# 使用索引器连接 DocumentDB 和 Azure 搜索

如果你希望对 DocumentDB 数据实现强大的搜索体验，请对 DocumentDB 使用 Azure 搜索索引器！ 在本文中，我们将展示如何将 Azure DocumentDB 与 Azure 搜索进行集成且无需编写代码以保持索引编制基础结构。

要对此进行设置，你需要[设置 Azure 搜索帐户](/documentation/articles/search-create-service-portal)（不需要升级到标准搜索），然后调用 [Azure 搜索 REST API](https://msdn.microsoft.com/library/azure/dn798935.aspx) 以创建 DocumentDB **数据源**和用于该数据源的**索引器**。

## Azure 搜素索引器概念

Azure 搜索支持创建和管理数据源（包括 DocumentDB）以及针对这些数据源操作的索引器。

**数据源**指定哪些数据需要编制索引、用于访问数据的凭据和启用 Azure 搜索的策略，以有效地标识数据 中的更改（如你的集合内已修改或已删除的文档）。数据源定义为独立的资源，以便它可以被多个索引器使用。

**索引器**描述数据从你的数据源流动到目标搜索索引的方式。你应该计划为每个目标索引和数据源组合创建一个索引器。虽然可以有多个索引器写入到相同的索引，但一个索引器只能写入到一个索引。索引器用于：

* 执行数据的一次性复制以填充索引。
* 在计划上同步索引与数据源中的更改。计划是索引器定义的一部分。
* 根据需要调用对索引的按需更新。

## 步骤 1：创建数据源

发出一个 HTTP POST 请求以在你的 Azure 搜索服务中创建新的数据源，包括以下请求标头。

POST https://[Search service name].search.windows.net/datasources?api-version=[api-version]  
Content-Type: application/json  
api-key: [Search service admin key]

api-version 是必需的。有效值包括 2015-02-28 或更高版本。

请求正文包含数据源定义，其中应包括以下字段：

* **名称**：数据源的名称。
* **类型**：使用 documentdb。
* **凭据**：
  + **connectionString**：必需。采用以下格式指定到 Azure DocumentDB 数据库的连接信息：AccountEndpoint=<DocumentDB endpoint url>;AccountKey=<DocumentDB auth key>;Database=<DocumentDB database id>
* **容器**：
  + **名称**：必需。指定要编制索引的 DocumentDB 集合。
  + **查询**：可选。你可以指定一个查询来将一个任意 JSON 文档平整成 Azure 搜索可编制索引的平面架构。
* **dataChangeDetectionPolicy**：可选。请参阅下面的[数据更改检测策略](#DataChangeDetectionPolicy)。
* **dataDeletionDetectionPolicy**：可选。请参阅下面的[数据删除检测策略](#DataDeletionDetectionPolicy)。

### 捕获已更改的文档

数据更改检测策略旨在有效识别已更改的数据项。目前，唯一支持的策略是 DocumentDB 提供的使用 \_ts 上次修改时间戳属性的 High Water Mark 策略 - 它按如下所示指定：

{  
 "@odata.type" : "#Microsoft.Azure.Search.HighWaterMarkChangeDetectionPolicy",  
 "highWaterMarkColumnName" : "\_ts"  
}

你还需要在投影中添加 \_ts 和用于查询的 WHERE 子句。例如：

SELECT s.id, s.Title, s.Abstract, s.\_ts FROM Sessions s WHERE s.\_ts > @HighWaterMark

### 捕获已删除的文档

在源表中删除行时，你也应该从搜索索引中删除这些行。数据删除检测策略旨在有效识别已删除的数据项。目前，唯一支持的策略是 Soft Delete 策略（删除标有某种类型的标志），它按如下所示指定：

{  
 "@odata.type" : "#Microsoft.Azure.Search.SoftDeleteColumnDeletionDetectionPolicy",  
 "softDeleteColumnName" : "the property that specifies whether a document was deleted",  
 "softDeleteMarkerValue" : "the value that identifies a document as deleted"  
}

[AZURE.NOTE] 如果你使用的是自定义投影，则将需要在 SELECT 子句中包含该属性。

### 请求正文示例

下面的示例创建具有自定义查询和策略提示的数据源：

{  
 "name": "mydocdbdatasource",  
 "type": "documentdb",  
 "credentials": {  
 "connectionString": "AccountEndpoint=https://myDocDbEndpoint.documents.azure.com;AccountKey=myDocDbAuthKey;Database=myDocDbDatabaseId"  
 },  
 "container": {  
 "name": "myDocDbCollectionId",  
 "query": "SELECT s.id, s.Title, s.Abstract, s.\_ts FROM Sessions s WHERE s.\_ts > @HighWaterMark"  
 },  
 "dataChangeDetectionPolicy": {  
 "@odata.type": "#Microsoft.Azure.Search.HighWaterMarkChangeDetectionPolicy",  
 "highWaterMarkColumnName": "\_ts"  
 },  
 "dataDeletionDetectionPolicy": {  
 "@odata.type": "#Microsoft.Azure.Search.SoftDeleteColumnDeletionDetectionPolicy",  
 "softDeleteColumnName": "isDeleted",  
 "softDeleteMarkerValue": "true"  
 }  
}

### 响应

如果成功创建数据源，你将收到 HTTP 201 已创建响应。

## 步骤 2：创建索引

如果你还没有目标 Azure 搜索索引，请创建一个。你可以从 [Azure 门户 UI](/documentation/articles/search-get-started#test-service-operations) 创建或通过使用[创建索引 API](https://msdn.microsoft.com/library/azure/dn798941.aspx)。

POST https://[Search service name].search.windows.net/indexes?api-version=[api-version]  
Content-Type: application/json  
api-key: [Search service admin key]

确保目标索引的架构与源 JSON 文档的架构或自定义查询投影的输出的架构兼容。

### 图 A：JSON 数据类型与 Azure 搜索数据类型之间的映射

|  |  |
| --- | --- |
| JSON 数据类型 | 兼容的目标索引字段类型 |
| Bool | Edm.Boolean、Edm.String |
| 类似于整数的数字 | Edm.Int32、Edm.Int64、Edm.String |
| 类似于浮点的数字 | Edm.Double、Edm.String |
| String | Edm.String |
| 基元类型的数组，如“a”、“b”、“c” | 集合 (Edm.String) |
| 类似于日期的字符串 | Edm.DateTimeOffset、Edm.String |
| GeoJSON 对象，如 { “type”: “Point”, “coordinates”: [ long, lat ] } | Edm.GeographyPoint |
| 其他 JSON 对象 | 不适用 |

### 请求正文示例

下面的示例创建带有 ID 和描述字段的索引：

{  
 "name": "mysearchindex",  
 "fields": [{  
 "name": "id",  
 "type": "Edm.String",  
 "key": true,  
 "searchable": false  
 }, {  
 "name": "description",  
 "type": "Edm.String",  
 "filterable": false,  
 "sortable": false,  
 "facetable": false,  
 "suggestions": true  
 }]  
 }

### 响应

如果成功创建索引，你将收到 HTTP 201 创建的响应。

## 步骤 3：创建索引器

可以通过使用具有以下标头的 HTTP POST 请求来在 Azure 搜索服务中创建新的索引器。

POST https://[Search service name].search.windows.net/indexers?api-version=[api-version]  
Content-Type: application/json  
api-key: [Search service admin key]

请求正文包含索引器定义，其中应包括以下字段：

* **名称**：必需。索引器的名称。
* **dataSourceName**：必需。现有数据源的名称。
* **targetIndexName**：必需。现有索引的名称。
* **计划**：可选。请参阅下面的[索引计划](#IndexingSchedule)。

### 按计划运行索引器

一个索引器可以选择指定一个计划。如果存在计划，则索引器将按计划定期运行。计划具有以下属性：

* **间隔**：必需。指定索引器运行的间隔或时段的持续时间值。允许的最小间隔为 5 分钟；最长为一天。必须将其格式化为 XSD “dayTimeDuration” 值（[ISO 8601 持续时间](http://www.w3.org/TR/xmlschema11-2/#dayTimeDuration)值的有受限的子集）。它的模式是：P(nD)(T(nH)(nM))。示例：PT15M 为每隔 15 分钟，PT2H 为每隔 2 小时。
* **startTime**：必需。指定索引器应何时开始运行的 UTC 日期时间。

### 请求正文示例

下面的示例创建一个索引器，该索引器将数据从 myDocDbDataSource 数据源引用的集合复制到于 UTC 2015 年 1 月 1 日开始并每小时运行一次的计划上的 mySearchIndex 索引。

{  
 "name" : "mysearchindexer",  
 "dataSourceName" : "mydocdbdatasource",  
 "targetIndexName" : "mysearchindex",  
 "schedule" : { "interval" : "PT1H", "startTime" : "2015-01-01T00:00:00Z" }  
}

### 响应

如果成功创建索引器，你将收到 HTTP 201 已创建响应。

## 步骤 4：运行索引器

除按计划定期运行以外，也可以通过发出以下 HTTP POST 请求来按需调用索引器：

POST https://[Search service name].search.windows.net/indexers/[indexer name]/run?api-version=[api-version]  
api-key: [Search service admin key]

### 响应

如果成功调用索引器，你将收到 HTTP 202 已接受的响应。

## 步骤 5：获取索引器状态

你可以发出 HTTP GET 请求来检索索引器的当前状态和执行历史记录：

GET https://[Search service name].search.windows.net/indexers/[indexer name]/status?api-version=[api-version]  
api-key: [Search service admin key]

### 响应

你将看到返回 HTTP 200 OK 响应，同时返回响应正文，它包含有关总体索引器运行状况状态、上次索引器调用以及最近索引器调用的历史记录（如果存在）的信息。

响应应类似于以下形式：

{  
 "status":"running",  
 "lastResult": {  
 "status":"success",  
 "errorMessage":null,  
 "startTime":"2014-11-26T03:37:18.853Z",  
 "endTime":"2014-11-26T03:37:19.012Z",  
 "errors":[],  
 "itemsProcessed":11,  
 "itemsFailed":0,  
 "initialTrackingState":null,  
 "finalTrackingState":null  
 },  
 "executionHistory":[ {  
 "status":"success",  
 "errorMessage":null,  
 "startTime":"2014-11-26T03:37:18.853Z",  
 "endTime":"2014-11-26T03:37:19.012Z",  
 "errors":[],  
 "itemsProcessed":11,  
 "itemsFailed":0,  
 "initialTrackingState":null,  
 "finalTrackingState":null  
 }]  
}

执行历史记录包含最多 50 个最近完成的执行，它们被按反向时间顺序排序（因此，最新执行出现在响应中的第一个）。

## 后续步骤

祝贺你！ 你已学习了如何使用 DocumentDB 索引器将 Azure DocumentDB 与 Azure 搜索进行集成。

* 要了解有关 Azure DocumentDB 的详细信息，请参阅 [DocumentDB 服务页](https://azure.microsoft.com/services/documentdb/)。
* 要了解有关 Azure 搜索的详细信息，请参阅[搜索服务页](https://azure.microsoft.com/services/search/)。