

در Angular می شود یک سری Template و ساختار از پیش تعریف شده داشت و در هر زمان که نیاز بود مدلی را به آنها پاس داد و نمای HTML مورد نظر را تحویل گرفت. بطور مثال در فرم سازها یا همان فرم‌های داینامیک ما نیاز داریم که مدل یک فرم (مثلا در فرمت JSON) را برای View ارسال کنیم و با استفاده از توانایی‌های Angular بتوانیم فرم مورد نظر را نمایش دهیم و در صورت امکان تغییر دهیم. ViewModel فرم شما در MVC میتواند چیزی شبیه این باشد

```
public class Form
{
    public string Name { get; set; }
    public string Title { get; set; }
    public List<BaseElement> Elements { get; set; }
}

public abstract class BaseElement
{
    public string Name { get; set; }
    public string Title { get; set; }
}

public class Section : BaseElement
{
    public List<TextBox> Elements { get; set; }
}

public class TextBox : BaseElement
{
    public string Value { get; set; }
    public string CssClass { get; set; }
}
```

یک کنترلر هم برای مدیریت فرم ایجاد میکنیم

```
public class FormBuilderController : Controller
{
    //
    // GET: /FormBuilder/

    public ActionResult Index()
    {
        var form = new Form();
        var section = new Section() { Title = "Basic Info", Name = "section01" };
        section.Elements.Add(new TextBox() { Name = "txt1", Title = "First Text Box" });
        form.Elements.Add(new TextBox() { Name = "txt1", Title = "Second Text Box" });
        var formJson = JsonConvert.SerializeObject(form);
        return View(formJson);
    }
}
```

در این کنترلر ما تنها یک اکشن داریم که در آن یک فرم خام ساده ایجاد کرده و سپس با استفاده از کتابخانه Json.net آنرا سریال و تبدیل به فرمت Json می‌کنیم و سپس آنرا برای View ایی که از Angular قدرت گرفته است، ارسال می‌نمائیم. پیاده سازی View با Angular به اشکال گوناگونی قابل پیاده سازی و استفاده است که در [اینجا](#) و [اینجا](#) می‌توانید ببینید. اما برای اینکه مشکل کنترلرهای تودرتو (Section) را حل کنید باید بصورت بازگشتی Template را فراخوانی کنید.

```
<script type="text/ng-template" id="ElementTemplate">
  <div ng-if="control.Type == 'JbSection'">
    <h2>{{control.Title}}</h2>
    <ul>
      <li ng-repeat="control in control.Elements" ng-include="'ElementTemplate'"></li>
    </ul>
  </div>
</script>
```

```
<script type="text/ng-template" id="element.html">
  {{data.label}}
  <ul>
    <li ng-repeat="element in data.elements" ng-include="'element.html'"></li>
  </ul>
</script>

<ul ng-controller="NestedFormCtrl">
  <li ng-repeat="field in formData" ng-include="'element.html'"></li>
</ul>
```

در اینجا صفحه element.html یک صفحه بیرونی است که Template ما در آن قرار دارد.