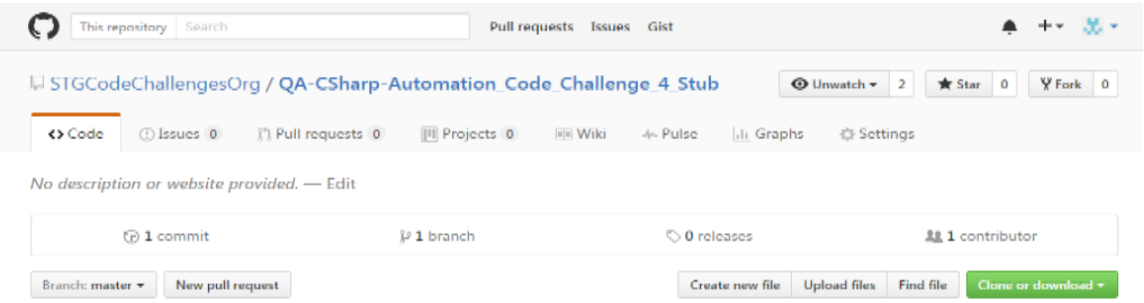
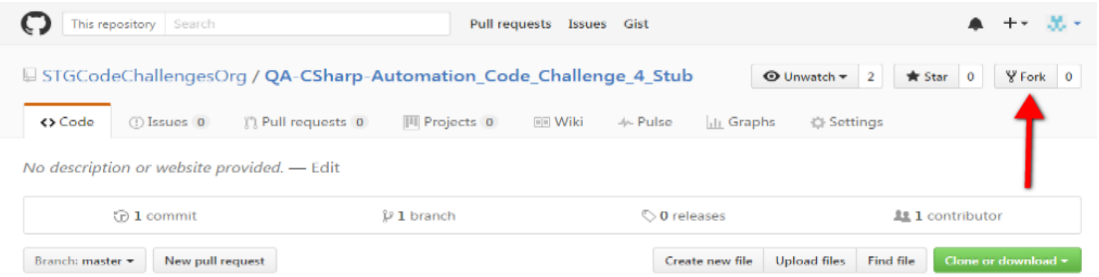
Git Procedure – Code Challenges

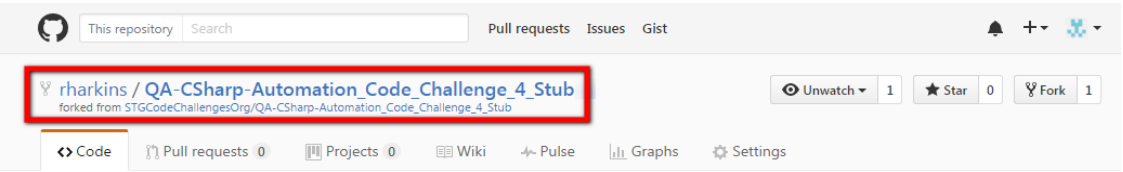
1. Go to the STGCodeChallengesOrg QA-CSharp-Automation_Code_Challenge_4_Stub GitHub repository - https://github.com/STGCodeChallengesOrg/QA-CSharp-Automation_Code_Challenge_4_Stub



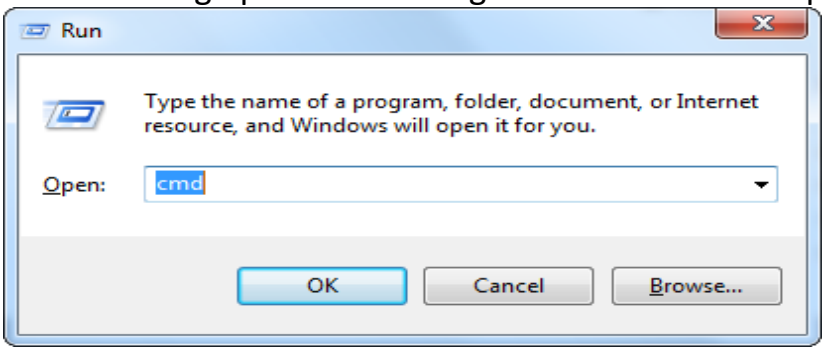
2. Click the Fork button – this creates a clone of the project on the remote server with you as the owner



3. Note that a new repository has been created under your personal GitHub account with a reference to the forked from repository

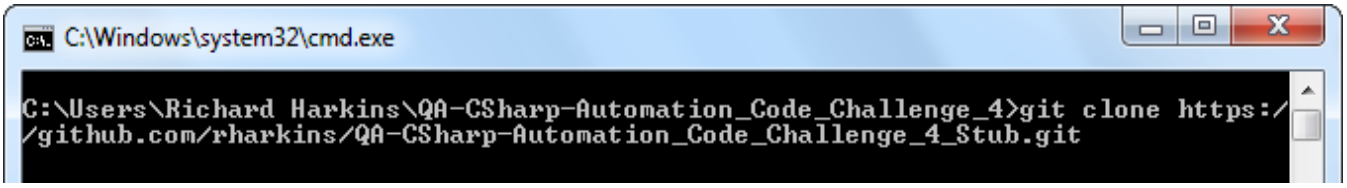


4. Start a terminal window by pressing the Window key and the R key on the keyboard. This will bring up the Run dialog box. Enter cmd and press the Enter key.

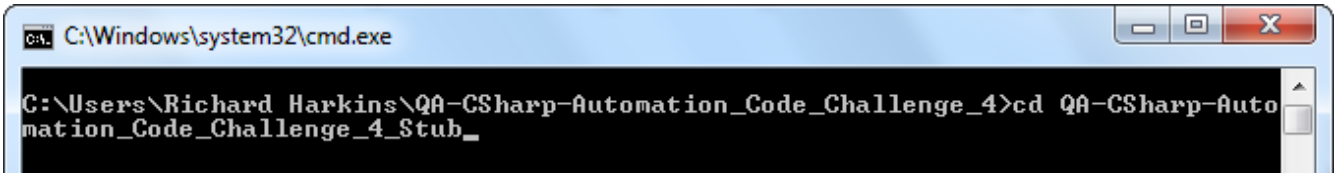


5. Change directory to the location of your local working directory – e.g. cd <local working directory path>

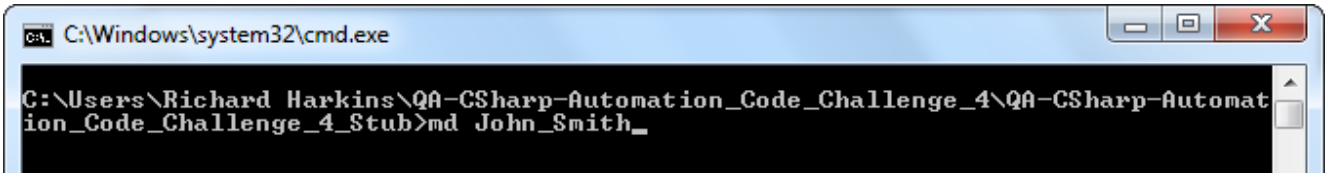
6. Clone the remote git repository to your local working directory – e.g. git clone <URL to remote git repository>



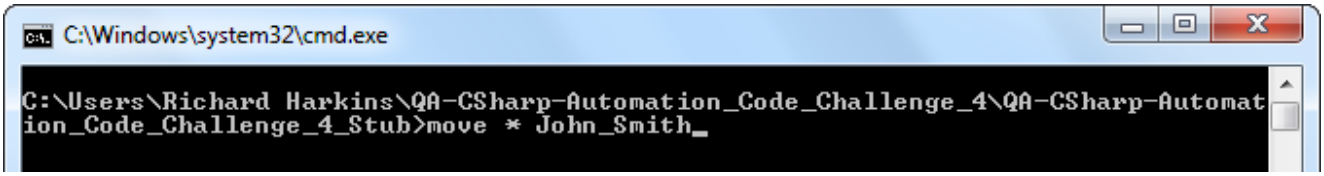
7. Change directory to the stub directory that was just cloned – e.g. cd QA-CSharp-Automation_Code_Challenge_4_Stub



8. Create a directory under the QA-CSharp-Automation_Code_Challenge_4_Stub directory named your Firstname and Lastname – e.g. md John_Smith



9. Move the files from your local working directory into the directory you just created – e.g. move * John_Smith

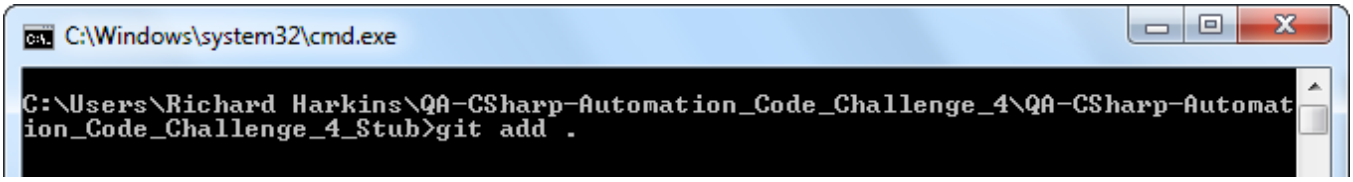


10. Using your IDE of choice, code your solution starting from the QA-CSharp-Automation_Code_Challenge_4_Stub directory

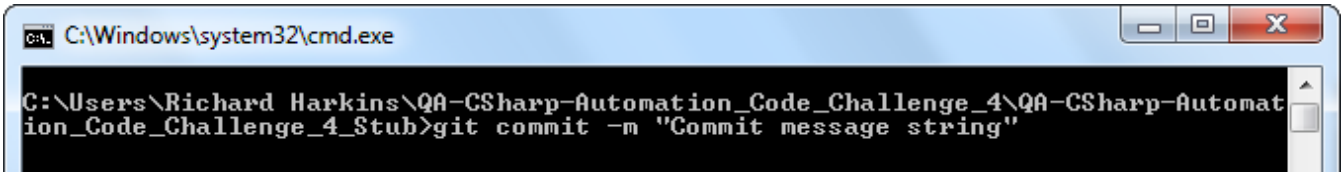
    The following table contains links for setup and configuration of git in several CSharp IDEs:

Language	IDE	Tutorial Website
CSharp	Visual Studio	https://www.visualstudio.com/en-us/docs/git/gitquickstart
CSharp	MonoDevelop	http://yasar.senturk.name.tr/blog/2014/03/29/how-to-use-github-with-monodevelopxamarin-studio/

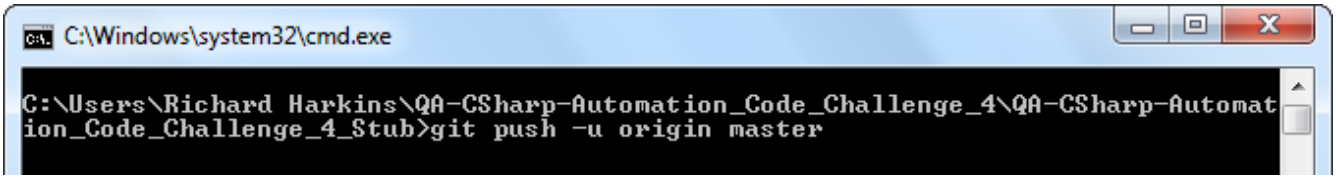
11. When you have finished coding your solution, add your FirstName_LastName directory to include in your commit (snapshot) to your local git repository – e.g. git add .



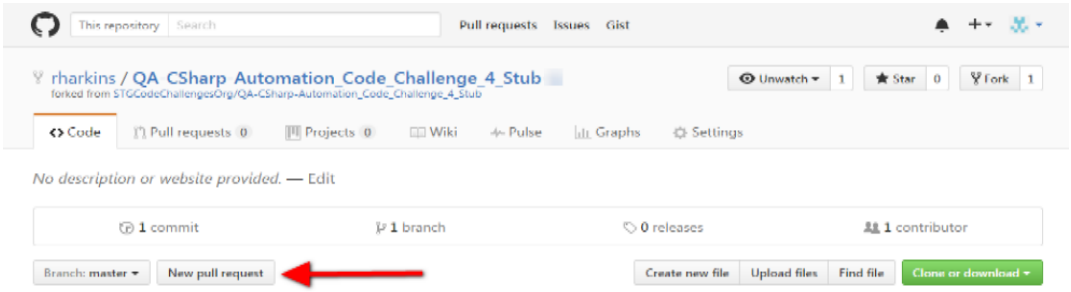
12. Commit added files/directories to local git repository, including a commit message (-m parameter) – e.g. git commit –m "commit message goes here"



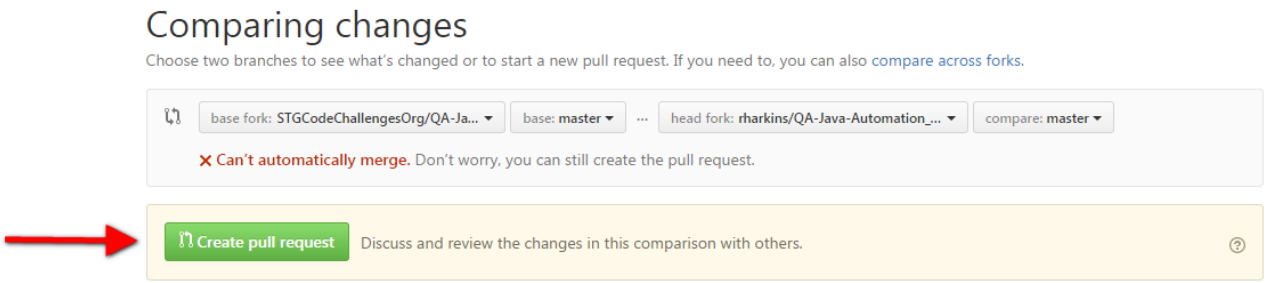
13. Push your files – e.g. push –u origin master



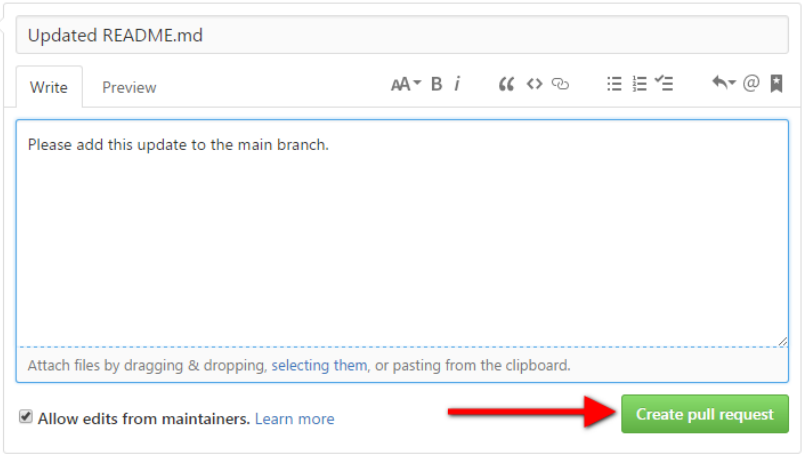
14. Click the New Pull Request button on GitHub to start the creation of a Pull request



15. Click the Create pull request button on GitHub to send a pull request to the project owner



16. Add text to the pull request message and click on the Create pull request button



17. The new pull request is now created and is pending approval from the remote repository owner. You can check the status of your pull request by clicking on the Pull Request tab on the remote repository's page: