おおまかな流れ

javadocを作成する

javadocのエクスポート

javadocのHTMLをWebで表示させる

javadocをgitbashでcommit pushさせる

githubでファイルのWeb表示許可する

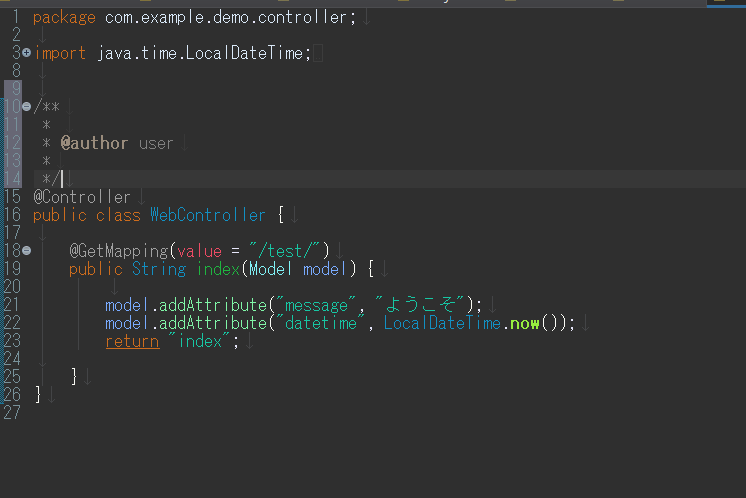
github内のJavadocをURLでWeb表示させる

Javadocの作成

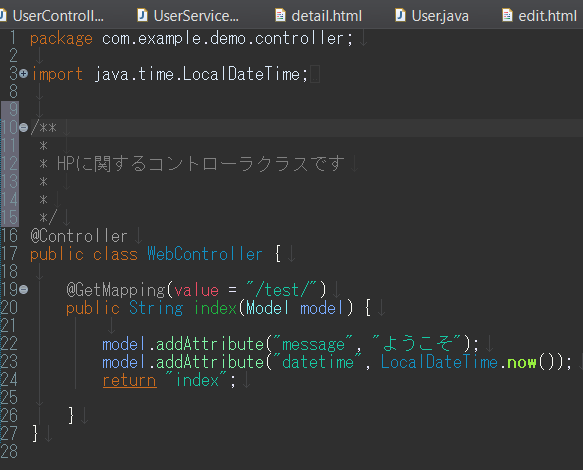
生成させたい場所に、/\*\* と入力するとJavadocが生成される

この時点でクラスに置きたいのかメソッドに置きたいのかを認識している

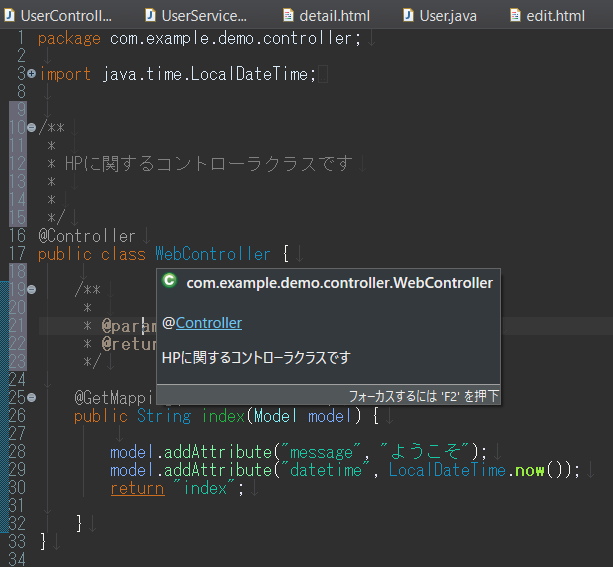
この場合はクラスの説明をしている



コメントアウトのような使い方をする



メソッドの説明をJavadocを使って記述する

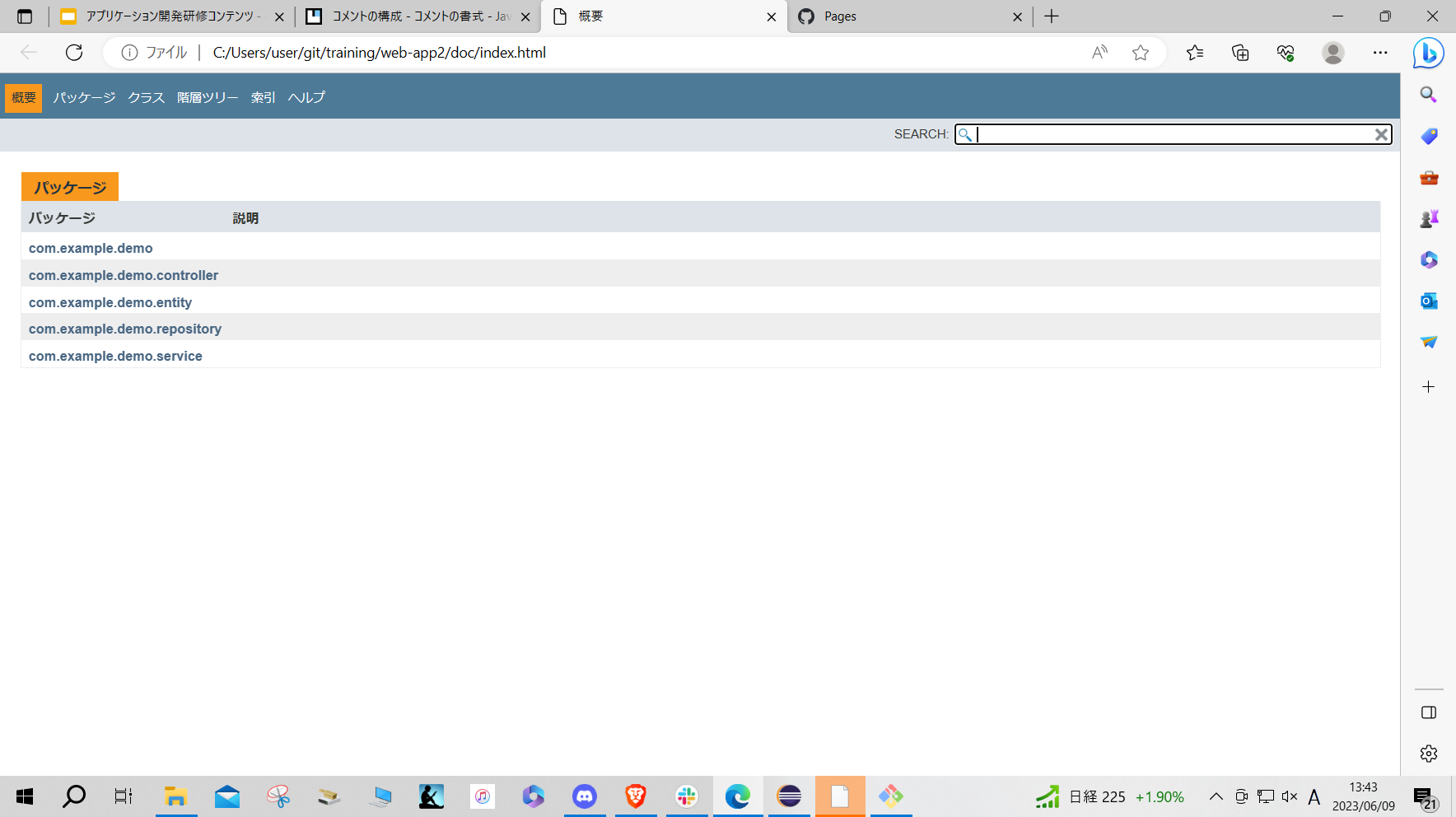


@ param が説明を意味している

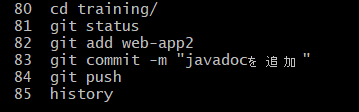
@ return が戻り値の説明を意味している

次にJavadocをWebでHTML表示させたいのでまずはJavadocをエクスポートする

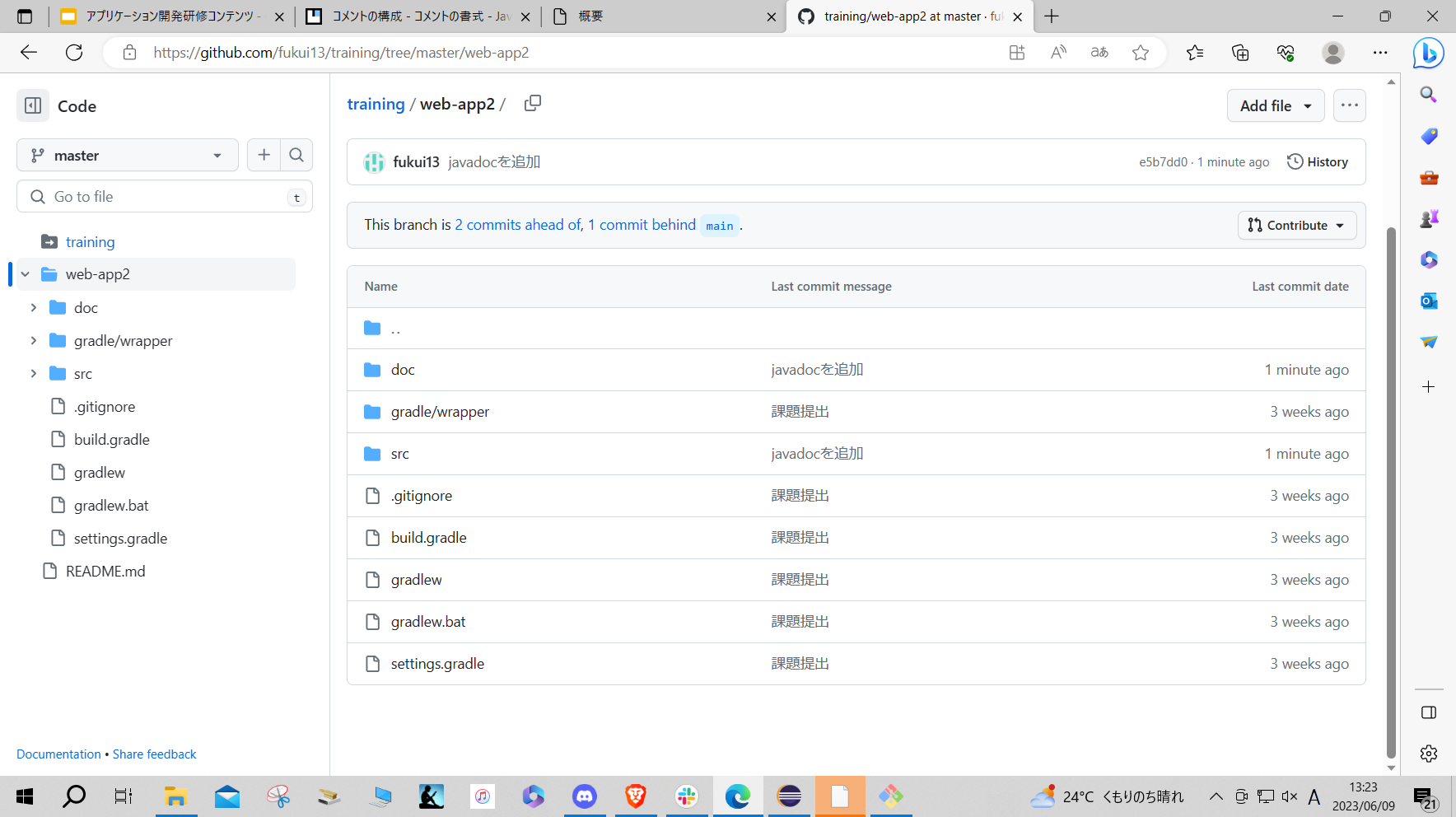
エクスポートした後、Web表示させる



エクスポートしたJavadocファイルをGtibashでcommit pushする

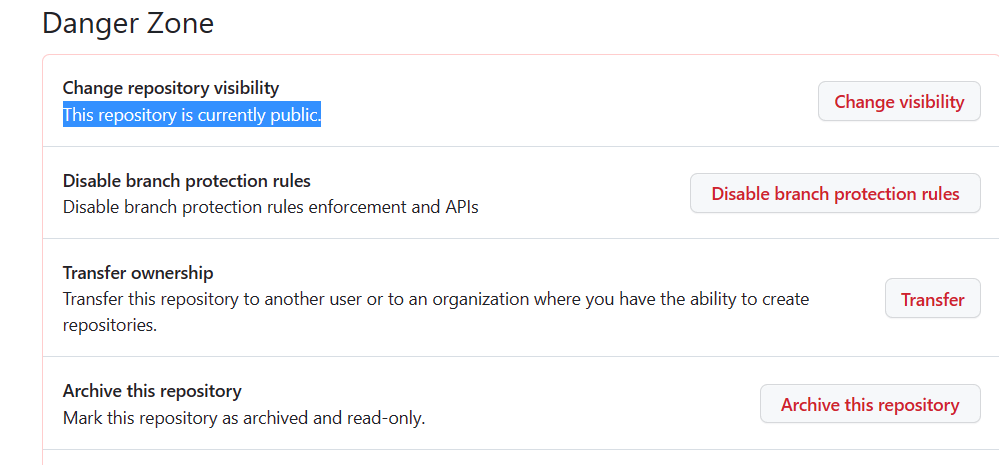


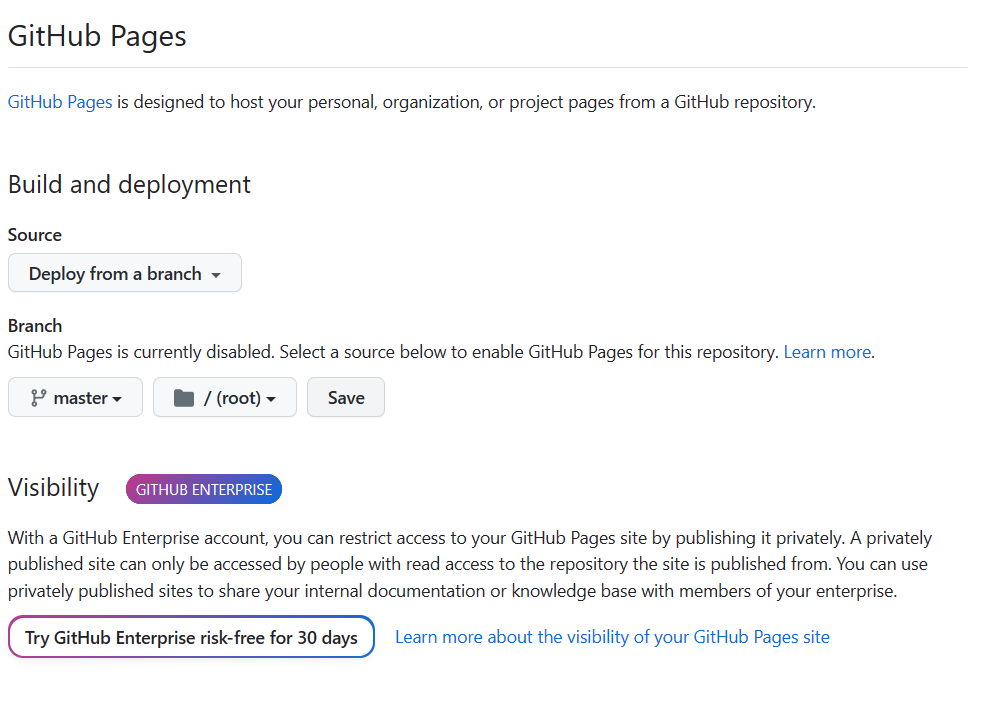
githubにプッシュできているか確認



次にGithub内でWeb表示させたいので

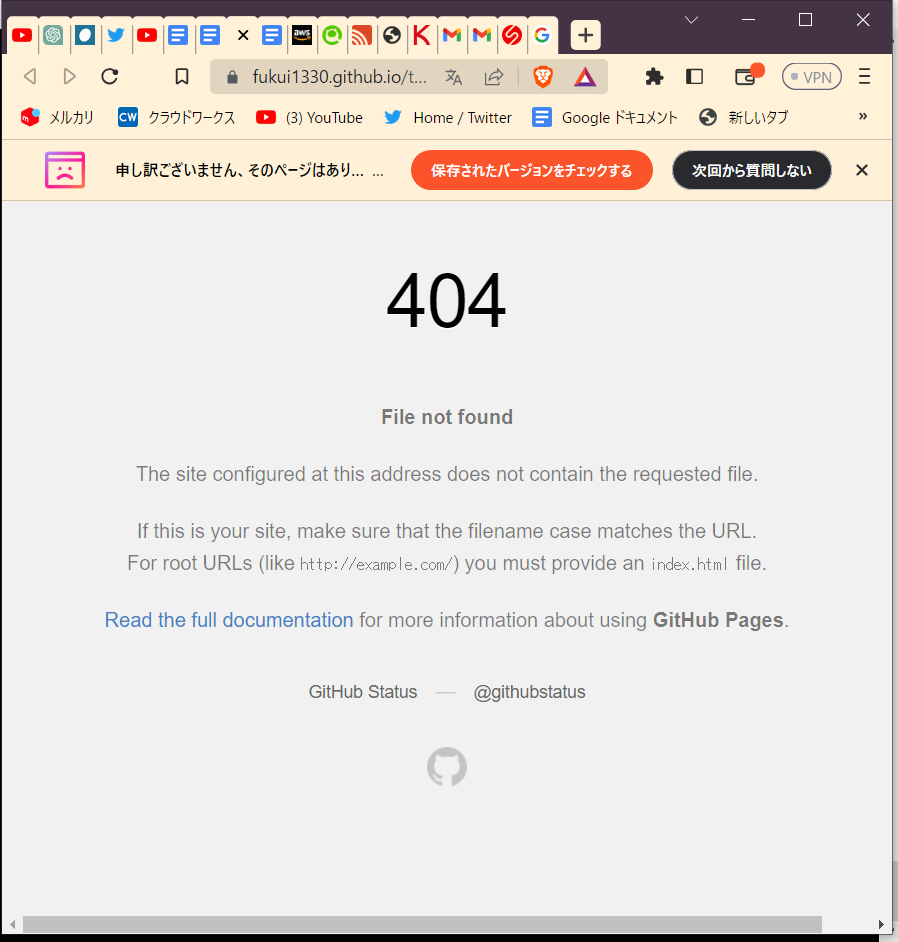
GithubでファイルのWeb表示許可するため、諸々設定していく





<https://fukui1330.github.io/training/web-app2/doc/index.html>

表示させてみたが、エラーとなってしまった



なので原因を特定していく

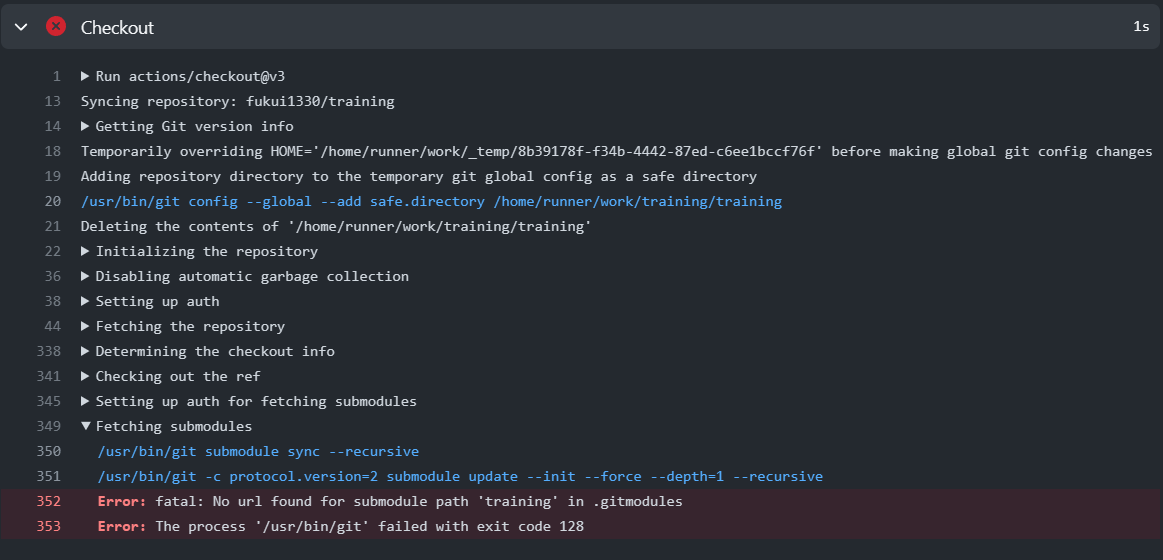
色々調べて見た結果

・githubに登録したユーザーネームがURLと相性が悪い場合があり

ユーザーネームを変更してみる　（検証済み、改善なし

・もう一度pushしてみる　（検証済み、改善無し

・ページビルドを確認してみる（pages build and deployment



案の定、エラーが出ていたのでエラー内容を確認してみる

Error: fatal: No url found for submodule path 'training' in .gitmodules

Error: The process '/usr/bin/git' failed with exit code 128

ログを要約すると、.gitmoduleというファイル内にtrainingのパスのサブモジュールのURLが見つからないという内容でした

問題解決のプロセスを組み立てる

1 gitbashで.gitmodule というファイルが存在するか確認する、隠しファイルとなるので $ls -a

2 存在しなかった場合 -v で.gitmoduleを作成する

中身は

[submodule "training"]

path = training

url = <サブモジュールのURL>

３作成したら変更をコミットしてプッシュする

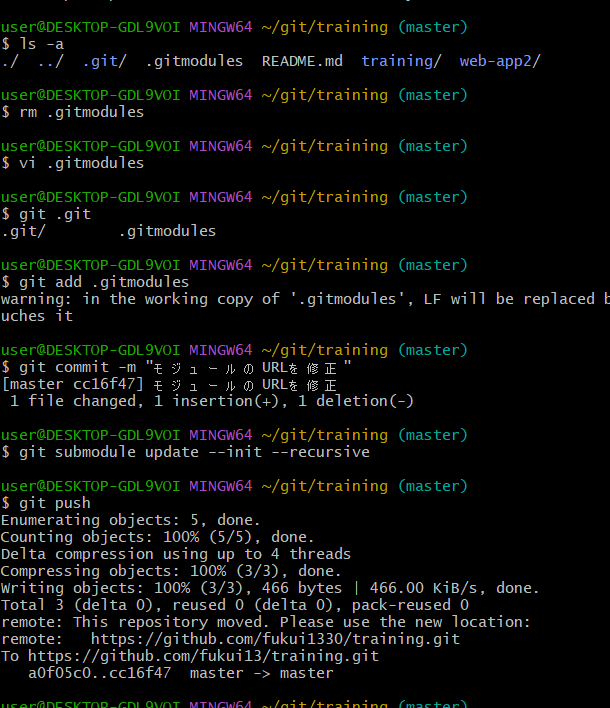
$ git add .gitmodule

$ git commit -m “サブモジュール作成”

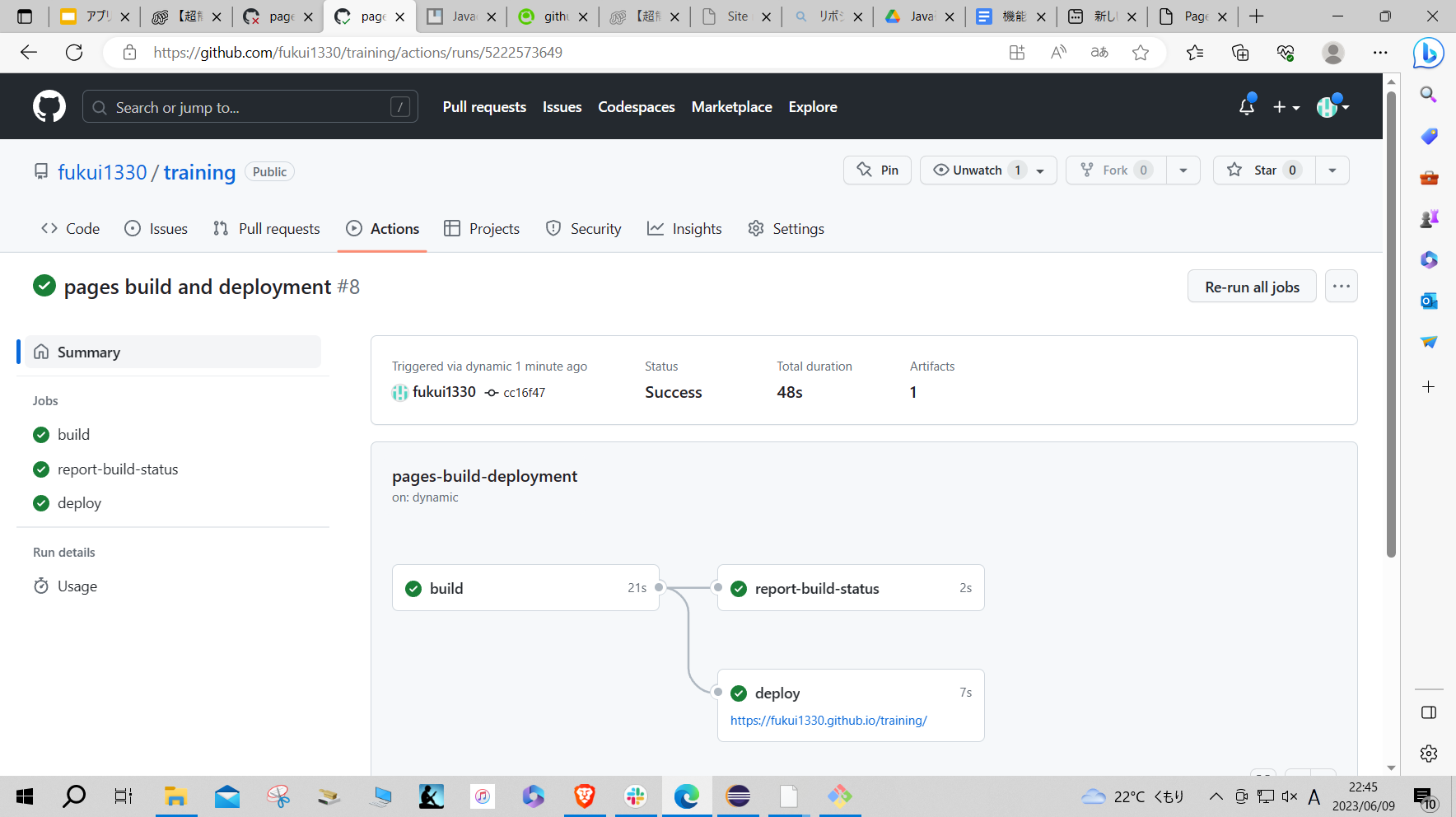
$ git push

記述した内容とは異なりますが、これは一度作成した.gitmoduleのURLが間違っていて、修正してアップデートしている流れで

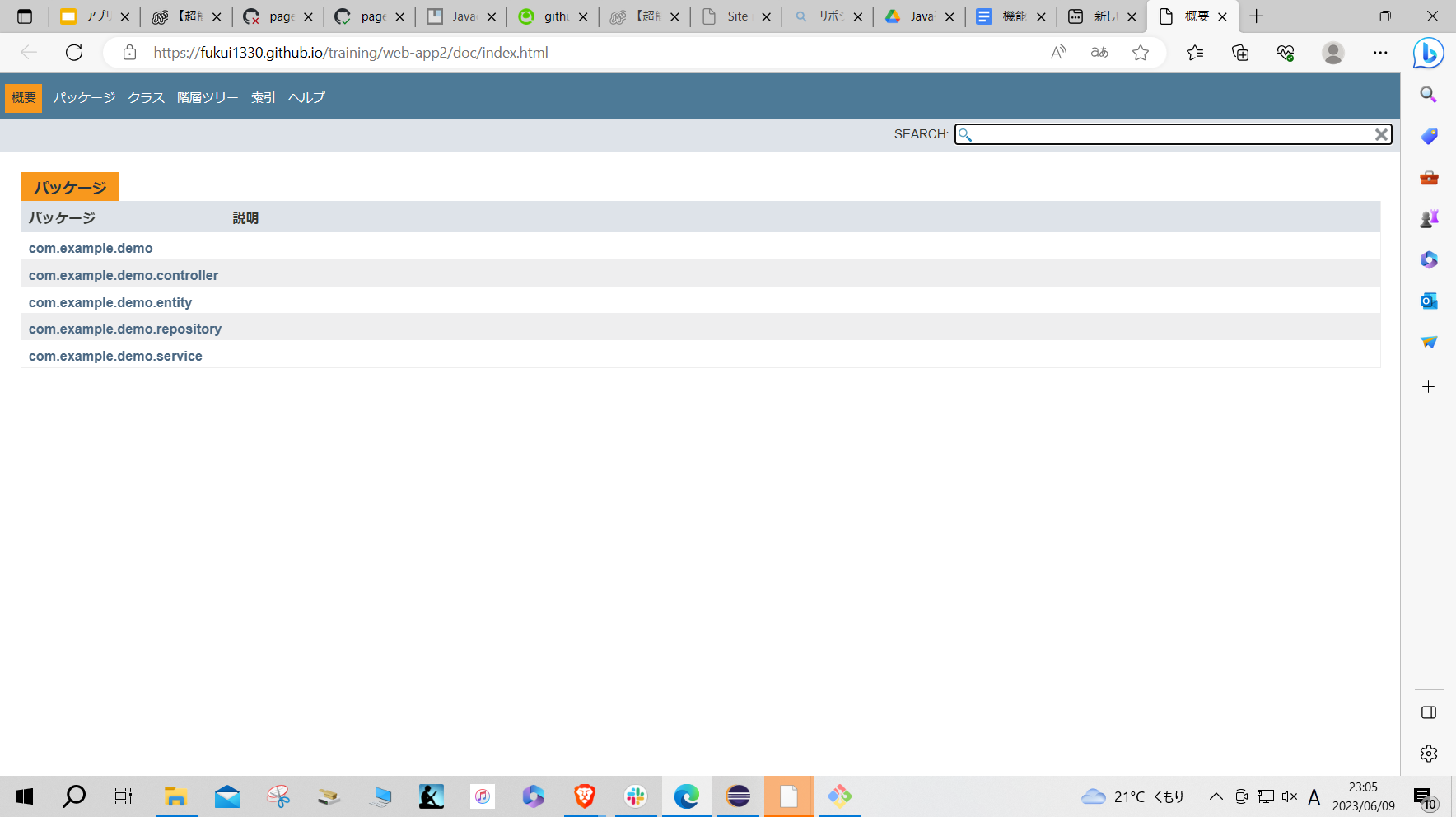
やっていることは似たようなものです



ビルドページに行きエラーが解決しているか確認する



無事、表示させる事ができました



まとめ

Javadocとは

・Javaのソースコード内に記述されるドキュメンテーションコメント形式

・プログラムのソースコードから自動的に生成されるドキュメントを作成するためのツール

・Javadocコメントはソースコード内に特定の形式で特別なコメントのようなもので、基本的にメソッドやクラスの前に記述される

・コメントにはクラスやメソッドの目的や使い方、パラメータの説明、戻り値の説明などの情報を含む（なんのクラスなのか分かりやすくコメントする

・様々なタグが存在する（author version see　などなど

.gitmoduleファイルとは

・サブモジュールを設定するためのファイル、このファイルがないとサブモジュールの情報が取得されずエラーが発生する場合がある

・ファイル内にはパスとURLが設定されている