

نویسنده



حسین احمدی

۳,۲۲۴

۴۴۸

۳۵۰

۲۸۱

حسین احمدی ، بنیانگذار TOSINSO ، توسعه دهنده وب و برنامه نویس ، بیش از ۱۲ سال سابقه فعالیت حرفه ای در سطح کلان ، مشاور ، مدیر پروژه و مدرس نهادهای مالی و اعتباری ، تخصص در پلتفرم دات نت و زبان سی شارپ ، طراحی و توسعه وب ، امنیت نرم افزار ، تحلیل سیستم های اطلاعاتی و داده کاوی ...



تخصص داری؟

میخواهی در آمد داشته باشی؟

توسینسو رو انتخاب کن

هم بازاریابی کن

هم تدریس کن

هم برند شخصیت و بساز

جشنواره تخفیف های زمستانه



نو سینسو

آموزش آنلاین به همراه پشتیبانی

و خرید اقساطی

تا ۵۰ درصد تخفیف

نحوه ارسال درخواست ها به صورت Ajax در ASP.NET MVC با کمک jQuery



۲۱۱۶



یکی از قابلیت های کاربردی در برنامه های تحت وب، قابلیت Ajax است. Ajax که مخفف Asynchronous JavaScript and XML است، به شما این اجازه را می دهد تا درخواست های خود را بدون این که نیاز به بارگذاری مجدد کل صفحه باشد، به سمت سرور ارسال کرده و پاسخ مورد نظر را دریافت کنید. در این مطلب قصد داریم به شما نحوه ارسال درخواست ها را به صورت Ajax با کمک jQuery در برنامه ASP.NET آموزش دهم. برای استفاده از کدهای ذکر شده در این مطلب باید به پروژه ای که قصد دارید در آن درخواست های Ajax را ارسال کنید، jQuery را اضافه کنید. برای اینکار می توانید از سایت jQuery فایل های مربوطه را دانلود کرده یا با کمک Nuget بسته jQuery را با دستور زیر نصب کنید:

```
install-package jQuery
```

بعد از نصب jQuery، یک Controller با نام Home به برنامه اضافه کنید که یک Action با نام Index در داخل آن تعریف شده باشد:

```
public class HomeController : Controller
{
    public ActionResult Index()
    {
        return View();
    }
}
```

محتویات View برای اکشن Index را به صورت زیر تغییر دهید:

```

<!DOCTYPE html>

<html>
<head>
  <title>Request Ajax with jQuery</title>
</head>
<body>
<input type="text" name="name" id="name"/>
<a href="#" id="get-message-link">Get Message</a>
<div id="message-container"></div>
<script src="~/scripts/jquery-3.0.0.js"></script>
</body>
</html>

```

دقت کنید که حتماً فایل jquery به صفحه شما اضافه شده باشد. کاری که قصد داریم انجام دهیم، ارسال مقدار وارد شده در input به سمت سرور و دریافت یک پیام مبتنی بر نام وارد شده و سپس نمایش آن در div ایست که id آن message-container است. برای ادامه، در ادامه اکشن با نام GetMessage را به HomeController به صورت زیر اضافه می کنیم:

```

public class HomeController : Controller
{
  public ActionResult Index()
  {
    return View();
  }

  public ActionResult GetMessage(string name)
  {
    return Content("Welcome dear " + name);
  }
}

```

برای قدم بعدی، باید درخواست را به سمت سرور و برای اکشن GetMessage ارسال کرده و نتیجه آن را گرفته و نمایش دهیم. Index را به صورت زیر تغییر می دهیم:

```

<!DOCTYPE html>

<html>
<head>
  <title>Request Ajax with jQuery</title>
</head>
<body>
<input type="text" name="name" id="name"/>
<a href="#" id="get-message-link">Get Message</a>
<div id="message-container"></div>

```

```

<script src="~/scripts/jquery-3.0.0.js"></script>
<script>
    $(function() {
        $('#get-message-link').on('click', function(e) {
            e.preventDefault();
            $.ajax({
                url: '@Url.Action("GetMessage", "Home")',
                data: { name: $('#name').val() },
                success: function(result) {
                    $('#message-container').html(result);
                }
            });
        });
    });
</script>
</body>
</html>

```

به اسکریپت نوشته شده دقت کنید، در jQuery برای ارسال درخواست ها به صورت ajax از دستور \$.ajax استفاده می کنیم. برای استفاده از این دستور، به عنوان پارامتر ورودی می بایست یک شیء که شامل پارامترهای درخواست ajax است را ارسال کنیم، این پارامترها عبارتند از:

۱. پارامتر url: آدرسی را مشخص می کنید که قصد داریم درخواست به صورت ajax برای آن ارسال شود. در این کد ما از Url.Action برای ایجاد آدرس اکشن GetMessage که در کنترلر Home قرار دارد استفاده کردیم.
۲. پارامتر data: این پارامتر که خود یک شیء است، مقادیری که می بایست به سمت سرور ارسال شوند را مشخص می کند، در این مثال ما یک پارامتر با نام name داریم که مقدار آن را از input با شناسه name خواندیم. اگر دقت کنید اکشن GetMessage یک پارامتر با نام name دارد که زمان دریافت درخواست، به صورت خودکار مقدار ارسال شده توسط درخواست ajax داخل آن قرار میگیرد. دقت کنید که نام پارامتر تعریف شده در بخش data برای ارسال درخواست، می بایست با نام پارامتر تعریف شده برای اکشن GetMessage یکسان باشد.
۳. پارامتر success: این پارامتر کدی را مشخص می کند که زمان دریافت پاسخ از سمت سرور باید اجرا شود، برای تابع success یک پارامتر ورودی مشخص کردیم و نام آن را result گذاشتیم، مقدار این پارامتر، دقیقاً همان مقداری است که اکشن GetMessage بر میگرداند. در داخل این متد مقدار result را داخل div ای با شناسه message-container قرار دادیم.

کدی که در این بخش با آن آشنا شدیم، یک کد بسیار ساده برای ارسال درخواست های ajax بود. دستور \$.ajax پارامترهای دیگری نیز قبول می کند، مانند beforeSend که می توان کدی را مشخص کرد تا قبل از ارسال درخواست اجرا شود یا error که در صورت وقوع خطا در طول ارسال درخواست، این کد اجرا شود. در صورت بروز خطا، کد success اجرا نخواهد شد. همچنین می توان بخشی را با نام complete در نظر گرفت که بعد از تکمیل درخواست اجرا شود، این کد چه در صورت موفقیت آمیز بودن درخواست و چه در صورت وقوع خطا اجرا خواهد شد. در ادامه کدی که از پارامترهای گفته شده استفاده شده را مشاهده می کنید:

```

<!DOCTYPE html>

<html>
<head>
    <title>Request Ajax with jQuery</title>
</head>

```

```
<body>
<input type="text" name="name" id="name"/>
<a href="#" id="get-message-link">Get Message</a>
<div id="message-container"></div>
<script src="~/scripts/jquery-3.0.0.js"></script>
<script>
$(function() {
    $('#get-message-link').on('click', function(e) {
        e.preventDefault();
        $.ajax({
            url: '@Url.Action("GetMessage", "Home")',
            data: { name: $('#name').val() },
            beforeSend: function () {
                alert('before submit request!');
            },
            success: function(result) {
                $('#message-container').html(result);
            },
            error: function() {
                alert('something was wrong!');
            },
            complete: function() {
                alert('request completed!');
            }
        });
    });
});
</script>
</body>
</html>
```

در این مطلب با نحوه ارسال و دریافت یک درخواست Ajax ساده در ASP.NET MVC آشنا شدیم. امیدوارم که این مطلب مورد توجه دوستان قرار گرفته باشد.

نویسنده: حسین احمدی

منبع: TOSINSO

[نظر](#) 

هیچ نظری ارسال نشده است! اولین نظر برای این مطلب را شما ارسال کنید...

[نظر شما](#) 

برای ارسال نظر باید وارد شوید.

از سرتاسر توسینسو

