عنوان: **پر کردن مقادیر Enum در کلاینت**
نویسنده: مهدی پایروند
تاریخ: ۱۳۹۲/۰۲/۱۱ ۱۲:۱۵
آدرس: www.dotnettips.info
برچسب‌ها: JSON, jQuery, jQuery-Tips, Enum, dropdown

در برنامه‌های وب امروز نیازی به فراخوانی ثوابت که در طول حیات برنامه انگشت شمار تغیر میکنند نیست و با توجه به استفاده از فرامین و متدهای سمت کلاینت احتیاج هست تا این ثوابت بار اول لود صفحه به کلاینت پاس داده شوند.

میتوان در این گونه موارد از قابلیت‌های گوناگونی استفاده کرد که در اینجا ما با استفاده از یک فیلد مخفی و json مقدار را به کلاینت پاس میدهیم و در این مثال در سمت کلاینت نیز دراپ دان را با این مقادیر پر میکنیم:

```
public enum PersistType
    {
        Persistable = 1,
        NotPersist = 2,
        AlwaysPersist = 3
    }
```

لیست را باید قبل از پر کردن در فیلد مخفی به json بصورت serialize شده تبدیل کرد، برای این منظور از JavaScriptSerializer موجود در اسمبلی‌های دات نت در متد زیر استفاده شده:

```
public static string ConvertEnumToJavascript(Type t)
        {
            if (!t.IsEnum) throw new Exception("Type must be an enumeration");

            var values = System.Enum.GetValues(t);
            var dict = new Dictionary<int, string>();

            foreach (object obj in values)
            {
                string name = System.Enum.GetName(t, obj);
                dict.Add(Convert.ToInt32(System.Enum.Format(t, obj, "D")), name);
            }

            return new JavaScriptSerializer().Serialize(dict);
        }
```

با توجه به اینکه در سمت کلاینت مقدار json ذخیره شده در فیلد مخفی را میتوان به صورت آبجکت برخورد کرد پس یک متد در سمت کلاینت این آبجکت را در loop قراد داده و درمتغییری در فایل جاوا اسکریپت نگهداری میکنیم:

```
var Enum_PersistType = null;

function SetEnumTypes() {
    Enum_PersistType = JSON.parse($('#hfJsonEnum_PersistType').val());
}
```

و در هر قسمت که نیاز به مقدار enum بود با توجه به ایندکس مقدار را برای نمایش ازاین متغییر بیرون میکشیم:

```
function GetPersistTypeTitle_Concept(enumId) {
    return Enum_PersistType[enumId];
}
```

برای مثال در dropdown در سمت کلاینت این نوع استفاده شده و در حالتی از صفحه فقط برای نمایش عنوان آن احتیاج به دریافت آن از سمت سرور باشد میتوان از این روش کمک گرفت

و یا در سمت javascript میتوان با استفاده از jQuery مقادیر متغییر را در dropdown پر کرد.

```
function FillDropdown() {
    $("#ddlPersistType").html("");
    $.each(Enum_PersistType, function (key, value) {
        $("#ddlPersistType").append($("<option></option>").val(key).html(value));
    });
}
```

FillDropdownListOnClient.zip

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۱۲ ۱۱:۵۴

با تشکر از شما.

فایل Newtonsoft.Json.dll در پروژه شما هست. JavaScriptSerializer توکار دات نت ازش استفاده نمی‌کنه. فقط از اسمبلی System.Web.Extensions.dll هست که استفاده می‌کنه.

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۲/۰۲/۱۲ ۱۱:۵۷

ممنون از شما، برای ادامه این سری لازم میشده که در آینده اضافه میشه.
شما میتونید در صورت دلخواه کد قسمت serialize رو با این کتابخانه بنویسید:

```
//return new JavaScriptSerializer().Serialize(dict);
            return Newtonsoft.Json.JsonConvert.SerializeObject(dict);
```

نویسنده: سام ناصری
تاریخ: ۱۳۹۲/۰۲/۱۲ ۱۳:۴۴

به نظر من موضوع رو خیلی پیچیده کردی. برای تولید json لازم نیست که از کتابخانه خاصی استفاده کنی با همون استرینگ منیوپولیشن ساده هم میشه این کار را کرد:

```
public static class EnumHelper
{
    public static Dictionary<string,EnumValueType> ToDictionary<EnumType,EnumValueType>()
    {
        return
Enum.GetValues(typeof(EnumType)).Cast<EnumValueType>().ToDictionary(i=>Enum.GetName(typeof(EnumType),i)
,i=>i);
    }
    public static string ToJson<EnumType>()
    {
        return "{" + string.Join(",",
ToDictionary<EnumType,int>().Select(i=>string.Format(@"{{""{0}"":{1}}}",i.Key,i.Value)).ToArray())+ "}"
;
    }
}
```

کلاس ساده بالا به سادگی json مورد نیاز را تولید میکند.
در ضمن برای اینجکت کردنش به صفحه هم لازم نیست که از فیلد مخفی استفاده کنی. به جاش json را مستقیم در محل مورد نظر رندر کن:
در Asp.Net MVC Razor:

```
var x = @EnumHelper.ToJson<MyEnum()
```

در Asp.Net Web Forms :

```
var x = <%=EnumHelper.ToJson<MyEnum>()%>
```

همچنین همانطور که در مثالهای فوق نشان داده ام حتی لازم نیست از JSON.Parse استفاده کنی.
البته من اینها را بر اساس ذهنیاتم خیلی سریع نوشتم و کدهای فوق را تست نکرده ام که ببینم درست کار میکنند یا نه. اما منظورم این بود که بپرسم چرا از فیلد مخفی استفاده کردی و چرا از JSON.Parse استفاده کردی و اینکه چرا از JavaScriptSerializer استفاده کردی؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۱۲ ۱۴:۲۶

در حالت کلی بهتره که از JavaScriptSerializer استفاده بشه چون می‌تونه یک سری escape حروف خاص رو لحاظ کنه.

نویسنده: سام ناصری
تاریخ: ۱۳۹۲/۰۲/۱۲ ۱۵:۶

موضوع این مقاله درباره Enum است و نه ارسال داده‌های کلی به کلاینت. پس آیا فکر میکنید در اینجا چیزی برای اسکیپ شدن وجود داره؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۱۲ ۱۶:۳۲

در مورد پیچیدگی صحبت کردید. راه شما به مراتب پیچیده‌تر است از روش مطرح شده و خوانایی کمتری داره. به علاوه هدف از ارائه مقالات بهتره ارائه راه حل‌هایی باشه تا حد امکان عمومی تا این که یک سری هک خاص مطرح بشه فقط مختص به یک روش خاص که فقط در یک مساله مشخص قابل استفاده باشه. بعد هم اگر کسی این هک رو جای دیگری استفاده کرد، چون نمی‌دونه یک سری از کاراکترها باید escape بشن، در ضمن کار گیر میفته. دید دادن برای حل مساله اینجا شاید بیشتر مطرح باشه تا حل مساله با یک هک ساده که فقط همینجا قابل استفاده است. همچنین زمانیکه یک سری متد تست شده داخل فریم ورک هست چرا باید رفت سراغ هک؟

ضمنا در ASP.NET MVC نیاز دارید که یک Html.Raw رو هم اضافه کنید و گرنه اطلاعات درج شده در صفحه encode می‌شن و در متغیر جاوا اسکریپتی قابل استفاده نخواهند بود.

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۲/۰۲/۱۲ ۲۳:۸

در موردی مطلبی که آقای ناصری فرمودند باید بگم زمانیکه برنامه به سمت چند زبانه میره اهمبت پیدا میکنه که میتونید برای مثال مقدار یا لیستی از مقادیر متنی برای زبان خاصی رو با resource خودش به فیلد مخفی پاس بدید و در نمایش پیغام‌های مختلف سمت کلاینت اسنفاده کنید. مثل متن پیغام هایی که خاص ارتباط ajax میباشد که به زبان‌های مختلف ارائه کرد.

نویسنده: سام ناصری
تاریخ: ۱۳۹۲/۰۲/۱۳ ۴:۳۳

من کلاً نمیفهمم. در ضمن من سه تا سوال مطرح کردم(پاراگراف آخر کامنتم) که من باز هم نمیفهمم این جواب کدومشونه.