

個人レベルで Github を使う

Seiichi Nukayama

2021-07-09

目次

1	Git for Windows	1
1.1	インストール	1
2	Eclipse で Github を使う	1
2.1	ワークスペースを Git で管理する	1

1 Git for Windows

1.1 インストール

ここ → <https://gitforwindows.org/> からダウンロードする。

Git-2.32.0.2-64-bit.exe が **ダウンロード** フォルダにダウンロードされる。^{*1}

Git-2.32.0.2-64-bit.exe をダブルクリックすることでインストールが始まる。

2 Eclipse で Github を使う

2.1 ワークスペースを Git で管理する

2.1.1 ワークスペース名の決定

C:ドライブの直下にワークスペースを作成し、それを Git で管理することにする。

ワークスペース名: workspace_git

2.1.2 Github でリポジトリを作成する

Github にサインインして、新しいリポジトリを作成する。

NEW をクリックして始める。

Create a new repository の画面が開く。

Repository name

まず、リポジトリ名を入力する。今回は workspace_git とした。

Description には、このリポジトリの簡単な説明を入れておく。今回は、「git 用ワークスペース」としておく。

Public / Private

このリポジトリを公開するかどうかを設定する。今回は Private とする。

Add a README file

README.md というファイルを作成するかどうかだが、作成しておいたほうが何かと便利。

Add .gitignore

.gitignore というファイルを作成するかどうかだが、これも作成しておいたほうが便利。

これにチェックを入れると、入力欄が表示される。ここでは Java を選択しておく。

Choose a license

どういうライセンスにするかを選択できる。今回は Private すなわち非公開なので、選択しなくてもかまわない。

ちなみに一番ゆるいライセンスは、MIT ライセンス。

Create repository ボタンをクリックすると、リポジトリが作成される。

新しくできたリポジトリの画面が表示される。Code ボタンをクリックする。

https://github.com/xxxxxx/workspace_git.git が表示されるので、コピーしておく。(コピーボタンがあるの

^{*1} 32bit 版が必要な場合は、ここ から Git-2.32.0.2-32-bit.exe をダウンロードできる。

で、それをクリック)

2.1.3 クローンする

今回は、C:ドライブ直下に Git 用ワークスペースを作成したいので、C:ドライブで コマンドプロンプトを開く。

C:> と表示されるので、git clone と入力してから、Ctrl-v として、先ほどコピーしておいたものを貼り付ける。

```
C:> git clone https://github.com/xxxxxx/workspace_git.git
```

これで、C:ドライブに workspace_git というフォルダが作成されている。

2.1.4 Eclipse でそのワークスペースで作業する

Eclipse を起動して、その今作成したワークスペースを指定する。

そして、そこで何らかのプロジェクトを作成し、コードを書く。

今回は、sample という Java プロジェクトを作成し、src に Main.java というクラスを作成して、以下のコードを書いた。

Listing 1 Main.java

```
1 public class Main {  
2     public static void main(String[] args) {  
3         System.out.println("テスト");  
4     }  
5 }
```

2.1.5 ワークスペースの変更を Github に反映させる

C:\workspace_git でコマンドプロンプトを開く。

```
C:\workspace_git> git status
```

とする。

```
C:\workspace_git>git status
```

On branch main

Your branch is up to date with 'origin/main'.

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: .gitignore

Untracked files:

(use "git add <file>..." to include in what will be committed)

sample/

no changes added to commit (use "git add" and/or "git commit -a")

git status -u

とすると、フォルダのなかのファイルも表示してくれる。

C:\workspace_git>git status -u

On branch main

Your branch is up to date with 'origin/main'.

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: .gitignore

Untracked files:

(use "git add <file>..." to include in what will be committed)

sample/.classpath

sample/.project

sample/.settings/org.eclipse.jdt.core.prefs

sample/src/sample/Main.java

no changes added to commit (use "git add" and/or "git commit -a")