.***GET***)  
**void** deleteWbxUser(@RequestParam(value=**"id"**) String id){  
 **try** {  
 **client**.deleteById(**"core1"**,id);  
 **client**.commit(**"core1"**);  
  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
  
}

*<!-- solrcloud -->*

**spring**:

**data**:  
 **solr**:  
 **repositories**:  
 **enabled**: **true  
 zk-host**: 127.0.0.1:2181,127.0.0.1:2182,127.0.0.1:2183

@Autowired  
**private** CloudSolrClient **cloudSolrClient**;