عنوان: **آشنایی بیشتر با AngularJS Directive**
نویسنده: عزیزخانی
تاریخ: ۱۳۹۳/۰۳/۰۵ ۱۲:۵۵
آدرس: www.dotnettips.info
گروه‌ها: AngularJS

در مطلب آشنایی با Directiveها در AngularJS با نحوه‌ی ایجاد Directive آشنا شدیم. هدف از این مطلب، آشنایی بیشتر با Directive در AngularJS است؛ یکی از بهترین فریم ورک‌های جاوااسکریپتی، با قابلیت ایجاد کتابخانه‌هایی از کامپوننت‌ها که می‌توانند به HTML اضافه شوند .

کتابخانه‌های جاوااسکریپتی زیادی وجود دارند. به عنوان مثال Bootstrap یکی از محبوب‌ترین "front-end framework" ها است که امکان تغییر در ظاهر المنت‌ها را فراهم می‌کند و شامل تعدادی کامپوننت جاوااسکریپتی نیز می‌باشد. مشکل کار، در هنگام استفاده از کامپوننت ها است. شخصی که در حال توسعه‌ی HTML است باید در کد جاوااسکریپتی خود از jQuery استفاده کند و بعنوان مثال یک Popover را فعال یا غیر فعال کند و این، یک فرآیند خسته کننده و مستعد خطا است.

یک مثال ساده از Directives AngularJS و بررسی آن

```
var m = angular.module("myApp");

myApp.directive("myDir", function() {
  return {
  restrict: "E",
  scope: {
   name: "@",
   amount: "=",
   save: "&"
  },
  template:
   "<div>" +
   "  {{name}}: <input ng-model='amount' />" +
   "  <button ng-click='save()'>Save</button>" +
   "</div>",
  replace: true,
  transclude: false,
  controller: [ "$scope", function ($scope) { …  }],
  link: function (scope, element, attrs, controller) {…}
  }
});
```

به الگوی نامگذاری directive دقت کنید. پیشوند my شبیه به یک namespace است. بنابراین اگر یک Application از دایرکتیوهای قرار گرفته در Module های متفاوت استفاده کند، به راحتی می‌توان محل تعریف یک directive را تشخیص داد. این نام می‌تواند نشان دهنده‌ی این باشد که این directive را خودتان توسعه داده‌اید یا از یک directive توسعه داده شده توسط شخص دیگری در حال استفاده هستید. به هر حال این نحوه‌ی نام گذاری یک اجبار نیست و به عنوان یک پیشنهاد است.

سازنده directive یک شیء را با تعدادی خاصیت باز می‌گرداند که تمامی آنها در سایت AngularJS توضیح داده شده‌اند. در اینجا قصد داریم تا توضیحی مختصر در مورد کاری که این خصوصیات انجام می‌دهند داشته باشیم.

· restrict : تشخیص می‌دهد که آیا directive در HTML استفاده خواهد شد. گزینه‌های قابل استفاده ‘C’ ، ‘E’ ، ‘A’ برای class ، element ، attribute و یا comment است . پیش فرض ‘A’ برای attribute است. اما ما بیشتر علاقه به استفاده از ویژگی element برای ایجاد المنت‌های UI داریم.

· scope : ایجاد یک scope ایزوله که متعلق به directive است و موجب ایزوله شدن آن از scope صدا زننده directive می‌شود. متغیرهای scope پدر از طریق خصوصیات تگ directive ارسال می‌شوند. این ایزوله کردن زمانی کاربردی است که در حال ایجاد کامپوننت هایی با قابلیت استفاده مجدد هستیم، که نباید متکی به scope پدر باشند. شیء scope در directive نام و نوع

متغیرهای scope را تعیین می‌کنند. در مثال بالا سه متغیر برای scope تعریف شده است:

- (name: "@") (by value, one-way : علامت @ مشخص می‌کند که مقدار متغیر ارسال می‌شود. Directive یک رشته را دریافت می‌کند که شامل مقدار ارسال شده از scope پدر می‌باشد. Directive می‌تواند از آن استفاده کند، اما نمی‌تواند مقدار آن را در scope پدر تغییر دهد.

- (amount: "=") (by reference, two-way : علامت = مشخص می‌کند این متغیر با ارجاع ارسال می‌شود. Directive یک ارجاع به مقدار متغیر در scope اصلی دریافت می‌کند. مقدار می‌تواند هر نوع داده ای، شامل یک شیء complex یا یک آرایه باشد. Directive می‌تواند مقدار را در scope پدر تغییر دهد. این نوع متغیر، زمانیکه نیاز باشد directive مقدار را در scope پدر تغییر دهد، استفاده می‌شود.

- (save: "&") (expression : علامت & مشخص می‌کند این متغیر یک expression را که در scope پدر اجرا می‌شود، نگهداری می‌کند. اکنون directive قابلیت انجام کارهایی فراتر از تغییر یک مقدار را دارد. به عنوان مثال می‌توان یک تابع را از scope پدر فراخوانی و نتیجه‌ی اجرا را دریافت کرد.

· template : الگوی رشته ای که جایگزین المنت تعریف شده می‌شود. فرآیند جایگزینی تمامی خصوصیات را از المنت قدیمی به المنت جدید انتقال می‌دهد. به نحوه استفاده از متغیرهای تعریف شده در scope ایزوله دقت کنید. این مورد به شما امکان تعریف directive های macro-style را می‌دهد که نیاز به کد اضافه‌ای، ندارند. اگرچه در بیشتر موارد الگو یک تگ ساده است که از کدهای link که در زیر توضیح داده شده است استفاده می‌کند.

· replace : تعیین می‌کند که آیا الگوی directive باید جایگزین المنت شود. مقدار پیش فرض false است.

· transclude : تعیین کننده این است که محتوای directive باید در المنت کپی شود یا خیر. در مثال زیر المنت tab شامل المنت‌های HTML دیگر است پس transclude برابر true است.

```
<body ng-app="components">
  <h3>BootStrap Tab Component</h3>
    <tabs>
       <pane title="First Tab">
          <div>This is the content of the first tab.</div>
       </pane>
       <pane title="Second Tab">
          <div>This is the content of the second tab.</div>
       </pane>
    </tabs>
</body>
```

· link : این تابع بیشتر منطق directive را شامل می‌شود. Link وظیفه دستکاری DOM ، ایجاد event listener ها و... را دارد. تابع Link پارامترهای زیر را دریافت می‌کند:

- scope : ارجاع به scope ایزوله شده directive دارد.

- element : ارجاع به المنت‌های DOM که directive را تعریف کرده اند. تابع link معمولا برای دستکاری المنت از jQuery استفاده می‌کند. (یا از **jqLite** Angular's در صورتی که jQuery بارگذاری نشده باشد)

- controller : در مواقعی که از دایرکتیوهای تو در تو استفاده می‌شود کاربرد دارد. این پارامتر یک directive فرزند با ارجاعی به پدر را فراهم می‌کند، بنابراین موجب ارتباط directive ها می‌شود.

به عنوان مثال، این directive که پیاده سازی bootstrap tab را انجام داده است، می‌توانید مشاهده نمایید.

موفق باشید