

在ElasticSearch中使用外部系统的版本

有一种非常常见的场景那就是使用其他数据库作为主要的数据存储，而Elasticsearch用来检索数据。这也意味着主数据库发生的一切变更都需要将其拷贝到Elasticsearch中。如果这时候有多个进程负责数据的同步，就会遇到[《Elasticsearch乐观锁并发控制\(optimistic concurrency control\)》](#)文章中提到的并发问题。

如果你的主数据库已经有了版本号码（Version numbers），或者诸如timestamp字段可以用于作为版本号码，这时候你可以在Elasticsearch中通过增加version_type=external来重新利用这些版本号码。版本号码必须是大于0并且小于9.2e+18的整数。

Elasticsearch处理外部的版本号码和处理内部版本号码的方式是不一样的。它不检测当前_version是否和请求参数中的版本一致；而是检测当前_version是否小于指定的版本。如果请求成功，那么外部的版本号就会被存储到文档中的_version中。

外部版本号不仅可以在索引(index)和删除(delete)请求时使用，还可以在创建新文档时使用。

比如，我们现在创建一个版本号为5的博客文章，我们可以如下操作：

```
PUT /website/blog/2?version=5&version_type=external
{
  "title": "My first external blog entry",
  "text": "Starting to get the hang of this..."
}
```

在响应消息中我们可以看到当前_version号码是5：

```
{
  "_index": "website",
  "_type": "blog",
  "_id": "2",
  "_version": 5,
  "created": true
}
```

现在我们更新这个文档，并指定新版本号为10

```
PUT /website/blog/2?version=10&version_type=external
{
  "title": "My first external blog entry",
  "text": "This is a piece of cake..."
}
```

上面的请求被成功执行，并且当前的_version为10

```
{
  "_index": "website",
  "_type": "blog",
  "_id": "2",
  "_version": 10,
  "created": false
}
```

当时如果你重新运行上面的请求，你会遇到之前一样的版本冲突的错误，因为你所指定的外部版本号并没有大于当前Elasticsearch中的版本号。

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

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。
本文链接：[【】（）](#)