

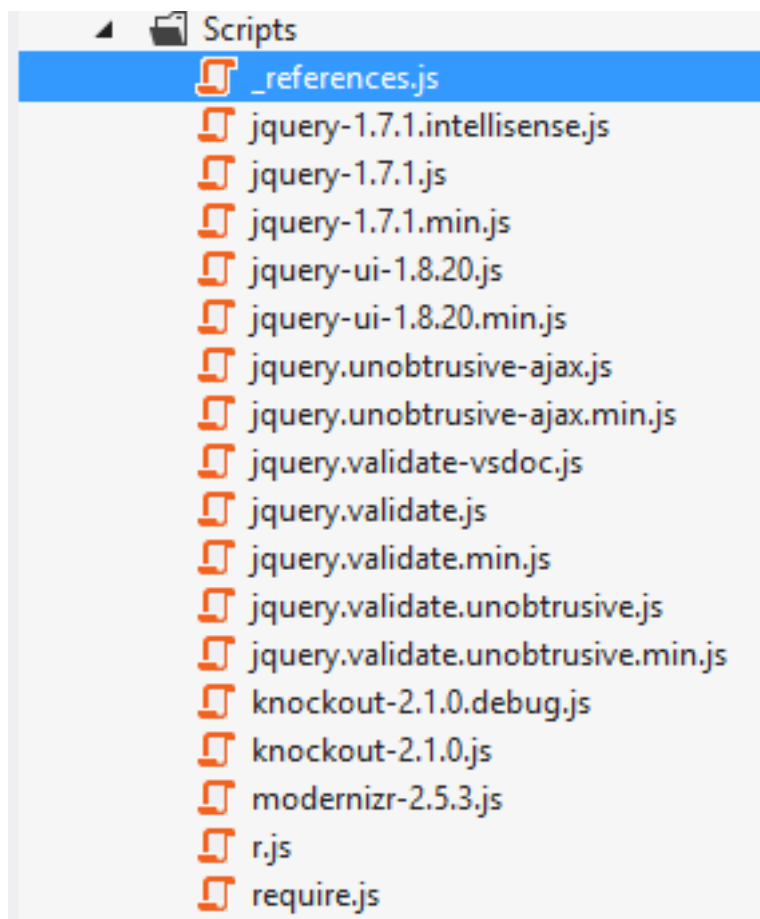
در پست قبلی با کلیات RequireJs آشنا شدید. در این به بررسی و پیاده سازی مثال قبل در قالب یک پروژه Asp.Net MVC می‌پردازم:

ابتدا یک پروژه Asp.Net MVC ایجاد کنید. در فولدر scripts تمام فایل‌های جاوااسکریپت پروژه قرار خواهند داشت. اگر قصد داشته باشیم که فایل‌های جاوااسکریپت سایر فریم ورک‌ها را استفاده نماییم (مثل ExtJs و backbone.js و...) برای طبقه بندی بهتر فایل‌ها، بهتر است که یک فولدر با نامی مشخص بسازیم و فایل‌های مورد نیاز را در آن قرار دهیم. البته اگر از nuget برای نصب این فریم ورک‌ها استفاده نماییم عموماً این کار انجام خواهد شد.

حال با استفاده از Package Manager Console و اجرای دستور زیر، اقدام به نصب requireJs کنید

```
PM> Install-package requireJs
```

ساختار فولدر scripts به صورت زیر خواهد شد (دو فایل r.js و require.js به این فولدر اضافه می‌شود)



یک فولدر به نام MyFiles در فولدر Scripts بسازید و فایل‌های purchase.js و product.js و credits.js در پروژه قبل را در آن کپی نمایید. کد فایل‌های پروژه قبل به صورت زیر بوده است:

purchase.js

```
define(["credits","products"], function(credits,products) {
    console.log("Function : purchaseProduct");
    return {
```

```

        purchaseProduct: function() {
            var credit = credits.getCredits();
            if(credit > 0){
                products.reserveProduct();
                alert('purchase done');
                return true;
            }
            alert('purchase cancel');
            return false;
        }
    });
});

```

در کد بالا از یک alert برای نمایش موفقیت یا عدم موفقیت عملیات استفاده کردم.

products.js

```

define(function(products) {
    return {
        reserveProduct: function() {
            console.log("Function : reserveProduct");
            return true;
        }
    }
});

```

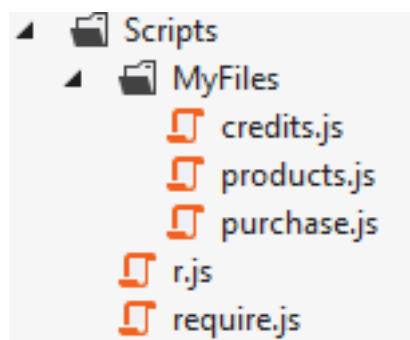
credits.js

```

define(function() {
    console.log("Function : getCredits");
    return {
        getCredits: function() {
            var credits = "100";
            return credits;
        }
    }
});

```

در نتیجه فایل‌های زیر به ساختار فولدر scripts اضافه شده است:



برای قدم بعدی، در متد RegisterBundles فایل bundleConfig پروژه دستور زیر را وارد نمایید:

```

bundles.Add( new ScriptBundle( "~/bundles/require" ).Include(
    "~/Scripts/require.js" ) );

```

کاملاً واضح است که نیاز به تغییر در فایل Layout_ پروژه نیز داریم؛ در نتیجه تغییرات زیر را در فایل اعمال نمایید:

```

@Scripts.Render("~/bundles/require")
<script type="text/javascript">
    require.config(
    {
        baseUrl:'Scripts/MyFiles'
    });
</script>
@Scripts.Render("~/bundles/jquery")
@RenderSection("scripts", required: false)

```

همان طور که مشاهده می‌کنید ابتدا با استفاده از دستور `Scripts.Render` فایل‌های `include` شده برای `requireJs` را در صفحه لود می‌کنید. سپس در تگ `scripts` که نوشته شده است با استفاده از دستور `require.config` مکان فایل‌های مورد نیاز را به فریم ورک `Require` معرفی می‌کنیم. این بدان معنی است که فریم ورک هر زمان که نیاز به لود یک وابستگی برای فایل‌های جاوااسکریپت داشته باشد، این مکان معرفی شده را جستجو خواهد کرد. حال برای استفاده و لود ماژول `purchase` در انتهای فایل `Index` فولدر `Home` تغییرات زیر را اعمال نمایید:

```

@section scripts
{
    <script type="text/javascript">
        require(['purchase'], function (purchase)
        {
            purchase.purchaseProduct();
        });
    </script>
}

```

در دستورات بالا با کمک دستور `require` (همان طور که در پست قبلی توضیح داده شد) ماژول `purchase` را لود می‌کنیم و بعد با فراخوانی تابع `purchaseProduct` به خروجی مورد نظر خواهیم رسید. در این جا من از دستور `alert` برای نمایش خروجی استفاده کردم! در نتیجه خروجی به صورت زیر خواهد بود:

