عنوان: **HTML5 Offline Web Applications**
نویسنده: شاهین کیاست
تاریخ: ۱۳۹۱/۰۵/۲۱ ۲۲:۵۷
آدرس: www.dotnettips.info
برچسب‌ها: HTML5

وب به سمتی پیش رفته که کاربران زیادی از تلفن همراه ، تبلت‌ها و دیگر عامل ها(Agent) جهت مرور صفحات وب استفاده می‌کنند. در نتیجه تعداد کاربرانی که مدام در حال حرکت به مرور صفحات وب و استفاده از سرویس‌های برخط می‌پردازند رو به افزایش است. برنامه‌های خارج از شبکه‌ی HTML 5 یا به عبارتی HTML5 Offline Web Applications توسعه دهنگان را قادر می‌سازد تا نرم افزارهای تحت وبی ارائه دهند که در حالت قطع بودن اینترنت و یا شبکه همچنان به سرویس دادن به کاربران ادامه دهد.
دیگر اینگونه نیست که وب تنها در حالت برخط بودن معنی پیدا کند. یک نرم افزار مدیریت هزینه‌ی تحت وب را بررسی کنید که روی تلفن همراه شما اجرا شده ، در محلی که دسترسی به اینترنت نیست قصد استفاده از آن را دارید. چه قدر خوب می‌شود که این نرم افزار به گونه ای پیاده سازی شده باشد که بتواند در حالت برون خطی (Offline) به سرویس دهی ادامه دهد به طور مثال قادر به ذخیره‌ی داده‌های شما به صورت برون خط و همزمان سازی آنها پس از اتصال به اینترنت باشد.
برنامه‌های تحت وب برون خط با کمک قابلیتی به نام نهانگاه برنامه (Application Cache) کار می‌کنند. این قابلیت می‌تواند تمامی بخش‌های سایت را به شکل برون خط و خارج از شبکه، ذخیره کند. با به کار گیری این ویژگی می‌توان تمامی فایل‌های ایستا (JavaScript , HTML , CSS , Image) بر روی ابزار کاربر ذخیره نمود.

نهانگاه برنامه چه تفاوتی با نهانگاه مرورگر (Browser cache) دارد ؟

مرورگرها برای افزایش سرعت بارگذاری سایت از نهانگاه مخصوص به خود استفاده می‌کنند. مرورگر اگر فایلی را در اختیار داشته باشد دیگر برای دفعات بعدی آن فایل را از سرور درخواست نمی‌کند. این قابلیت اگر دسترسی به شبکه قطع باشد کاربردی ندارد ، همچنین به عنوان توسعه دهنده کنترلی بر روی این قابلیت نداریم اما زمانی که از تکنولوژی برنامه‌های آفلاین استفاده می‌شود این توانایی در اختیار توسعه دهنده است که به مرورگر بگوید کدام فایل‌ها در نهانگاه نگهداری شود ، کدام فایل‌ها در هر بار اتصال از سرور درخواست شود و حتی اگر دریافت فایل‌ها از مخزن با مشکل مواجه شد چه اقدامی صورت پذیرد.

برای استفاده از این ویژگی اولین باید فایلی به نام cache.manifest ایجاد کرد. این فایل باید با نوع محتوای (Mime type) مناسب برای کاربر ارسال شود. این فایل یک فایل متنی می‌باشد که لیست فایل‌های مورد نظر ما با قواعد خاصی در آن قرار می‌گیرد.
همیشه اولین خط این فایل عبارت CACHE MANIFEST قرار دارد ، پس از این خط عبارت CACHE: وارد می‌شود و فایل‌های مد نظر ما لیست می‌شود.

```
CACHE MANIFEST

#Cache Section
CACHE:
/Content/Images/icons-18-white.png
/Content/Images/icons-36-white.png
/Content/Images/ajax-loader.png
/Content/css
/Scripts/js
```

ذخیره‌ی برخی فایل‌ها روی سیستم کاربر ضرورتی ندارد ، برای مثال اسکریپتی که از یک وب سرویس برخط اطلاعات آب و هوا را دریافت می‌کند در حالت Offline کاربردی ندارد. برای مشخص کردن این فایل‌ها یک لیست سفید آماده می‌کنیم.

```
NETWORK
webService.Js
```

در بخش معرفی فایل‌های آفلاین بازید همه‌ی فایل‌های مد نظر ما معرفی شوند اما در لیست سفید با گذاشتن ستاره به مرورگر اعلام می‌کنیم که هر فایل و مسیری که در بخش قبلی (CACHE) نیامده را همواره از سرور درخواست کن. درج ستاره در بخش NETWORK ضروری است زیرا همه‌ی آدرس‌های سایت باید در یکی از بخش‌های cache.manifest قرار بگیرد ، اگر آدرسی در cache.manifest قرار نگیرد هیچگاه بارگذاری نخواهد شد.

در فایل cache.manifest می‌توان یادداشت هم اضافه کرد ، تمامی کاراکتر هایی که بعد از # قرار گیرند پردازش نمی‌شوند.
یادداشت‌ها مفید هستند ، می‌تونید اطلاعات نسخه‌ی فعلی فایل cache.manifest را در آنها قرار دهید.
مثال :

```
CACHE MANIFEST
# 2010-06-18:v2

# Explicitly cached 'master entries'.
CACHE:
/favicon.ico
index.html
css
images/
stylesheet
.logo.png
scripts/main.js

# Resources that require the user to be online.
NETWORK:
login.php
/myapi
http://api.twitter.com
```

گفته شد فایل Cache.manifest باید با نوع محتوای مناسب ارسال شود ، برای تعیین نوع محتوا در وب سرور apache باید خط زیر به فایل .htaaccess اضافه شود :

```
AddType text/cache-manifest .appcache
```

در ASP.NET Web Forms می‌توان از یک Generic Handler برای ارسال فایل با نوع محتوای مناسب استفاده کرد :

```
using System.Web;

namespace JavaScriptReference {

    public class Manifest : IHttpHandler {

        public void ProcessRequest(HttpContext context) {
            context.Response.ContentType = "text/cache-manifest";
            context.Response.WriteFile(context.Server.MapPath("Manifest.txt"));
        }

        public bool IsReusable {
            get {
                return false;
            }
        }
    }
}
```

یا می‌توان در یک فایل ASPX به صورتی دستی نوع محتوا را مشخص کرد :

```
CACHE MANIFEST

# Version Jesus 3!

CACHE:

index.html
js/Custom.js
js/Utility.js
styles/index.css
styles/kendo.common.min.css
styles/BlueOpal/loading.gif
styles/BlueOpal/slider-h.gif
styles/BlueOpal/slider-v.gif

NETWORK:
*
```

```
<%@ Page Language="VB" ContentType="text/cache-manifest"  ResponseEncoding="utf-8"
AutoEventWireup="true" CodeFile="manifest.aspx.vb" Inherits="Configuration_manifest" %>
```

در ASP.NET MVC علاوه بر اینکه می‌توان دستی نوع محتوای Response را مشخص کرد، می‌توان یک ActionResult منحصر به فرد ایجاد کرد. یک نمونه پیاده سازی شده را اینجا مشاهده کنید.
پس از انجام همه‌ی این پیش نیازها باید فایل cache manifest را به خصیصه‌ی manifest برچسب html ارجاع دهیم.

```
<!DOCTYPE html>
<html manifest="/manifest.aspx">
```

اکنون قسمت‌های مد نظر سایت ما در حالت عدم دسترسی به شبکه نیز قابل استفاده می‌باشد. حتی این ویژگی در حالت برخط صفحات ما با سرعت بالاتری بارگذاری شوند.

خصیصه‌ی manifest تمامی فایل‌های HTML باید مقدار گیرد در غیر این صورت ممکن است صفحات درون نهانگاه قرار نگیرند.
برای بررسی مرورگرهایی که این ویژگی را پشتیبانی می‌کنند این لینک را مشاهده کنید.