Matcher Class

Matchers use Patterns to perform match operations on a character string.

Namespace

System

Matcher Methods

The following are methods for `Matcher`.

- end()
  Returns the position after the last matched character.
- end(groupIndex)
  Returns the position after the last character of the subsequence captured by the group index during the previous match operation. If the match was successful but the group itself did not match anything, this method returns -1.
- find()
  Attempts to find the next subsequence of the input sequence that matches the pattern. This method returns true if a subsequence of the input sequence matches this Matcher object's pattern.
- find(group)
  Resets the Matcher object and then tries to find the next subsequence of the input sequence that matches the pattern. This method returns `true` if a subsequence of the input sequence matches this Matcher object's pattern.
- group()
  Returns the input subsequence returned by the previous match.
- group(groupIndex)
  Returns the input subsequence captured by the specified group index during the previous match operation. If the match was successful but the specified group failed to match any part of the input sequence, `null` is returned.
- groupCount()
  Returns the number of capturing groups in this Matching object's pattern. Group zero denotes the entire pattern and is not included in this count.
- hasAnchoringBounds()
  Returns true if the Matcher object has anchoring bounds, false otherwise. By default, a Matcher object uses anchoring bounds regions.
- hasTransparentBounds()
  Returns true if the Matcher object has transparent bounds, false if it uses opaque bounds. By default, a Matcher object uses opaque region boundaries.
- hitEnd()
  Returns true if the end of input was found by the search engine in the last match operation performed by this Matcher object. When this method returns true, it is possible that more input would have changed the result of the last search.
- lookingAt()
  Attempts to match the input sequence, starting at the beginning of the region, against the pattern.
- matches()
  Attempts to match the entire region against the pattern.

the returned string match the sequence of characters in *inputString*.

- **region(start, end)**
  Sets the limits of this Matcher object's region. The region is the part of the input sequence that is searched to find a match.
- **regionEnd()**
  Returns the end index (exclusive) of this Matcher object's region.
- **regionStart()**
  Returns the start index (inclusive) of this Matcher object's region.
- **replaceAll(replacementString)**
  Replaces every subsequence of the input sequence that matches the pattern with the replacement string.
- **replaceFirst(replacementString)**
  Replaces the first subsequence of the input sequence that matches the pattern with the replacement string.
- **requireEnd()**
  Returns true if more input could change a positive match into a negative one.
- **reset()**
  Resets this Matcher object. Resetting a Matcher object discards all of its explicit state information.
- **reset(inputSequence)**
  Resets this Matcher object with the new input sequence. Resetting a Matcher object discards all of its explicit state information.
- **start()**
  Returns the start index of the first character of the previous match.
- **start(groupIndex)**
  Returns the start index of the subsequence captured by the group specified by the group index during the previous match operation. Captured groups are indexed from left to right, starting at one. Group zero denotes the entire pattern, so the expression `m.start(0)` is equivalent to `m.start()`.
- **useAnchoringBounds(anchoringBounds)**
  Sets the anchoring bounds of the region for the Matcher object. By default, a Matcher object uses anchoring bounds regions.
- **usePattern(pattern)**
  Changes the Pattern object that the Matcher object uses to find matches. This method causes the Matcher object to lose information about the groups of the last match that occurred. The Matcher object's position in the input is maintained.
- **useTransparentBounds(transparentBounds)**
  Sets the transparency bounds for this Matcher object. By default, a Matcher object uses anchoring bounds regions.

end()

Returns the position after the last matched character.

Signature

```
public Integer end()
```

Return Value

Type: Integer

end(groupIndex)

Returns the position after the last character of the subsequence captured by the group index during the previous match operation. If the match was successful but the group itself did not

Parameters

groupIndex

Type: Integer

Return Value

Type: Integer

Usage

Captured groups are indexed from left to right, starting at one. Group zero denotes the entire pattern, so the expression `m.end(0)` is equivalent to `m.end()`.

See Understanding Capturing Groups.

find()

Attempts to find the next subsequence of the input sequence that matches the pattern. This method returns true if a subsequence of the input sequence matches this Matcher object's pattern.

Signature

```
public Boolean find()
```

Return Value

Type: Boolean

Usage

This method starts at the beginning of this Matcher object's region, or, if a previous invocation of the method was successful and the Matcher object has not since been reset, at the first character not matched by the previous match.

If the match succeeds, more information can be obtained using the `start`, `end`, and `group` methods.

For more information, see Using Regions.

find(group)

Resets the Matcher object and then tries to find the next subsequence of the input sequence that matches the pattern. This method returns `true` if a subsequence of the input sequence matches this Matcher object's pattern.

Signature

```
public Boolean find(Integer group)
```

Parameters

group

Type: Integer

Return Value

Type: Boolean

Usage

If the match succeeds, more information can be obtained using the `start`, `end`, and `group` methods.

```
public String group()
```

Return Value

Type: String

Usage

Note that some groups, such as `(a*)`, match the empty string. This method returns the empty string when such a group successfully matches the empty string in the input.

group(groupIndex)

Returns the input subsequence captured by the specified group index during the previous match operation. If the match was successful but the specified group failed to match any part of the input sequence, `null` is returned.

Signature

```
public String group(Integer groupIndex)
```

Parameters

groupIndex

Type: Integer

Return Value

Type: String

Usage

Captured groups are indexed from left to right, starting at one. Group zero denotes the entire pattern, so the expression `m.group(0)` is equivalent to `m.group()`.

Note that some groups, such as `(a*)`, match the empty string. This method returns the empty string when such a group successfully matches the empty string in the input.

See Understanding Capturing Groups.

groupCount()

Returns the number of capturing groups in this Matching object's pattern. Group zero denotes the entire pattern and is not included in this count.

Signature

```
public Integer groupCount()
```

Return Value

Type: Integer

Usage

See Understanding Capturing Groups.

hasAnchoringBounds()

Returns true if the Matcher object has anchoring bounds, false otherwise. By default, a Matcher object uses anchoring bounds regions.

Type: Boolean

Usage

If a Matcher object uses anchoring bounds, the boundaries of this Matcher object's region match start and end of line anchors such as ^ and $.

For more information, see Using Bounds.

hasTransparentBounds()

Returns true if the Matcher object has transparent bounds, false if it uses opaque bounds. By default, a Matcher object uses opaque region boundaries.

Signature

```
public Boolean hasTransparentBounds()
```

Return Value

Type: Boolean

Usage

For more information, see Using Bounds.

hitEnd()

Returns true if the end of input was found by the search engine in the last match operation performed by this Matcher object. When this method returns true, it is possible that more input would have changed the result of the last search.

Signature

```
public Boolean hitEnd()
```

Return Value

Type: Boolean

lookingAt()

Attempts to match the input sequence, starting at the beginning of the region, against the pattern.

Signature

```
public Boolean lookingAt()
```

Return Value

Type: Boolean

Usage

Like the `matches` method, this method always starts at the beginning of the region; unlike that method, it does not require the entire region be matched.

If the match succeeds, more information can be obtained using the `start`, `end`, and `group` methods.

See Using Regions.

```
public Boolean matches()
```

Return Value

Type: Boolean

Usage

If the match succeeds, more information can be obtained using the `start`, `end`, and `group` methods.

See Using Regions.

pattern()

Returns the Pattern object from which this Matcher object was created.

Signature

```
public Pattern object pattern()
```

Return Value

Type: System.Pattern

quoteReplacement(inputString)

Returns a literal replacement string for the specified string *inputString*. The characters in the returned string match the sequence of characters in *inputString*.

Signature

```
public static String quoteReplacement(String inputString)
```

Parameters

inputString

Type: String

Return Value

Type: String

Usage

Metacharacters (such as `$` or `^`) and escape sequences in the input string are treated as literal characters with no special meaning.

region(start, end)

Sets the limits of this Matcher object's region. The region is the part of the input sequence that is searched to find a match.

Signature

```
public Matcher object region(Integer start, Integer end)
```

Parameters

start

Type: Integer

Type: Matcher

Usage

This method first resets the Matcher object, then sets the region to start at the index specified by `start` and end at the index specified by `end`.

Depending on the transparency boundaries being used, certain constructs such as anchors may behave differently at or around the boundaries of the region.

See Using Regions and Using Bounds.

regionEnd()

Returns the end index (exclusive) of this Matcher object's region.

Signature

```
public Integer regionEnd()
```

Return Value

Type: Integer

Usage

See Using Regions.

regionStart()

Returns the start index (inclusive) of this Matcher object's region.

Signature

```
public Integer regionStart()
```

Return Value

Type: Integer

Usage

See Using Regions.

replaceAll(replacementString)

Replaces every subsequence of the input sequence that matches the pattern with the replacement string.

Signature

```
public String replaceAll(String replacementString)
```

Parameters

replacementString

Type: String

Return Value

Type: String

Usage

Note that backslashes (\) and dollar signs ($) in the replacement string may cause the results to be different than if the string was treated as a literal replacement string. Dollar signs may be treated as references to captured subsequences, and backslashes are used to escape literal characters in the replacement string.

Invoking this method changes this Matcher object's state. If the Matcher object is to be used in further matching operations it should first be reset.

Given the regular expression `a*b`, the input `"aabxyzaabxyzabxyzb"`, and the replacement string `"-"`, an invocation of this method on a Matcher object for that expression would yield the string `"-xyz-xyz-xyz-"`.

replaceFirst(replacementString)

Replaces the first subsequence of the input sequence that matches the pattern with the replacement string.

Signature

```
public String replaceFirst(String replacementString)
```

Parameters

replacementString

Type: String

Return Value

Type: String

Usage

Note that backslashes (\) and dollar signs ($) in the replacement string may cause the results to be different than if the string was treated as a literal replacement string. Dollar signs may be treated as references to captured subsequences, and backslashes are used to escape literal characters in the replacement string.

Invoking this method changes this Matcher object's state. If the Matcher object is to be used in further matching operations it should first be reset.

Given the regular expression `dog`, the input `"zzzdogzzzdogzzz"`, and the replacement string `"cat"`, an invocation of this method on a Matcher object for that expression would return the string `"zzzcatzzzdogzzz"`.

requireEnd()

Returns true if more input could change a positive match into a negative one.

Signature

```
public Boolean requireEnd()
```

Return Value

Type: Boolean

Usage

If this method returns true, and a match was found, then more input could cause the match to be lost.

If this method returns false and a match was found, then more input might change the match but the match won't be lost.

Resets this Matcher object. Resetting a Matcher object discards all of its explicit state information.

Signature

```
public Matcher object reset()
```

Return Value

Type: Matcher

Usage

This method does not change whether the Matcher object uses anchoring bounds. You must explicitly use the `useAnchoringBounds` method to change the anchoring bounds.

For more information, see Using Bounds.

reset(inputSequence)

Resets this Matcher object with the new input sequence. Resetting a Matcher object discards all of its explicit state information.

Signature

```
public Matcher reset(String inputSequence)
```

Parameters

inputSequence

Type: String

Return Value

Type: Matcher

start()

Returns the start index of the first character of the previous match.

Signature

```
public Integer start()
```

Return Value

Type: Integer

start(groupIndex)

Returns the start index of the subsequence captured by the group specified by the group index during the previous match operation. Captured groups are indexed from left to right, starting at one. Group zero denotes the entire pattern, so the expression `m.start(0)` is equivalent to `m.start()`.

Signature

```
public Integer start(Integer groupIndex)
```

Parameters

groupIndex

Type: Integer

See Understanding Capturing Groups.

useAnchoringBounds(anchoringBounds)

Sets the anchoring bounds of the region for the Matcher object. By default, a Matcher object uses anchoring bounds regions.

Signature

```
public Matcher object useAnchoringBounds(Boolean anchoringBounds)
```

Parameters

anchoringBounds

Type: Boolean

If you specify `true`, the Matcher object uses anchoring bounds. If you specify `false`, non-anchoring bounds are used.

Return Value

Type: Matcher

Usage

If a Matcher object uses anchoring bounds, the boundaries of this Matcher object's region match start and end of line anchors such as ^ and $.

For more information, see Using Bounds.

usePattern(pattern)

Changes the Pattern object that the Matcher object uses to find matches. This method causes the Matcher object to lose information about the groups of the last match that occurred. The Matcher object's position in the input is maintained.

Signature

```
public Matcher object usePattern(Pattern pattern)
```

Parameters

pattern

Type: System.Pattern

Return Value

Type: Matcher

useTransparentBounds(transparentBounds)

Sets the transparency bounds for this Matcher object. By default, a Matcher object uses anchoring bounds regions.

Signature

```
public Matcher object useTransparentBounds(Boolean transparentBounds)
```

Parameters

transparentBounds

Type: Boolean

Type: Matcher

Usage

For more information, see Using Bounds.

DID THIS ARTICLE SOLVE YOUR ISSUE?
Let us know so we can improve!

Share your feedback

DEVELOPER CENTERS
Heroku
MuleSoft
Tableau
Commerce Cloud
Lightning Design System
Einstein
Quip

POPULAR RESOURCES
Documentation
Component Library
APIs
Trailhead
Sample Apps
Podcasts
AppExchange

COMMUNITY
Trailblazer Community
Events and Calendar
Partner Community
Blog
Salesforce Admins
Salesforce Architects

Privacy Information Terms of Service Legal Use of Cookies Trust Cookie Preferences

✓✕ Your Privacy Choices Responsible Disclosure Contact