

elasticsearch 插件编写: API 调用

作者: [flowaters](#)

原文链接: <https://ld246.com/article/1497943376586>

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>
 当我们需要扩容ES的功能时，就需要自己来开发插件。

</p>

<p>

</p>

<h2>
 最终的目录结构

</h2>

<p>
 一个开发好的插件的目录结构是：

</p>

<p>

</p>

```
class="prettyprint lang-js">plugins/  
├── abeffect  
│   ├── abeffect.jar  
│   └── plugin-descriptor.properties</pre>
```

<p>

</p>

<p>

</p>

<p>
 其中 plugin-descriptor.properties 中保存了运行的信息：如

</p>

<p>

</p>

<p>

</p>

```
class="prettyprint lang-js">description=abeffect for ElasticSearch  
version=1.0  
name=abeffect  
site=${elasticsearch.plugin.site}  
jvm=true  
classname=com.abeffect.es.plugin.MyPlugin  
java.version=1.8  
elasticsearch.version=5.4.0  
isolated=${elasticsearch.plugin.isolated}</pre>
```

<p>

</p>

<h2>
 加载过程

</h2>

<p>
 首先加载 plugin-descriptor.properties 中的 classname 类文件，这里为

</p>

```
class="prettyprint lang-js">com.abeffect.es.plugin.MyPlugin</pre>  
<pre class="prettyprint lang-js"></pre>
```

<p>

</p>
<h3>

MyPlugin 文件内容

</h3>

<p>

</p>

<pre class="prettyprint lang-js">

```
<pre class="prettyprint lang-js">import java.util.Collections;
import java.util.List;
import java.util.function.Supplier;
```

```
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.elasticsearch.cluster.metadata.IndexNameExpressionResolver;
import org.elasticsearch.cluster.node.DiscoveryNodes;
import org.elasticsearch.common.settings.ClusterSettings;
import org.elasticsearch.common.settings.IndexScopedSettings;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.settings.SettingsFilter;
import org.elasticsearch.plugins.ActionPlugin;
import org.elasticsearch.plugins.Plugin;
import org.elasticsearch.rest.RestController;
import org.elasticsearch.rest.RestController;
```

```
public class MyPlugin extends Plugin implements ActionPlugin {
    private final static Logger logger = LogManager.getLogger(MyPlugin.class);
```

```
    public MyPlugin() {
        super();
        logger.info("Load MyPlugin");
    }
```

```
    @Override
```

```
    public List<RestHandler> getRestHandlers(Settings settings, RestController restController,
```

```
        ClusterSettings clusterSettings, IndexScopedSettings indexScopedSettings, SettingsFilter
```

```
        settingsFilter, IndexNameExpressionResolver indexNameExpressionResolver, Supplier<DiscoveryNodes>
```

```
        nodesInCluster) {
        RestHandler handler = new MyRestAction(settings, restController);
        return Collections.singletonList(handler);
```

```
    }
```

```
}</pre>
```


</pre>

在 MyPlugin 中注册了 MyRestAction 实现，其示例内容为：

<p>


```
</p>
<p>
  <br />
</p>
<h3>
  MyRestController文件内容<br />
</h3>
<p>
  <br />
</p>
<pre class="prettyprint lang-js">
<p>
  <span>import</span> org.elasticsearch.action.admin.cluster.health.ClusterHealthRequest;
</p>

<p>
  <span>import</span> org.elasticsearch.action.admin.cluster.health.ClusterHealthResponse

</p>

<p>
  <span>import</span> org.elasticsearch.client.node.NodeClient;
</p>

<p>
  <span>import</span> org.elasticsearch.common.Table;
</p>

<p>
  <span>import</span> org.elasticsearch.common.settings.Settings;
</p>

<p>
  <span>import</span> org.elasticsearch.rest.RestController;
</p>

<p>
  <span>import</span> org.elasticsearch.rest.RestController;
</p>

<p>
  <span>import</span> org.elasticsearch.rest.RestRequest;
</p>

<p>
  <span>import</span> org.elasticsearch.rest.RestRequest.Method;
</p>

<p>
  <span>import</span> org.elasticsearch.rest.RestResponse;
</p>
```

```
<p>
  <span>import</span> org.elasticsearch.rest.action.RestResponseListener;
</p>
```

```
<p>
  <span>import</span> org.elasticsearch.rest.action.cat.AbstractCatAction;
</p>
```

```
<p>
  <span>import</span> org.elasticsearch.rest.action.cat.RestTable;
</p>
</pre>
```

```
<pre class="prettyprint lang-js">public class MyRestAction extends AbstractCatAction implements RestHandler {
    public MyRestAction(Settings settings, RestController restController) {
        super(settings);
        restController.registerHandler(Method.GET, "/_mycat/health", this);
    }
}
```

```
@Override
protected void documentation(StringBuilder sb) {
    sb.append("/_mycat/health\n");
}
```

```
@Override
public RestChannelConsumer doCatRequest(final RestRequest request, final NodeClient client) {
    ClusterHealthRequest clusterHealthRequest = new ClusterHealthRequest();

    return channel -> client.admin().cluster().health(clusterHealthRequest,
        new RestResponseListener<ClusterHealthResponse>(channel) {
            @Override
            public RestResponse buildResponse(final ClusterHealthResponse health) throws Exception {
                return RestTable.buildResponse(buildTable(health, request), channel);
            }
        });
}
```

```
@Override
protected Table getTableWithHeader(final RestRequest request) {
    Table t = new Table();
    t.startHeadersWithTimestamp();
    t.addCell("cluster", "alias:cl;desc:cluster name");
    t.addCell("status", "alias:st;desc:health status");
    t.addCell("node.total", "alias:nt,nodeTotal;text-align:right;desc:total number of nodes");
    t.addCell("node.data", "alias:nd,nodeData;text-align:right;desc:number of nodes that can store data");
    t.addCell("shards", "alias:t,sh,shards.total,shardsTotal;text-align:right;desc:total number of shards");
    t.addCell("pri", "alias:p,shards.primary,shardsPrimary;text-align:right;desc:number of primary shards");
}
```

```

        t.addCell("relo",
            "alias:r,shards.relocating,shardsRelocating;text-align:right;desc:number of relocating nodes");
        t.addCell("init",
            "alias:i,shards.initializing,shardsInitializing;text-align:right;desc:number of initializing nodes");
        t.addCell("unassign",
            "alias:u,shards.unassigned,shardsUnassigned;text-align:right;desc:number of unassigned shards");
        t.addCell("pending_tasks", "alias:pt,pendingTasks;text-align:right;desc:number of pending tasks");
        t.addCell("max_task_wait_time",
            "alias:mtwt,maxTaskWaitTime;text-align:right;desc:wait time of longest task pending");
        t.addCell("active_shards_percent",
            "alias:asp,activeShardsPercent;text-align:right;desc:active number of shards in percent");
    ;
    t.endHeaders();

    return t;
}

```

```

private Table buildTable(final ClusterHealthResponse health, final RestRequest request) {
    Table t = getTableWithHeader(request);
    t.startRow();
    t.addCell(health.getClusterName());
    t.addCell(health.getStatus().name().toLowerCase(Locale.ROOT));
    t.addCell(health.getNumberOfNodes());
    t.addCell(health.getNumberOfDataNodes());
    t.addCell(health.getActiveShards());
    t.addCell(health.getActivePrimaryShards());
    t.addCell(health.getRelocatingShards());
    t.addCell(health.getInitializingShards());
    t.addCell(health.getUnassignedShards());
    t.addCell(health.getNumberOfPendingTasks());
    t.addCell(health.getTaskMaxWaitingTime().millis() == 0 ? "-" : health.getTaskMaxWaitingTime());
    t.addCell(String.format(Locale.ROOT, "%1.1f%%", health.getActiveShardsPercent()));
    t.endRow();
    return t;
}

```

}</pre>

```

<p>
  <br />
</p>
<p>
  <br />
</p>
<p>
  <br />
</p>

```

```

<pre class="prettyprint lang-js">MyRestAction的内容参考了阅读资料2.
</pre>

```

```
<p>
  <br />
</p>
<p>
  <br />
</p>
<h2>
  打包过程
</h2>
<p>
  mvn package
</p>
<p>
  得到 abeffect.jar 文件
</p>
<p>
  <br />
</p>
<h2>
  运行过程和结果<br />
</h2>
<h4>
  启动
</h4>
<p>
  <br />
</p>
<pre class="prettyprint lang-js">[2017-06-20T15:20:09,719][INFO ][c.u.e.p.MyPlugin      ] Lo
d MyPlugin
[2017-06-20T15:20:09,728][INFO ][o.e.p.PluginsService ] [MwHlebl] loaded module [transpo
t-netty4]
[2017-06-20T15:20:09,729][INFO ][o.e.p.PluginsService ] [MwHlebl] loaded plugin [abeffect]
[2017-06-20T15:20:12,940][INFO ][o.e.d.DiscoveryModule ] [MwHlebl] using discovery type
zen]
[2017-06-20T15:20:14,292][INFO ][o.e.n.Node          ] initialized
[2017-06-20T15:20:14,301][INFO ][o.e.n.Node          ] [MwHlebl] starting ...</pre>
<p>
  <br />
</p>
<h4>
  调用API示例
</h4>
<p>
  <br />
</p>
<pre class="prettyprint lang-js">curl -s "localhost:9200/_mycat/health?v"
epoch   timestamp cluster   status node.total node.data shards pri relo init unassign pend
ng_tasks max_task_wait_time active_shards_percent
1497943235 15:20:35 elasticsearch green      1      1  0 0 0 0  0      0
-          100.0%</pre>
<p>
  <br />
</p>
<h2>
```

阅读资料

</h2>

<p>

</p>

<p>

1. ElasticSearch5.3插件开发（一）控制台打印信息

</p>

<p>

2. ElasticSearch5.3插件开发（二）获取集群健康信息

</p>

<p>

3. Creating a Plugin for Elasticsearch 5.0 Using Maven (Updated for GA) 推荐阅读，比本文写的清晰，使用Maven打包

</p>

<p>

4. Creating an Ingest Plugin for Elasticsearch (Updated for GA) 推荐阅读，比本文写的清晰，使用Gradle打包

</p>

<p>

5. https://github.com/jolicode/emoji-search/tree/master/esplugin 一个es plugin的例子, 可以参考

</p>

<p>

6. Building a plugin with Gradle and Elasticsearch 5.0

</p>

<p>

7. plugin gradle

</p>

<p>

</p>

<h2>

附录

</h2>

<p>

1. elasticsearch5.0.0分配的变化Http协议和REST接口的变化

</p>

<p>

</p>

</p>