عنوان: **خلاصه‌ای از روش‌های ارسال داده‌های سمت سرور به کدهای جاوا اسکریپتی در ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۱۳ ۱۳:۵
آدرس: www.dotnettips.info
گروه‌ها: JavaScript, MVC, JSON

روش‌های زیادی برای ارسال داده‌های سمت سرور تهیه شده در یک برنامه‌ی ASP.NET به کدهای سمت کاربر JavaScript ایی وجود دارند که تعدادی از مهم‌ترین‌های آن‌ها را در این مطلب بررسی خواهیم کرد.

روش اول: دریافت اطلاعات سمت سرور به کمک درخواست‌های Ajax

استفاده از Ajax یکی از روش‌های کلاسیک دریافت اطلاعات سمت سرور در کدهای جاوا اسکریپتی است.

```
<script type="text/javascript">
var products = [];
$(function() {
    $.getJSON("/home/products", function(response) {
        products = response.products;
    });
});
</script>
```

برای نمونه در اینجا با استفاده از امکانات jQuery، درخواست Ajax ایی به سرور ارسال شده و سپس نتیجه‌ی دریافتی، به آرایه‌ی جاوا اسکریپتی products نسبت داده شده‌است.
- مزایا: استفاده از Ajax، روشی بسیار متداول و شناخته شده‌است و به کمک انواع و اقسام روش‌های بازگشت JSON از سرور، می‌توان با آن کار کرد.
- معایب: درخواست Ajax، صرفا پس از بارگذاری اولیه‌ی صفحه به سمت سرور ارسال خواهد شد و در این بین، کاربر وقفه‌ای را مشاهده خواهد کرد. همچنین در اینجا بجای یک درخواست از سرور، حداقل دو درخواست باید ارسال شوند؛ یکی برای بارگذاری صفحه‌ی اصلی و دیگری برای دریافت اطلاعات Ajax ایی از سرور به صورت غیرهمزمان.

روش دوم: دریافت اطلاعات از یک فایل جاوا اسکریپتی خارجی

اطلاعات سمت کاربر را از یک فایل جاوا اسکریپتی خارجی الحاق شده‌ی به صفحه‌ی جاری نیز می‌توان تهیه کرد:

```
<script src="/file.js"></script>
```

یک چنین فایلی را می‌توان توسط کدهای سمت سرور نیز بازگشت داد و اطلاعات آن‌را تهیه و پر کرد. در این حالت فقط کافی است که ContentType شیء Response را از نوع application/javascript مشخص کنیم و سپس یک خروجی متنی جاوا اسکریپتی را از سمت سرور بازگشت دهیم.
این روش نیز تقریبا مانند حالت یک درخواست Ajax ایی کار می‌کند و اطلاعات مورد نیاز را در طی یک درخواست جداگانه، پس از بارگذاری صفحه‌ی اصلی، از سرور دریافت خواهد کرد. البته در حالت کار با Ajax، می‌توان در طی یک callback، نتیجه را دریافت کرد و سپس عکس العمل نشان داد؛ اما در اینجا callback ایی وجود ندارد.

روش سوم: استفاده از SignalR

در SignalR ابتدا سعی می‌شود تا با استفاده از Web Sockets ارتباطی ماندگار بین کلاینت و سرور برقرار شود و سپس در این حالت، سرور می‌تواند مدام اطلاعاتی، مانند تغییرات داده‌های خود را به سمت کاربر، جهت نمایش و یا محاسبات خاص خود ارسال کند. اگر حالت Web Socket میسر نباشد (توسط سرور یا کلاینت پشتیبانی نشود)، به حالت‌های دیگری مانند server events, forever frames, long polling سوئیچ خواهد کرد. اطلاعات بیشتر

روش چهارم: قرار دادن اطلاعات سمت سرور در کدهای HTML صفحه

روش متداول دیگری جهت تامین اطلاعات جاوا اسکریپتی سمت کاربر، قرار دادن آن‌ها در ویژگی‌های data-* ارائه شده در HTML5 است.

```
<ul>
@foreach (var product in products)
{
  <li id="product@product.Id" data-rank="@product.Rank">@product.Name</li>
}
</ul>
```

در اینجا لیستی از محصولات، به صورت متداولی از کنترلر دریافت گردیده و سپس ویژگی data-rank هر ردیف نمایش داده شده نیز توسط این اطلاعات سمت سرور مقدار دهی شده‌اند.
اکنون برای دسترسی به مقدار data-rank سطری مانند product1، در کدهای جاوا اسکریپتی صفحه می‌توان نوشت:

```
<script type="text/javascript">
var product1Rank = $("#product1").data("rank");
</script>
```

این روش برای قرار دادن اطلاعات ثابت اشیاء، روش مرسومی است.

روش پنجم: قرار دادن اطلاعات سمت سرور در کدهای جاوا اسکریپتی صفحه

این روش همانند روش چهارم است، با این تفاوت که اینبار اطلاعات مورد نیاز، مستقیما به یک متغیر جاوا اسکریپتی انتساب داده شده‌است:

```
<script type="text/javascript">
var product1Name = "@product1.Name";
</script>
```

مزیت این روش، عدم ارسال درخواست اضافه‌تری به سرور برای دریافت اطلاعات مورد نیاز است. مقدار product1Name در همان بار اول رندر صفحه از سمت سرور، مشخص می‌گردد.

روش ششم: انتساب یک شیء دات نتی به یک متغیر جاوا اسکریپتی

این روش همانند روش پنجم است، با این تفاوت که اینبار قصد داریم بجای یک مقدار ثابت رشته‌ای یا عددی، برای مثال، آرایه‌ای از اشیاء را به یک متغیر جاوا اسکریپتی انتساب دهیم. در اینجا ابتدا اطلاعات مورد نظر را به فرمت JSON تبدیل می‌کنیم:

```
//سمت سرور
[HttpGet]
public ActionResult Index()
{
    var array = new[]
    {
        "Afghanistan",
        "Albania",
        "Algeria",
        "American Samoa",
        "Andorra",
        "Angola",
        "Anguilla",
        "Antarctica",
        "Antigua and/or Barbuda"
     };
     ViewBag.JsonString = new JavaScriptSerializer().Serialize(array);
     return View();
}
```

سپس توسط متد Html.Raw می‌توان این رشته‌ی JSON را که اکنون حاوی آرایه جاوا اسکریپتی سمت سرور است، به یک متغیر جاوا اسکریپتی نسبت داد:

```
//سمت کلاینت
<script type="text/javascript">
var jsonArray = @Html.Raw(@ViewBag.JsonString);
</script>
```

استفاده از Html.Raw در این حالت از این جهت ضروری است که اطلاعاتی مانند [] به صورت encode شده در سمت کاربر نمایش داده نشوند؛ چون Razor به صورت پیش فرض اطلاعات را Encode می‌کند.
و یا اینکار را به صورت خلاصه به شکل زیر نیز می‌توان در سمت کاربر انجام داد:

```
<script type="text/javascript">
  var model = @Html.Raw(Json.Encode(Model));
  // your js code here
</script>
```

در اینجا کار تبدیل اطلاعات مدل به JSON، در همان سمت RazorView انجام شده‌است.