

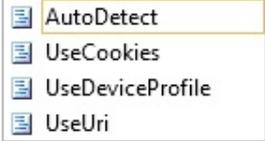
بسم الله الرحمن الرحيم

الدرس عشرون – الجزء الثاني

كيف نستعمل Cookieless ؟

قلت مسبقا ان (Forms Authentications) تستعمل الكوكيز للتحقق من المستخدم ، لكن في هذه الحالة ستواجهنا مشكلة ماذا لو كان متصفح المستخدم لا يدعم الكوكيز او معطل في الكوكيز ؟ طبعا هذه مشكلة تم حلها عن طريق اضافة الخاصية Cookieless الى (Forms Authentications) التي تعني التحقق من المستخدم حتى لو كان متصفح لا يدعم الكوكيز ، وفلسفتها هي عند تفعيل Cookieless ستقوم بأعطاء قيمة فريده يضاف الى رابط لكل مستخدم بدون الاعتماد على كوكيز المتصفح .

```
<?xml version="1.0"?>
<!--
For more information on how to configure your ASP.NET application, please visit
http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
  <system.web>
    <compilation debug="false" targetFramework="4.0" />
    <authentication mode="Forms">
      <forms cookieless=""
      </authentication>
    </system.web>
  </configuration>
```



الصورة اعلاه توضح ان Cookieless تمتلك اربعة خصائص :

- UseCookies : بمعنى دائما استعمل الكوكيز في التحقق .
- UseUri : يعني لا تستعمل كوكيز التحقق مطلقا .
- AutoDetect : تعني استعمل كوكيز التحقق بصورة اوتوماتيكية .
- UseDeviceProfile : تستعمل لمعرفة متى نستعمل كوكيز التحقق .

Cookies : عبارة عن ملف نصي صغير حجمه لايتجاوز عدة بايتات ويتقبل قيم نصية فقط ودائما يحوي بعض المعلومات عن المستخدم وهو من ابتكار مطوري Netscape ، حيث يقوم الويب سيرفر بإنشاءه ثم يضيفه الى هيدر البروتوكول HTTP ويرسل الى متصفح المستخدم على سبيل المثال يقوم السيرفر بأرسال الكوكيز ادناه الى المتصفح

Set –Cookie : Message = السلام عليكم

حيث Set-Cookie تجعل المتصفح ينشئ كوكيز باسم Message الذي يحوي الجملة (السلام عليكم) هذا في بداية تصفح الموقع ثم يقوم المتصفح بأرسال الكوكيز الى الويب سيرفر عند طلب اي صفحة ويكون هكذا

Cookie : Message = السلام عليكم

والكوكيز يكون على نوعين الاول (Session Cookies) هذا يخزن في الذاكرة الوقتيّة ويحذف بمجرد غلق الموقع ، والثاني (Persistent Cookies) هذا النوع يحفظ في المتصفح ويبقى موجود حتى بعد غلق صفحات الموقع وقد يبقى ايام او اسابيع بل حتى شهر .

ملخص : عندما يدخل المستخدم الاسم او الایمیل وكلمة السر يبدأ التحقق منه يعني هل هو موجود في قاعدة البيانات او لا فاذا كان موجود ، يرسل له كوكيز مشفر بحوي معلومات بأنه قد تم التحقق منه وبالتالي فعندما يطلب المستخدم صفحة ثانية فسوف يتم التحقق منه عن طريق الكوكيز الذي ارسل له هذا كل ما في الامر .

: Sliding Expiration

فكرته هي لو انك قمت بتسجيل الدخول في اي موقع أو منتدى وبعدها تركته لفترة من الوقت بدون ما تطلب اي صفحة فإنه بصورة اوتوماتيكية سيتم عمل تسجيل خروج لك .

```
<?xml version="1.0"?>
<!--
For more information on how to configure your ASP.NET application, please visit
http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
  <system.web>
    <compilation debug="false" targetFramework="4.0" />
    <authentication mode="Forms">
      <forms slidingExpiration="" />
    </authentication>
  </system.web>
</configuration>
```

بصورة افتراضية يكون (True) حيث يكون الوقت الافتراضي له (٣٠ دقيقة) يعني خلال نصف ساعة اذا لم يطلب المستخدم اي صفحة فانه بصورة اوتوماتيكية يعمل له تسجيل خروج ويمكن زيادة هذه الفترة من خلال الخاصية timeout .

```
<?xml version="1.0"?>
<!--
For more information on how to configure your ASP.NET application, please visit
http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
  <system.web>
    <compilation debug="false" targetFramework="4.0" />
    <authentication mode="Forms">
      <forms slidingExpiration="true" timeout="45"/>
    </authentication>
  </system.web>
</configuration>
```

، لكن في بعض الاحيان قد ترغب ان تجعل المستخدم يقوم بتسجيل الدخول كل فترة زمنية محددة حتى في حالة طلب صفحة خلال وقت اقل من نصف ساعة في هذه الحالة يكون :

```
<?xml version="1.0"?>
<!--
For more information on how to configure your ASP.NET application, please visit
http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
  <system.web>
    <compilation debug="false" targetFramework="4.0" />
    <authentication mode="Forms">
      <forms slidingExpiration="false" timeout="5"/>
    </authentication>
  </system.web>
</configuration>
```

الكود في الصورة اعلاه يجعل المستخدم يقوم بتسجيل الدخول كل ٥ دقائق .

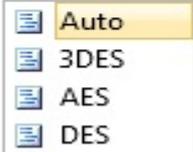
: Across Application

هذه الخاصية هي الاكثر تعقيدا وفكرتها لنفترض انك عملت أكثر من تطبيق (Web Application) لشركة او مؤسسة او اي شي آخر ، فليس من المعقول ان يقوم بتسجيل الدخول لكل هذه التطبيقات (يعني اذا قام بالتسجيل بالتطبيق الاول واحتاج شي في التطبيق الثاني او الثالث يستلزم ان يتم التحقق منه فهل يقوم بالتسجيل مرة اخرى ؟...)

طيب ماذا لو كانت هذه التطبيقات جميعها على استضافة واحدة ؟ وماذا اذا كانت على أكثر من استضافة (يعني سيرفرات مختلفة) ؟ يعني كيف نجعل هذه التطبيقات سواء على نفس الاستضافة او على استضافات مختلفة كأنها تطبيق واحد وعلى استضافة واحدة .

العقبة الاساسية هي انك لا تستطيع مشاركة كوكيز التحقق بين هذه التطبيقات لان كل تطبيق يستخدم (Decrypted Key) بالاضافة الى (Validation Key) فريد يختلف من تطبيق لآخر! .

```
<configuration>
  <system.web>
    <compilation debug="false" targetFramework="4.0" />
    <authentication mode="Forms">
      <forms slidingExpiration="false" timeout="5"/>
    </authentication>
    <machineKey decryption="" />
  </system.web>
</configuration>
```



طيب ، Decryption نقصد بها خوارزمية تشفير وفك تشفير الكوكيز وهي تأخذ احدى القيم :

- ❖ Auto : تعني اختيار خوارزمية تتوافق مع الويب سيرفر .
- ❖ AES
- ❖ DES
- ❖ 3DES (Triple DES)

اما بالنسبة (Validation) يقصد به تحديد خوارزمية التشفير التي تستعمل عندما كوكيز ارسل .

```

<configuration>
  <system.web>
    <compilation debug="false" targetFramework="4.0" />
    <authentication mode="Forms">
      <forms slidingExpiration="false" timeout="5"/>
    </authentication>
    <machineKey decryption="Auto" validation=""/>
  </system.web>
</configuration>

```

- AES
- MD5
- SHA1
- 3DES
- HMACSHA256
- HMACSHA384
- HMACSHA512

هذا بالإضافة الى (Decryption Key) وهو مفتاح المستعمل لتشفير وفك تشفير الكوكيز. اما (Validation Key) كذلك هو مفتاح لتشفير وفك التشفير عندما الكوكيز قد ارسل ، وبصورة افتراضية فأنهما يأخذان القيمة (AutoGenerate) والتي تجعل .net Framework تولد مفتاح عشوائي وتحفظه في LSA (Local Server Authority) .

```

<?xml version="1.0"?>
<!--
For more information on how to configure your ASP.NET application, please visit
http://go.microsoft.com/fwlink/?LinkId=169433
-->
<<configuration>
  <system.web>
    <compilation debug="false" targetFramework="4.0" />
    <authentication mode="Forms">
      <forms slidingExpiration="false" timeout="5"/>
    </authentication>
    <machineKey decryption="Auto" validation="SHA1" decryptionKey="AutoGenerate,IsolateApps" validationKey="AutoGenerate,IsolateApps" />
  </system.web>
<</configuration>

```

كما تلاحظ في الصورة اعلاه فان (Decryption Key) و (Validation Key) اضافة فيهما خاصية (IsolateApps) وهي التي تجعل لكل تطبيق (Decryption Key) و (Validation Key) قيم فريده وبالتالي لا يمكن ان تتشارك التطبيقات التي على نفس السيرفر بكوكيز التحقق .

(Decryption Key) و (Validation Key) بصورة افتراضية يحويان الخاصية (IsolateApp) لذلك يفضل ازالتها عند استخدام (Across Application) على نفس السيرفر .

```
<?xml version="1.0"?>
<!--
For more information on how to configure your ASP.NET application, please visit
http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
  <system.web>
    <compilation debug="false" targetFramework="4.0" />
    <authentication mode="Forms">
      <forms slidingExpiration="false" timeout="5"/>
    </authentication>
    <machineKey decryption="Auto" validation="SHA1" decryptionKey="AutoGenerate" validationKey="AutoGenerate" />
  </system.web>
</configuration>
```

نلتقي ان شاء الله في الجزء الثالث من الدرس عشرون

لاتنسونا بصالح دعواتكم

اخوكم محمد الساعدي