# Exercise: Version Control Using GitHub

## 1. Preparation – sign up for GitHub

#### Step 1

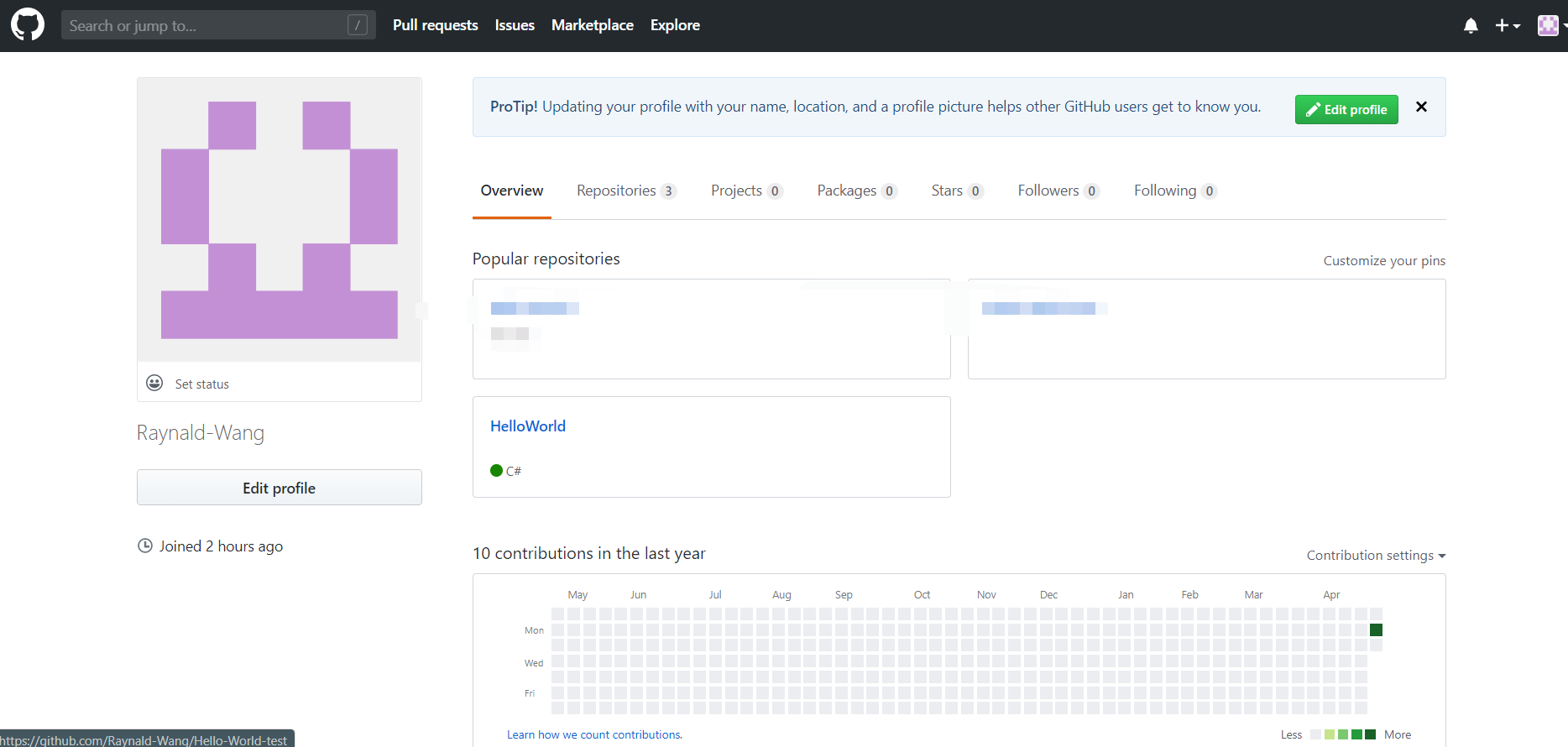
Open a browser and go to <https://github.com/>

#### Step 2

If you already had a GitHub account, you can just log on with it. Otherwise, you would need to create a new account. The registration process will be completed in a few seconds. Of course, email verification should be required.

#### Step 3

Now, you should see a dashboard as shown in Figure 1. Congratulations! Let’s carry on to the tasks.



## 2. Tasks

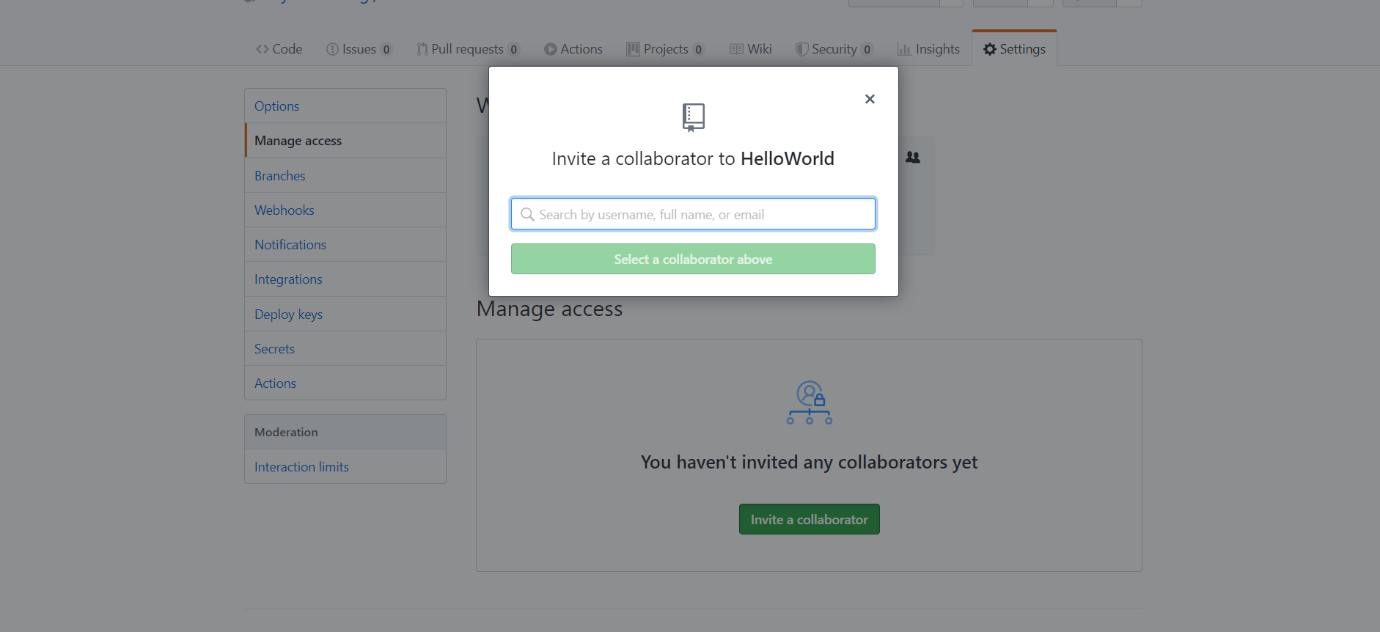
### ***2.1 Working with GitHub in Visual Studio 2019***

GitHub has multiple ways of teamwork projects, as shown below:

No1: Add Collaborators

Collaborators are similar to the Team model. The owner of the Repository can directly add collaborators to his warehouse, allowing collaborators to have almost the same rights as the owner.

Operation method: The Owner of the Repository clicks Settings in his repo to add the User who wants to cooperate



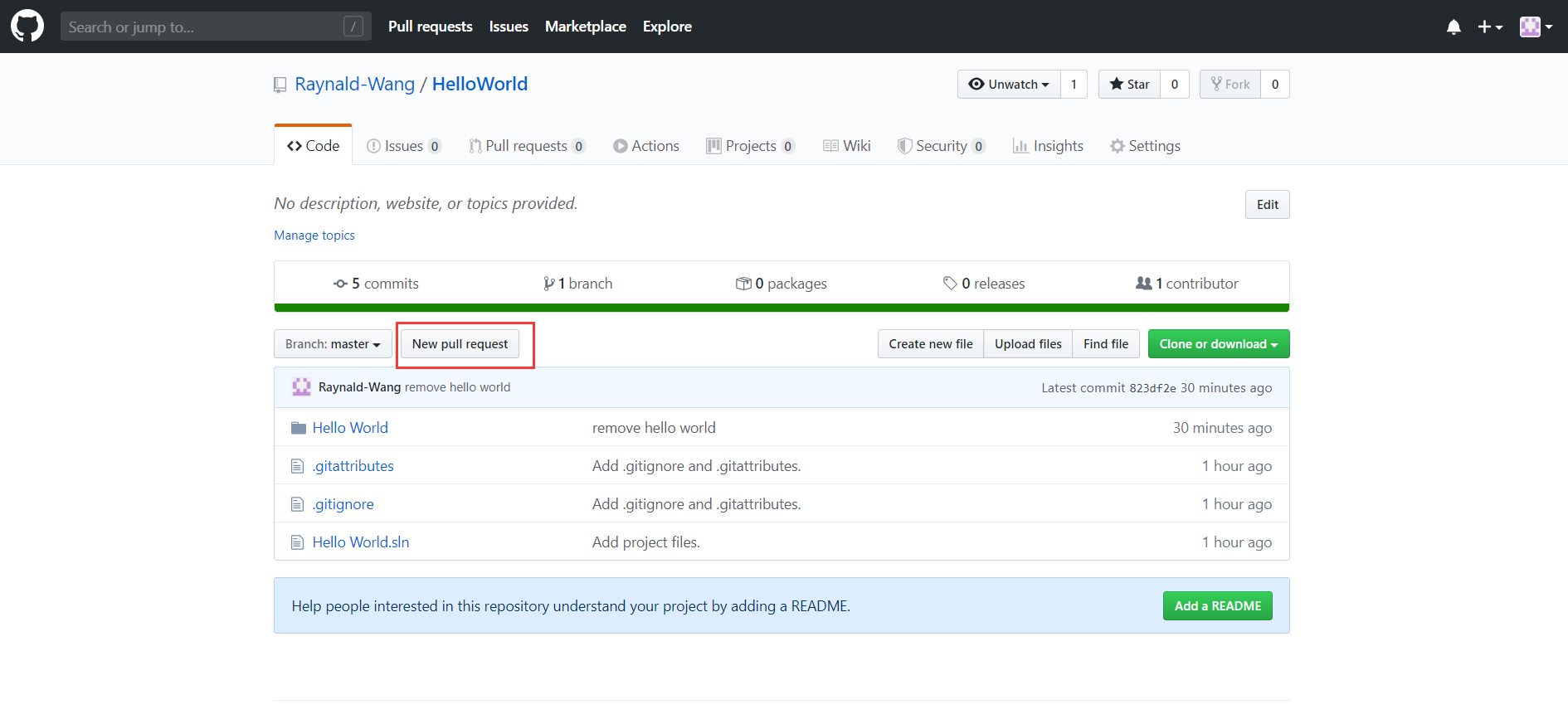
After the addition is successful, the invited collaborators will receive the invitation email sent by GitHub, and the email will be sent to the invitee's GitHub registered mailbox.

No2: Fork & Pull request

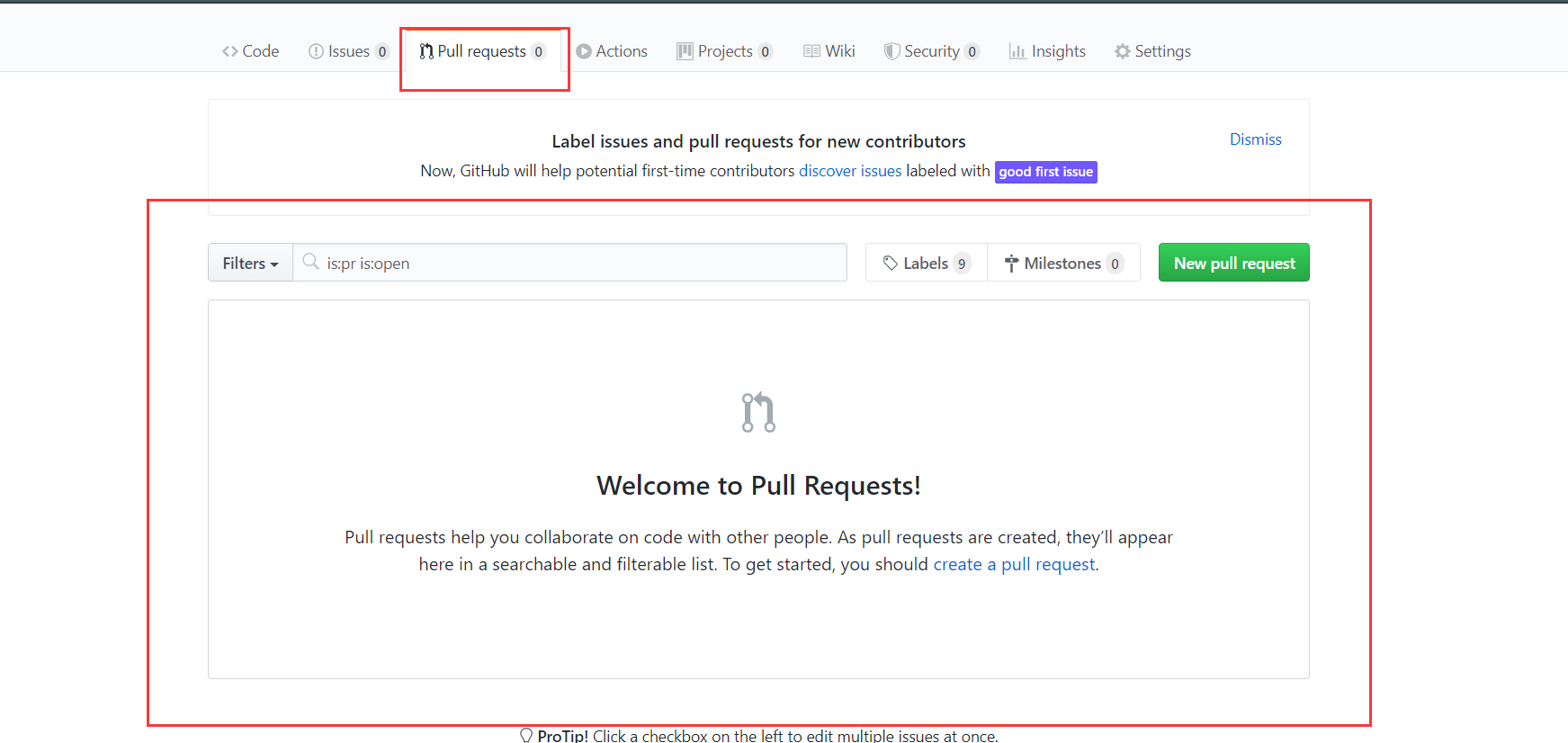
For cooperation between open source projects, the Fork & Pull request cooperation method is generally used. For example:

A has an open source Repository, B wants to participate in the open source project. As B, you must fork the A Repository of the open source project into the Repository of your account. In this way, B has a copy repo of A's open source project. Then, B can normally modify the project of A's open source project under his account (the changes here are all local).

It will not affect the open source repository of A itself, if one day B feels that it has added and improved a new function, or a bug of fixed A repo. Put a pull request and ask if A wants it.



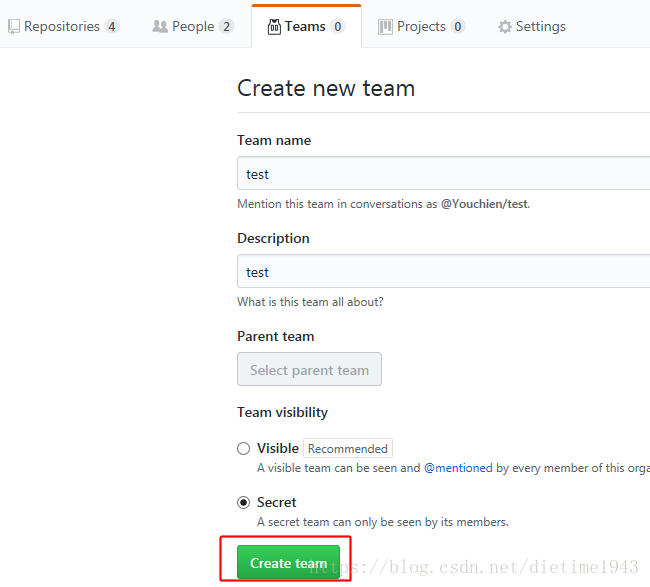
B Create pull request



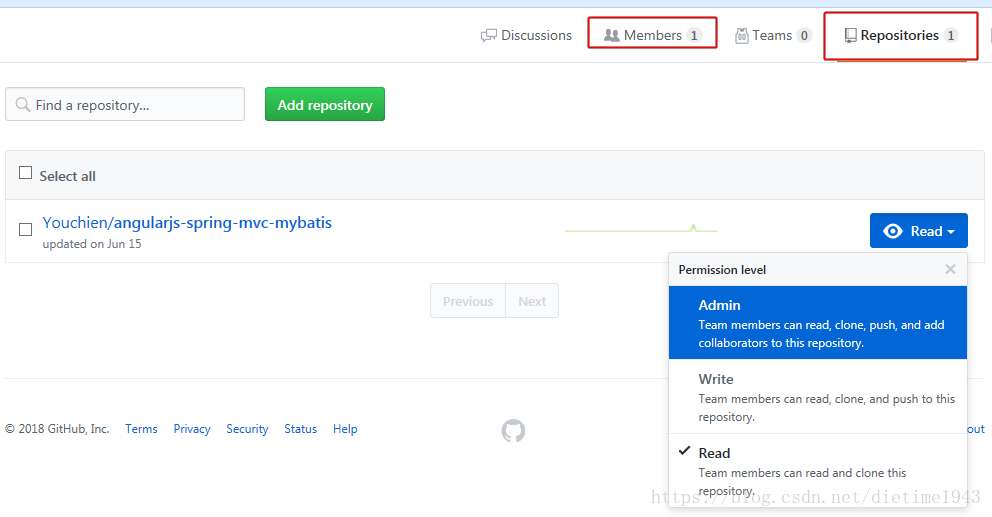
In this way, The pull request applied by B in the warehouse of A. Since I do not have two accounts, I cannot display the application. But the application will be displayed in a big red box.

No3: Organization & Team

If you have created an organization, you can create a team in the organization for cooperative development. The following illustrations are examples from other places.



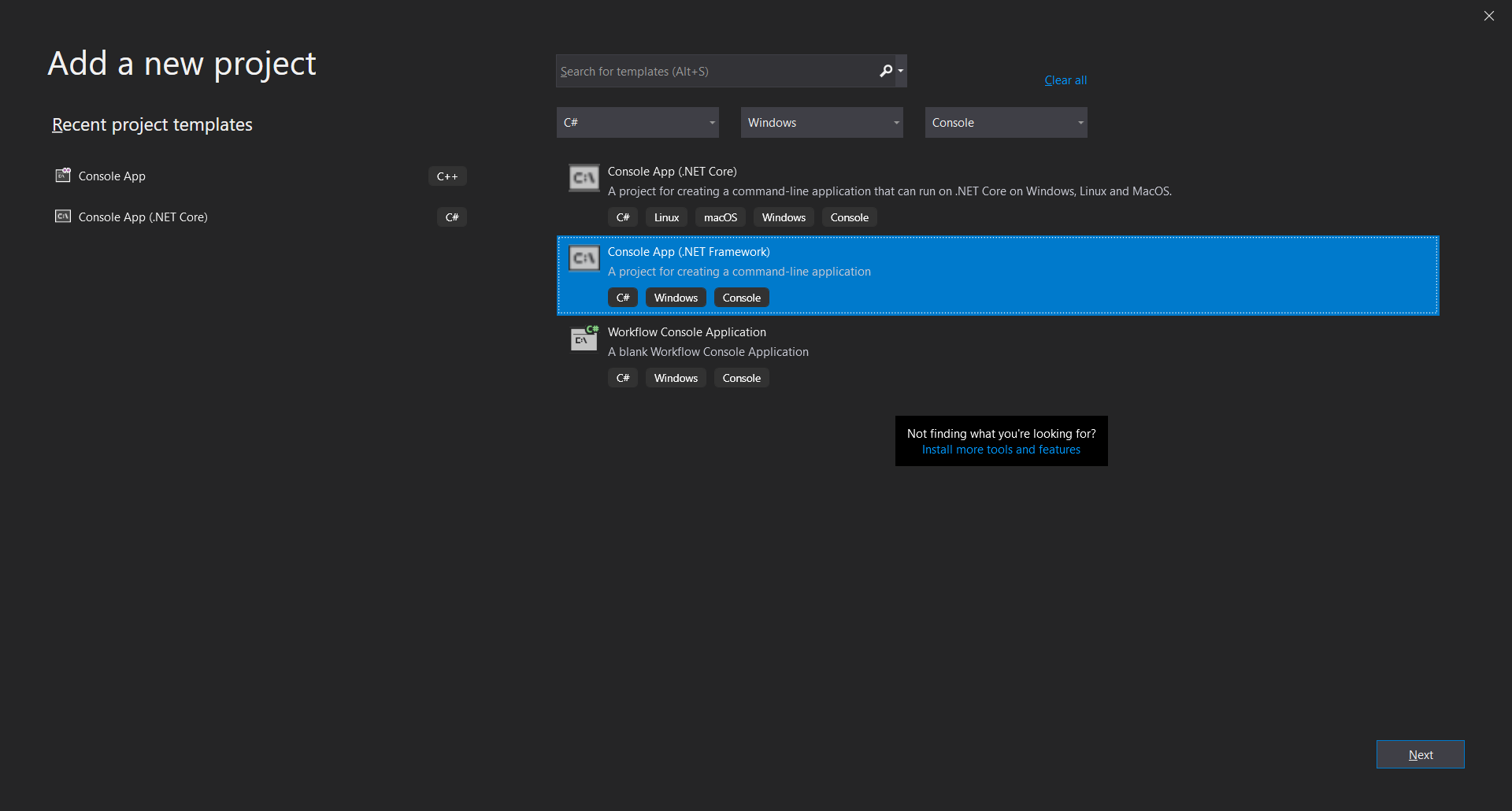
Manager A can add repo in the created team. Set its permissions and add team members, so that team members can cooperate to develop these projects.

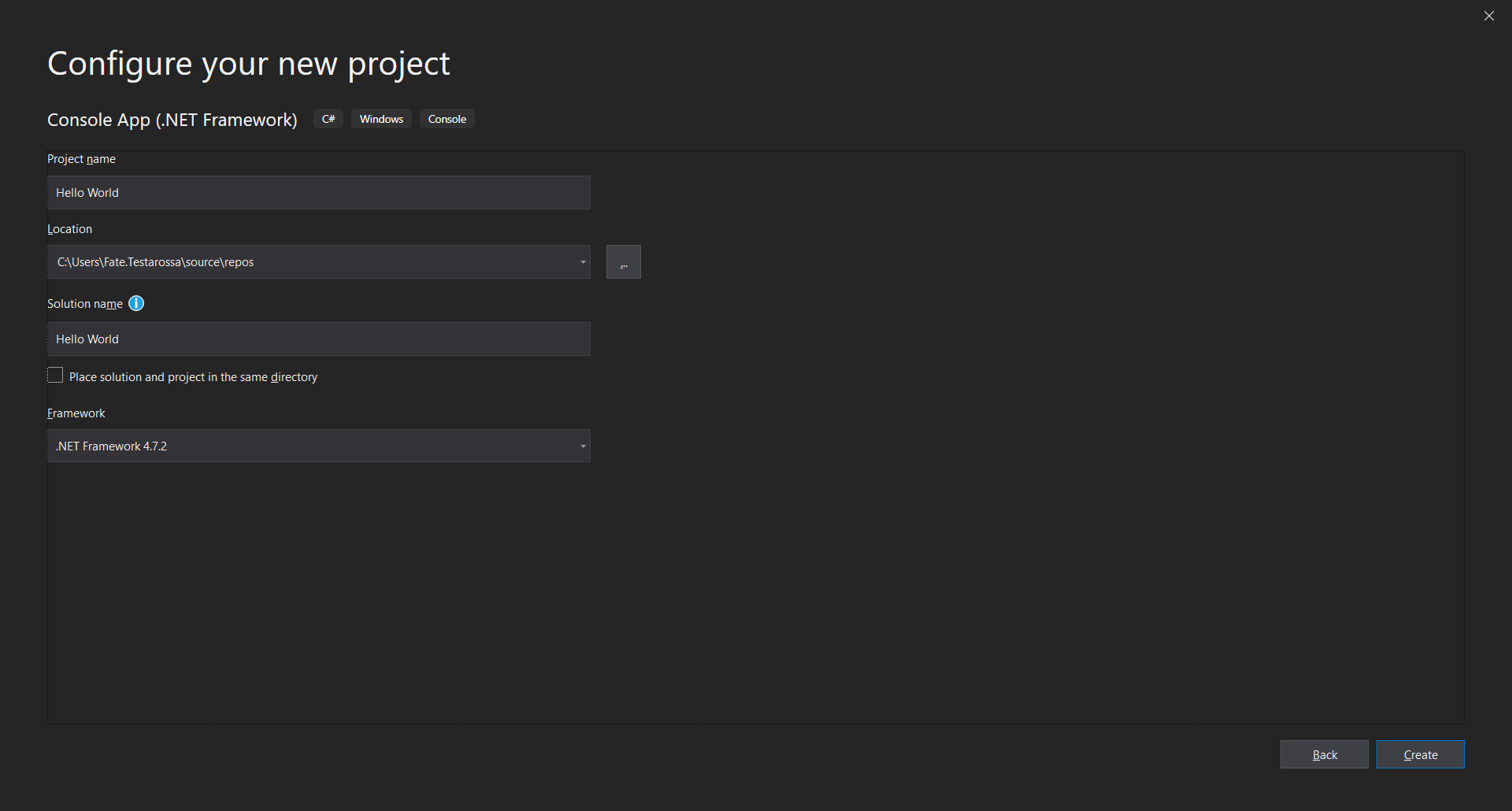


**After you have an account, you need to link with GitHub using visual Studio.**

#### Step 1

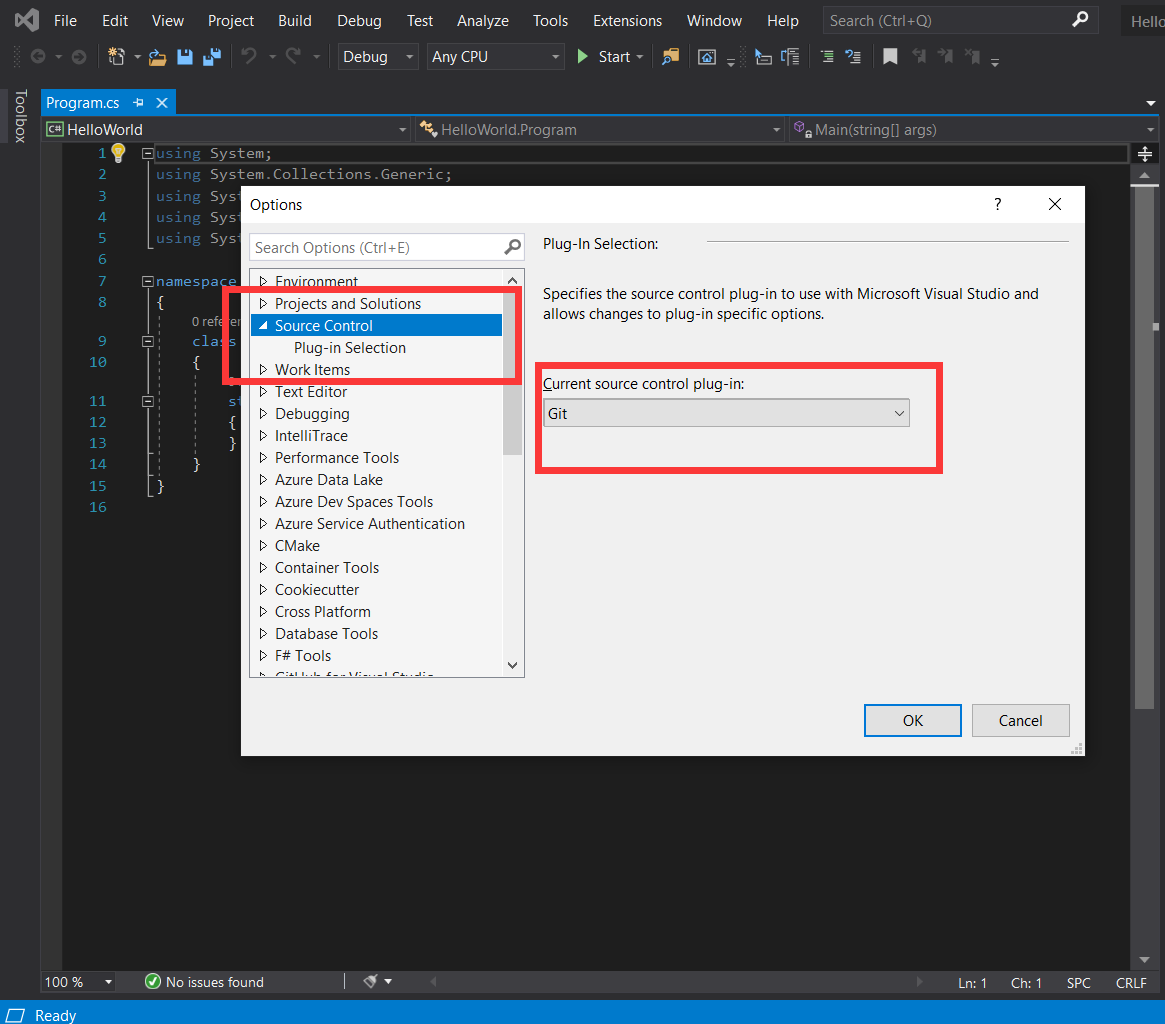
First you need a new project.





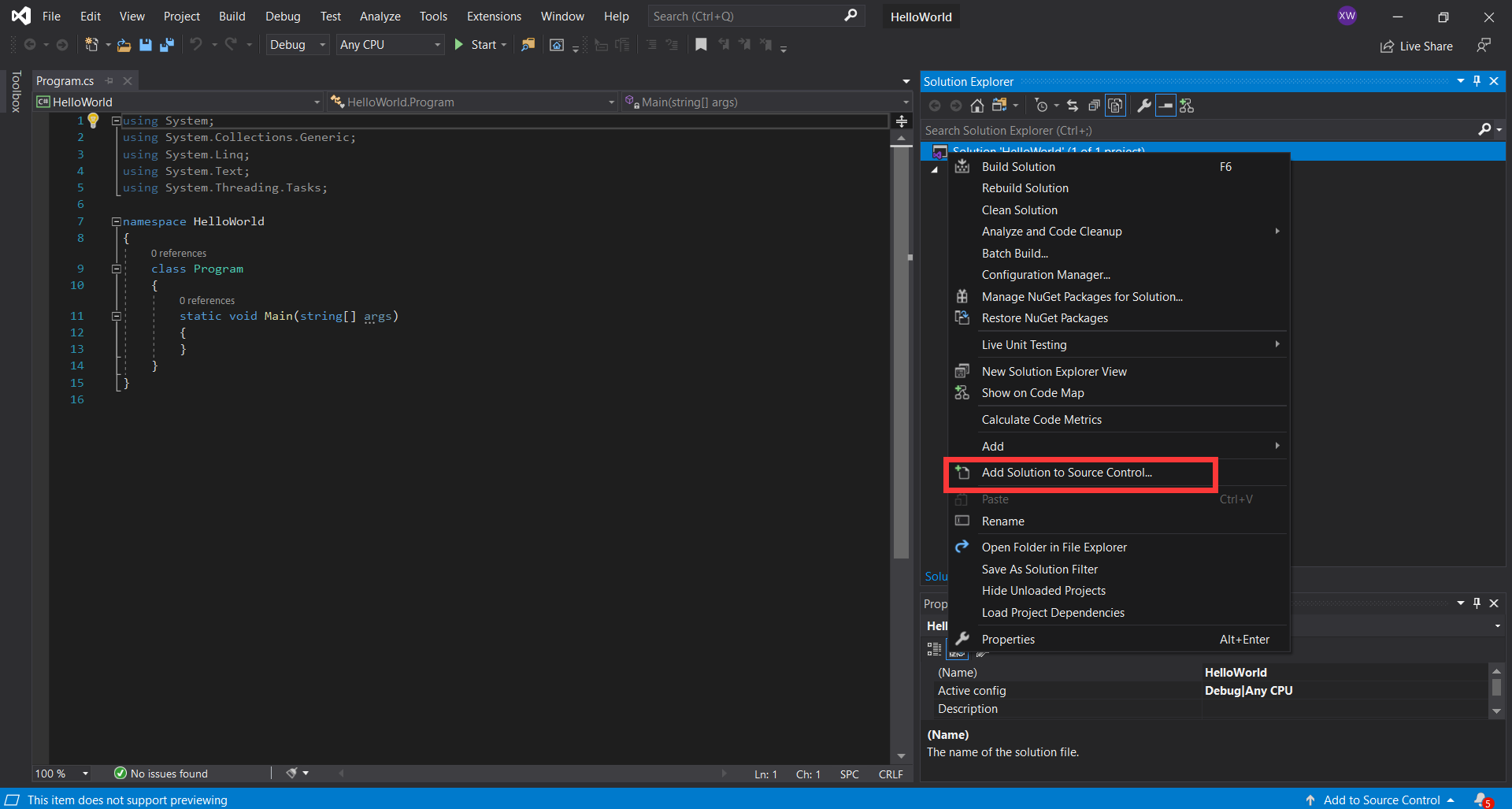
*Step 2*

After that, you need to setting up the Git source code plugin. In the top menu of VS, select [Tools]-[Options], then click [Source Control] and select "Git ", Then click “Ok”.



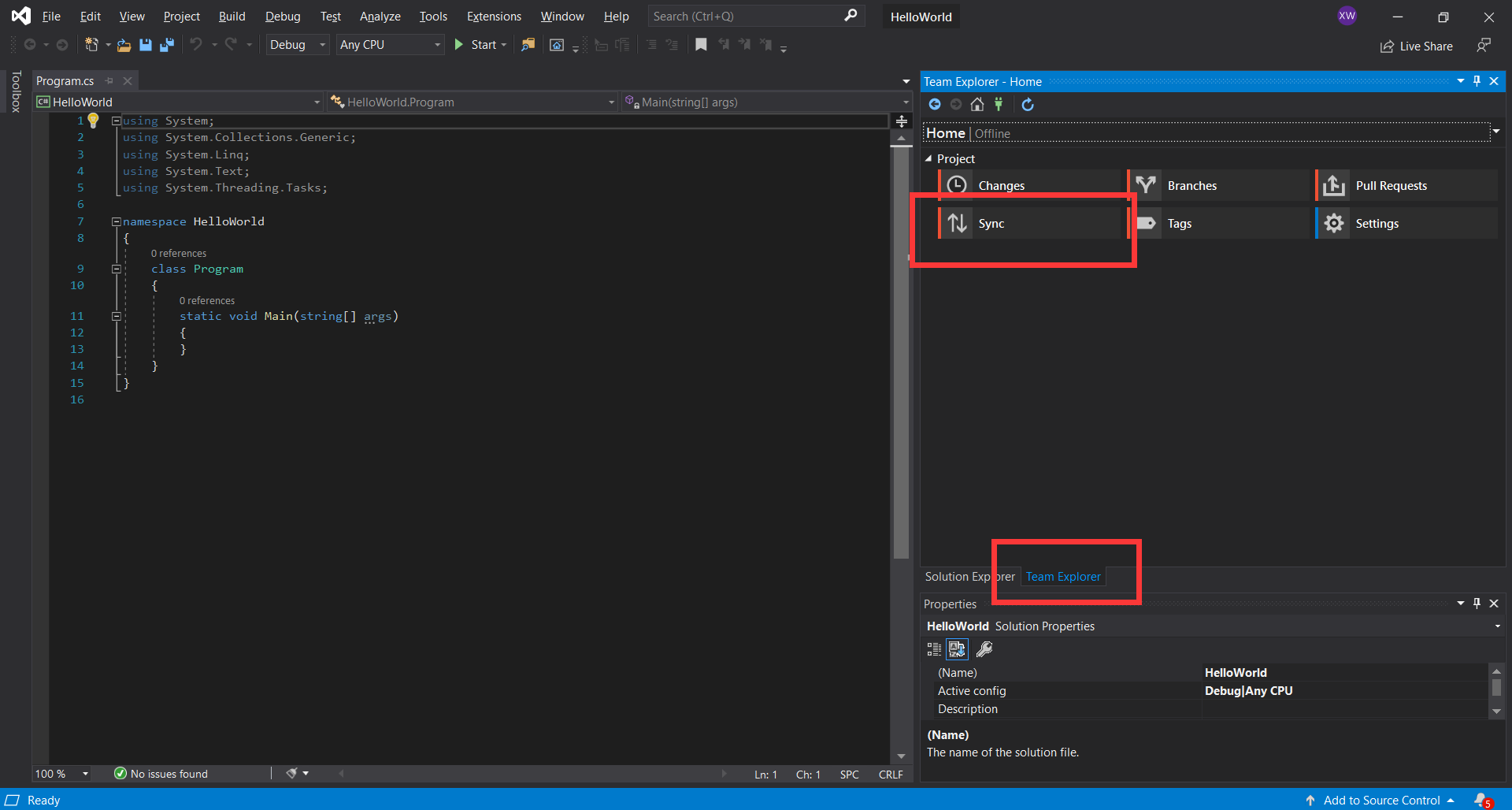
#### Step 3

Right click on the solution and add to source control



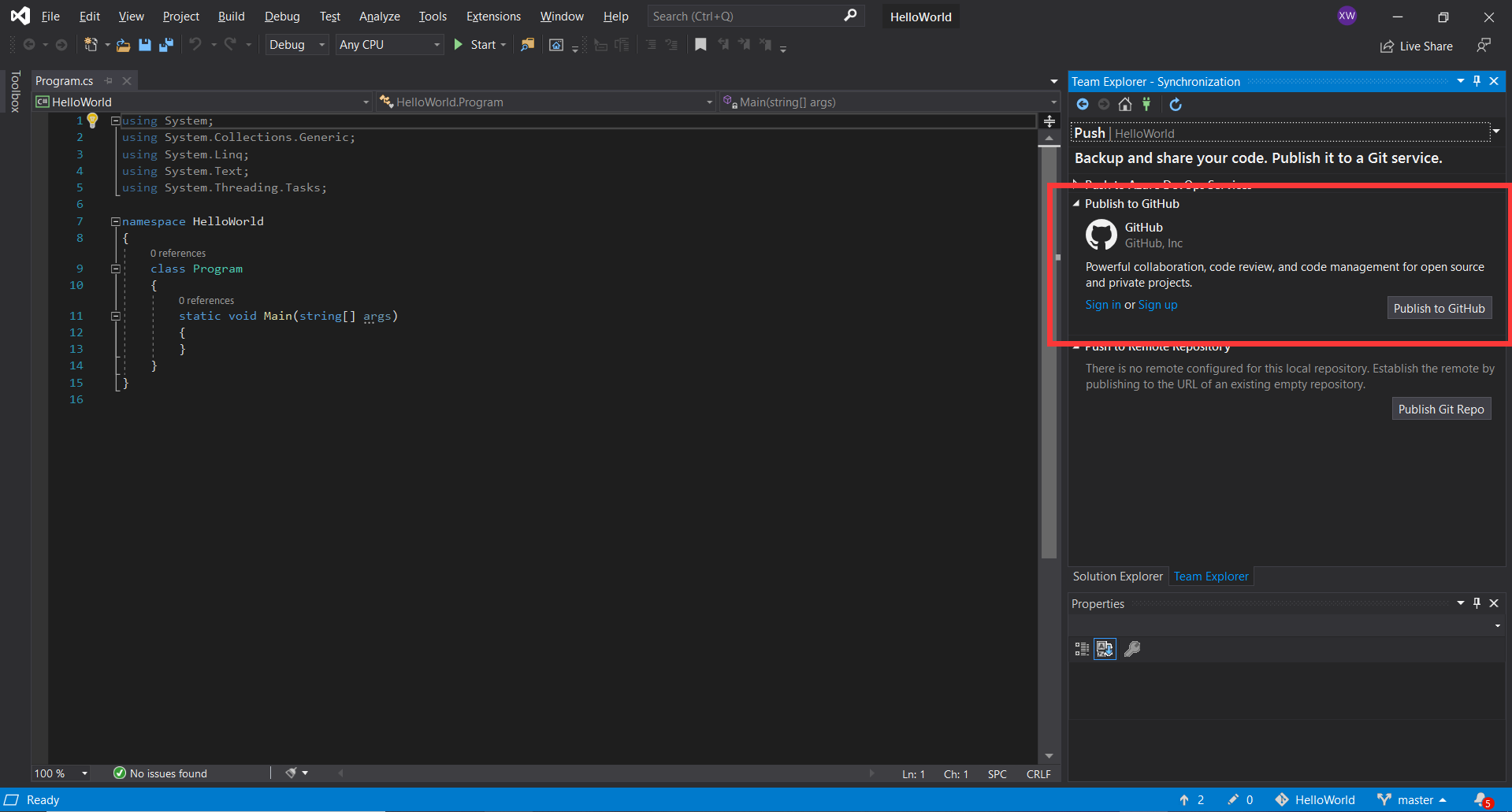
#### Step 4

Team Explorer-Synchronization

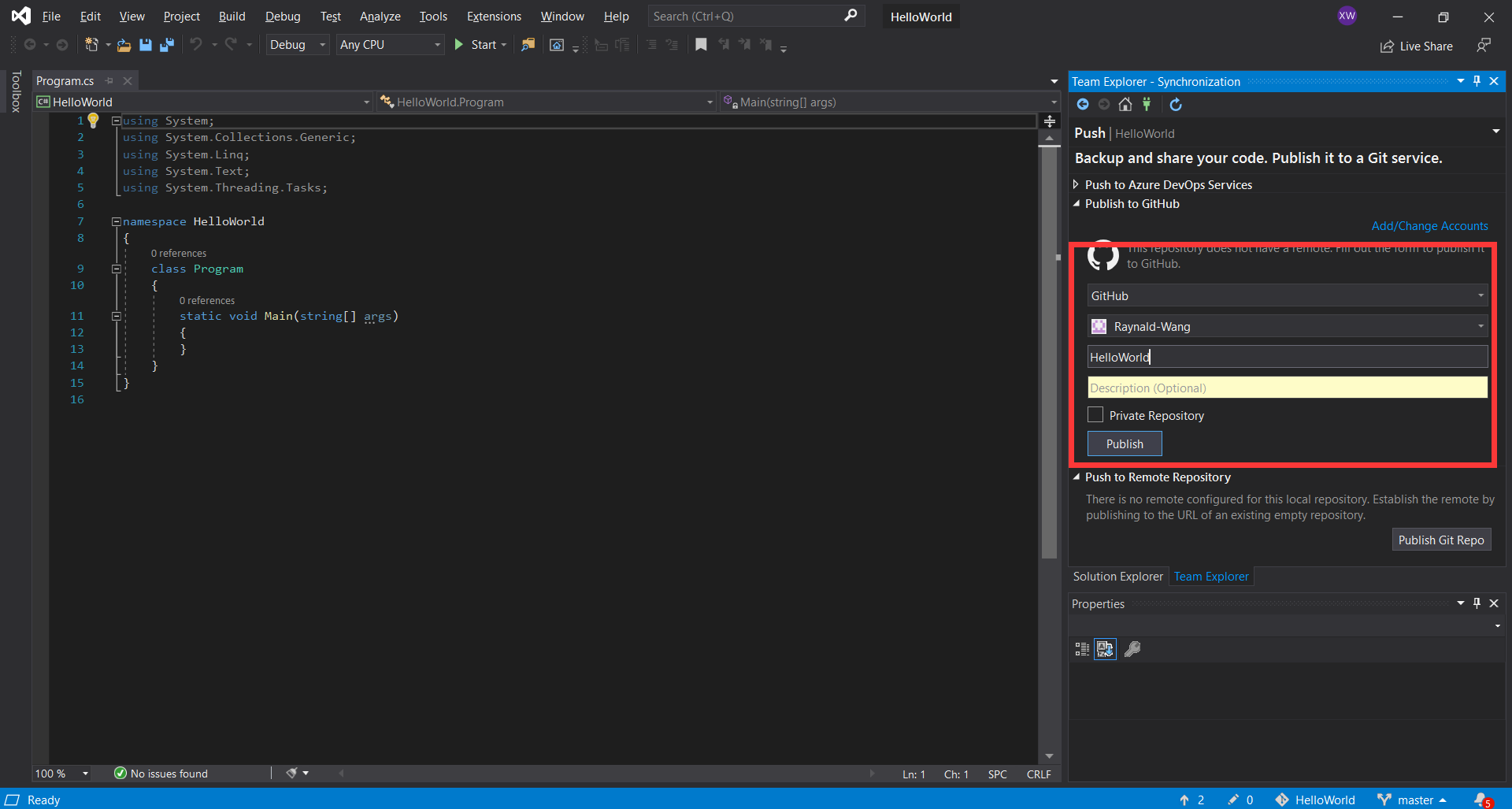


#### Step 5

Publish git repository-enter git address-publish

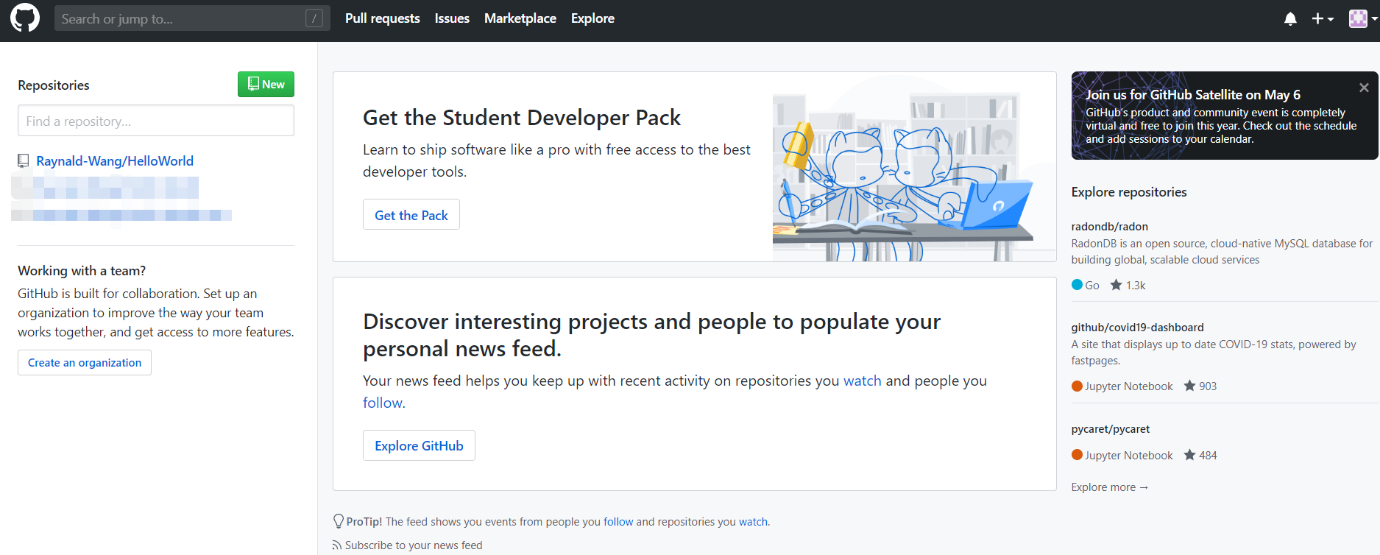


Here you need to fill in your GitHub account, and then vs will link to GitHub. After logging in to the account, fill in the warehouse name and warehouse description to be created, and click [Publish].



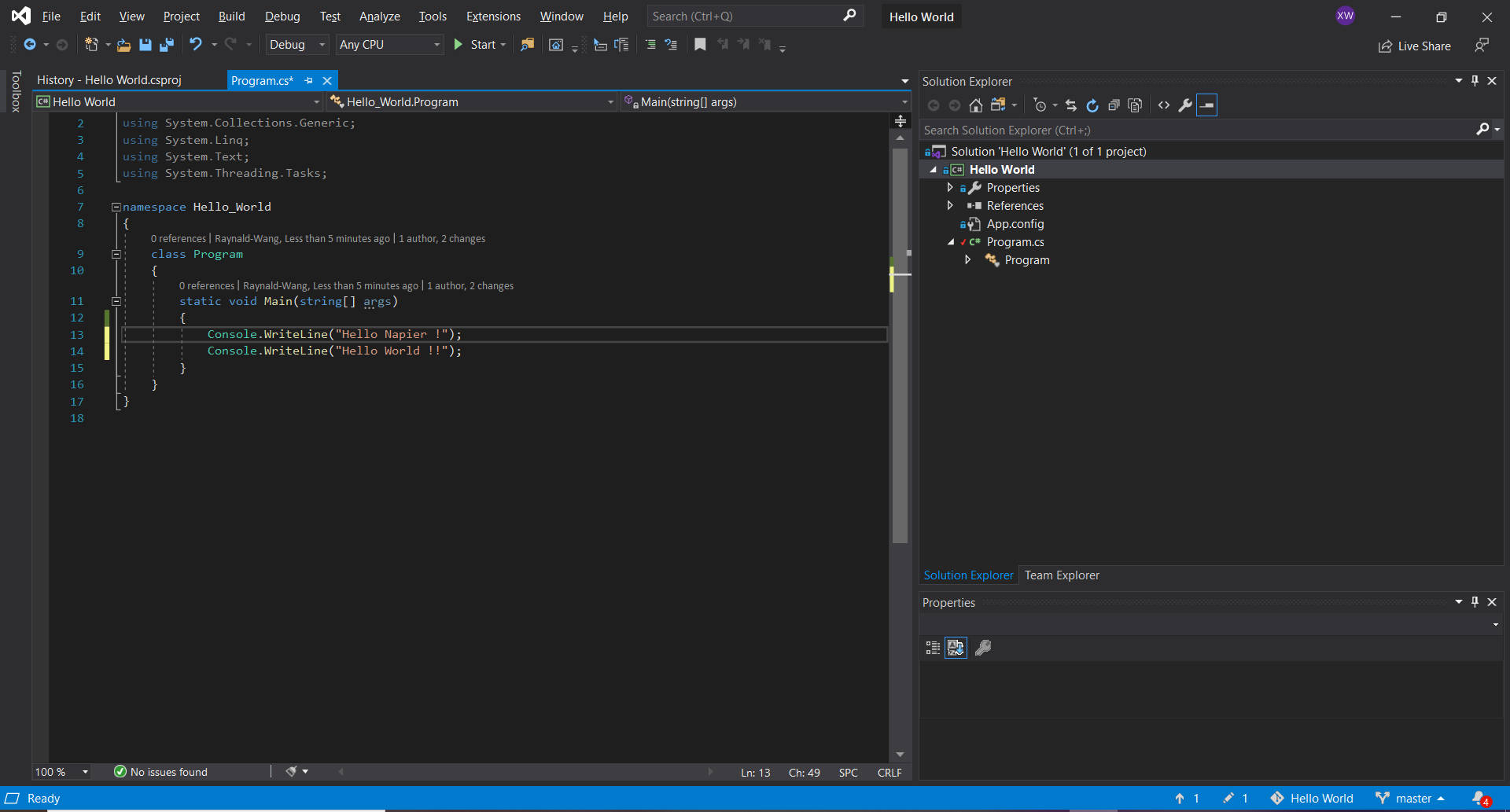
#### Step 6

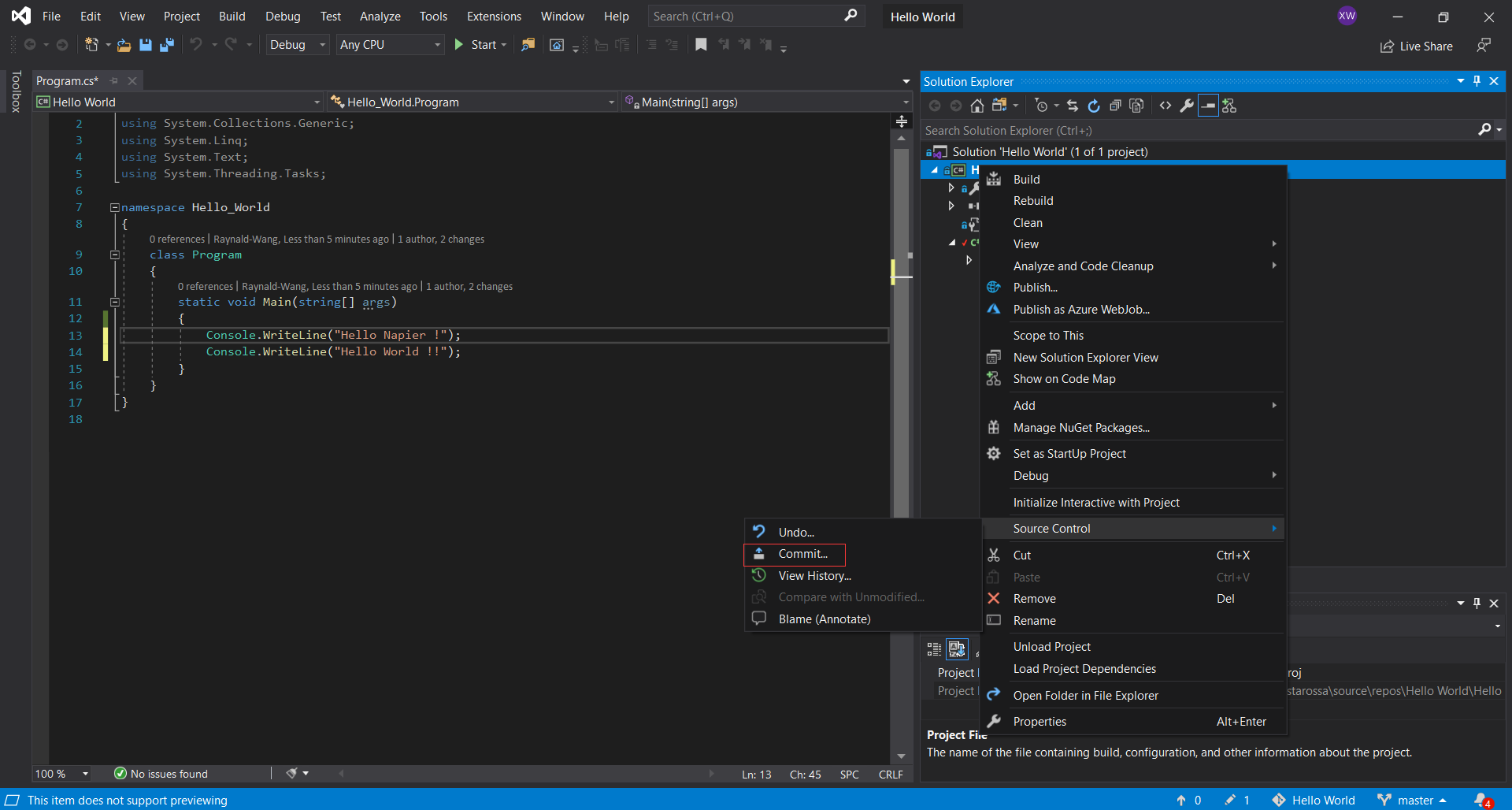
Open your GitHub homepage, you can see the warehouse you just created, and you can also see the files you just synchronized in the warehouse.



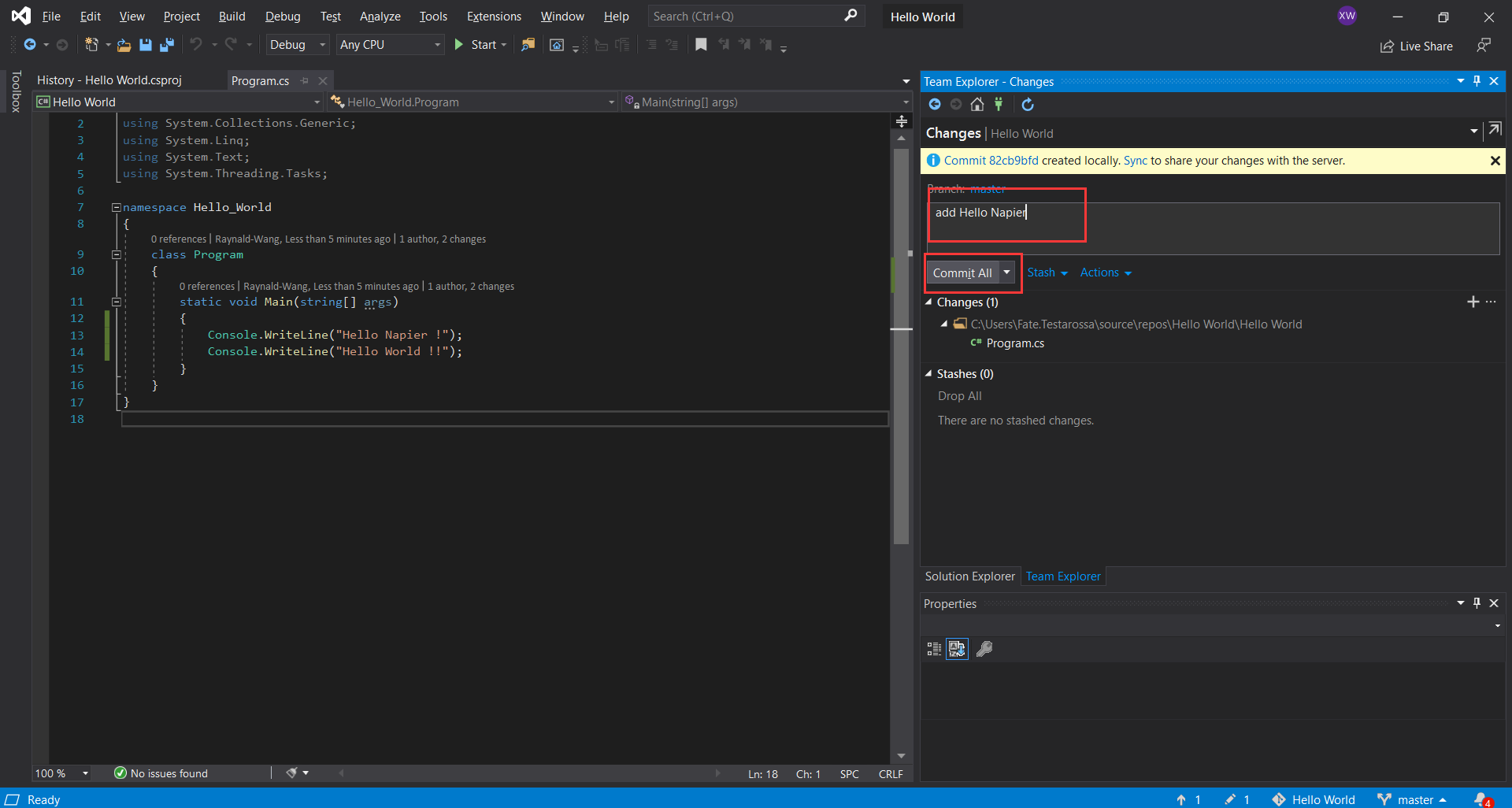
#### Step 7

When you update your project, you need to manually push the changes to GitHub. First, you must submit your changes to Git. If you modify some content in the Program.cs, you can see a red check mark in front of this file in the Solution Explorer, which means that this file has been modified, but has not been submitted to Git.( Right click on the project (note, not the right click solution))



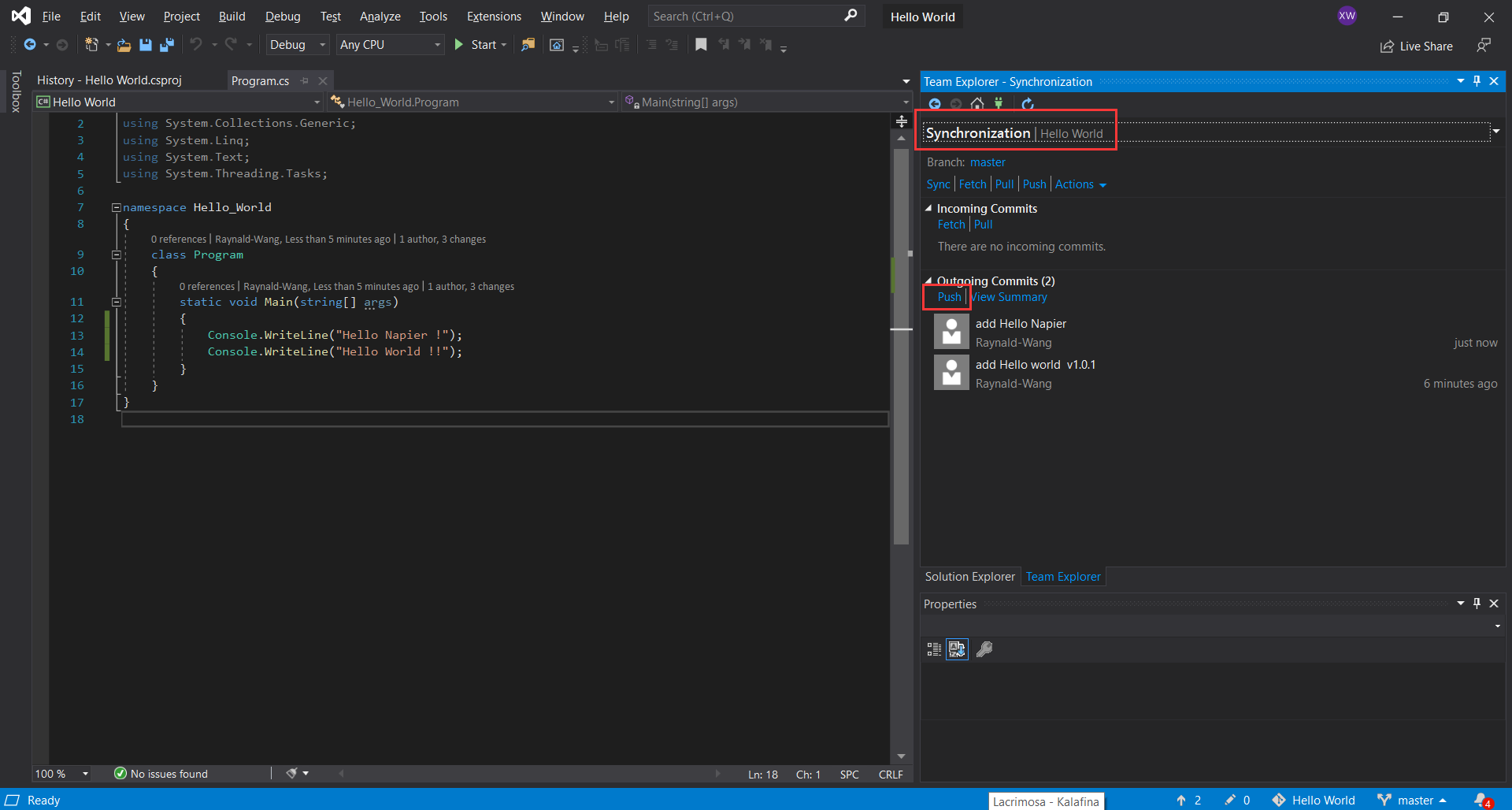


Fill in the submission message (commit), for example, here filled “add Hello Napier”. This message is used to mark this commit, so that when you check the historical version later, you can know what changes have been made this time. Click [Submit All], which just submits the changes to Git and then pushes them to GitHub.

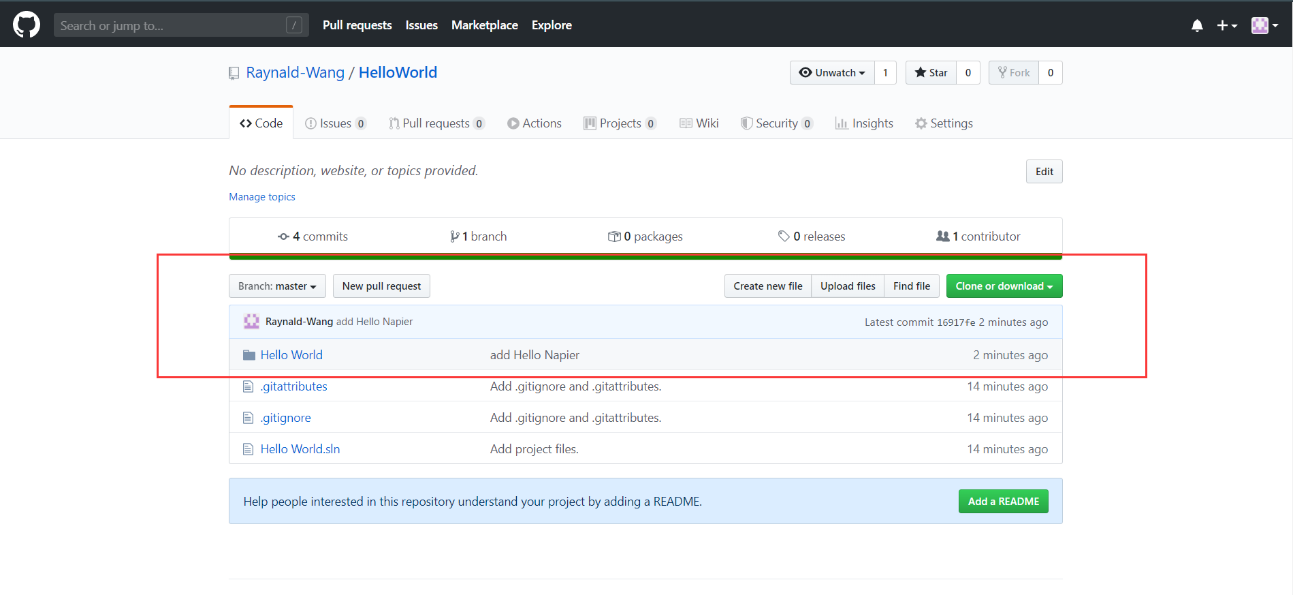


#### Step 8

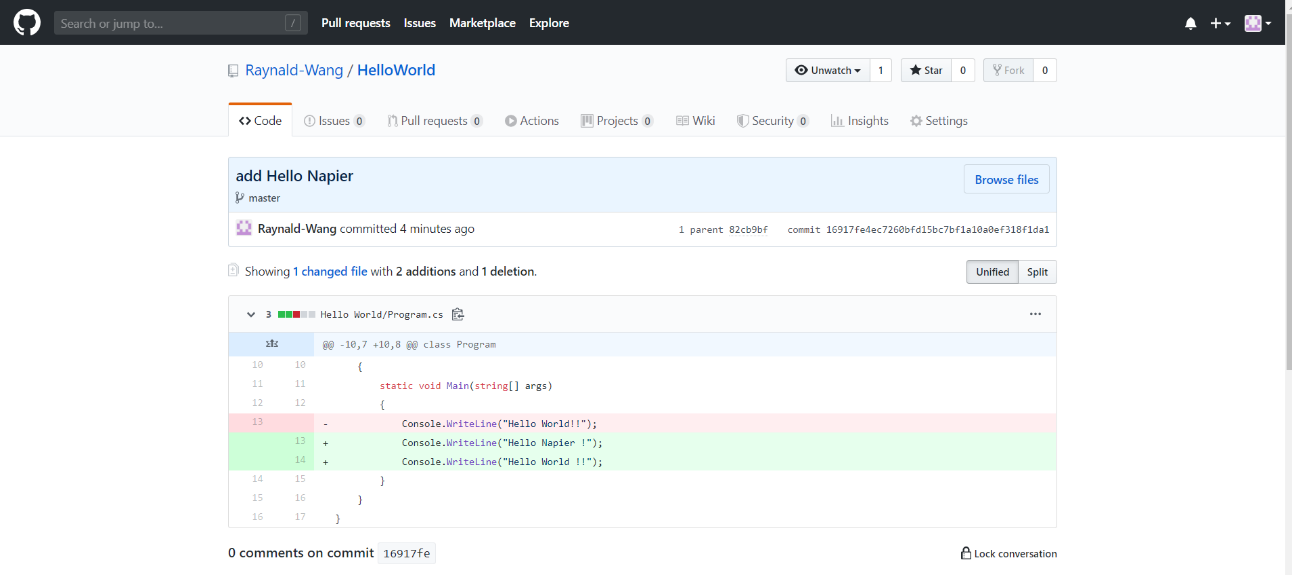
Click [Synchronization], and then [Push] the outgoing submission.



After completion, you can go to the GitHub page to see that the HelloWorld file has been updated.

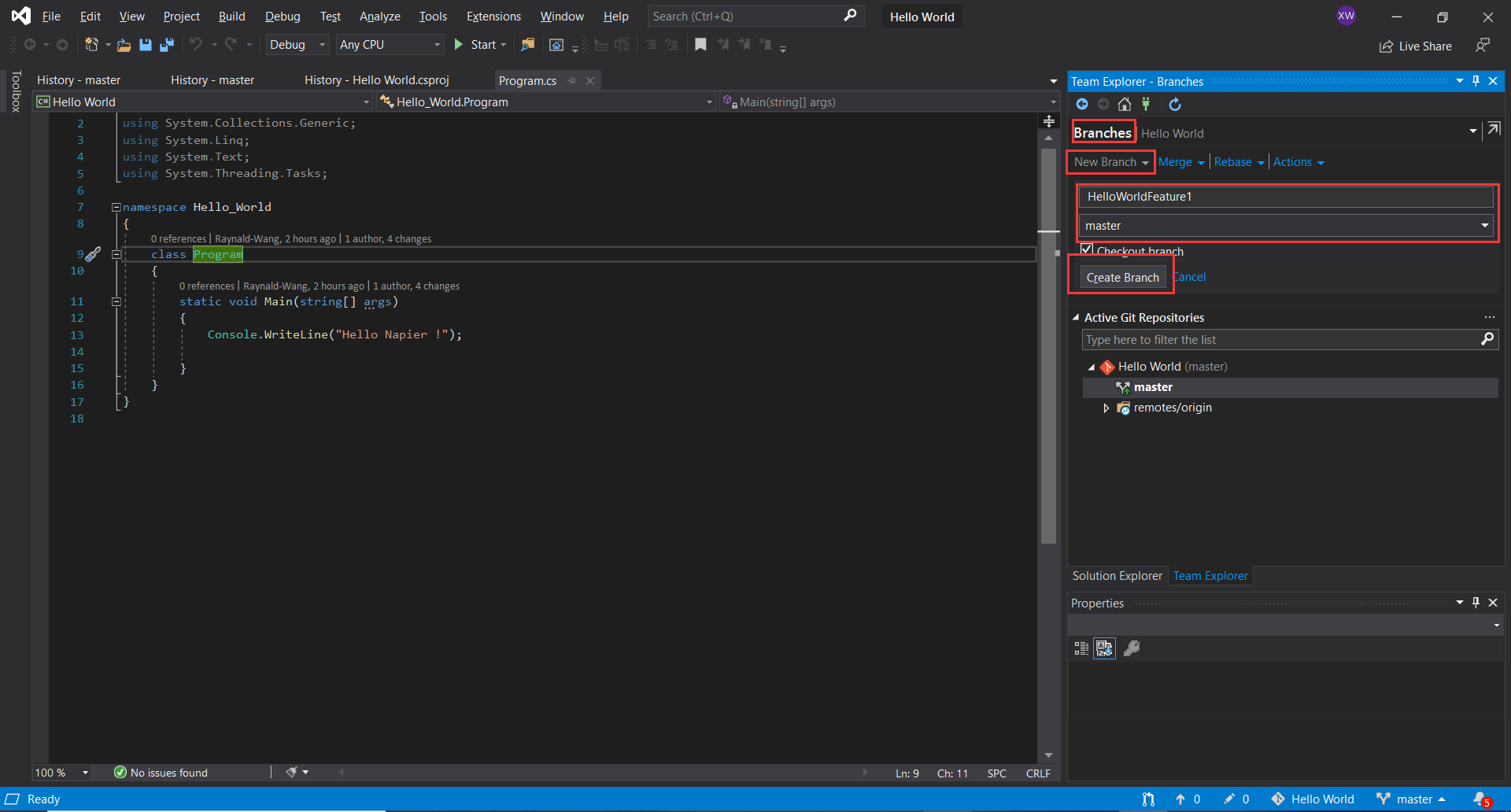


You can also check that part of the code has been modified on the GitHub page.

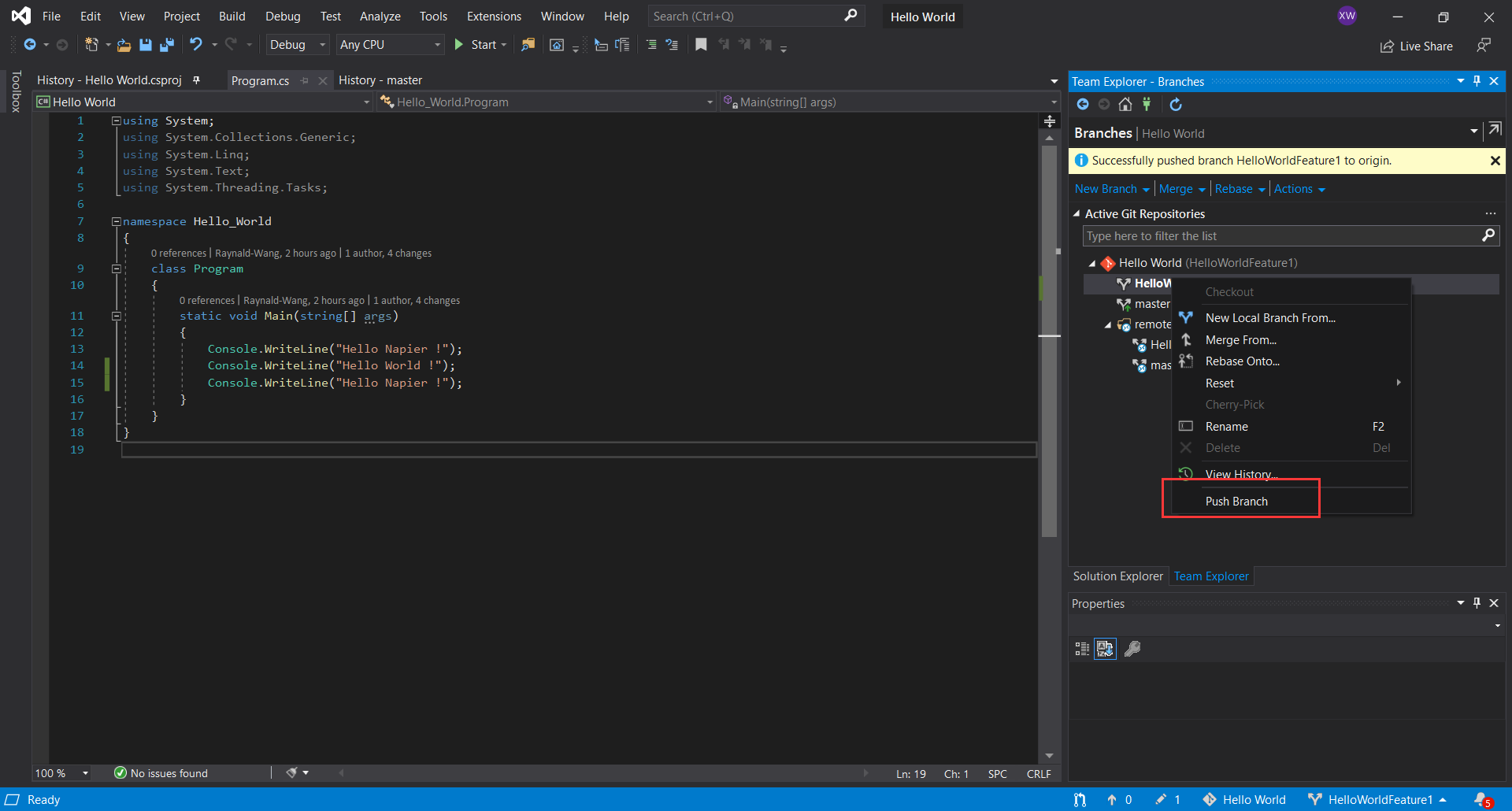


#### Step 9

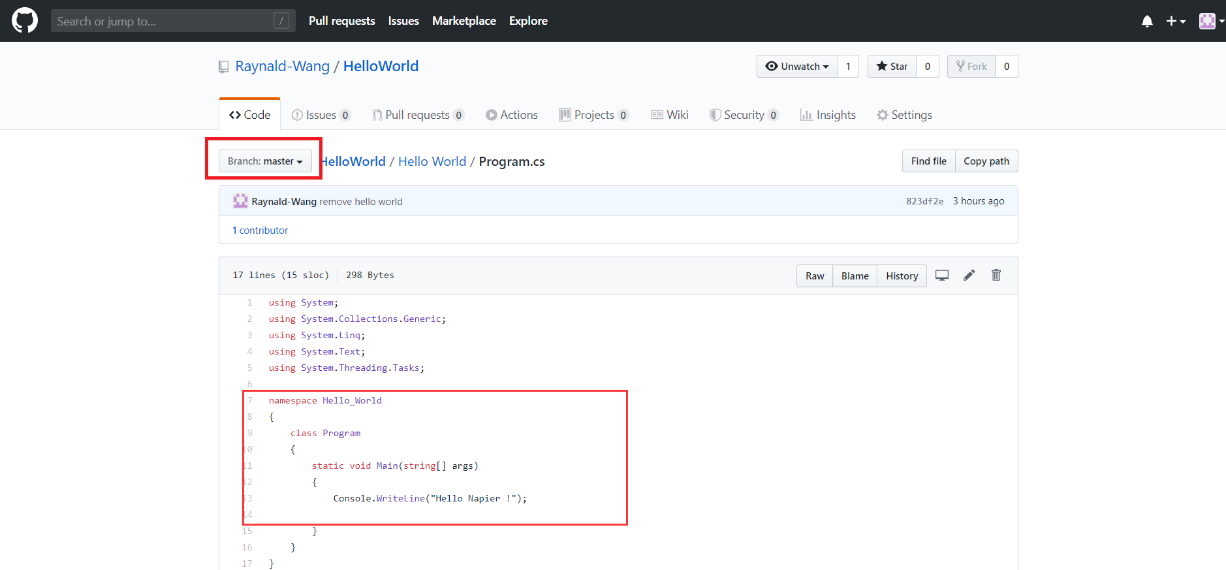
Now, let’s try to create a new branch of the project. In the Source Control Explorer, right click on the folder you want to branch from, e.g. HelloWorld, and then select **Branches** -> **Branch**, Give the target branch a wise name, e.g. HelloWorldFeature1.

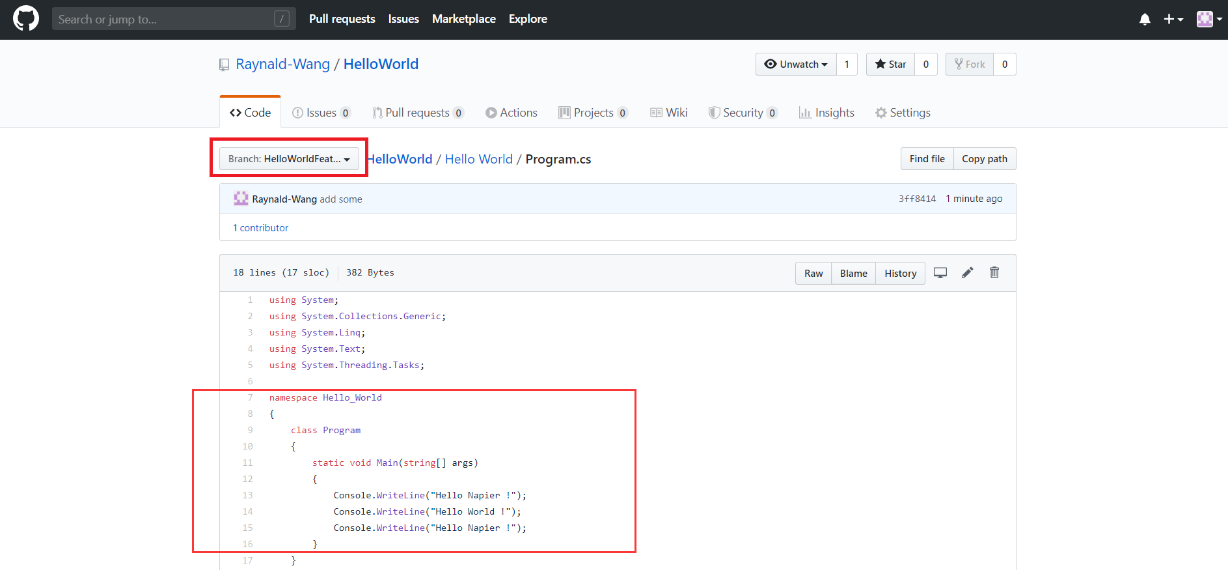


Since a new branch has been created, you can try new changes and upload them at will, which will not affect the original version.

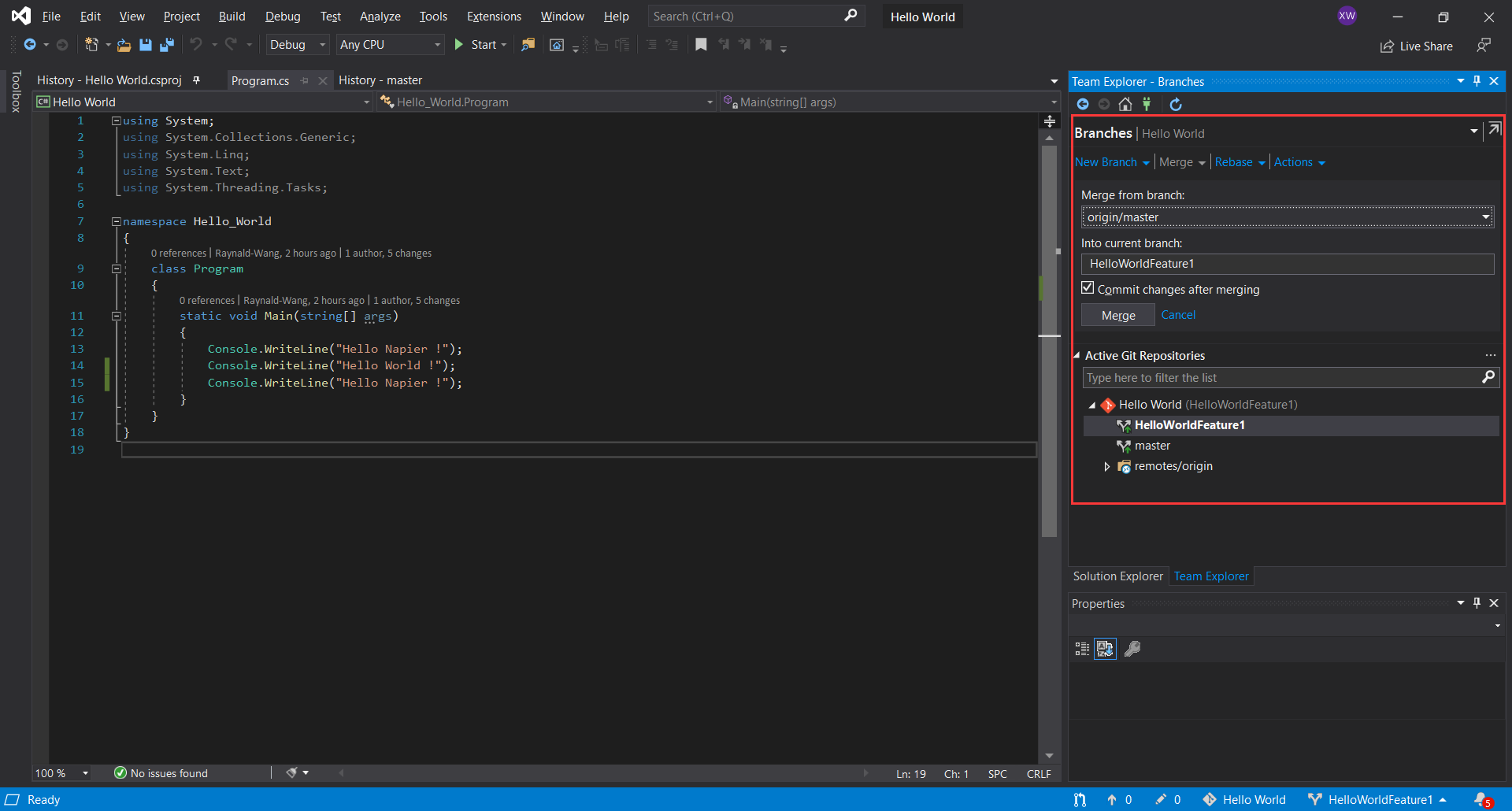


The following is a comparison in GitHub after the branch is uploaded.





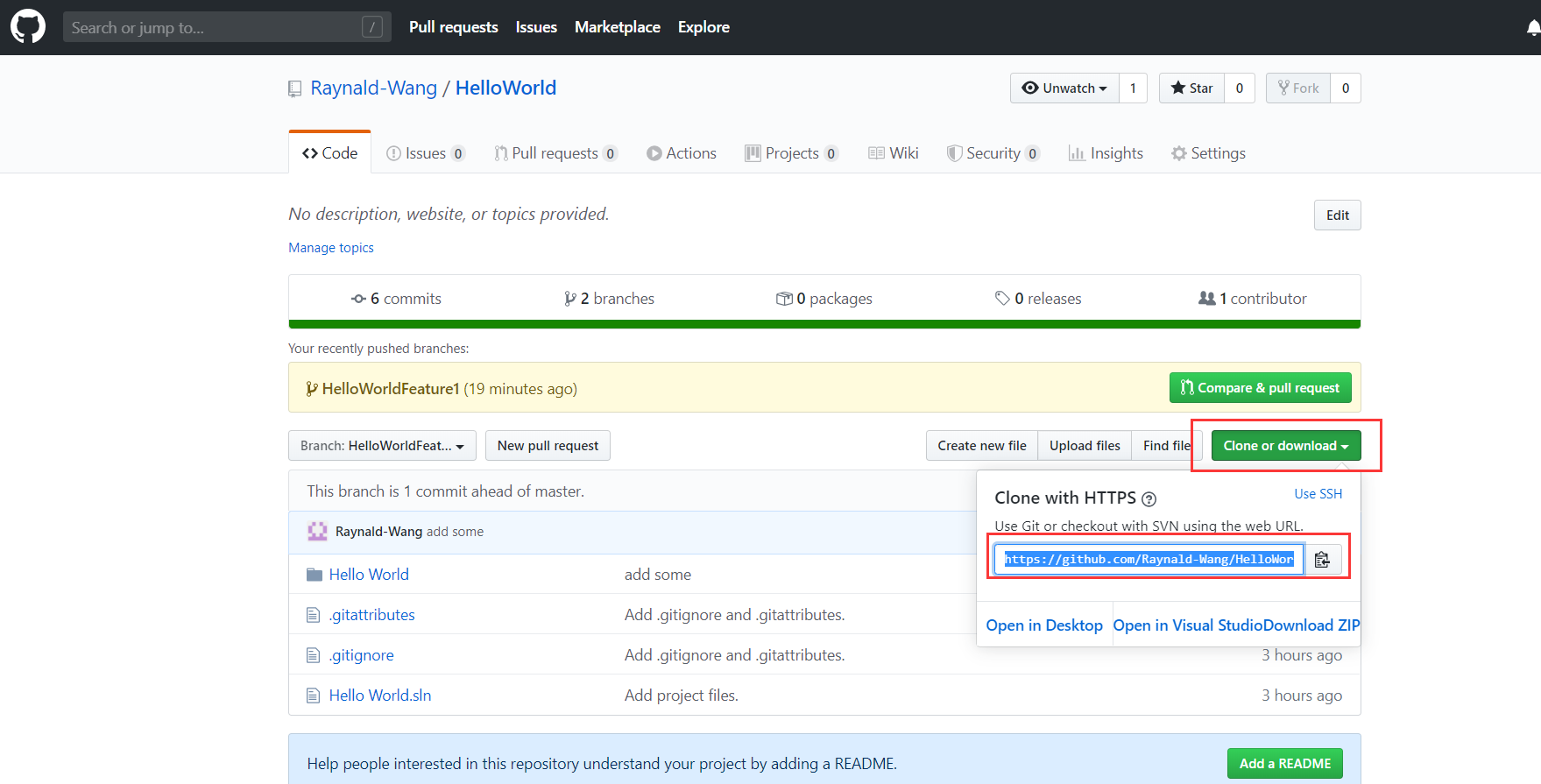
To merge changes from one source branch into a target branch, select the source branch in Source Control Explorer, right-click and select **Branches** -> **Merge** and the Merge wizard will show the form as shown below.

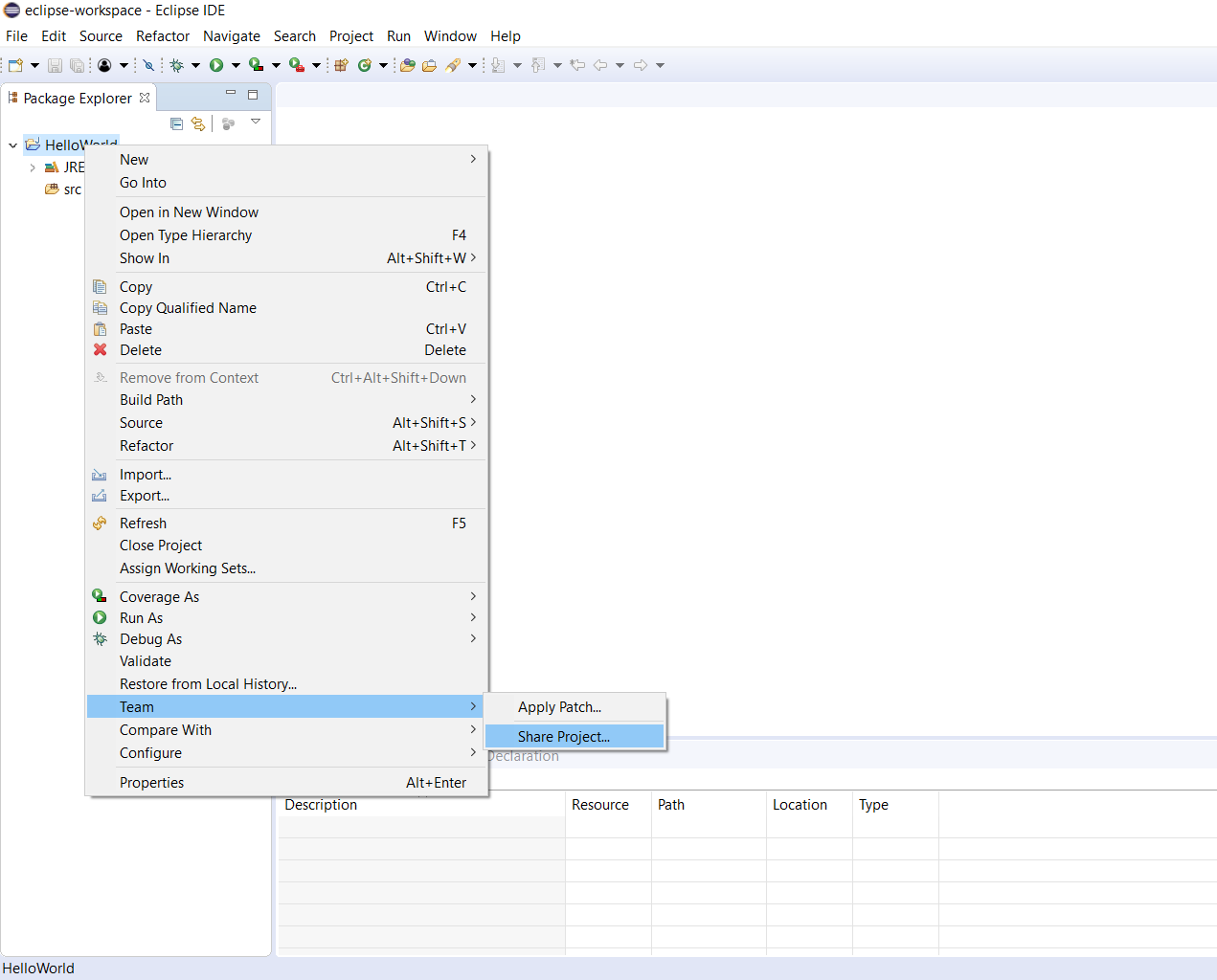


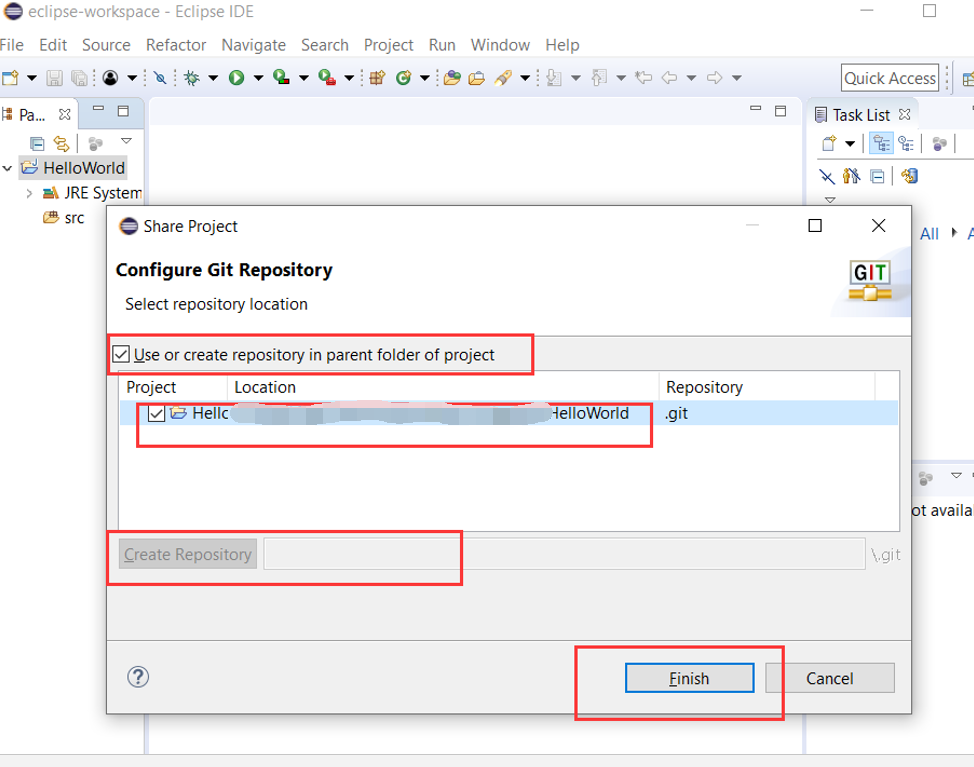
### ***2.2 Working with GitHub in Eclipse***

#### Step 1

First, create a new project with the same name as the GitHub repository. Because GitHub has links to directly copy the project code, just integrate the newly created project with the link.







#### Step 2

Now, you can try out the same activities that you have accomplished in the previous exercise. The workflow is quite similar to Visual Studio.