عنوان: **تبدیل یک View به رشته و بازگشت آن به همراه نتایج JSON حاصل از یک عملیات Ajax ایی در ASP.NET MVC**
نویسنده: تایماز قبادی
تاریخ: ۱۳۹۳/۱۱/۲۳ ۱۸:۴۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, JSON

ممکن است بخواهیم در پاسخ یک تقاضای Ajax ای ی، اگر عملیات در سمت سرور با موفقیت انجام شد، خروجی یک Controller action را به کاربر نهایی نشان دهیم. در چنین سناریویی لازم است که بتوانیم خروجی یک action را بصورت رشته برگردانیم. در این مقاله به این مسئله خواهیم پرداخت .
فرض کنید در یک سیستم وبلاگ ساده قصد داریم امکان کامنت گذاشتن بصورت Ajax را پیاده سازی کنیم. یک ایده عملی و کارآ این است: بعد از اینکه کاربر متن کامنت را وارد کرد و دکمه‌ی ارسال کامنت را زد، تقاضا به سمت سرور ارسال شود و اگر سرور پیغام موفقیت را صادر کرد، متن نوشته شده توسط کاربر را به کمک کدهای JavaScript و در همان سمت کلاینت بصورت یک کادر کامنت جدید به محتوای صفحه اضافه کنیم. بنده در اینجا برای اینکه بتوانم اصل موضوع مورد بحث را توضیح دهم، از یک سناریوی جایگزین استفاده می‌کنم؛ کاربر موقعیکه دکمه ارسال را زد، تقاضا به سرور ارسال میشود. سرور بعد از انجام عملیات، تحت یک شی JSON هم نتیجه‌ی انجام عملیات و هم محتوای HTML نمایش کامنت جدید در صفحه را به سمت کلاینت ارسال خواهد کرد و کلاینت در صورت موفقیت آمیز بودن عملیات، آن محتوا را به صفحه اضافه می‌کند.

با توجه به توضیحات داده شده، ابتدا یک شیء نیاز داریم تا بتوانیم توسط آن نتیجه‌ی عملیات Ajax ایی را بصورت JSON به سمت کلاینت ارسال کنیم:

```
public class MyJsonResult
{
  public bool success { set; get; }
  public bool HasWarning { set; get; }
  public string WarningMessage { set; get; }
  public int errorcode { set; get; }

  public string message {set; get; }
  public object data { set; get; }
 }
```

سپس به متدی نیاز داریم که کار تبدیل نتیجه‌ی action را به رشته، انجام دهد:

```
public static string RenderViewToString(ControllerContext context,
    string viewPath,
    object model = null,
    bool partial = false)
{
    ViewEngineResult viewEngineResult = null;
    if (partial) viewEngineResult = ViewEngines.Engines.FindPartialView(context, viewPath);
    else viewEngineResult = ViewEngines.Engines.FindView(context, viewPath, null);
    if (viewEngineResult == null) throw new FileNotFoundException("View cannot be found.");
    var view = viewEngineResult.View;
    context.Controller.ViewData.Model = model;
    string result = null;
    using(var sw = new StringWriter()) {
        var ctx = new ViewContext(context, view, context.Controller.ViewData,
context.Controller.TempData, sw);
        view.Render(ctx, sw);
        result = sw.ToString();
    }
    return result;
}
```

در اینجا موتور View را بر اساس اطلاعات یک View، مدل و سایر اطلاعات Context جاری کنترلر، وادار به تولید معادل رشته‌ای آن می‌کنیم.

فرض کنیم در سمت Controller هم از کدی شبیه به این استفاده میکنیم:

```
public JsonResult AddComment(CommentViewModel model) {
    MyJsonResult result = new MyJsonResult() {
        success = false;
    };
    if (!ModelState.IsValid) {
        result.success = false;
        result.message = "لطفاً اطلاعات فرم را کامل وارد کنید";
        return Json(result);
    }
    try {
        Comment theComment = model.toCommentModel();
        //EF service factory
        Factory.CommentService.Create(theComment);
        Factory.SaveChanges();
        result.data = Tools.RenderViewToString(this.ControllerContext, "/views/posts/_AComment", model,
true);
        result.success = true;
    } catch (Exception ex) {
        result.success = false;
        result.message = "اشکال زمان اجرا";
    }
    return Json(result);
}
```

و در سمت کلاینت برای ارسال Form به صورت Ajax ایی خواهیم داشت:

```
@using (Ajax.BeginForm("AddComment", "posts",
new AjaxOptions()
{
   HttpMethod = "Post",
   OnSuccess = "AddCommentSuccess",
   LoadingElementId = "AddCommentLoading"
}, new { id = "frmAddComment", @class = "form-horizontal" }))
{
    @Html.HiddenFor(m => m.PostId)
    <label for="fname">@Texts.ContactName</label>
    <input type="text" id="fname" name="FullName" class="form-control" placeholder="@Texts.ContactName
">
    <label for="email">@Texts.Email</label>
    <input type="email" id="InputEmail" name="email" class="form-control" placeholder="@Texts.Email">
    <br><textarea name="C_Content" cols="60" rows="10" class="form-control"></textarea><br>
    <input type="submit" value="@Texts.SubmitComments" name="" class="btn btn-primary">
    <div class="loading-mask" style="display:none">@Texts.LoadingMessage</div>
}
```

در اینجا در صورت موفقیت آمیز بودن عملیات، متد جاوا اسکریپتی AddCommentSuccess فراخوانی خواهد شد.
باید توجه شود Texts در اینجا یک Resource هست که به منظور نگهداری کلمات استفاده شده در سایت، برای زبانهای مختلف استفاده می‌شود (رجوع شود به مفهوم بومی سازی در Asp.net) .

و در قسمت script ها داریم:

```
<script type="text/javascript">
  function AddCommentSuccess(jsData) {
   if (jsData && jsData.message)
    alert(jsData.message);
   if (jsData && jsData.success) {
    document.getElementById("frmAddComment").reset();
      //افزودن کامنت جدید ساخته شده توسط کاربر به لیست کامنتهای صفحه
    $("#divAllComments").html(jsData.data + $("#divAllComments").html());
   }
  }
</script>
```

متد AddCommentSuccess اطلاعات شیء JSON بازگشتی از کنترلر را دریافت و سپس پیام آن‌را در صورت موفقیت آمیز بودن عملیات، به DIV ایی با id مساوی divAllComments اضافه می‌کند.