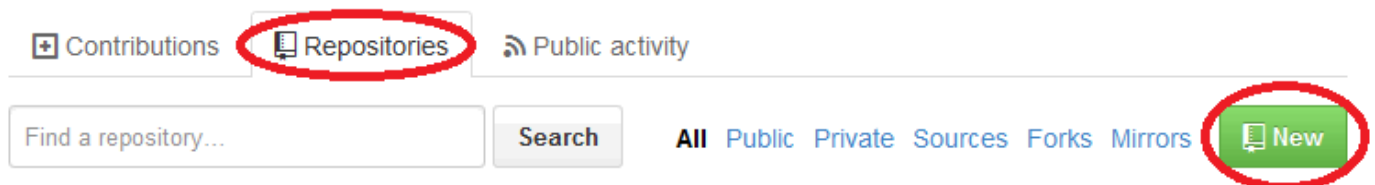



از نگارش 2012 ویژوال استودیو، امکان کار با مخازن Git، به صورت یکپارچه و توکار و بدون نیاز به ابزارهای جانبی، توسط آن فراهم شده‌است. در ادامه قصد داریم به کمک این ویژگی توکار، نحوه‌ی ارسال یک پروژه‌ی از پیش موجود VS.NET را برای اولین بار به GitHub بررسی کنیم.

## تنظیمات مقدماتی GitHub

در ابتدا نیاز است یک مخزن کد خالی را در GitHub ایجاد کنید. برای این منظور به برگه‌ی Repositories در اکانت GitHub خود مراجعه کرده و بر روی دکمه‌ی New کلیک کنید:



سپس در صفحه‌ی بعدی، نام پروژه را به همراه توضیحاتی وارد نمائید و بر روی دکمه‌ی Create repository کلیک کنید. در اینجا سایر گزینه‌ها را انتخاب نکنید. نیازی به انتخاب گزینه‌ی READ ME و یا انتخاب مجوز و غیره نیست. تمام این کارها را در سمت پروژه‌ی اصلی می‌توان انجام داد و یا VS.NET فایل‌های ignore را به صورت خودکار ایجاد می‌کند. در اینجا صرفاً هدف، ایجاد یک مخزن کد خالی است.

**Owner**  VahidN / **Repository name**

Great repository names are short and memorable. Need inspiration? How about [turnt-octo-bugfixes](#).

**Description** (optional)

☒ **Public**  
Anyone can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.


☐ **Initialize this repository with a README**  
This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.

Add .gitignore: **None** | Add a license: **None** ⓘ

**Create repository**

از اطلاعات صفحه‌ی بعدی، تنها به آدرس مخصوص GitHub آن نیاز داریم. از این آدرس در VS.NET برای ارسال اطلاعات به سرور استفاده خواهیم کرد:

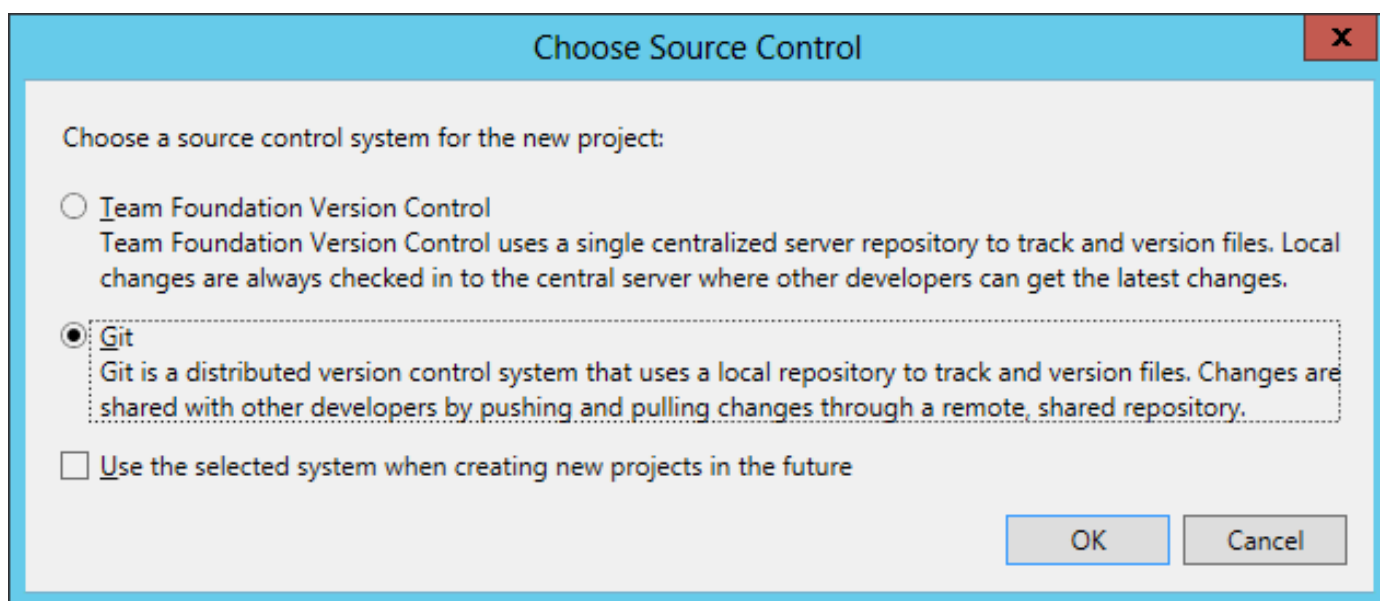
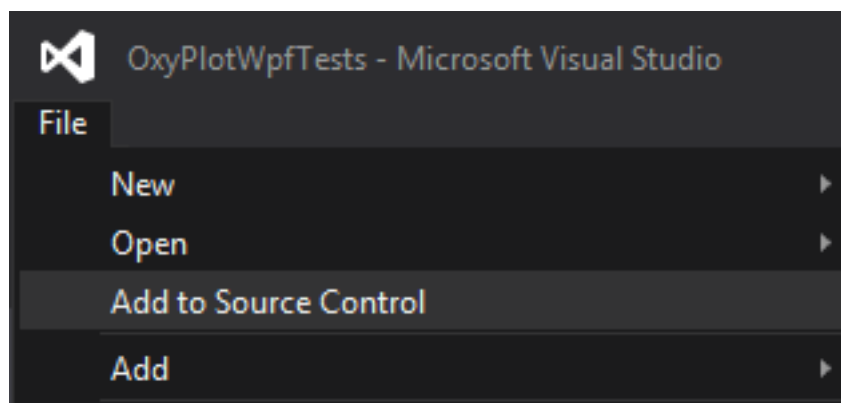
#### Quick setup — if you've done this kind of thing before

 **Set up in Desktop** or **HTTPS** **SSH**

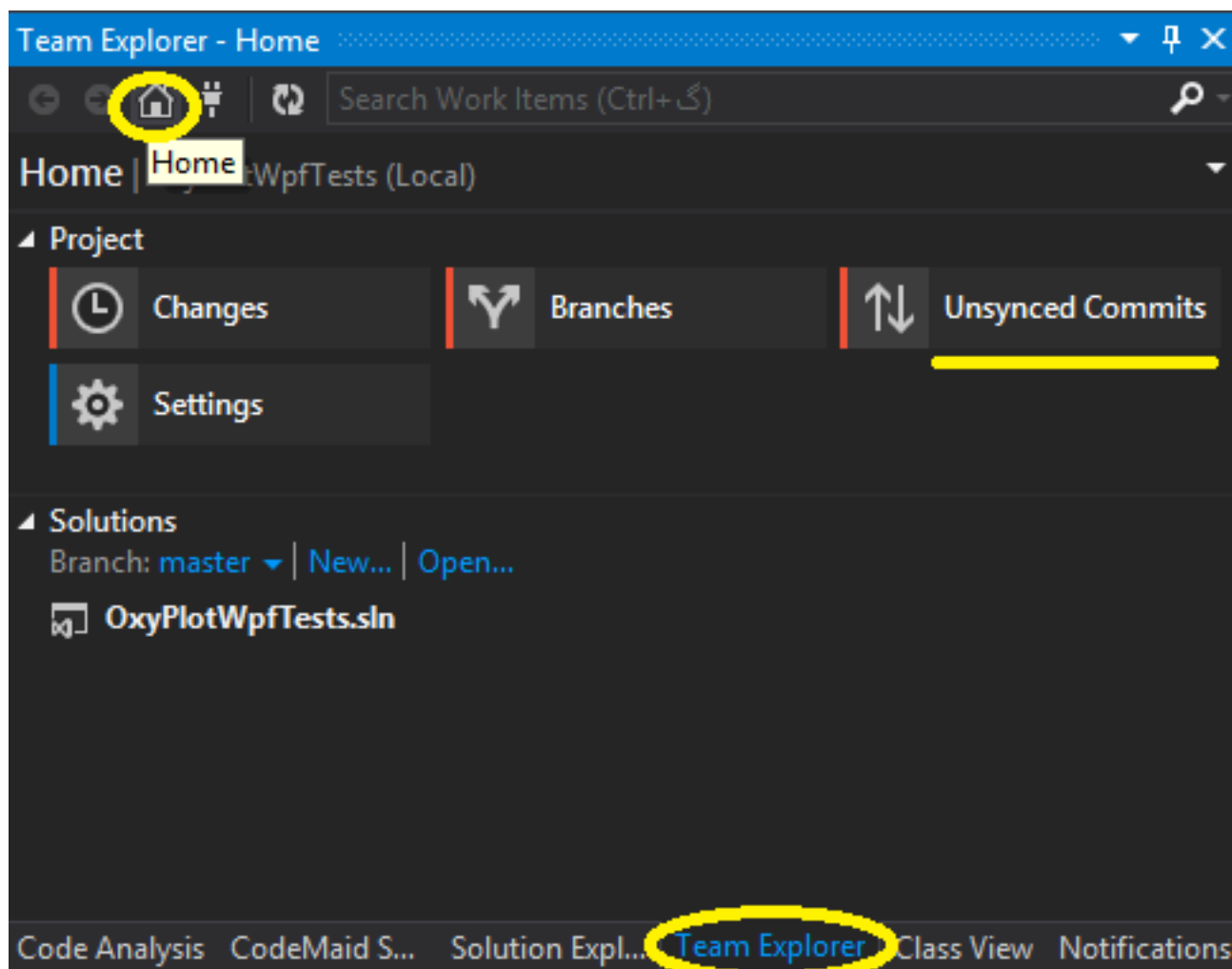
We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

#### تنظیمات VS.NET برای ارسال پروژه به مخزن GitHub

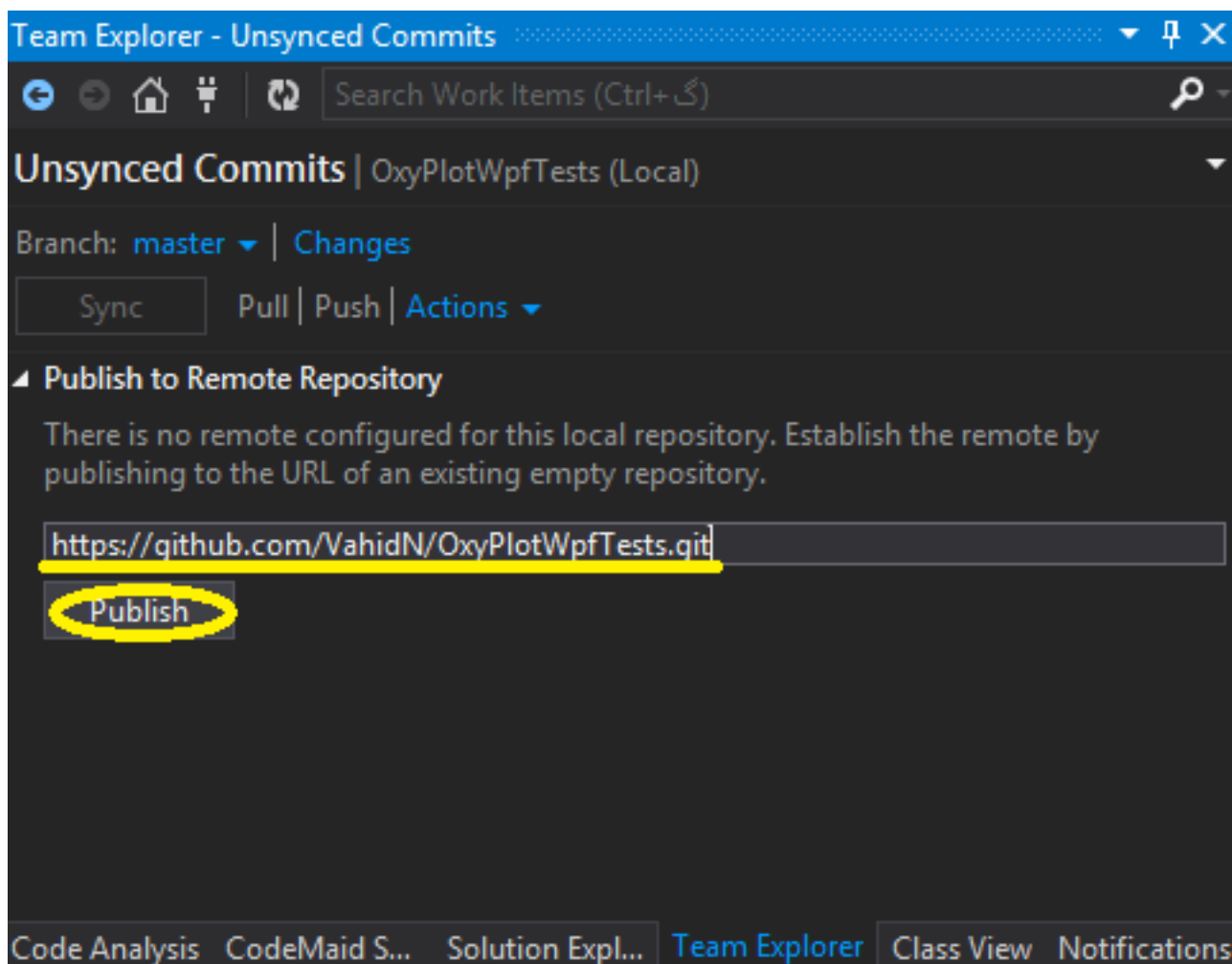
پس از ایجاد یک مخزن کد خالی در GitHub، اکنون می‌توانیم پروژه‌ی خود را به آن ارسال کنیم. برای این منظور از منوی File، گزینه‌ی Add to source control را انتخاب کنید و در صفحه‌ی باز شده، گزینه‌ی Git را انتخاب نمایید:



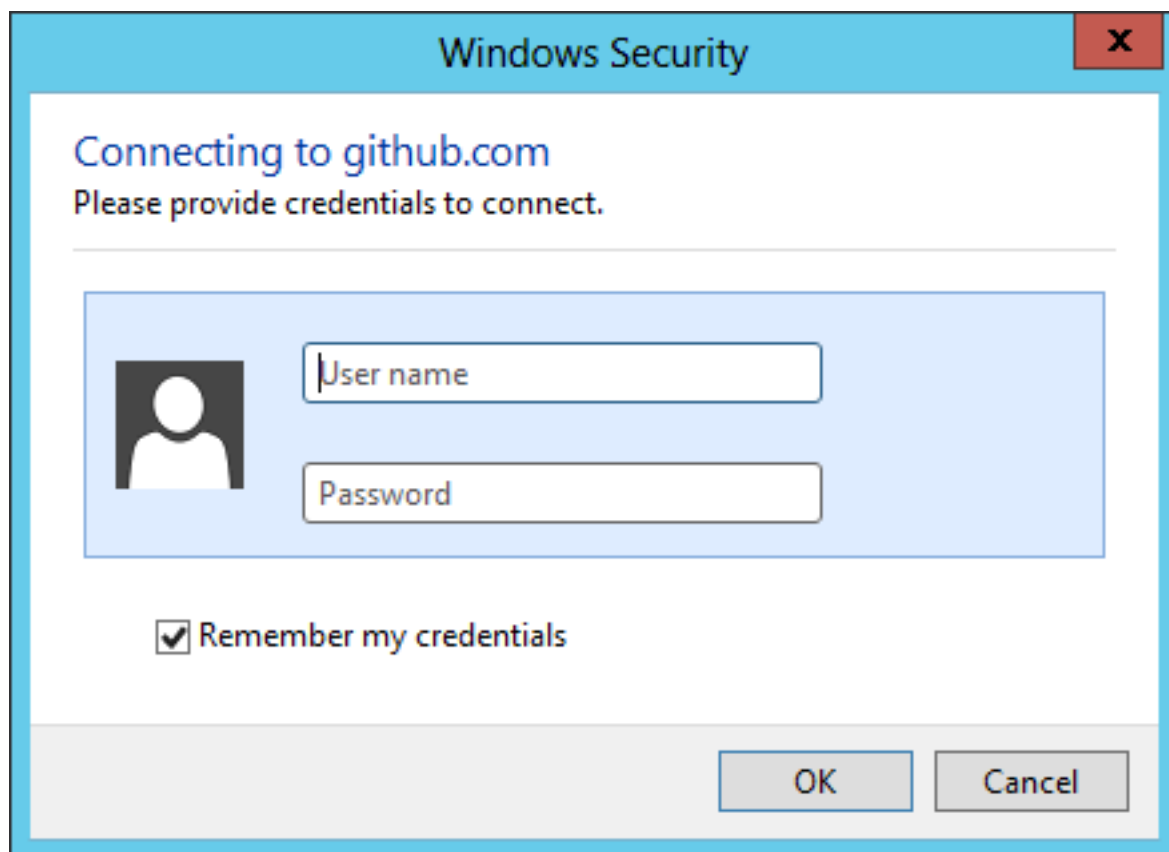
سپس در کنار برگه‌ی Solution Explorer، برگه‌ی Team Explorer را انتخاب کنید. در اینجا بر روی دکمه‌ی Home در نوار ابزار آن کلیک کرده و سپس بر روی دکمه‌ی Unsynced commits کلیک نمایید.



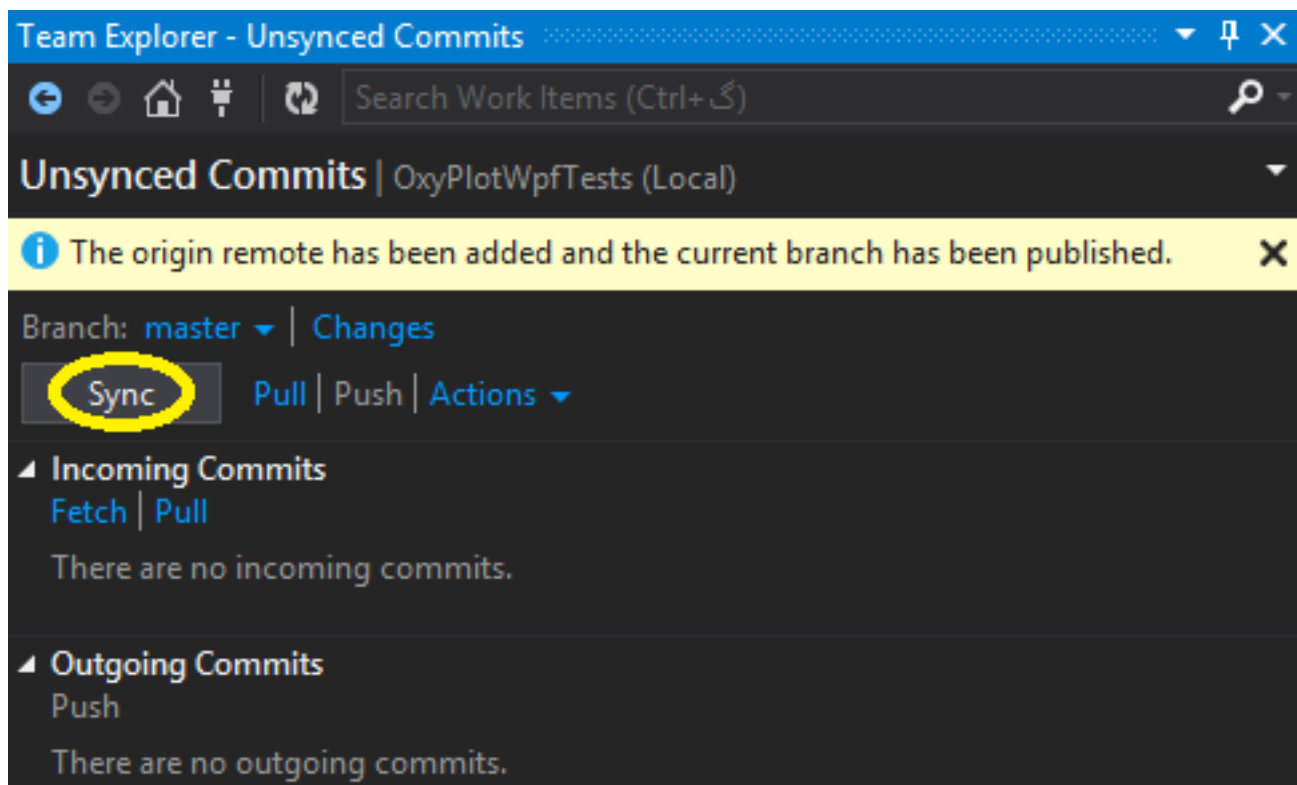
در ادامه در صفحه‌ی باز شده، همان آدرس مخصوص مخزن کد جدید را در GitHub وارد کرده و بر روی دکمه‌ی Publish کلیک کنید:



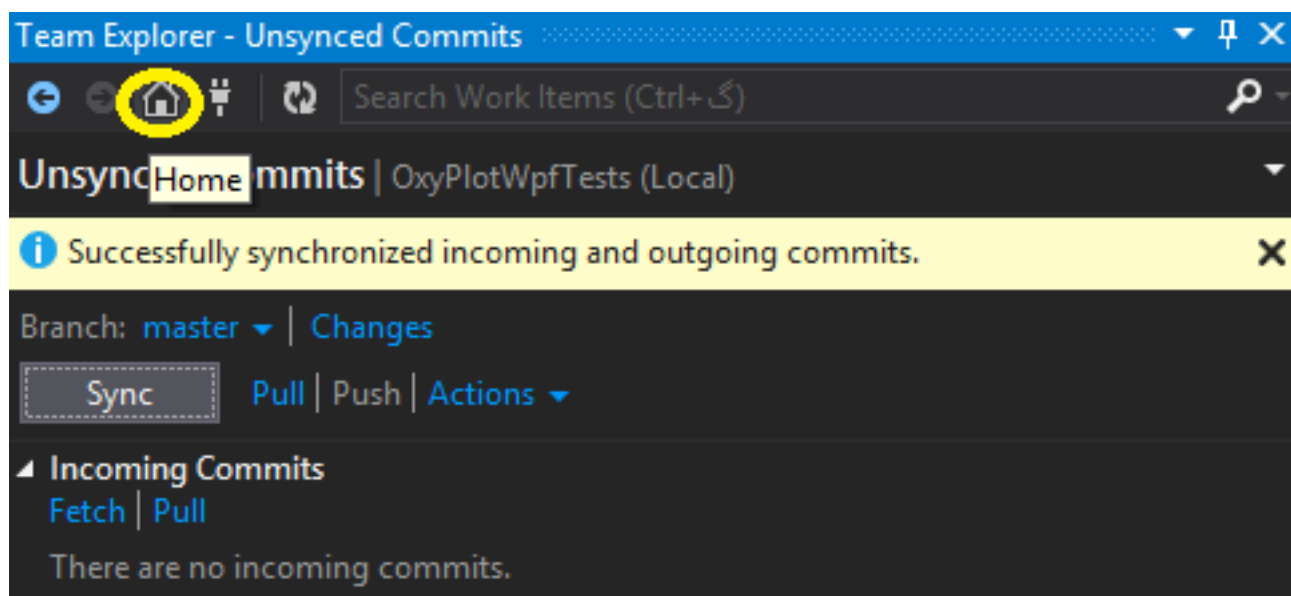
در اینجا بلافاصله صفحه‌ی لاگینی ظاهر می‌شود که باید در آن مشخصات اکانت GitHub خود را وارد نمایید:



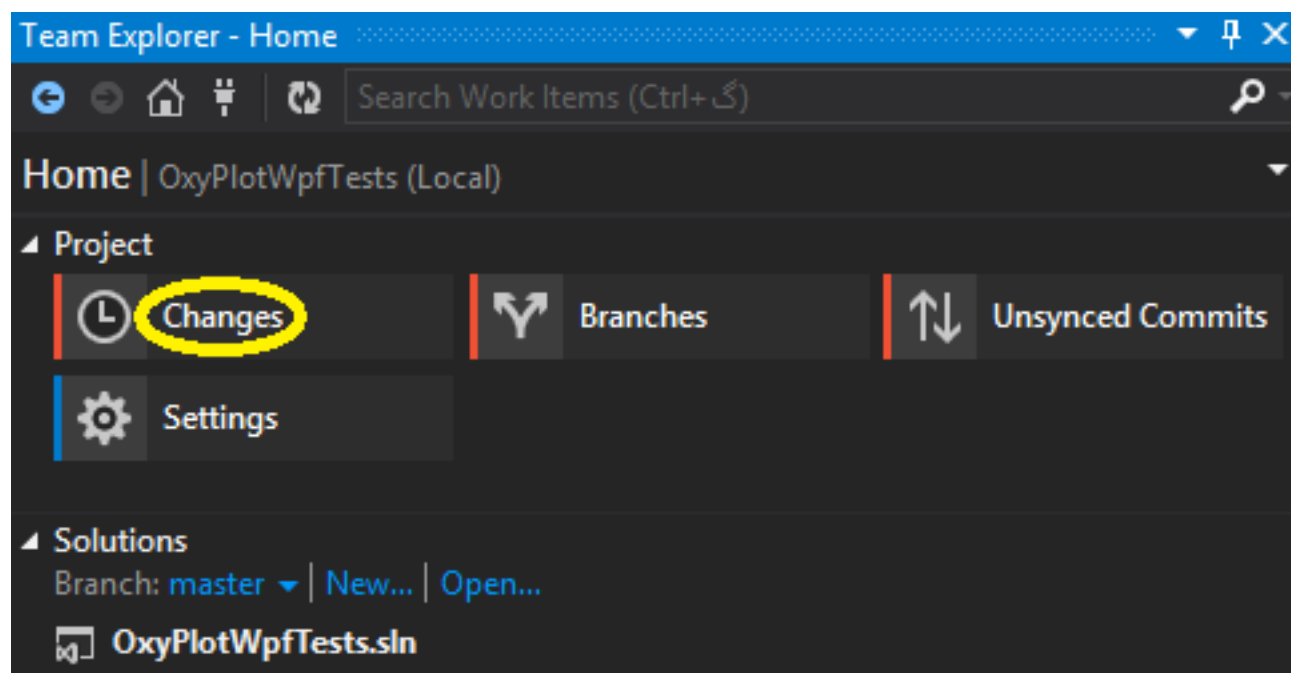
به این ترتیب عملیات Publish اولیه انجام شده و تصویر ذیل نمایان خواهد شد:



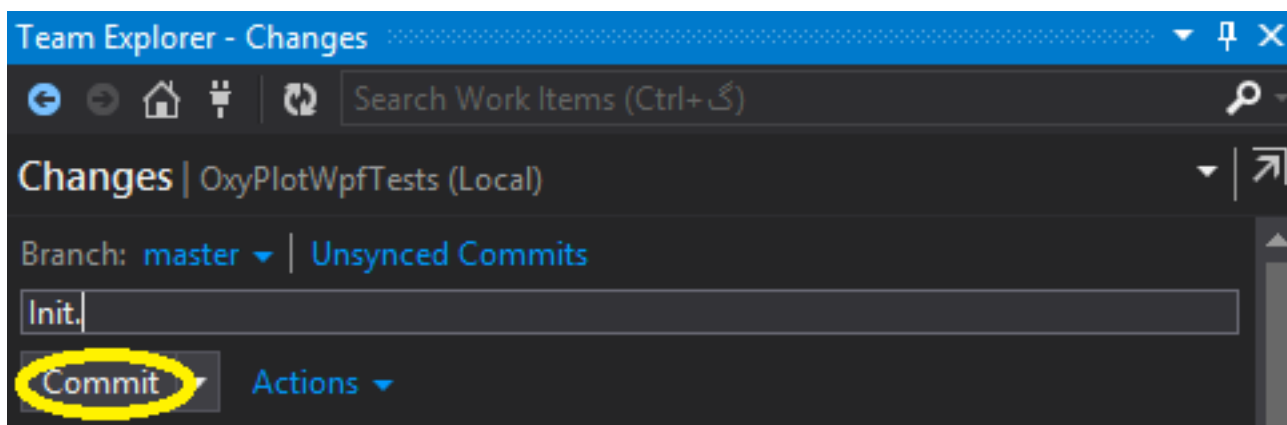
در اینجا بر روی دکمه‌ی Sync کلیک کنید. به این ترتیب مخزن کد GitHub به پروژه‌ی جاری متصل خواهد شد:



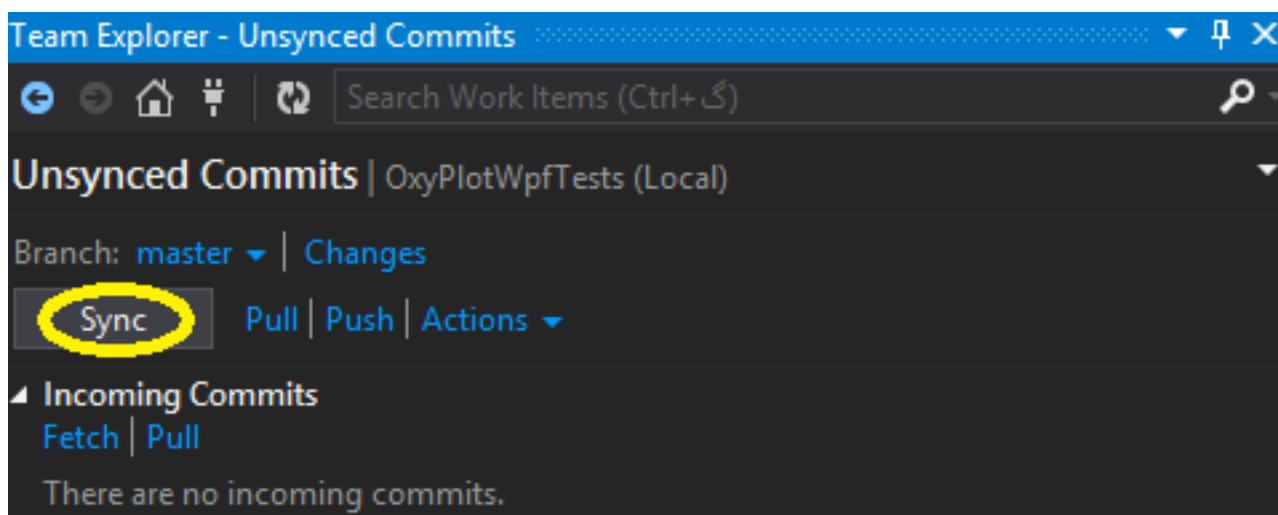
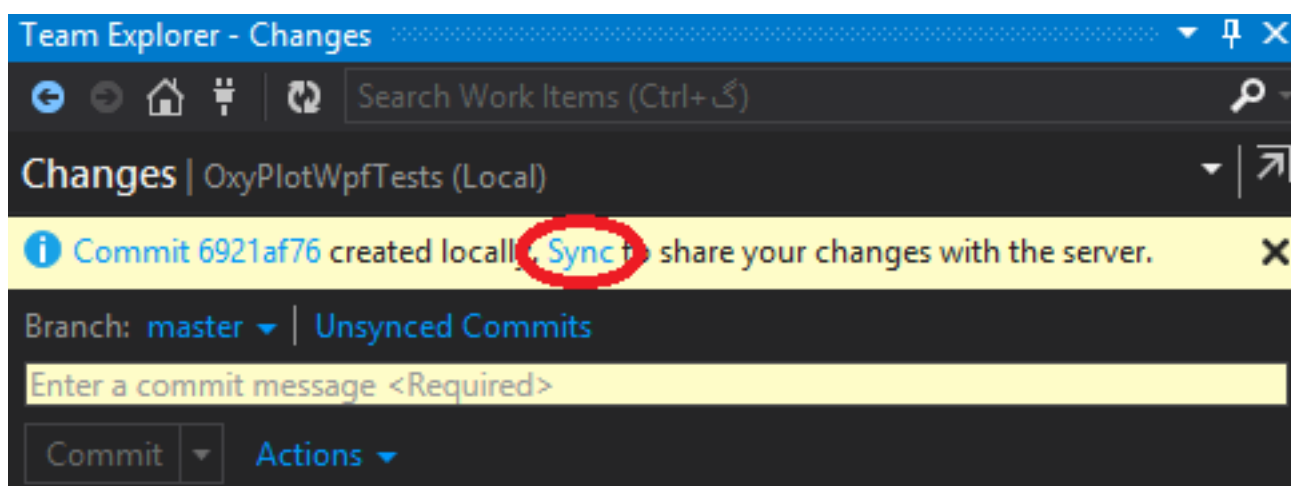
سپس نیاز است فایل‌های موجود را به مخزن کد GitHub ارسال کرد. بنابراین پس از مشاهده‌ی پیام موفقیت آمیز بودن عملیات همگام‌سازی، بر روی دکمه‌ی Home در نوار ابزار کلیک کرده و اینبار گزینه‌ی Changes را انتخاب کنید:



در اینجا پیام اولین ارسال را وارد کرده و سپس بر روی دکمه‌ی Commit کلیک کنید:



پس از مشاهده‌ی پیام موفقیت آمیز بودن commit محلی، نیاز است تا آنرا با سرور نیز هماهنگ کرد. به همین جهت در اینجا بر روی لینک Sync کلیک کرده و در صفحه‌ی بعدی بر روی دکمه‌ی Sync کلیک کنید:





اندکی صبر کنید تا فایل‌ها به سرور ارسال شوند. اکنون اگر به GitHub مراجعه کنید، فایل‌های ارسالی قابل مشاهده هستند:

An OxyPlot Sample — Edit

2 commits   1 branch   0 releases   1 contributor

branch: master   OxyPlotWpfTests / +

Init.

VahidN authored 3 minutes ago   latest commit 6921af7608

OxyPlotWpfTests	Init.	3 minutes ago
.gitattributes	Initial commit to add default .gitignore and .gitAttribute files.	18 minutes ago
.gitignore	Initial commit to add default .gitignore and .gitAttribute files.	18 minutes ago
OxyPlotWpfTests.sln	Init.	3 minutes ago

We recommend adding a README to this repository to help give people an overview of your project.   Add a README

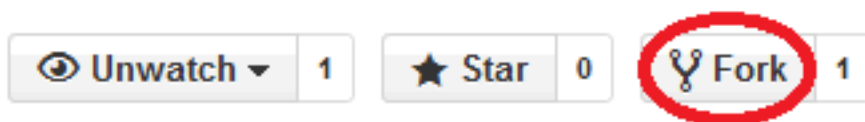
### اعمال تغییرات بر روی پروژه‌ی محلی و ارسال به سرور

در ادامه می‌خواهیم دو فایل README.md و LICENSE.md را به پروژه اضافه کنیم. پس از افزودن آن‌ها، یا هر تغییر دیگری در پروژه، اینبار برای ارسال تغییرات به سرور، تنها کافی است به برگه‌ی Team explorer مراجعه کرده و ابتدا بر روی دکمه‌ی Home کلیک کرد تا منوی انتخاب گزینه‌های آن ظاهر شود. در اینجا تنها کافی است گزینه‌ی Changes را انتخاب و دقیقاً همان مراحل عنوان شده‌ی پیشین را تکرار کرد. ابتدا ورود پیام Commit و سپس Commit. در ادامه Sync محلی و سپس Sync با سرور.

فرض کنید برای رفع باگی در پروژه‌ای از GitHub، ایده‌ای دارید. روند کاری اعلام آن، روش‌های مختلفی می‌تواند داشته باشند؛ از باز کردن یک Issue جدید تا فرستادن یک فایل zip و غیره. اما روش استاندارد مشارکت در پروژه‌های Git، ارسال یک PR یا Pull Request است. در ادامه نحوه‌ی انجام این کار را به کمک امکانات توکار VS.NET بررسی خواهیم کرد.

### ایجاد یک Fork جدید در GitHub

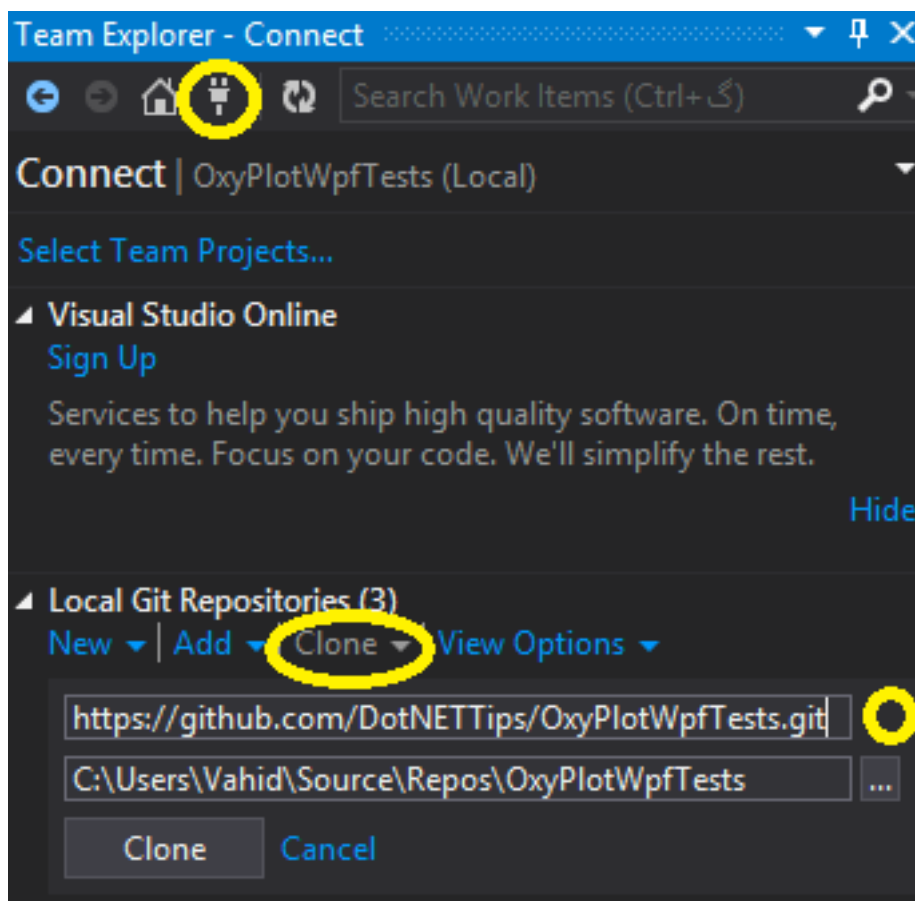
برای ارسال تغییرات انجام شده بر روی یک پروژه، نیاز است به صاحب یا مسئول آن مخزن در GitHub مراجعه و سپس درخواست دسترسی اعمال تغییرات را نمود. در این حالت، احتمال اینکه جواب منفی دریافت کنید، بسیار زیاد است. جهت مدیریت یک چنین مواردی، قابلیت به نام ایجاد یک Fork پیش بینی شده است.



در بالای هر مخزن کد در GitHub، یک دکمه به نام Fork موجود است. بر روی آن که کلیک کنید، یک کپی از آن پروژه را به مجموعه‌ی مخزن‌های کد شما در GitHub اضافه می‌کند. بدیهی است در این حالت، مجوز ارسال تغییرات خود را به GitHub و در اکانت خود خواهید داشت. نحوه‌ی اطلاع رسانی این تغییرات به صاحب اصلی این مخزن کد، ارسال همان PR یا Pull Request است.

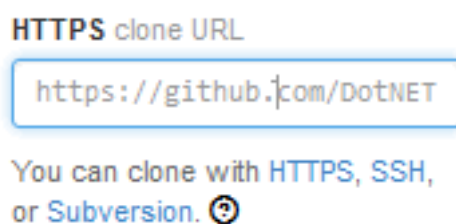
### دریافت مخزن کد Fork شده از GitHub به کمک Visual Studio

پس از اینکه Fork جدیدی را از پروژه‌ای موجود ایجاد کردیم، نیاز است یک Clone یا کپی مطابق اصل آن را جهت اعمال تغییرات محلی، تهیه کنیم. برای اینکار VS.NET را گشوده و به برگه‌ی Team Explorer آن که در کنار Solution Explorer قرار دارد، مراجعه کنید.



در اینجا بر روی دکمه‌ی Connect در نوار ابزار آن، کلیک کرده و در صفحه‌ی باز شده، بر روی لینک Clone کلیک نمایید. در اینجا می‌توان آدرس مخزن کد Fork شده را جهت تهیه یک Clone مشخص کرد؛ به همراه محلی که قرار است این Clone در آن ذخیره شود.

آدرس HTTPS وارد شده، در کنار تمام مخازن کد GitHub قابل مشاهده هستند:

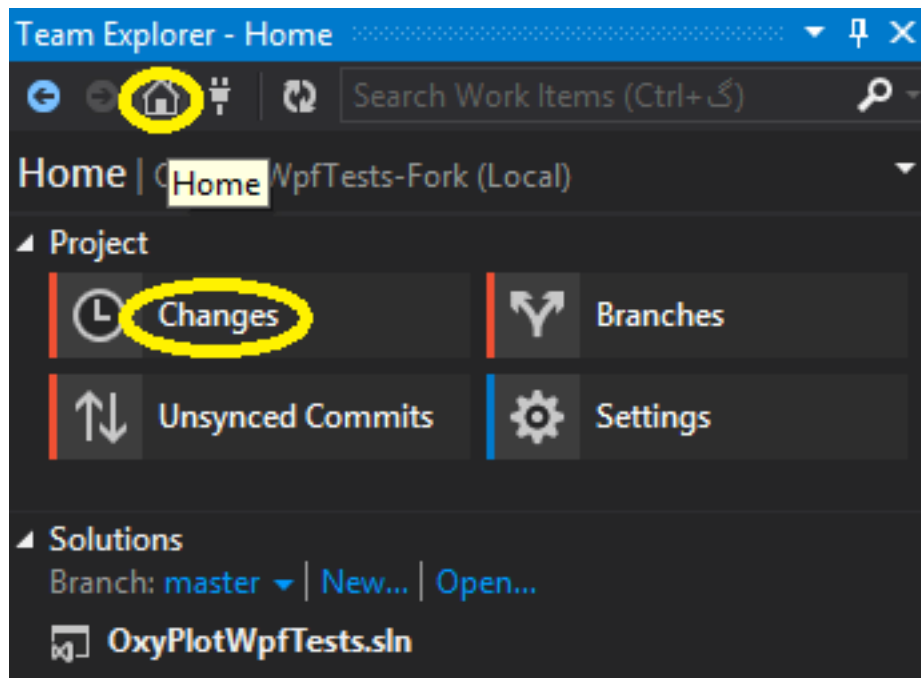


پس از تکمیل این دو آدرس، بر روی دکمه‌ی Clone کلیک نمایید. پس از پایان کار، اگر به آدرس محلی داده شده بر روی کامپیوتر خود مراجعه کنید، یک کپی از فایل‌های این مخزن، قابل مشاهده هستند.

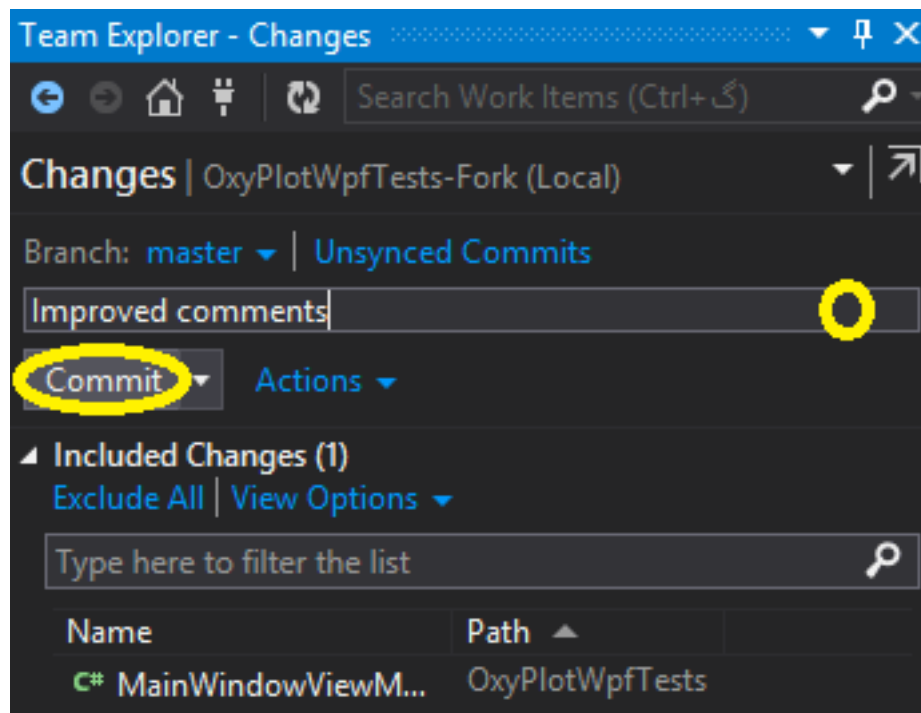
#### اعمال تغییرات محلی و ارسال آن به سرور GitHub

در ادامه، این پروژه‌ی جدید را در VS.NET باز کرده و تغییرات خود را اعمال کنید. اکنون نوبت به ارسال این تغییرات به سرور GitHub است. برای این منظور به برگه‌ی Team Explorer مراجعه کرده و بر روی دکمه‌ی Home آن کلیک کنید. سپس گزینه‌ی

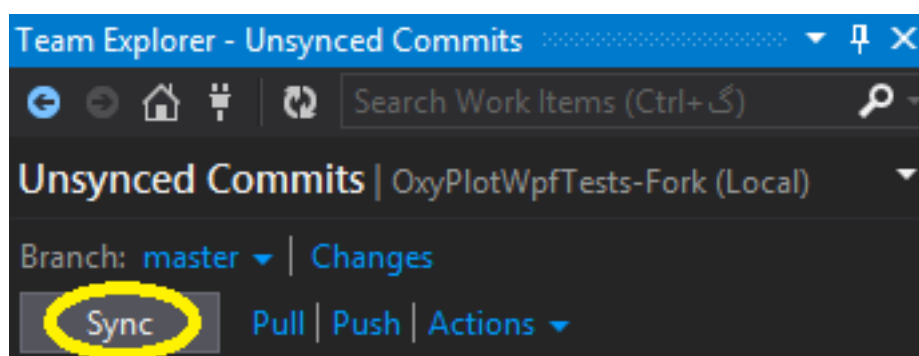
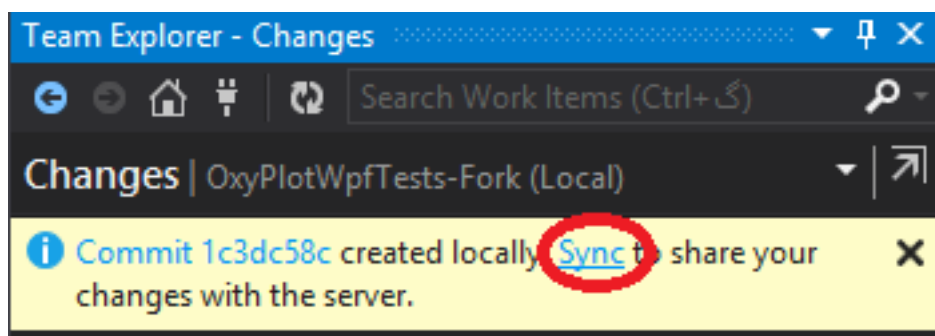
Changes را انتخاب نمایید:



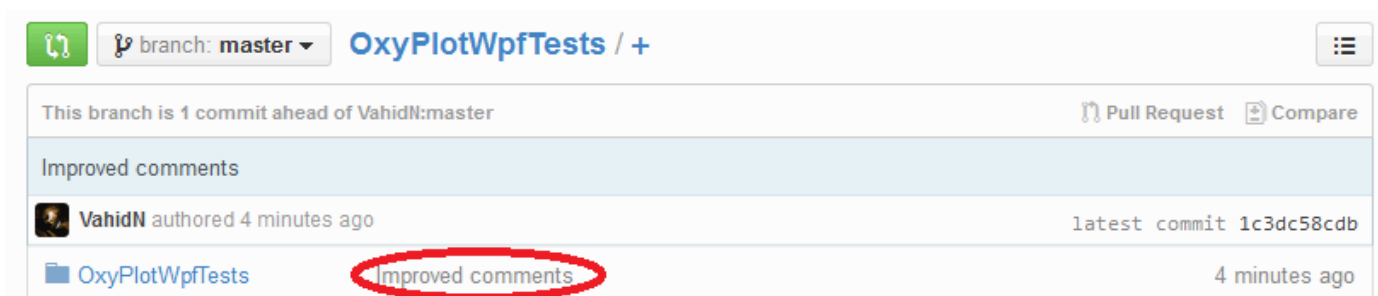
در اینجا توضیحاتی را نوشته و سپس بر روی دکمه‌ی Commit کلیک کنید.



پس از هماهنگ سازی محلی، اکنون نوبت به هماهنگ سازی این تغییرات با مخزن کد GitHub است. بنابراین بر روی لینک Sync در پیام ظاهر شده کلیک کنید و در صفحه‌ی بعدی نیز بر روی دکمه‌ی Sync کلیک نمایید:



اکنون اگر به پروژه‌ی GitHub خود مراجعه کنید، این تغییر جدید قابل مشاهده‌است:



مطلع سازی صاحب اصلی مخزن کد از تغییرات انجام شده

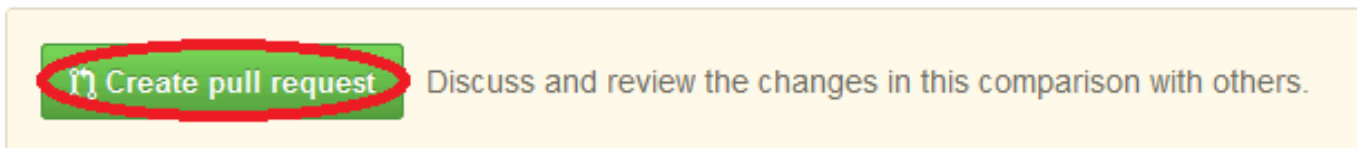
تا اینجا کسی از تغییرات جدید انجام شده‌ی توسط ما باخبر نیست. برای اطلاع رسانی در مورد این تغییرات، به مخزن کد Fork شده که اکنون تغییرات جدید به آن ارسال شده‌اند، مراجعه کنید. سپس در کنار صفحه بر روی لینک Pull request کلیک نمائید:



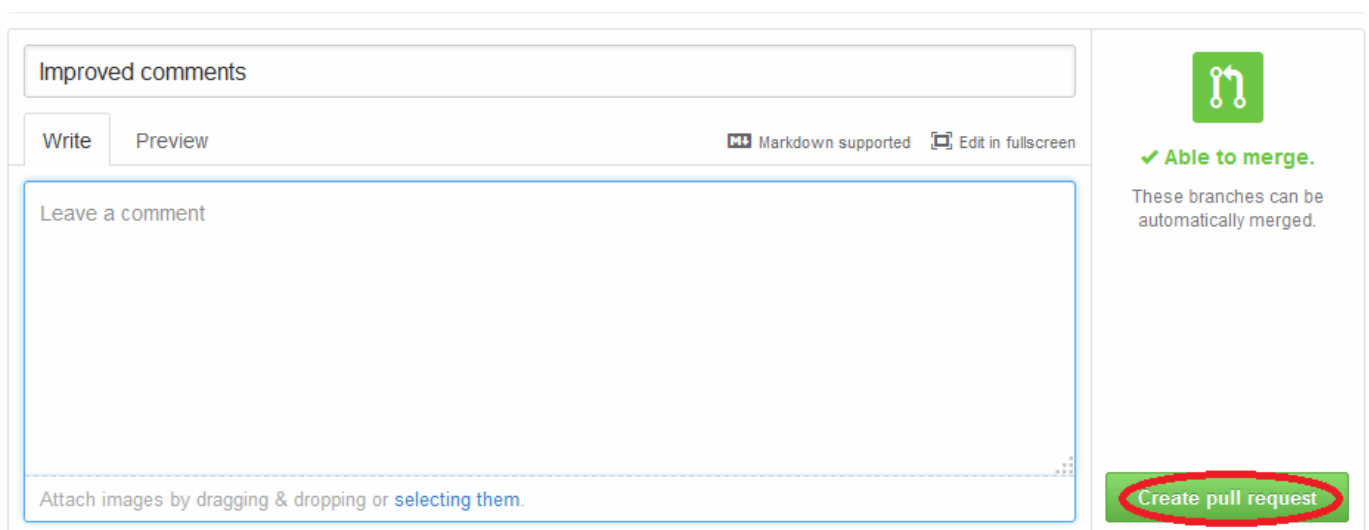
در اینجا بر روی دکمه‌ی New pull request کلیک کنید:



در ادامه تغییرات ارسال شما نمایش داده خواهند شد. آن‌ها را بررسی کرده و مجدداً بر روی دکمه‌ی Create pull request کلیک کنید:

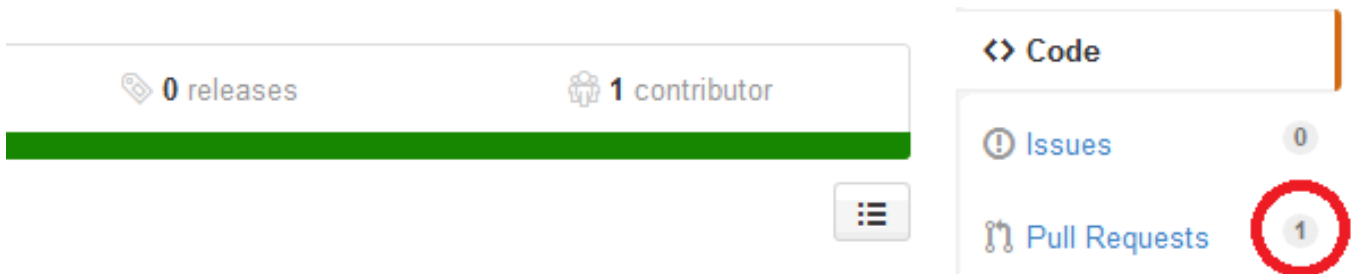


در اینجا عنوان و توضیحاتی را وارد کرده و سپس بر روی دکمه‌ی Create pull request کلیک نمایید:



## یکی سازی تغییرات با مخزن اصلی

اکنون صاحب اصلی مخزن کد یک ایمیل را دریافت خواهد کرد؛ همچنین اگر به مخزن کد خود مراجعه نماید، آمار Pull requests دریافتی قابل مشاهده است:

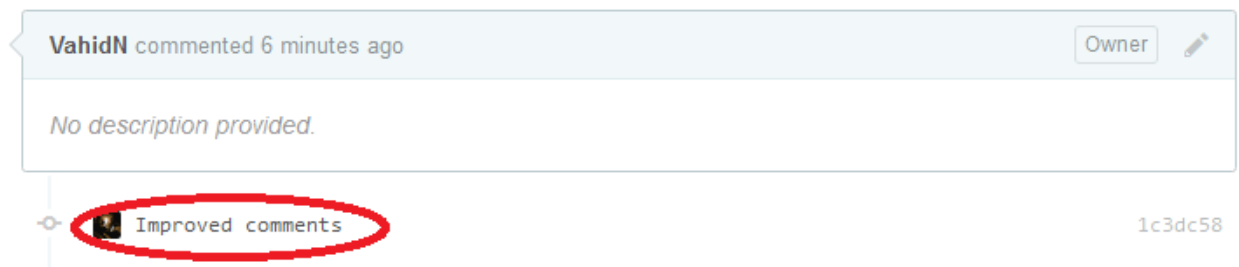


پس از انتخاب یکی از آن‌ها، لینکی برای بررسی تغییرات انجام شده و همچنین دکمه‌ای برای یکی سازی آن‌ها با پروژه‌ی اصلی وجود دارد:

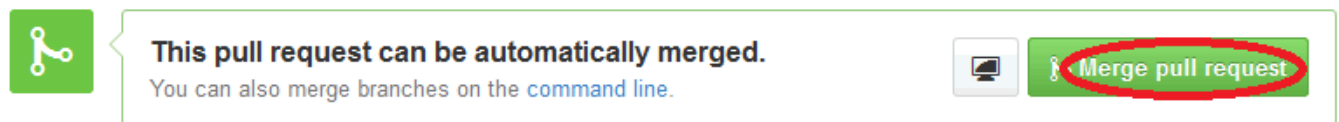
## Improved comments #1

**Open** VahidN wants to merge 1 commit into `VahidN:master` from `DotNETTips:master`

Conversation 0 Commits 1 Files changed 1

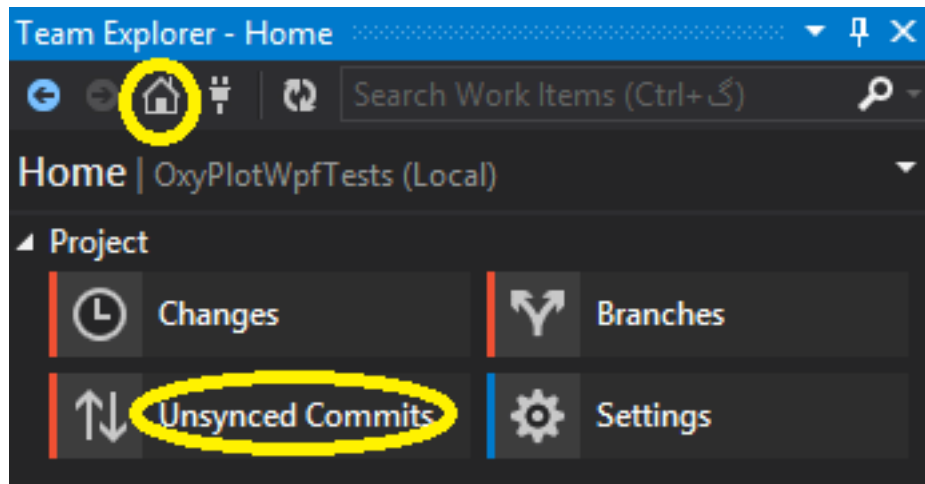


Add more commits by pushing to the **master** branch on **DotNETTips/OxyPlotWpfTests**.



## دریافت این تغییرات در مخزن کد محلی توسط صاحب اصلی پروژه

اکنون که این تغییرات با پروژه‌ی اصلی Merge و یکی شده‌اند، صاحب اصلی پروژه جهت تهیه‌ی یک کپی محلی و بهبود یا تغییر آن‌ها می‌تواند به صورت ذیل عمل کند:



ابتدا به برگه‌ی Team explorer مراجعه کرده و بر روی دکمه‌ی Home آن کلیک کنید. سپس گزینه‌ی Unsynced commits را انتخاب نمایید. در صفحه‌ی باز شده بر روی دکمه‌ی Sync کلیک نمایید. به این ترتیب آخرین تغییرات را از مخزن کد GitHub به صورت خودکار دریافت خواهید کرد:

