**Solr Python API：SolrCloudpy与Pysolr的对比**

目录

[1. SolrCloudpy 1](#_Toc423445418)

[1.1 使用优势 1](#_Toc423445419)

[1.2 安装 2](#_Toc423445420)

[1.3 使用示例 2](#_Toc423445421)

[1.4 使用缺陷 3](#_Toc423445422)

[2．Pysolr 3](#_Toc423445423)

[2.1 使用优势 3](#_Toc423445424)

[2.2安装 3](#_Toc423445425)

[2.3 使用示例 4](#_Toc423445426)

[2.4 使用缺陷 4](#_Toc423445427)

本文简单介绍两种Solr的pythoy API ，用于基于solr服务器进行检索， 文中不但介绍其具体的使用方法，也详细的对比了两种API的优缺点和适宜使用的场景。

# SolrCloudpy

SolrCloudpy是为与SolrCloud交互而设计的库文件。

## 1.1 使用优势

直接与SolrCloud交互的API，对于安装于集群中的solr服务十分方便，不需要人为判断索引位置，可根据SolrCloud与Zookeeper交互由索引名直接获取索引地址，十分高效，不会出现索引重建时不在原集群节点而导致的获取不到索引数据的错误。

## 安装

使用pip进行安装，安装命令为：

pip install solrcloudpy

## 使用示例



上图是使用solrcloudpy的一个简单示例，首先导入相关的python文件，使用类：

solrcloudpy.connection.SolrConnection(*server='host:8983'*,*detect\_live\_nodes=False*, *user=None*, *password=None*, *timeout=10*)

连接solr服务器；使用类：

solrcloudpy.parameters.**SearchOptions**(\*\*kwargs)

管理查询选项，可用“.”连接不同的查询参数：



构建好的查询参数传递给search方法，连接固定索引进行查询：



其中，response为返回的查询结果，可进行下一步的处理。

## 使用缺陷

无法传递高亮参数，不能实现界面上的高亮显示。

# 2．Pysolr

Pysolr是solr的较为轻量的python应用。

## 2.1 使用优势

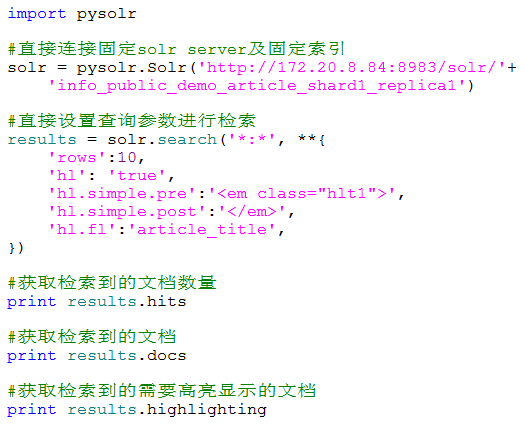
Pysolr设计为与固定服务器上的固定索引进行直接对话，连接后可设置查询参数，直接进行检索，方便易用，同时，提供的参数比较全面，避免了solrcloudpy不能传递高亮参数的劣势，可提供高亮功能，更好的发挥了solr的搜索体验，令用户实现效果更优的检索界面。

## 2.2安装

使用pip进行安装，安装命令为：

pip install pysolr

## 2.3 使用示例



上图是使用solrcloudpy的一个简单示例，首先导入相关的python文件，使用pysolr.Solr方法连接solr 服务器并指定要检索的索引；使用solr.search方法设置查询参数，其中，第一个参数为solr查询q，\*\*{}中添加的参数为其他查询参数，例如，“rows”返回结果数量，“hl”是否启用高亮，“hl.fl”高亮的返回字段等；results即为得到的检索数据，可进行进一步分析，如获取文档数量，文档内容，某字段的内容等等。

## 2.4 使用缺陷

与solrcoudpy相比，在多节点的集群中，pysolr不能通过其中任意一个节点判断索引的位置，只能由用户指定某个节点上的某个索引，一旦指定不能改变，若索引位置移动，则无法进行检索，这一点对于集群中的solr服务造成很多不便，不利于开发。