

**معرفی:**

امروزه تست کردن کدها به دلیل وجود ابزارهای مختلف زیادی، کار آسانی شده است. اما بعضی‌ها در web application ها، یکی از تست‌هایی را که خیلی هم مهم است را فراموش می‌کنند که آن هم تست UI است. شما را در این مقاله با یکی از روش‌های خوب تست UI آشنا خواهیم کرد. ابزارهای زیادی برای تست UI وجود دارد که کار کردن با آنها نه تنها زمان بر بلکه بسیار خسته کننده می‌باشند و به خاطر همین خیلی‌ها از انجام تست UI صرف نظر می‌کنند.

**WatIn چیست؟**

WatIn مخفف Web Application Testing in .Net می‌باشد؛ که یک فریم ورک تست web application ها است. WatIn این اجازه را به شما می‌دهد که با استفاده از IE ویا FireFox عناصر داخل صفحات را مقدار دهی کنید و یا حتی رویدادی را برای عناصر فراخوانی کنید.

**شروع کار با WatIn:**

در زیر یک نمونه از کار با WatIn را می‌توانید مشاهده کنید:

```
[TestMethod]
public void SearchForWatinOnGoogle()
{
    using (var browser = new IE("http://www.google.com"))
    {
        browser.TextField(Find.ByName("q")).TypeText("Watin");
        browser.Button(Find.ByName("btnG")).Click();
        Assert.IsTrue(browser.ContainsText("Watin"));
    }
}
```

WatIn یک فریم ورک کاربر پسند است و در ادامه متوجه می‌شوید که استفاده از این فریم ورک چه مزایایی دارد. برای نصب، WatIn را می‌توانید [از اینجا](#) دانلود کنید ویا اگر خواستید می‌توانید با NuGet هم این فریم ورک را دانلود کرده و نصب نمایید. برای شروع کار با Watin باید reference هایی را به پروژه تان اضافه کنید که یکی از این reference ها Watin.Core.dll می‌باشد و برای استفاده از IE ویا FireFox باید فضای نام Watin.Core را اضافه کنیم. Watin چند فضای نام دیگری را هم به همراه دارد که در زیر به توضیح مختصری از آنها می‌پردازیم:

**1-Watin.Core.DialogHandlers:** این فضای نام این امکان را به شما می‌دهد تا دیالوگ هایی را که مرورگر می‌تواند به کاربر نمایش دهد، مدیریت کنید. از handlerهای این فضای نام، AlertDialogHandler، ConfirmDialogHandler، LoginDialogHandler و FileUploadDialogHandler، PrintDialogHandler می‌باشد.

**2-Watin.Core.Exceptions:** این فضای نام دارای یک سری exception می‌باشد و این امکان را به ما می‌دهد تا یک سری رفتارهای ناخواسته را کنترل کنیم. بعضی از این exception ها، ElementNotFoundException، IElementNotFoundException، TimeoutException و WatinException می‌باشد.

**3-Watin.Core.Logging:** این فضای نام کلاس هایی را در اختیار ما می‌گذارد تا بتوانیم عملیاتی را که در کدمان انجام می‌دهیم log کنیم.

مثالی از watin که در بالا نشان دادیم به این صورت عمل می‌کند که مرورگر IE را باز کرده و به سایت google خواهد رفت. در این صفحه جعبه متنی یا TextBox با نام "q" را پیدا کرده و عبارت "Watin" را در آن تایپ می‌کند و همچنین Button ی با نام "btnG" پیدا کرده و آن را کلیک می‌نماید و در آخر بررسی می‌کند که در مرورگر متنی شامل Watin وجود دارد یا خیر. مشاهده کردید که به همین سادگی یک تست UI نوشتیم. به نظر شما جالب نبود؟ فرض کنید که اگر می‌خواستید با مثلاً Microsoft Test Manager این کار را انجام دهید چه دردسرهایی را باید تحمل می‌کردید. حالا تست UI برای همه برنامه نویسی‌ها جذاب خواهد شد.

به جای مثال بالا می‌توانیم به صورت زیر هم عمل کنیم:

```
[TestMethod]
public void SearchForWatiNOnGoogle()
{
    using (var browser = new IE("http://www.google.com"))
    {
        browser.TextField(Find.ByName("q")).Value="WatiN";
        browser.Button(Find.ByName("btnG")).ClickNowait();
        Thread.Sleep(3000);
        Assert.IsTrue(browser.ContainsText("WatiN"));
    }
}
```

تفاوت کد دوم با کد اول این است چون در کد اول از متد `TypeText` استفاده کردیم یک مقدار سرعت تست را پایین می‌آورد ولی اگر از `Value` و یا از `SetAttribute` استفاده کنیم دیگر عمل تایپ را انجام نداده و مقدار را مستقیماً در مقدار `TextField` قرار می‌دهد. شاید بپرسید چرا بعد از متد `ClickNowait` چند ثانیه صبر می‌کنم؟ چون صفحه برای اینکه بارگذاری شود و نتیجه جستجو را نشان دهد کمی طول کشیده و `Assert.IsTrue` شما `Failed` می‌شود. البته به جای `Thread.Sleep` می‌توانیم از متدهای مربوط به `Watin` هم استفاده کنیم مانند `WaitUntilComplete` و یا از `WaitUntilContainsText`.

## نظرات خوانندگان

نویسنده: آرش خوشبخت  
تاریخ: ۲۱:۳۰ ۱۳۹۲/۰۱/۲۴

اگر سوالی یا مشکلی در کارکردن با این فریم ورک داشتین می‌توانید اینجا بپرسین