Elasticsearch乐观锁并发控制(optimistic concurrency control)

Elasticsearch是一个分布式系统。当documents被创建、更新或者删除，其新版本会被复制到集群的其它节点。Elasticsearch既是异步的(asynchronous)也是同步的(concurrent)，其含义是复制请求都是并行发送的，但是到达目的地的顺序是无序的。Elasticsearch系统需要一种方法使得老版本的文档永远都无法覆盖新的版本。

每当文档被改变的时候，文档中的_version将会被增加（+1）。Elasticsearch使用_version确保所有的修改都会按照正确的顺序执行。如果文档旧的版本在新的版本之后到达，它会被简单的忽略。

我们可以充分利用_version这个特点确保数据不会因为修改冲突而丢失，我们可以指定文档的version来做想要的更改。如果那个版本号不是现在的，我们的请求就失败了。让我们创建一个新的blog post：

```
PUT /website/blog/1/_create
{
  "title": "My first blog entry",
  "text":  "Just trying this out..."
}
```

从响应的内容可以看出这个新创建文档的_version为1。现在假如我们需要修改这个文档：我们将这个文档的数据加载到web表单中，然后我们对其进行修改，然后保存这份数据。首先我们来检索这份文档：

```
GET /website/blog/1
```

响应内容包含一个_version的内容

```
{
  "_index" :   "website",
  "_type" :    "blog",
  "_id" :      "1",
  "_version" : 1,
  "found" :    true,
  "_source" :  {
```

```
    "title": "My first blog entry",
    "text":  "Just trying this out..."
  }
}
```

现在我们将上面web修改的内容进行保存，并重新索引(reindexing)，我们在请求的URL中加入了version使得我们的修改被应用：

```
PUT /website/blog/1?version=1
{
  "title": "My first blog entry",
  "text":  "Starting to get the hang of this..."
}
```

我们只希望文档的_version为1的时候更新才生效。
当然，本次的请求会成功，请求返回的内容告诉我们_version已经变成2了，如下所示：

```
{
  "_index":   "website",
  "_type":    "blog",
  "_id":      "1",
  "_version": 2
  "created":  false
}
```

然而，当我们重新运行上面的修改请求，也就是version=1，这时候Elasticsearch将会返回409 Conflict状态的HTTP响应码，响应内容如下：

```
{
   "error": {
      "root_cause": [
         {
            "type": "version_conflict_engine_exception",
            "reason": "[blog][1]: version conflict, current [2], provided [1]",
            "index": "website",
            "shard": "3"
         }
```

```
    ],
    "type": "version_conflict_engine_exception",
    "reason": "[blog][1]: version conflict, current [2], provided [1]",
    "index": "website",
    "shard": "3"
  },
  "status": 409
}
```


上面的相应内容告诉我们当前Elasticsearch中该文档的_version是2，但是我们指定想要更新的版本是1。

Elasticsearch对409响应码的解释是：

> The request could not be completed due to a conflict with the current state of the resource. This code is
> only allowed in situations where it is expected that the user might be able to resolve the conflict and
> resubmit the request. The response body SHOULD include enough information for the user to recognize the
> source of the conflict. Ideally, the response entity would include enough information for the user or user
> agent to fix the problem; however, that might not be possible and is not required.
>
> Conflicts are most likely to occur in response to a PUT request. For example, if versioning were being
> used and the entity being PUT included changes to a resource which conflict with those made by an earlier
> (third-party) request, the server might use the 409 response to indicate that it can't complete the request.
> In this case, the response entity would likely contain a list of the differences between the two versions in
> a format defined by the response Content-Type.

我们需要做什么取决于程序的需求。我们可以告知用户其他人修改了文档，你应该在保存前再看一下。

所有更新和删除文档的请求都接受version参数，它可以允许在你的代码中增加乐观锁控制(optimistic concurrency control)。

本文翻译自：https://www.elastic.co/guide/en/elasticsearch/guide/current/optimistic-concurrency-control.html