

همانطور که از نامش پیداست jcookie یک پلاگین jquery است. این پلاگین به شما این اجازه را می‌دهد تا هر نوع داده ای را که مایل هستید از قبیل رشته‌ها، آرایه‌ها و object را در قالب json با رمزگذاری base 64 ذخیره نمایید. استفاده از این رمزگذاری باعث کوچکتر شدن حجم کوکی تا 70 درصد می‌شود. در این مقاله شما یاد می‌گیرید که چطور برای ذخیره و بازیابی کوکی از آن استفاده کرده و چگونه در یک زبان سمت سرور، مثل سی شارپ نیز کوکی مورد نظر را با همان فرمت بخوانید. جهت دانلود فایل jcookie به [اینجا](#) مراجعه کنید. **ذخیره کوکی**

برای ساخت یک کوکی به روش زیر اقدام می‌کنیم. استفاده از \$.jCookies دو خاصیت به نام‌های نام کوکی و مقدار کوکی را name & Value در دسترس ما می‌گذارد:

```
var d = new Date();
$.jCookies({
    name: 'dotnettips.info',
    value: { Title: 'ساخت کوکی با jcookie', Author: 'علی یگانه مقدم', Seen:
d.getDate(), Favorite: true }
});
```

همانطور که می‌بینید ذخیره اطلاعات توسط jcookie بسیار ساده و راحت بوده و هر نوع داده ای در آن به راحتی قابل ذخیره سازیست. برای مثال می‌توانید اطلاعات یک کلاس را خیلی راحت و سریع با آن ذخیره کنید. به طور پیش فرض تاریخ انقضای کوکی 27 روز بعد از ایجاد آن می‌باشد. در صورتی که تمایل دارید این تاریخ را تغییر دهید یکی از خاصیت‌های seconds, minutes, hours و days در دسترس شماست و مقادیری که جلوی آن‌ها به کار گرفته می‌شود باید نوع صحیح بوده و در صورتی که مقدار نامعتبر وارد شود خاصیت مورد نظر نادیده گرفته می‌شود.

```
$.jCookies ({ name : 'User', value : { username : 'Bob' , level : 5 }, minutes : 60 });
```

برای تغییر پیش فرض‌های ساخت کوکی مانند انقضای 27 روز به عدد پیش فرض خودتان فایل jcookies.js را باز کرده و تنظیمات پیش فرض آن را تغییر بدهید. برای تغییر دنبال کد زیر بگردید:

```
$.jCookies.defaults =
{
name : '',
value : '',
days : 27
}
```

بازیابی کوکی

برای بازیابی کوکی مجدداً از \$.jCookies استفاده می‌شود ولی تنها باید یک خاصیت get که نام کوکی هست را بنویسید:

```
var values = $.jCookies ({ get: 'dotnettips.info' });
```

در صورتی که نام کوکی‌ای که درخواست کرده اید وجود نداشته باشد یا اینکه تاریخ انقضای آن سر رسیده است و از سیستم کلاینت حذف شده است یا اینکه هنگام درخواست کوکی با خطایی مواجه شده باشد، مقدار برگشتی false خواهد بود و اگر نیاز دارید که بدانید آیا نوع برگشتی false به خاطر خطا بوده است یا خیر یک خاصیت نوع بول به نام error هم اضافه می‌شود:

```
var values = $.jCookies({ get: 'Rutabaga', error: true });
```

در صورتی که خطایی داده شود response مقدار values در مرورگر کروم به شکل زیر خواهد بود. در هر مرورگر نحوه نمایش خطا می‌تواند متفاوت باشد.

```
Error : {
  arguments : undefined,
  message : "Invalid base64 data",
  stack : "-",
```

```
type : undefined
}
```

بازیابی همه کوکی ها

در صورتی که به خاصیت get مقدار * را بدهید تمامی کوکی ها برگشت داده خواهند شد و به صورت آرایه ای از نام کوکی ها در دسترسی خواهند بود:

```
var values = $.jCookies({ get: '*' });
alert(values["dotnettips.info"].Title);
alert(values["data2"].Title);
```

حذف کوکی

نحوه کدنویسی حذف کوکی هم دقیقا مشابه خواندن کوکی است؛ با این تفاوت که به جای استفاده از خاصیت get از خاصیت erase استفاده می کنیم و با دادن نام کوکی به این خاصیت، کوکی حذف خواهد شد:

```
var value = $.jCookies({ erase: 'dotnettips.info' });
```

در صورتی که کوکی وجود داشته باشد، آن را حذف کرده و مقدار true را برگشت خواهد زد و در صورتی که کوکی وجود نداشته باشد مقدار false را بر میگرداند.

بازیابی کوکی در سمت سرور با سی شارپ

در این روش ما ابتدا با همان دستور معمولی دات نت یعنی `page.request.cookie` درخواست دریافت کوکی را می دهیم ولی از آنجا که در jcookie دو عمل روی داده ها صورت گرفته است باید دو کار اضافه تر را انجام داد: برگشت داده ها از حالت رمزگذاری base64 داده ها در فرمت json هستند و باید به حالتی قابل استفاده در محیط شی گرا تبدیل شوند.

برای بازگردانی از حالت base64 از کلاس و متد `Convert.FromBase64String` در فضای نام `system.convert` استفاده می کنیم که آرایه ای از نوع بایت را بر میگرداند و از `Encoding.UTF8.GetString` هم برای decode کردن آرایه به نوع رشته استفاده می کنیم. تا به اینجای کار داده های ما به صورت یک json خوانا با فرمت string درآمده است. برای دسترسی به داده های موجود در این فرمت باید آن ها را `Deserialize` کنیم که این کار را از طریق کلاس [JavaScriptSerializer](#) در فضای نام `System.Web.Script.Serialization` انجام می دهیم و از کلاس `دیکشنری` برای ذخیره داده های برگشتی استفاده می کنیم که نوع string برای نام خاصیت و نوع آجکت برای ذخیره مقدار خاصیت خواهد بود. یعنی برای بازگردانی اولین مثال بالا باید داده های در نوع `دیکشنری` به صورت زیر لیست شوند

ایجاد کوکی با jcookie	Title
علی یگانه مقدم	Author
2015/1/14	Seen
true	Favorite

```
byte[] from64 = Convert.FromBase64String(Page.Request.Cookies["dotnettips.info"].Value);
string json = Encoding.UTF8.GetString(from64);
Dictionary<string, object> article = new
JavaScriptSerializer().Deserialize<Dictionary<string, object>>(json);
Page.Response.Write("Title: " + (string)article["Title"]);
```

پشتیبانی از یونیکد

موقعی که من اولین مثال بالا را نوشتم و مقادیر را به صورت فارسی وارد کردم متوجه شدم که رشته های یونیکد را انکود نمی کند و در نتیجه زبان فارسی در آن پشتیبانی نمی شود. برای همین تغییراتی در فایل jz ایجاد کرده و عبارت value قبل از تبدیل به base64 را به صورت utf-16 انکود کردم و در هنگام خواندن کوکی هم به صورت utf-16 دیکود کردم و مشکل زبان فارسی هم در این حالت حل شد. البته کدی که اضافه کردم قابلیت های انکودینگ بیشتری هم دارد. فقط تنها مورد این هست که برای خواندن کوکی در سمت سرور باید یک تغییر کوچک یک کلمه ای بدهیم؛ باید کلمه UTF8 را به

Unicode که می‌شود همان UTF-16 در کد تغییر دهیم، که به کد زیر تغییر خواهد یافت:

```
byte[] from64 = Convert.FromBase64String(Page.Request.Cookies["dotnettips.info"].Value);
string json = Encoding.Unicode.GetString(from64);
Dictionary<string, object> article = new
JavaScriptSerializer().Deserialize<Dictionary<string, object>>(json);
Page.Response.Write("Title: " + (string)article["Title"]);
```

برای دریافت jcookie با پشتیبانی از زبان فارسی به [اینجا](#) مراجعه کنید.
کدهای بالا در فایل زیر قرار گرفته اند. [jcookie.zip](#)