

در برنامه‌های مبتنی بر وب رایج، معمولاً تبدیل تاریخ میلادی به شمسی در سمت سرور انجام می‌گیرد و تاریخ شمسی حاصل از تبدیل، به کاربر نمایش داده می‌شود. اما در برنامه‌های Single Page و یا به اختصار SPAها که کلاینت فقط با یک سری داده به فرمت JSON درگیر است، برای نمایش تاریخ شمسی به چه طریقی باید عمل کرد؟ آیا باید تاریخ را در سمت سرور به فرمت مورد نظر تبدیل کرد و یا در سمت کلاینت؟ همه‌ی این‌ها از جمله سوالاتی هست که به هنگام توسعه‌ی SPAها با آن‌ها حتماً درگیر خواهید شد.

شاید بتوان گفت که در SPAها، هدف این است که از بار سرور تا حد ممکن کم کرد و آن را در بین کلاینت‌ها توزیع کرد. در SPAها نقش اصلی سرور تامین داده هاست و بیشتر پردازش‌ها در صورت امکان در سمت کلاینت انجام می‌شود و می‌بینید که حتی رندر کردن HTML نیز به عهده‌ی قالب‌های سمت کلاینت است. البته هنوز هم می‌توان قبل از اینکه داده را به فرمت JSON سریالایز کرد، سمت سرور بر روی آن‌ها پیمایش انجام داده و تاریخ‌های میلادی را به شمسی تبدیل کرد که هدف ما این نیست و می‌خواهیم این کار را بر عهده‌ی مرورگر کاربر قرار دهیم. معرفی **moment.js**

برای کار با داده‌هایی از جنس تاریخ در سمت کلاینت، کتابخانه‌ی جاوا اسکریپتی قدرتمندی به نام **moment.js** وجود دارد. این کتابخانه دارای انواع و اقسام API برای نمایش و پردازش تاریخ هست. حتی می‌تواند relative time را نیز نمایش دهد. منظور از relative time این هست که به جای نمایش تاریخ، اختلاف آن را با زمان حال نمایش دهد. برای مثال می‌نویسند فلان پست در دو ساعت پیش ارسال شده و زمان دقیق ارسال پست را نمایش نمی‌دهد.

خوشبختانه برای افزودن تاریخ شمسی به این کتاب خانه، افزونه‌ای به نام **moment-jalaali** برای آن تدارک دیده شده است. کار با آن نیز بسیار راحت است. کافی است در همان APIهایی که برای فرمت کردن تاریخ در **moment.js** استفاده می‌کردید؛ یک **z** در ابتدای آن‌ها قرار دهید که مثال‌های کامل استفاده از آن را در مستندات آن می‌توانید مشاهده کنید.

نحوه‌ی استفاده از **moment.js** در **AngularJS** و **ASP.NET**

در ASP.NET فیلد هایی که از جنس **DateTime** هستند به شکل زیر به فرمت JSON سریالایز می‌شوند:

```
\Date(1374222094520)\
```

در **moment.js** احتیاج به کدنویسی برای **parse** کردن این نوع فرمت و تبدیل کردن آن به تاریخ وجود ندارد؛ چرا که **moment.js** به صورت **تو کار** از این نوع فرمت نیز پشتیبانی می‌کند و احتیاجی به کار اضافه‌تر نیست.

```
moment("/Date(1198908717056-0700)/"); // December 28 2007 10:11 PM
```

در AngularJS هر گاه قصد داشته باشیم که فرمت نمایش داده‌ها را تغییر دهیم از **filter**ها استفاده می‌کنیم. برای مثال فیلتر **uppercase** داده **name** را با حروف بزرگ نمایش می‌دهد.

```
{{ name | uppercase }}
```

حال برای تاریخ نیز می‌خواهیم چنین کاری انجام دهیم؛ بدین صورت که یک فیلتر سفارشی به شکل زیر تعریف کرده تا تاریخ میلادی را به صورت شمسی و با فرمت دلخواهی که می‌خواهیم نمایش دهد:

```
{{post.date | jalaliDate:'jYYYY/jMM/jDD hh:mm' }}
```

تعریف فیلتر **jalaliDate** نیز به شکل زیر است:

```
app.filter('jalaliDate', function () {
    return function (inputDate, format) {
        var date = moment(inputDate);
        return date.fromNow() + " " + date.format(format);
    };
});
```

```
}  
});
```

خروجی این فیلتر نیز به شکل "4 ماه پیش 03:10 07/12/1392" است و مشاهده می‌کنید که به کمک filterها در AngularJS انجام این گونه از کارها بسیار ساده و لذت بخش است.

توجه کنید که این فقط یک ایده‌ی ابتدایی و ساده از پیاده سازی فیلتر فوق است. قطعا با کمک APIهای متنوع momentjs و پارامترهای ورودی فیلتر، می‌توان فیلتری بسیار پیشرفته‌تر تعریف کرد. دریافت کدهای یک مثال پیاده سازی شده با استفاده از [moment-jalali-AngularJs.rar](#) کدهای فوق