عنوان	**توسعه نرم افزار برای سیستم عامل ویندوز موبایل، بخش هفتم: توسعه برای وب موبایل**
عنوان اصلی	**Windows Mobile App Development Part 7: Mobile Web Development**
کلمات کلیدی	C#, Win Mobile, AJAX, .NET, Dev
مؤلف	**mstruys, dougturn**
مرجع	http://www.codeproject.com
سطح	مبتدی
مترجم	مهدی عبداللهی http://m0911.wordpress.com
تاریخ انتشار	۱۶ فروردین ۱۳۸۹
تعداد صفحه	۸
مطالب مرتبط	توسعه ی نرم افزار برای سیستم عامل ویندوز موبایل، بخش اول: ایجاد نخستین برنامه توسعه ی نرم افزار برای سیستم عامل ویندوز موبایل، بخش دوم: شبیه ساز دستگاه و مدیریت شبیه ساز توسعه ی نرم افزار برای سیستم عامل ویندوز موبایل، بخش سوم: توسعه ی برنامه با WinForm توسعه ی نرم افزار برای سیستم عامل ویندوز موبایل، بخش چهارم: کنترل سفارشی و GPS توسعه ی نرم افزار برای سیستم عامل ویندوز موبایل، بخش پنجم: مقدمه ای بر SQL Server CE توسعه ی نرم افزار برای سیستم عامل ویندوز موبایل، بخش ششم: امنیت دستگاه و نصب نرم افزار
فایل های ضمیمه	

مقدمه

زمانی که برای ویندوز موبایل برنامه می نویسید یک پرسش را همان اول باید پاسخ دهید. آیا برنامه به صورت محلی روی دستگاه اجرا خواهد شد؟ با پاسخ به این پرسش مشخص خواهد شد که آیا نرم افزار تحت وب مناسب تر هست یا نه. اگر دستگاه مورد نظر فاقد اتصال مطمئن به اینترنت است، نرم افزار تحت وب اصلا به درد نخواهد خورد. به عبارت بهتر اگر دستگاه مورد نظر همیشه می تواند به شبکه ی اینترنت متصل شود، نرم افزار تحت وب، انتخاب بهتری است. یک امتیاز نرم افزار تحت وب این است که نیازی ندارد حتما روی دستگاه ویندوز موبایل اجرا شود و از طرفی هم روی طیف وسیعی از دستگاه ها قابل اجرا است. دستگاه های ویندوز موبایل جدید همگی از نرم افزار های تحت وب پشتیبانی می کنند و این را مدیون IEMobile هستند که بر پایه ی برخی قابلیت های اینترنت اکسپلورر ساخته شده است. در دستگاه های اخیر ویندوز موبایل، کاربران درست همانند دستگاه های دسکتاپ از اینترنت استفاده می کنند. تنها تفاوت فقط اندازه ی کوچک تر صفحه ی نمایش است. در ضمن از امکانات ASP.NET 3.5 و AJAX برای کمتر شدن تبادل داده بین کلاینت (دستگاه) و سرور استفاده می نمایند. این امکان به ویژه در اتصال اینترنت با سرعت کم مانند GPRS بسیار کار راه انداز است.

در این مقاله اطلاعاتی را در باره ی توسعه ی نرم افزار های تحت وب موبایل، تنظیم برنامه ی وب برای پشتیبانی از AJAX و در نهایت استفاده از کنترل های مرورگر داخل پروژه ی Smart Device ویژوال استودیو خواهید دید.

نرم افزار های تحت وب و پشتیبانی از ویندوز موبایل

دستگاه های ویندوز موبایل از همان اول مرورگر وب داشتند ولی تا پیش از ارایه ی ویندوز موبایل ۶، امکانات مرورگر مذکور بسیار محدود و بر مبنای نگارش های قدیمی تر اینترنت اکسپلورر بود. آخرین نگارش (در زمان تألیف این مقاله) مرورگر ویندوز موبایل با نام Internet Explorer Mobile 6 یک مرورگر با امکانات کامل است که کیفیت بالای کار با مرورگر وب در دستگاه های دسکتاپ را برای کاربر به ارمغان می آورد. این مرورگر بهترین سازگاری را با تمامی نگارش های مرورگر های ویندوز موبایل دارد. امکانات جدید و پیشرفته ی آن به کاربران کمک می کند که کار شان را به سرعت انجام دهند که در زیر به آن ها اشاره می کنیم:

- پشتیبانی از Jscript نگارش ۵.۷ اینترنت اکسپلورر ۸ که به توسعه دهنده گان امکان می دهد از قابلیت AJAX همانند دسکتاپ در ویندوز موبایل هم استفاده نمایند.
- اینترنت اکسپلورر موبایل ۶ امکان نمایش هر دو حالت ویژه ی موبایل و معمولی وب سایت ها را دارد.
- اینترنت اکسپلورر موبایل ۶ بسته به تنظیم کاربر می تواند به صورت مرورگر دسکتاپ یا مرورگر موبایل عمل کند.
- کاربران می توانند از امکانات صفحه ی لمسی و همچنین حالت های مختلف بزرگ نمایی (Zoom) استفاده نمایند.

حالت های مختلف نمایش وب سایت

زمانی که کاربر یک وب سایت را باز یا یک برنامه ی تحت وب را اجرا مـی نمایـد، مرورگـر کلاینـت توسـط یـک متغیـر رشـته ای UserAgent شناسایی می شود و بسته به مقدار متغر مذکور، نرم افزار تحت وب امکانات بیشتر یا کمتری (مانند پشتیبانی از AJAX) را بـرای کلاینـت فـراهم می کند. آخرین نگارش ویندوز موبایل، نگارشی از اینترنت اکسپلورر را استفاده می نماید که به راحتی می تواند به جای مرورگر دسکتاپ یا ویندوز موبایل شناسایی گردد. برای این کار کافی است که از منوی داخل اینترنت اکسپلورر موبایل گزینه ی مربوط به آن را انتخاب کنید.

شکل ۱: برنامه ی اینترنت اکسپلورر ۶ موبایل در حالت نمایش دسکتاپ و موبایل

در شکل ۱ نحوه سوئیچ کردن بین حالت های دسکتاپ و موبایل را می بینید. در بخش وسطی شکل ۱ اینترنت اکسپلورر در حالت ویـنـدوز موبایـل اجرا می شود که نمایش وب سایت را برای دستگاه ویندوز موبایل بهینه کرده است. در بخش سمت راست شکل ۱ نیز همان وب سایت را ایـن بـار در حالت دسکتاپ مشاهده می کنید. با اینکه صفحه ی نمایش کوکتر است لیکن کـاربر همـان حالـت مرورگـر ویـنـدوز دسـکتاپ را دارد. در یـک دستگاه با امکان صفحه ی لمسی جا به جایی در صفحه ی وب خیلی راحت تر انجام می شود.

تشخیص قابلیت های مرورگر

برای تشخیص امکانات مرورگر کلاینت می توانید از یک متغیر رشته ای UserAgent استفاده نمایید. مقادیر زیر برای متغیر مذکور صرف نظر از نوع دستگاه (استاندارد یا پروفشنال) برگردانده می شوند:

- Mozilla/4.0 (compatible; MSIE 6.0; Windows CE; IEMobile 8.12; MSIEMobile 6.0)
- Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)

که اولی برای حالت موبایل و دومی برای حالت دسکتاپ است.

اگر مرورگر در حالت دسکتاپ اجرا شود به طور خودکار امکانات AJAX را خواهد داشت. اگر مرورگر در حالت موبایل اجرا شود باید کلاینت به سرور اعلام کند که می تواند در حالت سازگار با ie5 اجرا شود. هر مرورگر روی ویندوز موبایل که نگارش IEMobile آن ۶.۱۲ یا بالاتر باشد این سازگاری را دارد. برای اعلام سازگای می توانید از قطعه کد زیر استفاده نمایید:

```
public static bool IsIEMobileWithAjaxSupport(string input)
{
   const string mobileBrowser = "IEMobile ";
   bool ajaxSupported = false;
   if (input.Contains(mobileBrowser))
   {
      string version = input.Substring(input.IndexOf(mobileBrowser) +
         mobileBrowser.Length);
      version = version.Remove(version.IndexOf(';'));
      string[] versionNr = version.Split(new char[] { '.' });
      int IEMobileMajor = Convert.ToInt32(versionNr[0]);
      int IEMobileMinor = Convert.ToInt32(versionNr[1]);
      ajaxSupported = IEMobileMajor > 6 || (IEMobileMajor == 6 && IEMobileMinor >= 12);
   }
   return ajaxSupported;
}
```

کد ۱: تشخیص امکانات اینترنت اکسپلورر موبایل

اگر نگارش اینترنت اکسپلورر موبایل ۶.۱۲ یا بالاتر باشد می توانیم توسط قطعه کد زیر امکانات مرورگر کلاینت را به سرور اطلاع دهیم. این کار را در بخشی از متد FrameworkInitialized داخل نرم افزار تحت وب انجام می دهیم. توجه داشته باشید که این کار یک بار انجام می شود. هر بار که پیغام postback دریافت شود برای نمونه به هنگام بارگذاری مجدد (refresh) بخشی از صفحه نیازی به مقدار دهی ClientTarget نیست چون سرور از مقدار آن مطلع شده است.

```
protected override void FrameworkInitialize()
{
    base.FrameworkInitialize();
    if (!IsPostBack)
    {
        if (IsIEMobileWithAjaxSupport(Request.UserAgent))
        {
            ClientTarget = "ie5";
        }
    }
}
```

کد ۲: ارسال اطلاعات مرورگر کلاینت به سرور

تشخیص ویندوز موبایل استاندارد از پروفشنال

از آن جا که هر دو نگارش استاندارد و پروفشنال ویندوز موبایل مقدار یکسان برای UserAgent برمی گردانند شما باید از روش دیگری برای تشخیص نوع دستگاه استفاده نمایید. در یک درخواست مرورگر اطلاعات اضافی نیز به سرور ارسال می گردد. یکی از این اطلاعات اندازه ی صفحه بر مبنای پیکسل است. تفاوت اصلی هم در این جا است. بر خلاف برنامه های Smart Client ، نرم افزار های تحت وب برای هر دو نوع استاندارد و پروفشنال ویندوز موبایل از کنترل های یکسانی استفاده می نمایند. دلیلش هم آن است که مرورگر اینترنت در هر دوی آن ها یکسان است. بنا بر این یک نرم افزار تحت وب در دستگاه ویندوز موبایل استاندارد می تواند دارای دکمه (button) باشد. شما به راحتی با دکمه های جابجایی می توانید روی دکمه ی مورد نظر تان رفته، با دکمه ی اکشن روی ان کلیک کنید. فیلد هدر ("UA-Pixels") تعداد واقعی پیکسل ها را بر می گرداند یعنی در دو حالت تفکیک گرافیکی بالا و معمولی مقادیر درست را بر می گرداند.

برای تنظیم مرورگر در حالت نمایش بهینه باید از برچسب MobileOptimized (tag) استفاده نمایید که اندازه ی صفحه ی نمایشی را که نرم افزار طبق آن توسعه داده شده است، مشخص نماید. کد زیر داخل برنامه برچسب MobileOptimized را بسته به عرض صفحه ی نمایش مقدار دهی می کند:

```
protected void Page_Load(object sender, EventArgs e)
{
   if (!IsPostBack)
   {
       if (IsIEMobileWithAjaxSupport(Request.UserAgent))
       {
           if (this.Header != null)
           {
               HtmlMeta mobileOptimized = new HtmlMeta();
               mobileOptimized.Name = "MobileOptimized";
               mobileOptimized.Content =
                  Request.Headers["UA-pixels"].Split(new char[] { 'X', 'x' })[0];
               Header.Controls.Add(mobileOptimized);
           }
       }
   }
}
```

کد ۳: تنظیم عرض سمت راست صفحه برای راحتی کاربر

با استفاده از اطلاعات Request.Header می توانید دستگاه های مختلف را تشخیص داده، حالت نمایش را برای همه نوع دستگاه بهینه کنید.

در شکل ۲ همان صفحه ی وب را می بینید که برای نمایش در دو دستگاه استاندارد و پروفشنال ویندوز موبایل بهینه شده است.

شکل ۲: بارگذاری صفحه در اندازه ی فونت متفاوت در دستگاه های مختلف

قطعه کد زیر نشان می دهد که چگونه می توانید اندازه ی قلم کنترل های وب را بسته به دستگاه و به ویژه عرض صفحه ی نمایش دستگاهی که برنامه روی آن اجرا می گردد، تغییر دهید. در این جا باید از امکانات ASP.NET 3.5 ممنون باشید و این که می توانید مشخصات کنترل ها را در فایل کد پس زمینه (code – behind) تغییر دهید.

```
protected void Page_Load(object sender, EventArgs e)
{
    string currentTime = DateTime.Now.ToLongTimeString();
    if (!IsPostBack)
    {
        if (IsIEMobileWithAjaxSupport(Request.UserAgent))
        {
            if (this.Header != null)
            {
                HtmlMeta mobileOptimized = new HtmlMeta();
                mobileOptimized.Name = "MobileOptimized";
                mobileOptimized.Content = Request.Headers["UA-pixels"].Split(
                    new char[] { 'X', 'x' })[0];
                Header.Controls.Add(mobileOptimized);
                pageWidth = Convert.ToInt32(mobileOptimized.Content);
            }
        }
    }
    if (pageWidth < 240)
    {
        Label1.Font.Size = System.Web.UI.WebControls.FontUnit.Small;
        Label2.Font.Size = System.Web.UI.WebControls.FontUnit.Small;
        Label3.Font.Size = System.Web.UI.WebControls.FontUnit.Small;
        Label4.Font.Size = System.Web.UI.WebControls.FontUnit.Small;
```

```
        Label5.Font.Size = System.Web.UI.WebControls.FontUnit.Small;
        DropDownList1.Font.Size = System.Web.UI.WebControls.FontUnit.Small;
        Button1.Font.Size = System.Web.UI.WebControls.FontUnit.Small;
        Menu1.Font.Size = System.Web.UI.WebControls.FontUnit.Small;
    }
}
```

کد ۴: تنظیم اندازه ی فونت متناسب با مدل دستگاه کلاینت

می بینید که اینترنت اکسپلورر موبایل آخرین تکنولوژی های موجود در نگارش دسکتاپ را تیز پشتیبانی می کند که اغلب امکانات ASP.NET 3.5 را در موبایل در اختیار تان قرار داده است.

ترکیب وب و اسمارت کلاینت

اگر یک برنامه ی اسمارت کلاینت (Smart Client) توسعه می دهید باید از یک کنترل مرورگر (Browser Control) استفاده کنید. کنترل مرورگر در آخرین نگارش ویندوز موبایل به شما امکان استفاده ار مرورگر وب در دسترس را می دهد و این یعنی امکان استفاده از ASP.NET 3.5 و امکانات AJAX از داخل یک برنامه ی اسمارت کلاینت. ترکیب وب و اسمارت کلاینت این امکان را می دهد که کاربر حتی بدون اتصال به شبکه نیز برنامه را اجرا نماید. اگر اتصال شبکه برقرار باشد کاربر می تواند از کنترل مرورگر استفاده نماید و مثلا به داده های بانک اطلاعاتی روی سرور دسترسی پیدا کند. این ویژگی مثلا در حالتی که چند کاربر به طور هم زمان روی داده های یکسانی در بانک اطلاعاتی کار می کنند از تداخل پیش گیری می نماید چون این کار کاملا توسط سرور انجام می شود. امتیاز دیگر این روش به هنگام پردازش های سنگین از داخل یک کنترل مرورگر خودش را نشان می دهد مثلا برای دریافت اطلاعات نقشه های زمینی. به جای استفاده از وب سرویس شما می توانید تمام عملیات سروری را داخل یک کنترل مرورگر انجام دهید.

کنترل مرورگر که در برنامه ی اسمارت کلاینت خود از آن استفاده می کنید همان مقدار UserAgent مرورگر مستقل اینترنت اکسپلورر موبایل را بر می گرداند. اما کنترل مرورگر را نمی توان به حالت موبایل معرفی کرد. از آن جا که دستگاه ویندوز موبایل صفحه نمایش کوچکی دارد ممکن است بخواهید صفحات خاصی را برای ترکیب کنترل مرورگر در داخل یک برنامه ی اسمارت کلاینت ایجاد نمایید. کد نمونه ی زیر یک برنامه ی کامل را تشان می دهد که اتصال شبکه اینترنت را تشخیص می دهد و در صورت وصل بودن دستگاه امکان نمایش یک صفحه ی وب حاوی اطلاعات در داخل یک کنترل مرورگر هست. اگر شبکه قطع شود کاربر می تواند آدرس های جدید را وارد نماید ولی نمی تواند صفحات آن ها را مرور نماید. البته این مثال بسیار ساده است و در نمونه ی برنامه ی واقعی امکانات بیشتری می توانید قرار دهید.

```
public partial class Form1 : Form
{
   private SystemState nrNetWorkConnections;
   private SystemState deviceCradled;
   public Form1()
   {
       InitializeComponent();
       nrNetWorkConnections = new SystemState(SystemProperty.ConnectionsCount);
       nrNetWorkConnections.Changed +=
          new ChangeEventHandler(nrNetWorkConnections_Changed);
       deviceCradled = new SystemState(SystemProperty.CradlePresent);
       deviceCradled.Changed += new ChangeEventHandler(deviceCradled_Changed);
   }
   private void Form1_Load(object sender, EventArgs e)
   {
       bool isCradled = (int)deviceCradled.CurrentValue != 0;
       int nrNetworks = (int)nrNetWorkConnections.CurrentValue;
       tbURL.Select(tbURL.Text.Length, 0);
       btnGo.Enabled = isCradled || nrNetworks > 0;
```

```
    }
    void deviceCradled_Changed(object sender, ChangeEventArgs args)
    {
        bool isCradled = (int)args.NewValue != 0;
        int nrNetworks = (int)nrNetWorkConnections.CurrentValue;
        btnGo.Enabled = isCradled || nrNetworks > 0;
    }
    void nrNetWorkConnections_Changed(object sender, ChangeEventArgs args)
    {
        bool isCradled = (int)deviceCradled.CurrentValue != 0;
        int nrNetworks = (int)args.NewValue;
        btnGo.Enabled = isCradled || nrNetworks > 0;
    }
    private void btnGo_Click(object sender, EventArgs e)
    {
        webBrowser1.Navigate(new Uri(tbURL.Text));
    }
}
```

کد ۵: یک برنامه ی کامل اسمارت کلاینت که اتصالات شبکه را تشخیص می دهد

ترکیب اسمارت کلاینت همراه با وب در وقع بهترین حالت از دو روش را در اختیار تان قرار می دهد. شما می توانید بهره وری کاربر همراه با توان پردازش را روی سرور قرار دهید. اگر داده ها به طور مرکزی ذخیره شوند می توانید یک کپی محلی از آن ها را (حتی از طریق وب سرویس) دانلود کنید. این روش مسأله ی تداخل را حل می کند و میزان داده های قابل دسترس روی دستگاه می تواند به همان مقدار مورد نیاز کاربر محدود گردد. این کار صرفا برای صرفه جویی فضای حافظه دستگاه نیست بلکه به لحاظ امنیتی نیز مهم است چون باعث می شود داده های حساس کمتری در هر زمان روی دستگاه قابل دسترس باشند.

شکل ۳: برنامه ی تحت وب از داخل یک مرورگر و به صورت بخشی از یک برنامه ی اسمارت کلاینت