

C# .Net Core Developer Exercise: International City Search Feature

You are asked to write code to support the user interface of an international city search feature.

You will not be writing any actual UI code, but are asked to develop a search algorithm to help the user entering the name of a city.

The requirements from E-Commerce are for a user experience which works as follows: As the user types each character of the city's name in a textbox, options should update to show all valid choices for the next character and a list of possible matching cities.

The requirements are:

Typing a search string will return

- All cities that start with the search string
- All valid next characters for each matched city.
- Runtime performance is extremely important, the initial load time is not.
- Your application will need to be able to handle data sets in excess of 1,000,000 records.
- Spaces and dashes are valid characters when returning a list of next characters.
- The application will be used by a single user at a time; there is no need to code for concurrency.
- 3rd Party interface `ICityFinder` and `ICityResult` should be used (provided below).

Examples:

- Given the input 'BANG' and a list of cities 'BANDUNG', 'BANGUI', 'BANGKOK', 'BANGALORE', the application should return next characters of 'U', 'K', 'A' and the cities 'BANGUI', 'BANGKOK', 'BANGALORE'.
- Given the input 'LA' and a list of cities 'LA PAZ', 'LA PLATA', 'LAGOS', 'LEEDS' the application should return next characters of ' ' and 'G' and the cities 'LA PAZ', 'LA PLATA', 'LAGOS'
- Given the input 'ZE' and a list of cities 'ZARIA', 'ZHUGHAI', 'ZIBO' the application will return no next characters and no cities

The 3rd Party Interfaces are:

```
namespace CitySearch
{
    using System.Collections.Generic;

    public interface ICityResult
    {
        ICollection<string> NextLetters { get; set; }

        ICollection<string> NextCities { get; set; }
    }
}

namespace CitySearch
{
    public interface ICityFinder
    {
        ICityResult Search(string searchString);
    }
}
```