



链滴

Elasticsearch源码学习：1、环境搭建&am p;rest入口

作者：[liuyu](#)

原文链接：<https://ld246.com/article/1443023701862>

来源网站：[链滴](#)

许可协议：[署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>一、环境搭建</p>

<p>0、搭建好maven环境，装好eclipse及maven插件</p>

<p>1、从github下载源码：https://github.com/elastic/elasticsearch</p>

<p>2、可能会报错：Plugin execution not covered by lifecycle configuration
 则需要修改pom文件，将报错插件配置移到pluginManagement下：</p>

```
<pre class="brush: xml">&lt;build&gt;
```

```
<p>&lt;pluginManagement&gt;</p>
```

```
<p>&lt;plugins&gt;<br>
```

```
&lt;plugin&gt;</p>
```

```
<p>&lt;!--报错插件 --&gt;</p>
```

```
<p>&lt;/plugin&gt;<br>
```

```
&lt;/plugins&gt;<br>
```

```
&lt;/pluginManagement&gt;<br>
```

```
&lt;/build&gt;</pre><p></p>
```

<p>
如果错误节点在parent上，修改父项目的pom</p>

<p>3、一切正常后，运行运行
org.elasticsearch.bootstrap.Elasticsearch即可启动es。</p>

<p>(抛出异常：NoSuchFileException[target\eclipse_run\config],先不管它)</p>

<p>
访问http://localhost:9200就能看到熟悉的"you know, for search"了</p>

<p> </p>

<p>二、rest访问入口：Action</p>

<p>
先来看看es是怎么提供rest服务的吧。</p>

<p>es用Action类来处理rest请求，所有Action实现类在org.elasticsearch.rest.action.RestActionModule中注册自己负责处理的rest url：
</p>

```
<pre class="brush: java"> protected void configure() {
```

```
....
```

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> bind(RestMainAction.class).asEagerSingleton();
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> bind(RestNodesInfoAction.class).asEagerSingleton();
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> bind(RestNodeStatsAction.class).asEagerSingleton();
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> ....
```

```
</span></span></code></pre>
```

```
<p></p></pre><p></p>
```

```
<p><span>&nbsp;</span></p>
```

```
<p><span>&nbsp;</span></p>
```

```
<pre class="brush: java">public class RestSearchAction extends BaseRestHandler {
```

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> @Inject
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> public RestSearchAction(Settings settings, RestController controller, Client client) {
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> super(settings, controller, client);
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> controller.registerHandler(GET, "/"_search", this);
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> controller.registerHandler(POST, "/"_search", this);
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> controller.registerHandler(GET, "{index}/_search", this);
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> controller.registerHandler(POST, "{index}/_search", this);
```

```
</span></span></code></pre>
```

```
<p>...</p></pre><p></p>
```

当发送请求时，Action的parseSearchRequest进行解析，然后进入handleRequest处</p>

 </p>

所以，在想了解的Action服务类上打个断点，看看它是怎么工作的吧~~
<p>