The Essentials

With Logsene, we expose the Elasticsearch API so you can:

- send log events through it directly from your application, using any Elasticsearch library
- send log events by using existing application such as Logstash, or Apache Flume, or Fluentd Elasticsearch plugin, or anything that can output to Elasticsearch. Or you can implement your own "log shipper".
- search for logs from your own application, or by configuring/adapting existing Elasticsearch UIs, such as Kibana
- optionally define custom mappings for your log types, so you can tweak the way your logs are indexed

When you use the API, here are the things you need to know:

- host name: **logsene-receiver.sematext.com**
- port: **80** or **443** (depending on whether you want to use plain HTTP or HTTPS)**
  **
- index name: your Logsene application token - note: **this token should be kept secret** (n.b. you can have N Logsene Apps, each with its own token)

Indexing

With the same REST API, you can index logs directly from your application, or you can craft your own "log sender".

NOTE:
If you are sending logs from your application use the Elasticsearch HTTP API. If you are sending logs from a Java application use a library like log4j2-elasticsearch-http or Jestinstead of Elasticsearch TransportClient.

Besides specifying your Logsene app token as the index name, it's nice to have a field named "@timestamp". Its value should be a valid ISO 8601 timestamp. This will be used for searching and sorting when/if you use Kibana with Logsene. If you don't provide a timestamp, Logsene will add one when it receives your message.

For example, you can send a log like this:

```
NOW=`date "+%Y-%m-%dT%H:%M:%S"`
curl -XPOST https://logsene-receiver.sematext.com/$YOUR_TOKEN_HERE/mytype/ -d '
{
  "@timestamp": "'$NOW'",
  "message": "Hello World!"
}'
```

This will index a simple "hello world" message to Logsene. That event would have the current timestamp and will go to your Logsene app (provided that the $YOUR_TOKEN_HERE variable contains your token), within a type named "mytype". The type is a logical division of events. Typically, you'd put events with different structures in different types. For example, syslog messages in a type called "syslog", apache logs in a type called "apache". Essentially, the type can be anything, it's the token of your application that has to match.

For performance reasons, we highly recommend using the Bulk API, because it allows you to send multiple events with a single request. For example, the following request sends three events:

```
NOW=`date "+%Y-%m-%dT%H:%M:%S"`

echo '{ "index" : { "_index": "LOGSENE_APP_TOKEN_GOES_HERE", "_type" : "mytype" } }
{ "@timestamp": "'$NOW'", "severity_numeric" : 1 }
{ "index" : { "_index": "LOGSENE_APP_TOKEN_GOES_HERE", "_type" : "mytype" } }
{ "@timestamp": "'$NOW'", "message" : "hello again" }
{ "index" : { "_index": "LOGSENE_APP_TOKEN_GOES_HERE", "_type" : "mytype" } }
{ "@timestamp": "'$NOW'", "alternate_message": "fields can be added and removed at will" }' > re

curl -XPOST https://logsene-receiver.sematext.com/_bulk --data-binary @req; echo
```

Default Log Index Mapping

A mapping is a way to define how your logs are indexed - which fields are in each log event and how each field is indexed. Logsene provides a default mapping that works well for most use-cases:

- the **@timestamp** field is an ISO 8601 date
- the **geoip** field is an object that contains a **location** geo point field (this works well if you're using Logstash)
- the predefined fields **host**, **facility**, **severity**, **syslog-tag**, **source** and **tags** **** are not analyzed, which enables only exact matches (you can still use wildcards, for example to search for **web-server*** and get **web-server01**)
- all string fields are analyzed by whitespace and lowercased by default, enabling a search for **message:hello** to match an event with **Hello World** in the **message** field

Custom Log Index Mapping

If the default log index fields (also known as index mapping) don't fit your needs you can create completely custom index mapping. See Custom Logsene Mapping Template How-To. Note that if you have N different log structures,

the best way to handle that is by creating N Logsene Apps, each with its own index mapping. For example, you may have web server logs, your system logs in /var/log/messages, and your custom application logs. Each of these 3 types of logs has a different structure. The web server logs probably use Apache Common Log format, the logs in /var/log/messages have syslog structure, and your own application's logs can be in any format your application happens to use. To handle all 3 log formats elegantly simply create 3 separate Logsene Apps and use a different format for each of them. See Custom Logsene Mapping Template How-To for details.