

در این مقاله قصد داریم با استفاده از ماژول [Angular-Translate](#) امکان ایجاد یک سیستم چند زبانه را تشریح کنیم. angular-translate یک ماژول توسعه داده شده AngularJS میباشد که با استفاده از i18n و l10n، قابلیت چند زبانه را به صورت Lazy Loading برای شما فراهم می‌کند. شما میتوانید با خط فرمان زیر، در بخش package-manager، کتابخانه‌های مربوط به angular-translate را به نرم افزار خود اضافه نمایید:

Install-Package AngularTranslate

تکه کد زیر یک مثال ساده از angular translate است که در صفحه‌ی اصلی این ماژول قرار داده شده است.

```
var app = angular.module('at', ['pascalprecht.translate']);

app.config(function ($translateProvider) {
  $translateProvider.translations('en', {
    TITLE: 'Hello',
    FOO: 'This is a paragraph.',
    BUTTON_LANG_EN: 'english',
    BUTTON_LANG_DE: 'german'
  });
  $translateProvider.translations('de', {
    TITLE: 'Hallo',
    FOO: 'Dies ist ein Paragraph.',
    BUTTON_LANG_EN: 'englisch',
    BUTTON_LANG_DE: 'deutsch'
  });
  $translateProvider.preferredLanguage('en');
});

app.controller('Ctrl', function ($scope, $translate) {
  $scope.changeLanguage = function (key) {
    $translate.use(key);
  };
});
```

با نگاهی ساده به تکه کد فوق می‌توان مراحل افزودن این ماژول و استفاده از آن را به صورت زیر ساده کرد:

1. وابستگی pascalprecht.translate را در بخش angular.module قرار می‌دهیم.
2. در بخش app.config وابستگی \$translateProvider را تزریق می‌کنیم.
3. با استفاده از رویه‌ی translations زبان‌های مختلف را به همراه لیبل آنها اضافه می‌نماییم. توجه کنید که امکان خواندن این ریسورس‌ها از فایل txt و json نیز وجود دارد.
4. با استفاده از رویه‌ی preferredLanguage زبان پیش فرض سیستم تعیین می‌گردد.
5. در کنترلر فراخوانده‌ی تغییر زبان، باید وابستگی \$translate را اضافه نماییم.
6. با استفاده از رویه‌ی use زبان مورد نظر را تغییر می‌دهیم. تمامی مراحل فوق در قالب یک [پروژه نمونه در Plunker](#) قرار داده شده است. در بخش بعدی به بررسی اجمالی قابلیت‌های این ماژول خواهیم پرداخت.