







### **Developers**



Apex Reference Guide / System Namespace / OrgLimits Class

# **Pattern Class**

Represents a compiled representation of a regular expression.

## **Namespace**

System

## **Pattern Methods**

The following are methods for Pattern.

compile(regExp)

Compiles the regular expression into a Pattern object.

• matcher(stringtoMatch)

Creates a Matcher object that matches the input string *stringtoMatch* against this Pattern object.

• matches(regExp, stringtoMatch)

Compiles the regular expression *regExp* and tries to match it against the specified string. This method returns true if the specified string matches the regular expression, false otherwise.

• pattern()

Returns the regular expression from which this Pattern object was compiled.

quote(yourString)

Returns a string that can be used to create a pattern that matches the string *yourString* as if it were a literal pattern.

• split(regExp)

Returns a list that contains each substring of the String that matches this pattern.

• split(regExp, limit)

Returns a list that contains each substring of the String that is terminated either by the regular expression *regExp* that matches this pattern, or by the end of the String.

## compile(regExp)

Compiles the regular expression into a Pattern object.

#### Signature

public static Pattern compile(String regExp)

### **Parameters**

#### regExp

Type: String

#### **Return Value**

Type: System.Pattern

## matcher(stringtoMatch)



~

#### **Parameters**

### stringtoMatch

Type: String

#### **Return Value**

Type: Matcher

## matches(regExp, stringtoMatch)

Compiles the regular expression *regExp* and tries to match it against the specified string. This method returns true if the specified string matches the regular expression, false otherwise.

#### Signature

public static Boolean matches(String regExp, String stringtoMatch)

#### **Parameters**

#### regExp

Type: String

#### stringtoMatch

Type: String

#### **Return Value**

Type: Boolean

#### Usage

If a pattern is to be used multiple times, compiling it once and reusing it is more efficient than invoking this method each time.

#### Example

Note that the following code example:

```
Pattern.matches(regExp, input);
```

produces the same result as this code example:

```
Pattern.compile(regex).
matcher(input).matches();
```

## pattern()

 $\hbox{Returns the regular expression from which this Pattern object was compiled.}$ 

#### Signature

public String pattern()

#### **Return Value**

Type: String



V

#### Signature

public static String quote(String yourString)

#### **Parameters**

#### yourString

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.

## split(regExp)

Returns a list that contains each substring of the String that matches this pattern.

#### Signature

public String[] split(String regExp)

#### **Parameters**

#### regExp

Type: String

#### Return Value

Type: String[]



#### Note

In API version 34.0 and earlier, a zero-width *regExp* value produces an empty list item at the beginning of the method's output.

#### Usage

The substrings are placed in the list in the order in which they occur in the String. If *regExp* does not match the pattern, the resulting list has just one element containing the original String.

## split(regExp, limit)

Returns a list that contains each substring of the String that is terminated either by the regular expression *regExp* that matches this pattern, or by the end of the String.

#### Signature

public String[] split(String regExp, Integer limit)

#### Parameters

#### regExp

Type: String

#### limit





- The pattern is applied a maximum of (limit 1) times.
- The list's length is no greater than *limit*.
- o The list's last entry contains all input beyond the last matched delimiter.
- If *limit* is non-positive, the pattern is applied as many times as possible, and the list can have any length.
- If limit is zero, the pattern is applied as many times as possible, the list can have any length, and trailing empty strings are discarded.

#### **Return Value**

Type: String[]



#### Note

In API version 34.0 and earlier, a zero-width regExp value produces an empty list item at the beginning of the method's output.

#### 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

© Copyright 2025 Salesforce, Inc. All rights reserved. Various trademarks held by their respective owners. Salesforce, Inc. Salesforce Tower, 415 Mission Street, 3rd Floor, San Francisco, CA 94105, United States

<u>Privacy Information</u> <u>Terms of Service</u> <u>Legal</u> <u>Use of Cookies</u> <u>Trust</u> <u>Cookie Preferences</u>



Your Privacy Choices

Responsible Disclosure