

OData 查询选项

1 简介

这是 OData 的查询选项列表。

目前我们仅支持此处所述的选项。

2 检索对象

2.1 检索所有对象

可通过指定 URI 来检索所有对象。例如：`/odata/myservice/myresource`。如果在浏览器中指定 URI，则可以看到此结果。

2.2 检索单个对象

可通过在 URI 中传递对象标识符来检索单个对象。例如：`/odata/myservice/myresource(8444249301330581)`。

2.3 检索关联对象

可通过传递 `$expand` 查询参数来检索关联对象。例如：`/odata/myservice/Employees?$expand=Cars,Address/City`。

3 统计对象数量

3.1 检索对象计数

可通过传递 `$count` 查询选项来了解有多少个对象。在这种情况下，结果是一个整型，即对象的数量。例如：`/odata/myservice/myresource/$count`。

3.2 内联计数

通过将 `$inlinecount` 查询选项设为“所有页面”，结果中将包含返回项的数量计数。例如：`?$inlinecount=allpages`。

4 筛选

通过向请求附加 `$filter=...` 参数来应用筛选器。例如：`/Employees?$filter=Name eq 'John'`。

4.1 传递属性

下表说明了如何为不同属性类型传递值：

类型	传递方式
字符串和枚举	用单引号括起来（例如， <code>'John'</code> ）
日期时间	以 <code>datetime</code> 开头，并用单引号括起来（例如， <code>datetime' 2015-01-01'</code> 或 <code>datetime' <epoch value here>'</code> ）
其他	普通值（例如， <code>15</code> ）

4.2 比较运算符

我们支持以下比较运算符：

运算符	含义	示例
<code>eq</code>	等于	<code>/Employees?\$filter=Name eq 'John'</code>
<code>ne</code>	不等于	<code>/Employees?\$filter=Name ne 'John'</code>
<code>gt</code>	大于	<code>/Employees?\$filter=Age gt 15</code>
<code>lt</code>	小于	<code>/Employees?\$filter=Age lt 15</code>
<code>ge</code>	大于等于	<code>/Employees?\$filter=Age ge 15</code>
<code>le</code>	小于等于	<code>/Employees?\$filter=Age le 15</code>

4.3 算术运算符

运算符	含义	示例	返回值
<code>add</code>	加	<code>/Products?\$filter=Prices add 2 eq 10</code>	价格为 8 的所有产品
<code>sub</code>	减	<code>/Products?\$filter=Prices sub 2 eq 10</code>	价格为 12 的所有产品
<code>mul</code>	乘	<code>/Products?\$filter=Prices mul 2 eq 10</code>	价格为 5 的所有产品
<code>div</code>	除	<code>/Products?\$filter=Prices div 2 eq 10</code>	价格为 20 的所有产品

mod 模数 /Products?\$filter=Prices mod 5 eq 0 价格能被 5 整除的所有产品

4.4 函数

函数	示例	返回值
substringof	/Employees?\$filter=substringof('f', Name)	名字中包含“f”的所有员工
endswith	/Employees?\$filter=endswith(Name, 'f')	名字以“f”结尾的所有员工
startswith	/Employees?\$filter=startswith(Name, 'f')	名字以“f”为开头的所有员工
length	/Employees?\$filter=length(Name) eq 5	名字长度为 5 的所有员工
year	/Employees?\$filter=year(DateOfBirth) eq 1990	出生于 1990 年的所有员工
month	/Employees?\$filter=month(DateOfBirth) eq 5	出生于五月的所有员工
day	/Employees?\$filter=day(DateOfBirth) eq 31	出生于某个月 31 号的所有员工
hour	/Employees?\$filter=hour(Registration) eq 13	在 13:00（下午 1 点）至 13:59（下午 1:59）之间注册的所有员工
minute	/Employees?\$filter=minute(Registration) eq 55	在几点 55 分注册的所有员工
second	/Employees?\$filter=second(Registration) eq 55	在几点几分 55 秒注册的所有员工

4.5 组合筛选器

筛选器可与 and、or、not 和 () 组合。例如：?\$filter=Name eq 'John' and (Age gt 65 or Age lt 11)。

组合 示例

and /Employees?\$filter=Name eq 'John' and Age gt 65

or /Employees?\$filter=Age gt 65 or Age lt 11

not /Employees?\$filter=not(Name eq 'John')

() /Employees?\$filter=Name eq 'John' and (Age gt 65 or Age lt 11)

4.6 按关联筛选

可筛选关联实体的属性。执行该操作的方式取决于关联是公开一个对象还是对象列表。

类型	示例
筛选关联对象	People?filter=BirthPlace/CityName eq 'Rotterdam'
筛选关联列表	City?filter=BornIn/any(person:person/Year le 1919)

将关联公开为链接时，可通过这种方式筛选关联对象或列表。将关联公开为关联对象 ID 时，无法执行上述筛选。

5 排序

可使用 `$orderby` 查询选项对结果排序。例如：`?$orderby=Name or ?$orderby=BirthPlace/CityName`。

默认方向为升序，可将此确定为显式设置。例如：`?$orderby=Name asc`。

也可按降序方向对结果排序。例如：`?$orderby=Name desc`。

可对多个属性进行排序，这些属性必须用逗号分隔。例如：`?$orderby=Name asc, Age desc`。

6 选择字段

可通过指定 `$select` 查询选项来选择要返回哪些属性和关联。例如：`?$select=Name, Age`。

7 分页

7.1 设置上限（限值）

可使用 `$top` 查询选项来限制返回对象的数量，其中限值为一个正整型。例如：`?$top=100`。

7.2 跳过（偏移）

可使用 `$skip` 查询选项来跳过多个对象，然后检索结果，其中偏移为一个正整型。例如：`?$skip=100` 将返回从列表中第 101 个对象开始的对象。

8 空文本

可将值与 `null` 文本进行比较。例如：`?$filter=Name eq null`。

此示例中，`Name` 是在数据库中不能有任何分配值的字符串属性。请注意，`null` 表示没有值，而不是 `''`（`''` 是一个空字符串）。

根据关联进行筛选时，空文本可能相当有用。例如：`?$filter=Association_A_B ne null`。此示例中，查询实体类型 `A` 的对象，这些对象具有至少一个针对实体类型 `B` 对象的关联集。