Filtering Assignment

# Environment

You have 1,000,000 records like the ones shown in the table below.

The user has the ability to filter data by zero or many columns.

Each column may have zero or one filter.

A filter field can be any of the column titles displayed.

Available operators for an integer column are: equal to, not equal to, greater than, greater than or equal to, less than, less than or equal to.

Available operators for a string column are: equal to, not equal to, contains, does not contain, starts with, ends with.

Available operators for a date column are: equal to, not equal to, after, after or equal to, before, before or equal to.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **employeeId** | **firstName** | **lastName** | **jobDescription** | **departmentName** | **managerId** | **dateHired** | **lastPasswordChange** |
| **1223** | Steve | Bob | Developer | Development | 1004 | 3/5/2015 | 5/5/2015 8:35:00 AM |
| **2287** | Joe | Smith | Developer | Development | 1004 | 2/10/2009 | 2/5/2012 7:45:00 AM |
| **1004** | George | Coco | Manager | Development | 2258 | 12/30/2013 | 8/1/2015 0:00:00 AM |
| **2867** | Gary | Bobby | Tester | Quality Assurance | 1587 | 5/17/2015 | 8/4/2015 7:55:00 AM |
| **1335** | Stacy | Terry | Developer | Development | 1004 | 9/21/2012 | 8/3/2015 8:01:00 AM |

# Problem

Assume a cache of 100,000 records is kept in memory.

Please create a .Net project with an employee object, a filter object, and a filtering utility class that accepts **two input parameters (List<Employee>, List<Filter>)** and returns the filtered list of employees. Please email the source code so that we can run it. The source code should reflect your style / code quality.

**Please address performance and latency issues.**

## Focus on meeting the following filtering criteria:

### Display results where employeeId is 1004.

#### Example Request

GET …/api/employee?filter={"logic":"and","filters":[{"employeeId":"entityTypeId","operator":"eq","value":1004}]}

#### Example Response

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1004** | George | Coco | Manager | Development | 2258 | 12/30/2013 | 8/1/2015 8:07:00 AM |

### Display results where lastPasswordChange was before or at 8/1/2015 at midnight.

#### Example Request

GET …/api/employee?filter={"logic":"and","filters":[{"lastPasswordChange":"entityTypeId","operator":"lte","value":"8/1/2015 00:00:00"}]}

#### Example Response

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1223** | Steve | Bob | Developer | Development | 1004 | 3/5/2015 | 5/5/2015 8:35:00 AM |
| **2287** | Joe | Smith | Developer | Development | 1004 | 2/10/2009 | 2/5/2012 7:45:00 AM |
| **1004** | George | Coco | Manager | Development | 2258 | 12/30/2013 | 8/1/2015 0:00:00 AM |

### Display results where departmentName contains the text “Dev” and dateHired is after 1/1/2015 midnight.

#### Example Request

GET …/api/employee?filter={"logic":"and","filters":[{"departmentName":"entityTypeId","operator":"contains","value":"Dev"},{"dateHired":"entityTypeId","operator":"gt","value":"1/1/2015 00:00:00"}]}

#### Example Response

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1223** | Steve | Bob | Developer | Development | 1004 | 3/5/2015 | 5/5/2015 8:35:00 AM |

# Bonus Question:

Each column may have 0, 1 or 2 filters.

A column that has two filters may use the logic “and” or “or” between the two filters.

## Example filters:

Display results where departmentName contains “Dev” and lastPasswordChange is between 6/30/2015 midnight **and** 8/1/2015 midnight.

Display results where departmentName starts with “Dev” **or** “Test”.