Applying Techniques for Searching Strings

Steve Gordon

.NET Engineer and Microsoft MVP

@stevejgordon www.stevejgordon.co.uk

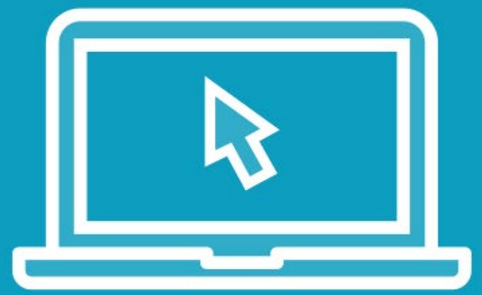
Overview

- Check if strings start or end with specific characters
- Identify if strings contain specific characters or substrings
- Locate character positions within strings
- Extract substrings
- Low allocation search techniques

Requirements

- Skip processing for strings which begin with malformed data.

Demo



Use the StartsWith method on the string type

```
// Definitions:

public bool StartsWith (char value);
```

StartsWith

Determines whether a string instance starts with the specified character.

```
// Definitions:

public bool StartsWith (char value);

public bool StartsWith (string value);
```

StartsWith

Determines whether a string instance starts with the specified character.

Determines whether the beginning of a string instance matches the specified string.

```
// Definitions:

public bool StartsWith (char value);

public bool StartsWith (string value);

// Use:

var myString = "A string literal";

bool result = myString.StartsWith('A'); // true
```

StartsWith

Determines whether a string instance starts with the specified character.

Determines whether the beginning of a string instance matches the specified string.

```
// Definitions:

public bool StartsWith (string value, StringComparison
comparisonType);


public bool StartsWith (string value, bool ignoreCase,
CultureInfo? culture);
```

StartsWith

Determines whether the beginning of a string instance matches the specified string when using a specific StringComparison option or Culture.

```
// Definitions:

public bool EndsWith (char value);
```

EndsWith

Determines whether a string instance ends with the specified character.

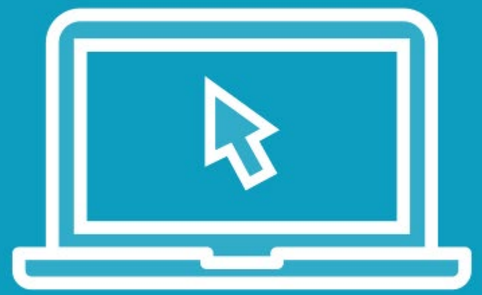
```
// Definitions:

public bool EndsWith (char value);

public bool EndsWith (string value);
```

EndsWith

Determines whether a string instance ends with the specified character.

Determines whether the end of a string instance matches the specified string.

Demo

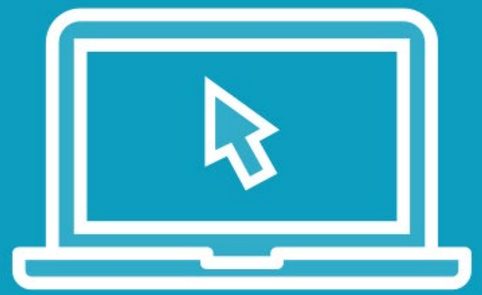


Identify if a string contains a specific character or string of characters

Requirements

- Provide an implementation of the Parse method for the ProductInfo type.
- Consider the data invalid if it does not contain a hyphen.
- Split out the sales code and stock keeping unit (SKU) code.
- Data should be considered invalid if it contains more than one hyphen.

Demo



Locate characters within strings

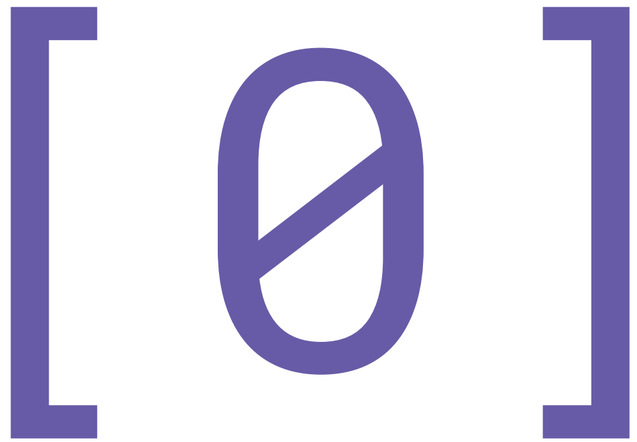
Extract portions of strings (substrings)

Requirements

- Update the processor to handle some additional rules for sales category data.
- Ensure that the numeric digits of the code are limited to 0, 1, 2 and 3.
- Implement the TryParse method on the Category type.

Locate the position of the colon character within the source string.

IndexOf



Accepts a character to locate

Returns an integer of its first position (index) within the string

Returns minus one (-1) if the character was not found

Like arrays, a character indexer is zero-based

- First character appears at index 0

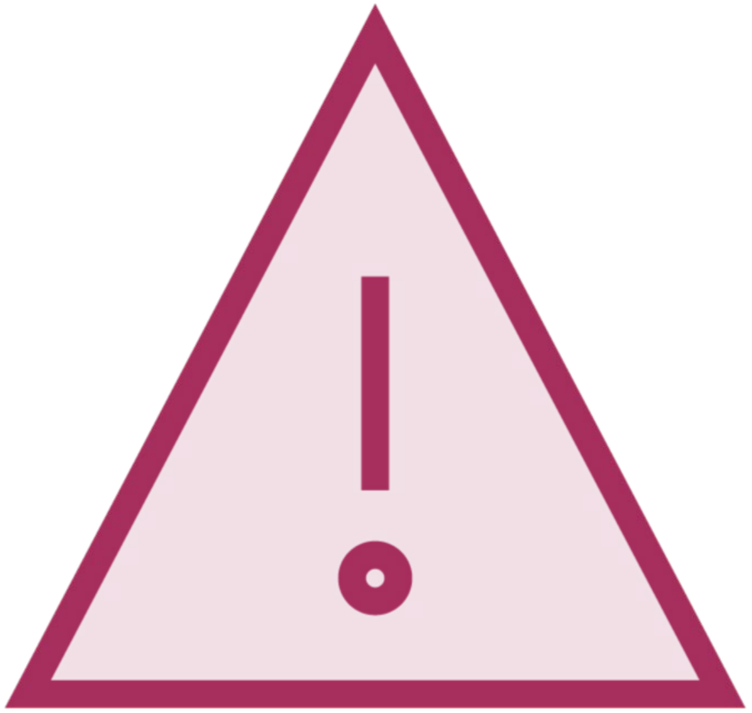
Also accepts a string to locate

- Returns the index of the strings first character if it is located in the source string

```
// Definitions:

public string Substring (int startIndex, int length);


// Use:

var myString = "A string literal";

string substring = myString.Substring(0, 8); // "A string"
```

Substring

Retrieves a substring from this instance. The substring starts at a specified character position and has a specified length.



IsLetter may not always return the expected result

String data may contain surrogate pairs of UTF-16 chars

IsLetter won’t work correctly for all languages

To more accurately detect letters, consider enumerating Runes

Runes represent Unicode scalar values

In our situation where all characters in the category code are expected to be UK English letters, calling char.IsLetter is safe.

Ranges

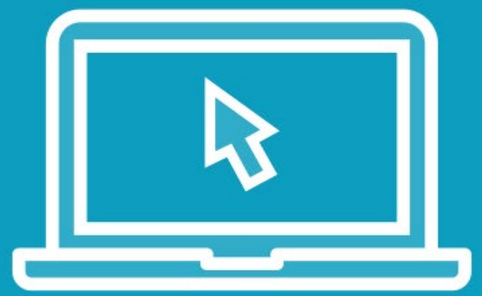
A range can be used to represent a sub-range within a sequence

Strings are a sequence of characters and therefore support ranges

The range operator was added in C# 8

- Indicated by using two period characters within an indexer

Demo



Apply searching techniques

Process string data without additional allocations

Requirements

- Provide additional diagnostics information about invalid data.
- Report common causes for invalid rows.
- Limit performance impact by avoiding allocation of additional strings.

We are required to accept, but ignore leading and trailing whitespace. We want to count the non-whitespace characters between the pipe and the colon.

Up Next:
Applying Techniques to Modify Strings