

چکیده:

Ajax یک تکنیک است که در آن از مجموعه ابزارهای مختلف (جاوا اسکریپت و XML) برای ایجاد پویایی در صفحات وب استفاده میشود.

Ajax سرواژه Asynchronous JavaScript And XML است که می‌شود جاوا- اسکریپت و XML ناهمگام! در مورد Ajax تا بخواهید کتاب هست و وب سایت اما من قصد دارم بدون درگیر شدن با جزئیات یک برنامه کوچک را با شما سهیم شوم: تاریخچه: ما یک فرم وب داشتیم که در آن باید در فیلد اول کد شعبه را وارد میکردیم و در فیلد دوم نام شعبه را مطابق شکل زیر

	کد شعبه
	نام و آدرس شعبه

روزی یکی از همکاران گرانقدر فرمودند کاری بکنید که پس از وارد کردن کد شعبه نام شعبه و آدرس آن ظاهر شود و طبیعتاً این کار باید بدون refresh شدن صفحه بانجام رسد. من هم گفتم این کار مثال کلاسیک Ajax است و آن را پیاده سازی کردم.

طرح مسئله:

فرض کنید یک جدول داریم دارای سه ستون 1- کد شعبه 2- نام شعبه 3- آدرس شعبه مطابق شکل زیر

1	branchaddress	branchname	branchcode
2	بلوار میرداماد - شماره ۲۰۰	مرکزی	201
3	شهرک قدس - بالاتر از تقاطع ایران زمین و دادمان	بلوار خوردین	202
4	سعادت آباد - میدان کاج	میدان سعادت آباد	203
5	جنب صائب شرقی - شماره ۲	خیابان شیراز شمالی	204
6	بین خیابان گلشهر و گلغام - شماره ۹۰	خیابان آفریقا	205
7	خیابان زرتشت غربی - جنب خیابان دوم مجتمع گیان - شماره ۷۷	خیابان زرتشت	206
8	جنب خیابان هارندران - شماره ۵۲	میدان هفت تیر	207
9	الهیة - برج A گلنار	الهیة	208
10	جنب خیابان ۲۷ - مقابل بیمارستان کسری - شماره ۲۵	خیابان الوید	209

هدف آنست که پس از وارد کردن کد سه رقمی شعبه در فرم وب بدون آنکه صفحه بارگزاری مجدد (refresh) گردد نام و آدرس شعبه ظاهر گردد.

اجزای پروژه:

این پروژه برای سادگی، بسیار خلاصه شده است 3 جزء اصلی آن عبارتند از :

1- پایگاه داده که برای سهولت یک فایل Access دارای یک جدول سه ستونی می باشد اطلاعات این جدول شامل کد، نام و آدرس برخی شعب تهران بانک پاسارگاد است.

2- یک فایل html شامل میانای کاربر (user interface) یا همان رابط کاربر که در آن کدهای جاوا اسکریپت قرار داده شده.

3- یک فایل ASPX برای پردازش سمت سرور شامل دستیابی به اطلاعات و ایجاد خروجی مناسب برای صفحه html .

■ فایل پایگاه داده:

یک فایل access است که در آنرا در فلدر private/db\_ با نام DBBranch.mdb ذخیره کرده ایم یعنی در root سایت معمولا یک فلدر private\_ هست که معمولا مجوز write را به آن می دهند من بطور کاملا سنتی و دلیخواه فایل های پایگاه داده ای را در آن قرار می دهم البته در یک زیرفلدر مشخص بنام db .

■ فایل html به نام Ajax.htm در root وب سایت ذخیره می شود. در این فایل که رابط یا میانای کاربر است دو فیلد متنی قرار می دهیم:

1- فیلد اول ورودی است بشرح زیر:

```
<input name="BrCodeIn" size="9" onkeyup="Trigger1();" maxlength="3" style="float: right">
```

این فیلد همانطور که می بینید حداکثر سه کاراکتر می گیرد و به محض آنکه دست شما از روی کلید برداشته شود تابع Trigger1 را فراخوانی میکند.

2- فیلد دوم خروجی است بشرح زیر:

```
<input name="OutPut" size="70" dir="rtl" readonly="true" style="float: right">
```

نکته آن بودن آنست که آنرا به یک فیلد خروجی شبیه کرده است.

دو تابع اصلی در فایل **Ajax.htm** وجود دارد یکی همان تابع **Trigger1** است که وقتی کلیدی زده می شود فراخوانی می شود و دیگری تابع **ajaxFunction** است که کد آن بشرح زیر است:

```
function Trigger1()
{
    var branchcode=document.Form1.BrCodeIn.value;

    if(branchcode.length==3)
    {
        ajaxFunction();
    }
}

function ajaxFunction()
{
    var xmlHttp;
    try
    {
        xmlHttp=new XMLHttpRequest();
    }
    catch (e)
    {
        try
        {
            xmlHttp=new ActiveXObject("Msxml2.XMLHTTP");
        }
        catch (e)
        {
            try
            {
                xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
            }
        }
    }
}
```

```

catch (e)
{
    alert("سازگاری مرورگر");
    return false;
}
}
}
}
//****
xmlHttpRequest.onreadystatechange=function()
{
    if((xmlHttpRequest.readyState > 0)&&(xmlHttpRequest.readyState < 4))
    {
        document.getElementById("img1").style.visibility = 'visible';
    }

    if(xmlHttpRequest.readyState==4)
    {
        document.Form1.OutPut.value=xmlHttpRequest.responseText;
        document.getElementById("img1").style.visibility = 'hidden';
    }
}

xmlHttpRequest.open("GET","http://babaktavakoli.com/AjaxSample.aspx?
BrCode="+document.Form1.BrCodeIn.value,true);
xmlHttpRequest.send(null);
}

```

خطوط اول تابع برای آنست که سازگاری مرورگر شما با Ajax چک شود و شیء **xmlHttpRequest** مناسب انتخاب شود همانطور که می بینید سه بار تلاش می شود و هر بار اگر شیء **xmlHttpRequest** ایجاد نشود سعی می شود که به نحو دیگری (سازگار با مرورگر دیگری) ایجاد شود اگر در نهایت ایجاد نشد با نمایش پیام خطا مناسب به کاربر اعلام میگردد که مرورگر وی از قابلیت Ajax پشتیبانی نمیکند. این قسمت از کد جاوا اسکریپت در اغلب برنامه های Ajax به طور استاندارد وجود دارد صرف نظر از اینکه برنامه قرار است چه کاری انجام بدهد.

پس از **//\*\*\*\*** تابعی قرار دارد که زمانی که وضعیت شیء **xmlHttpRequest** تغییر کند فراخوانی می شود (**xmlHttpRequest.onreadystatechange=function**) و چک میکند که

آیا داده آماده است در صورت آماده بودن (وضعیت=4) خروجی را روی فیلد نمایش خروجی قرار میدهد. اگر وضعیت بین یک تا سه باشد یعنی در حال ارسال و دریافت اطلاعات است و تصویر مناسبی را نمایش می دهد.

خط ماقبل آخر هم شیء را باز میکند در واقع به شیء میگوید ورودی و خروجی خود را از کجا بگیرد این ورودی بصورت یک پارامتر به یک فایل ASPX که برای پردازش سمت سرور بکار میرود ارسال میشود و پاسخ آن از طریق همان فایل در اختیار شیء قرار میگیرد.

#### ■ فایل AjaxSample.aspx

این فایل نیز در root ذخیره شده و کد آن از سه تابع اصلی تشکیل شده است:

تابع MyQueryMethod که از پایگاه دادهها (همان فایل Access) اطلاعات را خارج میکند.

تابع دوم GetBranchData نام دارد که از درون تابع page load فراخوانی می شود. وظیفه این تابع فراخوانی تابع MyQueryMethod و رساندن جواب به تابع بالاتر است در این تابع مدیریت استثنا بانجام می رسد و خروجی هم فرمت داده می شود.

تابع سوم page\_load است که به محض بارگزاری صفحه در سمت سرور اجرا می شود. در این تابع ابتدا ورودی سنجیده شده و در صورت معتبر بودن ورودی و امکان تبدیل آن به double ورودی به تابع GetBranchData جهت پردازش داده می شود.

```
System.Data.IDataReader MyQueryMethod(double branchcode) { // تابع جستجو در پایگاه داده
string connectionString = "Provider=Microsoft.Jet.OLEDB.4.0; Ole DB Services=-4; Data Source=" + Server.MapPath
("\\_private\\db\\DBBranch.mdb"); // رشته اتصال به پایگاه داده ها
System.Data.IDbConnection dbConnection = new System.Data.OleDb.OleDbConnection(connectionString);
// رشته پرسمان
string queryString = "SELECT [branch].* FROM [branch] WHERE ([branch].[branchcode] = @branchcode)";
System.Data.IDbCommand dbCommand = new System.Data.OleDb.OleDbCommand();
dbCommand.CommandText = queryString;
dbCommand.Connection = dbConnection;
// تعریف پارامتر و مقدار دهی
System.Data.IDataParameter dbParam_branchcode = new System.Data.OleDb.OleDbParameter();
```

```

dbParam_branchcode.ParameterName = "@branchcode";
dbParam_branchcode.Value = branchcode;
dbParam_branchcode.DbType = System.Data.DbType.Double; // نوع داده مضاعف
dbCommand.Parameters.Add(dbParam_branchcode);

dbConnection.Open();
System.Data.IDataReader dataReader = dbCommand.ExecuteReader(System.Data.CommandBehavior.CloseConnection);

return dataReader;
}

// تابع بارگذاری صفحه در سمت سرور
void Page_Load(object sender, EventArgs e)
{
Response.Clear();

string BrCode = Request.QueryString["BrCode"]; // گرفتن پارامتر ورودی از

if (BrCode != null) // اگر خالی نباشد
{
try
{
double d1 = double.Parse(BrCode); // تبدیل رشته به عدد
string strNameAddress = GetBranchData(d1); // فراخوانی تابع فرمت دهنده
Response.Write(strNameAddress); // نوشتن در خروجی
}
catch
{

}

}
Response.End();
}
// این تابع برای فرمت دادن به خروجی و مدیریت استثنا میباشد
private string GetBranchData(double d1)
{
try
{

```

```
System.Data.IDataReader dr = MyQueryMethod(d1);
if(dr.Read())
{
return dr["branchname"].ToString() + " - " + dr["branchaddress"].ToString(); // نام و آدرس به هم متصل میشوند
}
else
{
return "";
}

}
catch
{
return "";
}

}
```

خروجی برنامه در این آدرس قابل مشاهده است : (عددی بین 201 تا 290 وارد کنید)



بانک پاسپورت گاد



287

کد شعبه

نام و آدرس شعبه

خیابان نفت - بلوار میرداماد - خیابان نفت - جنب خیابا