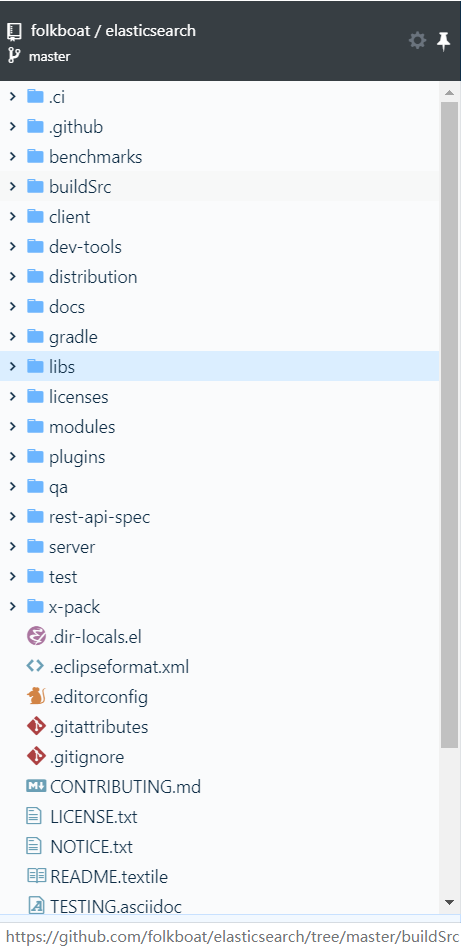
**Development view**

This view highlights the concerns and interests of the developers and testers of the project. In this view, we’ll focus on the main modules of Elasticsearch and the relations between them.

**Modules:**

Following is the picture of the whole source file structure of Elasticsearch (Figure 1). First we exclude all those folders about documents, licenses, source codes building, testing, benchmarking, some plugins and reusable codes. Now only two folders are left: client, server. They are the core of Elasticsearch. We’ll discuss them separately first, and move to the connections of them.



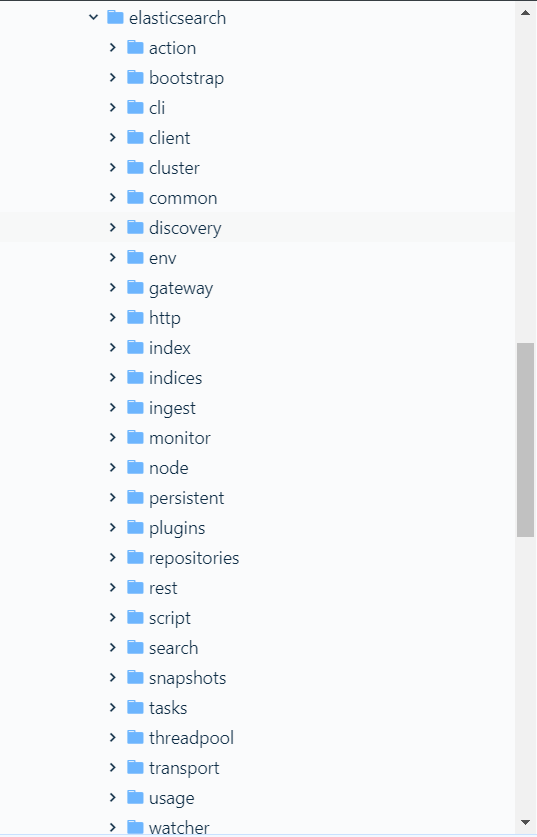
(Figure 1)

**Server:**

There are three outermost folders in this module: apache, elasticsearch, and joda (Figure 2). All sub packages and java classes are organized in them. In the apache folder, it contains the classes acted as the communicators between Elasticsearch and its underlying project – apache lucene. The joda folder contains nothing but a java class about time format. Then all the major packages are in the elasticsearch folder (Figure 3), so it is also the place we spend most time on, and figuring out the components in this folder and connections among them, you can get most of the major knowledge of the server module.

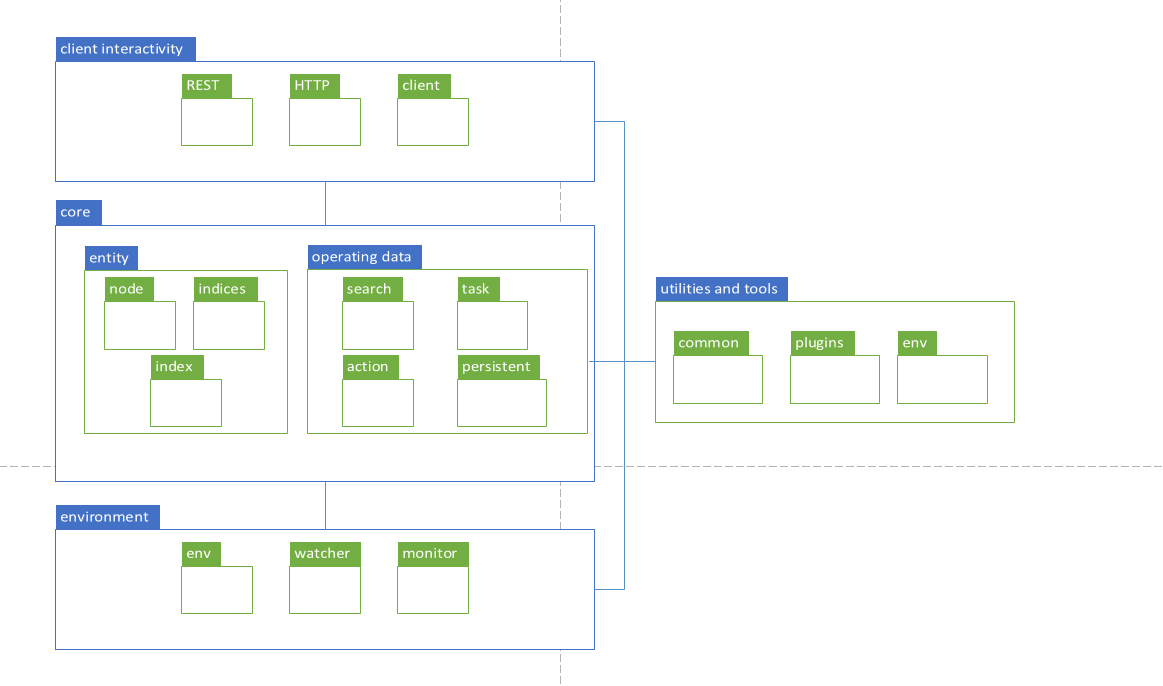


(Figure 2)



(Figure 3)

To have a more clear view of the content structure of the elasticsearch folder (also the main structure of server module), we draw the following diagram (Figure 4).



(Figure 4)

We divide the server module into four parts: client interactivity, core, environment, utilities and tools.

The client interactivity part is responsible for the communication between client and server. It provides the RESTful API and other interfaces for client to interactive with.

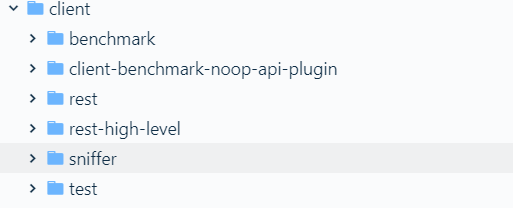
Then comes the core components of this module. It contains two sub parts: entity and operating data.

Next, the environment part includes elements that communicate with the platform environment.

And at last, the utilities and tools part provides common codes for other parts to use.

**Client**

Here’s the components layout of Client module. (Figure 5)



(Figure 5)

The main packages are rest, rest-high-level, sniffer (Figure 6). The rest package constructs REST requests and sends them to the server. To simplify the generation of REST requests, the client module provides a rest-high-level package, which wraps the lower-level rest package.



(Figure 6)