# Cookie Class

The `Cookie` class lets you access cookies for your Salesforce site using Apex.

## Namespace

System

## Usage

Use the `setCookies` method of the PageReference Class to attach cookies to a page.

**Important**

- Cookie names and values set in Apex are URL encoded, that is, characters such as @ are replaced with a percent sign and their hexadecimal representation.
- The `setCookies` method adds the prefix “`apex__`” to the cookie names.
- Setting a cookie's value to `null` sends a cookie with an empty string value instead of setting an expired attribute.
- After you create a cookie, the properties of the cookie can't be changed.
- Be careful when storing sensitive information in cookies. Pages are cached regardless of a cookie value. If you use a cookie value to generate dynamic content, you should disable page caching. For more information, see Configure Site Caching in Salesforce Help.

Consider the following limitations when using the `Cookie` class:

- The `Cookie` class can only be accessed using Apex that is saved using the Salesforce API version 19 and above.
- The maximum number of cookies that can be set per Salesforce Sites domain depends on your browser. Newer browsers have higher limits than older ones.
- Cookies must be less than 4K, including name and attributes.
- The maximum header size of a Visualforce page, including cookies, is 8,192 bytes.

For more information on sites, see “Salesforce Sites” in the Salesforce online help.

## Example

The following example creates a class, `CookieController`, which is used with a Visualforce page (see markup below) to update a counter each time a user displays a page. The number of times a user goes to the page is stored in a cookie.

```
// A Visualforce controller class that creates a cookie
// used to keep track of how often a user displays a page
public class CookieController {

    public CookieController() {
        Cookie counter = ApexPages.currentPage().getCookies().get('counter');
        // If this isn't the first time the user is accessing the page
        // create a new cookie, incrementing the value of the original count by 1
            Integer count = Integer.valueOf(counter.getValue());
            counter = new Cookie('counter', String.valueOf(count+1),null,-1,true);
        }

        // Set the new cookie for the page
        ApexPages.currentPage().setCookies(new Cookie[]{counter});
    }

    // This method is used by the Visualforce action {!count} to display the current
    // value of the number of times a user had displayed a page.
    // This value is stored in the cookie.
    public String getCount() {
        Cookie counter = ApexPages.currentPage().getCookies().get('counter');
        if(counter == null) {
            return '0';
        }
        return counter.getValue();
    }
}
```

```
// Test class for the Visualforce controller
@isTest
private class CookieControllerTest {
  // Test method for verifying the positive test case
  static testMethod void testCounter() {
    //first page view
    CookieController controller = new CookieController();
    System.assert(controller.getCount() == '1');

    //second page view
    controller = new CookieController();
    System.assert(controller.getCount() == '2');
  }
}
```

The following is the Visualforce page that uses the `CookieController` Apex controller above. The action `{!count}` calls the `getCount` method in the controller above.

```
<apex:page controller="CookieController">
You have seen this page {!count} times
</apex:page>
```

- Cookie Constructors
- Cookie Methods

## Cookie Constructors

The following are constructors for `Cookie`.

- **Cookie(name, value, path, maxAge, isSecure)**
  Creates a new instance of the `Cookie` class using the specified name, value, path, age, and the secure setting.
- **Cookie(name, value, path, maxAge, isSecure, SameSite)**
  Creates a new instance of the `Cookie` class using the specified name, value, path, and age, and settings for security and cross-domain behavior.
- **Cookie(name, value, path, maxAge, isSecure, SameSite, isHttpOnly)**
  Creates a new instance of the `Cookie` class using the specified name, value, path, age, and settings for security, cross-domain behavior, and JavaScript access.

#### Signature

```
public Cookie(String name, String value, String path, Integer maxAge, Boolean isSecure)
```

#### Parameters

***name***

Type: String

The cookie name. It can’t be `null`.

***value***

Type: String

The cookie data, such as session ID.

***path***

Type: String

The path from where you can retrieve the cookie.

***maxAge***

Type: Integer

A number representing how long a cookie is valid for in seconds. If set to less than zero, a session cookie is issued. If set to zero, the cookie is deleted.

***isSecure***

Type: Boolean

A value indicating whether the cookie can only be accessed through HTTPS (`true`) or not (`false`).

## Cookie(name, value, path, maxAge, isSecure, SameSite)

Creates a new instance of the `Cookie` class using the specified name, value, path, and age, and settings for security and cross-domain behavior.

#### Signature

**Note**

Google Chrome 80 introduces a new default cookie attribute setting of `SameSite`, which is set to `Lax`. Previously, the `SameSite` cookie attribute defaulted to the value of `None`. When `SameSite` is set to `None`, cookies must be tagged with the `isSecure` attribute indicating that they require an encrypted HTTPS connection.

```
public Cookie(String name, String value, String path, Integer maxAge, Boolean isSecure, String
SameSite)
```

#### Parameters

***name***

Type: String

The cookie name. It can’t be `null`.

***value***

Type: String

The cookie data, such as session ID.

#### *maxAge*

Type: Integer

A number representing how long a cookie is valid for in seconds. If set to less than zero, a session cookie is issued. If set to zero, the cookie is deleted.

#### *isSecure*

Type: Boolean

A value indicating whether the cookie can only be accessed through HTTPS (`true`) or not (`false`).

#### *SameSite*

Type: String

The `SameSite` attribute on a cookie controls its cross-domain behavior. The valid values are `None`, `Lax`, and `Strict`. After the Chrome 80 release, a cookie with a `SameSite` value of `None` must also be marked secure by setting a value of `None; Secure`.

### See Also

- *Salesforce Spring '20 Release Notes:* Prepare for Google Chrome's Changes in SameSite Cookie Behavior That Can Break Salesforce Integrations
- *Chrome Platform Status*: Reject insecure SameSite=None cookies

## Cookie(name, value, path, maxAge, isSecure, SameSite, isHttpOnly)

Creates a new instance of the `Cookie` class using the specified name, value, path, age, and settings for security, cross-domain behavior, and JavaScript access.

### Signature

```
public Cookie(String name, String value, String path, Integer maxAge, Boolean isSecure, String
SameSite, Boolean isHttpOnly)
```

### Parameters

#### *name*

Type: String

The cookie name. It can't be `null`.

#### *value*

Type: String

The cookie data, such as session ID.

#### *path*

Type: String

The path from where you can retrieve the cookie.

#### *maxAge*

Type: Integer

A number representing how long a cookie is valid for in seconds. If set to less than zero, a session cookie is issued. If set to zero, the cookie is deleted.

#### *isSecure*

Type: Boolean

The `SameSite` attribute on a cookie controls its cross-domain behavior. The valid values are `None`, `Lax`, and `Strict`. After the Chrome 80 release, a cookie with a `SameSite` value of `None` must also be marked secure by setting a value of `None; Secure`.

#### *isHttpOnly*

Type: Boolean

A value indicating whether the HttpOnly attribute for the cookie is set (`true`) or not (`false`). If `true`, client-side JavaScript can't access the cookie.

#### See Also

- *MDN Web Docs*: Set-Cookie HTTP Response Header

## Cookie Methods

The following are methods for `Cookie`. All are instance methods.

- getDomain()
  Returns the name of the server making the request.
- getMaxAge()
  Returns a number representing how long the cookie is valid for, in seconds. If set to `< 0`, a session cookie is issued. If set to `0`, the cookie is deleted.
- getName()
  Returns the name of the cookie. Can't be `null`.
- getPath()
  Returns the path from which you can retrieve the cookie. If `null` or blank, the location is set to root, or "/".
- getSameSite()
  Returns the value for the `SameSite` attribute of the cookie.
- getValue()
  Returns the data captured in the cookie, such as Session ID.
- isSecure()
  Returns `true` if the cookie can only be accessed through HTTPS, otherwise returns `false`.
- isHttpOnly()
  Returns `true` if client-side JavaScript is forbidden from accessing the cookie; otherwise returns `false`.

### getDomain()

Returns the name of the server making the request.

#### Signature

```
public String getDomain()
```

#### Return Value

Type: String

### getMaxAge()

Returns a number representing how long the cookie is valid for, in seconds. If set to `< 0`, a session cookie is issued. If set to `0`, the cookie is deleted.

#### Signature

## getName()

Returns the name of the cookie. Can't be `null`.

#### Signature

```
public String getName()
```

#### Return Value

Type: String

## getPath()

Returns the path from which you can retrieve the cookie. If `null` or blank, the location is set to root, or “/”.

#### Signature

```
public String getPath()
```

#### Return Value

Type: String

## getSameSite()

Returns the value for the `SameSite` attribute of the cookie.

#### Signature

```
public String getSameSite()
```

#### Return Value

Type: String

#### See Also

- *web.dev*: SameSite Cookies Explained

## getValue()

Returns the data captured in the cookie, such as Session ID.

#### Signature

```
public String getValue()
```

#### Return Value

Type: String

## isSecure()

Returns `true` if the cookie can only be accessed through HTTPS, otherwise returns `false`.

#### Signature

## isHttpOnly()

Returns `true` if client-side JavaScript is forbidden from accessing the cookie; otherwise returns `false`.

### Signature

```
public Boolean isHttpOnly()
```

### Return Value

Type: Boolean

### See Also

- *MDN Web Docs*: Set-Cookie HTTP Response Header

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

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

Your Privacy Choices Responsible Disclosure Contact