APBD - Tutorial 1

pgago@pja.edu.pl

March 5, 2020

1 Task 1

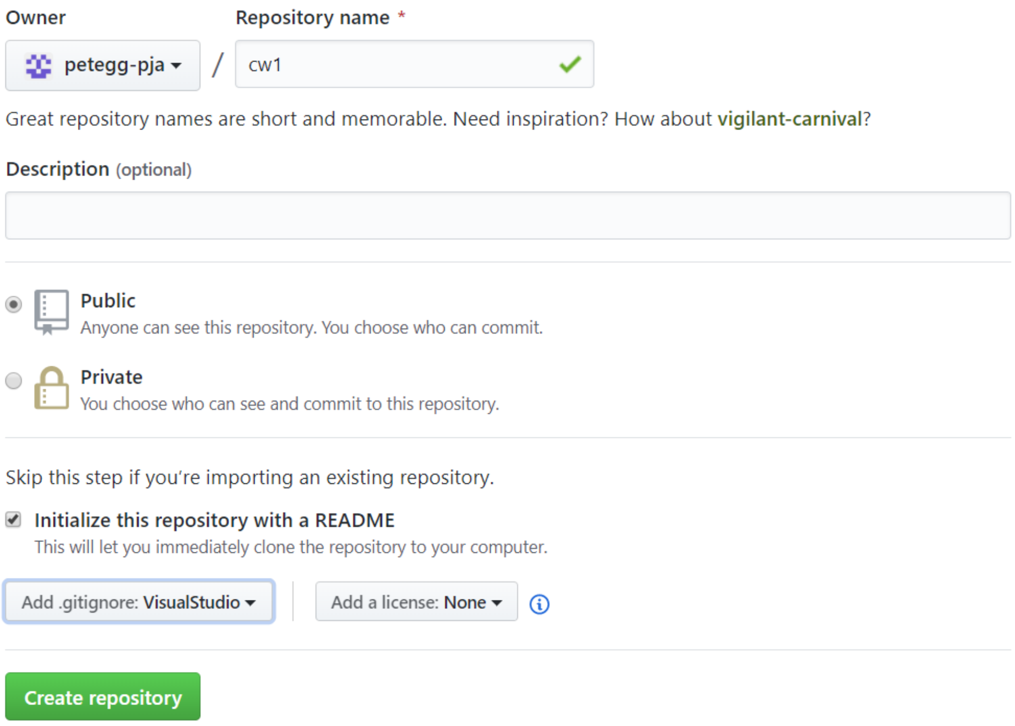
During the first task you need to prepare a new account on the site GitHub (`https://github.com/`). You will use this tool until the end of the semester.

- Create a new account using school mail. The account name should consist of an index number.
- Next, create a repository called 'tut1' for the C# code (Visual Studio). In the repository place the appropriate **.gitignore** file along with the **README.MD** file (tick the 'Initialize this repository with a README' checkbox).
- The readme file should contain name, surname, student group number and number index.
- Then use the following command to clone the repository to the desktop.

```
git clone [repository address]
```

- To do this, start the PowerShell console on Windows or Terminal on Mac/Linux. The command should be run on the Desktop.

Figure 1: Create repository screen



Owner

petegg-pja / Repository name * cw1

Great repository names are short and memorable. Need inspiration? How about vigilant-carnival?

Description (optional)

● Public
Anyone can see this repository. You choose who can commit.

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

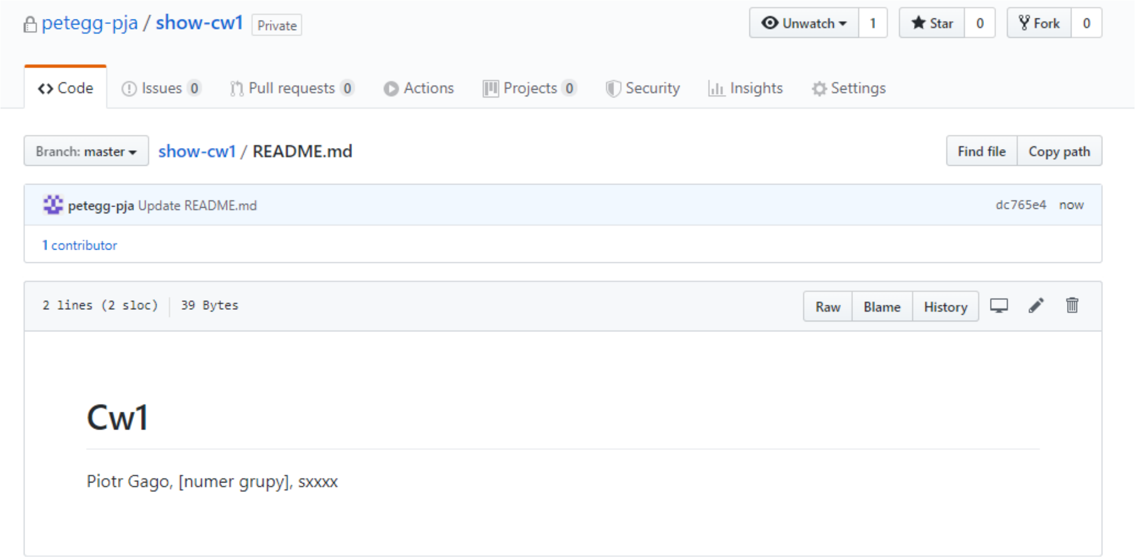
Skip this step if you're importing an existing repository.

☑ Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: VisualStudio

Add a license: None

Figure 2: Example of README.MD file



show-cw1 / README.md

petegg-pja Update README.md dc765e4 now

1 contributor

2 lines (2 sloc) | 39 Bytes

Cw1

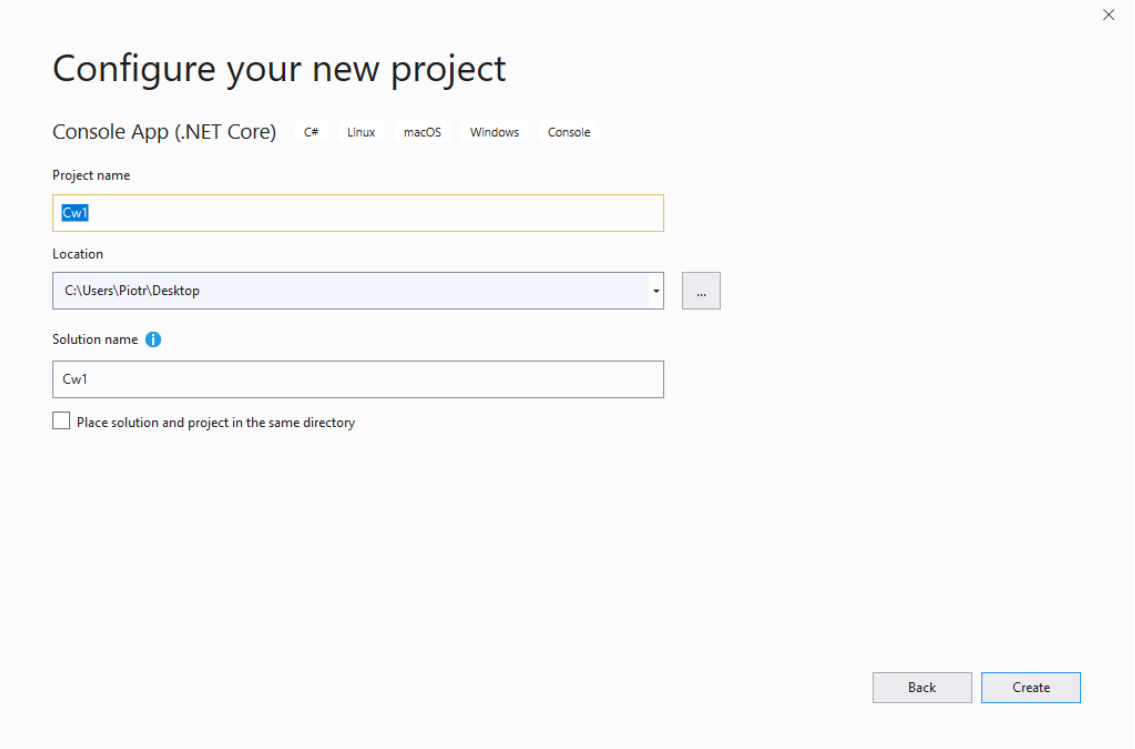
Piotr Gago, [numer grupy], sxxxx

2 Task 2

We are going to implement this exercise using Microsoft Visual Studio 2019.

- You need to create a new console application project using the .NET Core 3.1 (or 2.2).

Figure 3: New project screen in Microsoft Visual Studio 2019



Configure your new project

Console App (.NET Core) C# Linux macOS Windows Console

Project name: Cw1

Location: C:\Users\Piotr\Desktop

Solution name: Cw1

☐ Place solution and project in the same directory

- The new project should be placed in a folder that is traced by the version control system. This is the folder that you cloned in the previous task.
- After creating the project, it's worth to run it and see if "Hello World!" was displayed in the console.
- Your task is to prepare a simple "crawler" that will search the selected website and find all email addresses on it.

- The program should take a single parameter that stores the URL of the page being searched.

- Then, using the HttpClient class, we make an HTTP GET request and get the content of the page's source code.

- In the last step, we search the content of the page and write on the console all the email addresses that we found.

- Please view the project content in Solution Explorer. In addition, please run the application in debug mode and study how breakpoints work (they are extremely useful in the debugging process).

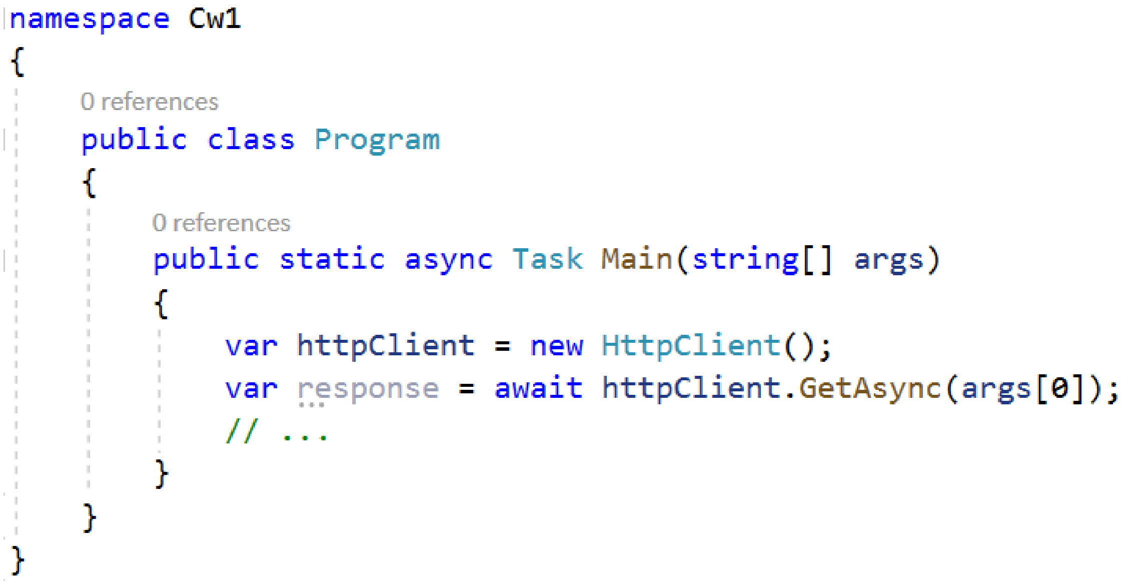
- When finished, perform "commit" using the console. To do this, open the console and navigate to the folder where you cloned your repository. Then use the following commands appropriately:

  ```
  git status
  git add .
  git status
  git commit -m "initial␣commit"
  git push
  ```

- Please make sure on your GitHub site that your commits are visible.

- During exercises, please always remember to "push" (commit) your local commit to the server. Your work will only be checked through your repository.

Figure 4: The example of the code



```
namespace Cw1
{
    public class Program
    {
        public static async Task Main(string[] args)
        {
            var httpClient = new HttpClient();
            var response = await httpClient.GetAsync(args[0]);
            // ...
        }
    }
}
```

3 Task 3

In this task, please check the contents of the compiled application on the disk. Please review the contents of the **bin** folder of the application written in Task 2. Then, with the help of the console, start the application and pass the parameter in the form of a URL.

4 Task 4

Imagine that we're suddenly got additional requirements related to the application written in Task 2.

- In the case when parameter 1 has not been passed, we should throw the `ArgumentNullException`.
- When the parameter is `null`. we should throw the `ArgumentException`.
- We should properly release the resources (using the `Dispose()` method) associated with the use of the HttpClient class
- In the case when an error occurred while downloading - we display the information "Error while downloading the page".
- In the case when no email addresses were found - we display the information "No email addresses found".
- When we found addresses - we display them on the console. We would like to display only unique email addresses.

5 Task 5

Finally, the code from Task 4 is placed on a separate branch in our repository. At this point, we will no longer use the console to work with the Git tool. We will use the "Team Explorer" window. Git is a console tool, but almost every IDE has a built-in graphical Git client. However, it is always worth remembering that graphics tools usually allow you to use only a small percentage of all the commands that Git offers.

- Create a new branch called "additional-requirements".
- Place your code as "commit" on the "additional-requirements" branch.
- Perform the "push" command and place the branch on the server.
- After completing the task, please check on Github.com if you see the newly added branch in the "Branches" tab.