

課題の提出手順

課題の提出には以下の作業が必要になります

Commit(コミット)

- Commitは、追加・変更したファイルをGitに登録するための操作です。

Push(プッシュ)

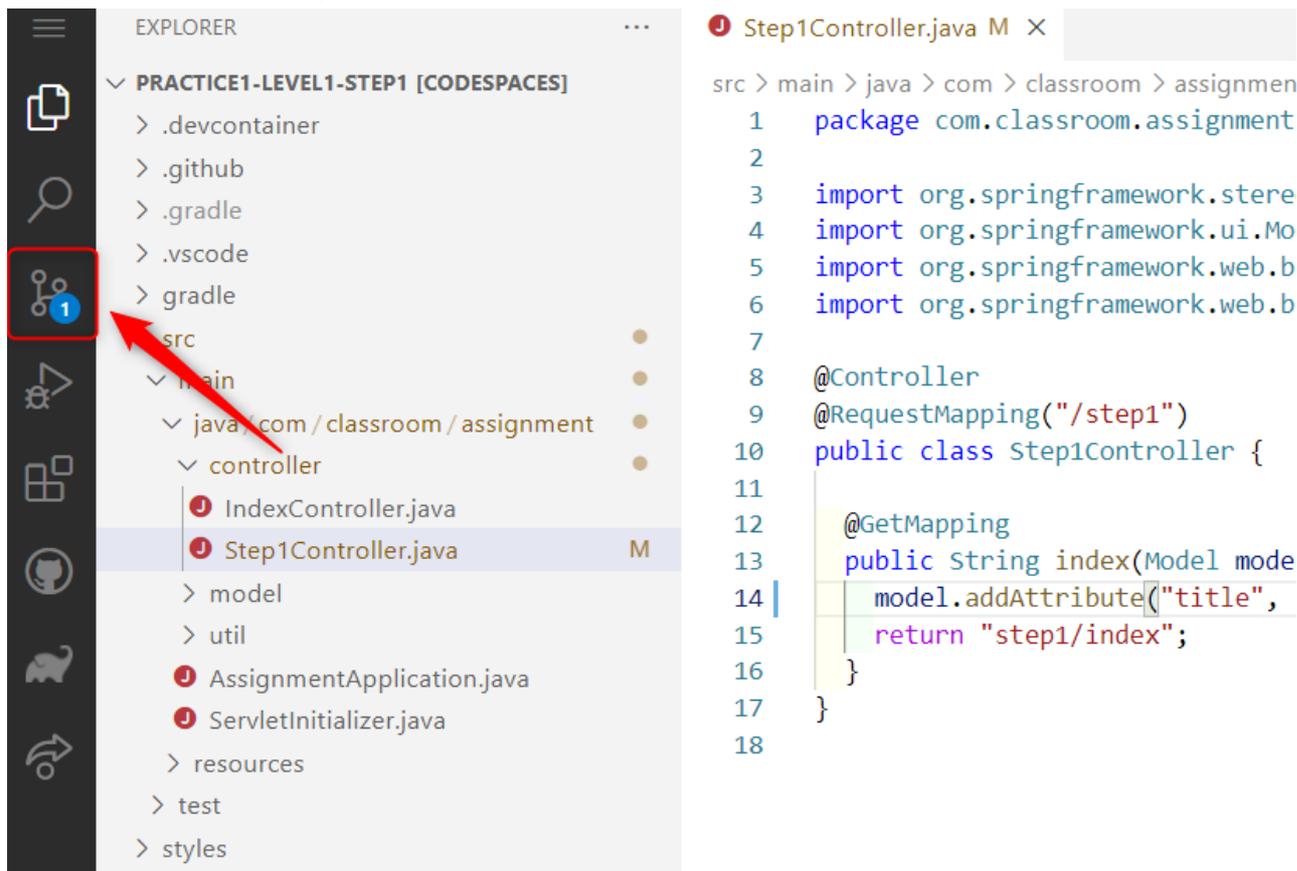
- pushは、コミットした内容をローカルレポジトリからリモートレポジトリに送信する操作です。

Pull Request(プルリクエスト)

- Pull Requestは、開発者のローカルレポジトリでの変更を他の開発者に通知する機能です。

作業手順

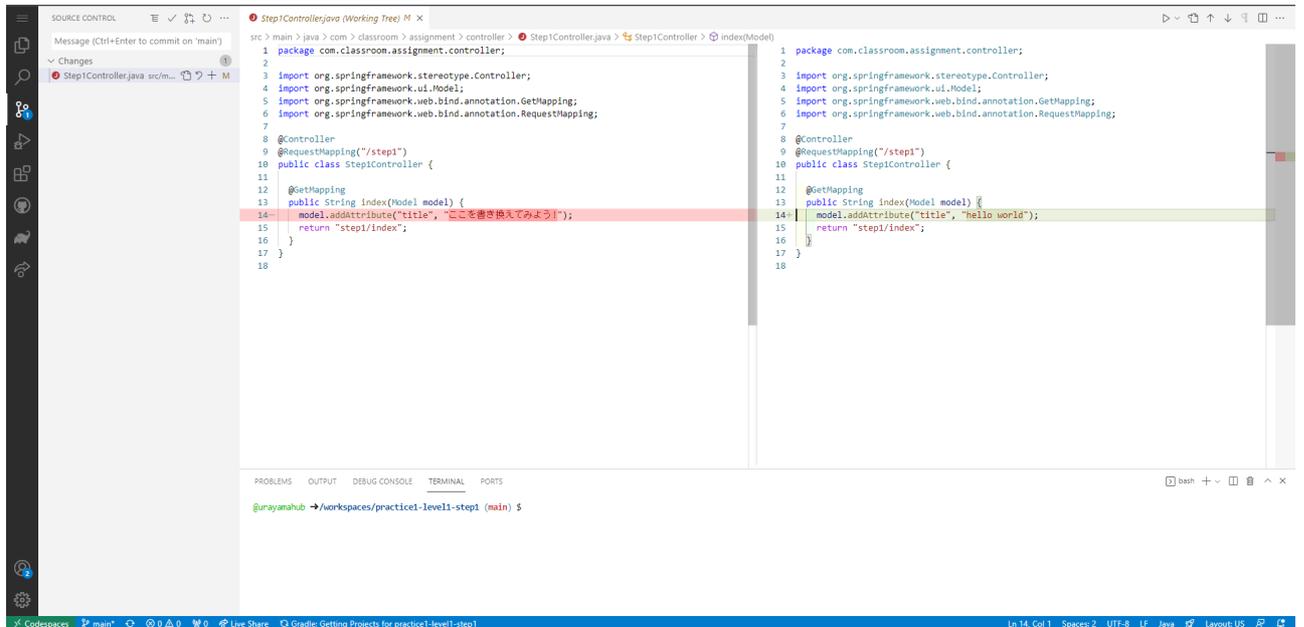
1. 木の枝のようなマークを押します。



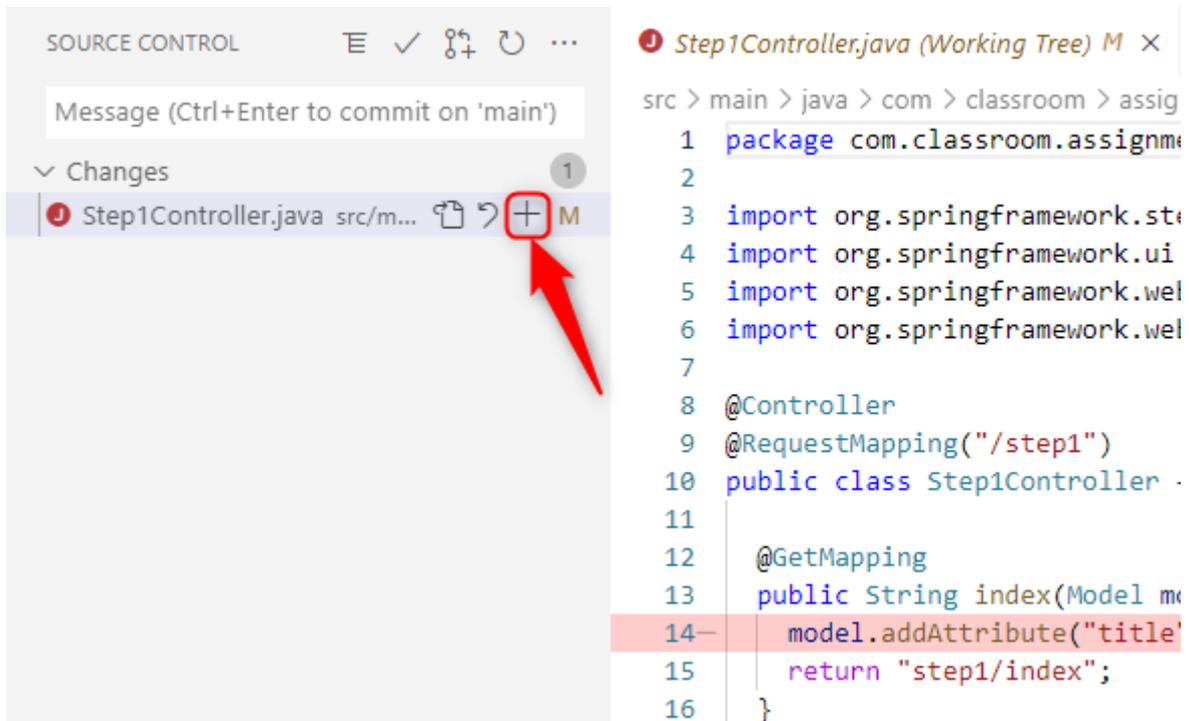
The screenshot shows the Visual Studio Code interface. On the left, the Explorer view displays a file tree for a project named 'PRACTICE1-LEVEL1-STEP1 [CODESPACES]'. The tree structure includes folders like '.devcontainer', '.github', '.gradle', '.vscode', 'gradle', 'src', 'main', and 'java/com/classroom/assignment'. Under 'assignment', there is a 'controller' folder containing 'IndexController.java' and 'Step1Controller.java'. The 'Step1Controller.java' file is selected and highlighted in blue. A red box highlights the Git icon in the left sidebar, with a red arrow pointing to it. To the right, the code editor shows the content of 'Step1Controller.java' with the following code:

```
src > main > java > com > classroom > assignment
1  package com.classroom.assignment
2
3  import org.springframework.stereotype.Controller;
4  import org.springframework.ui.Model;
5  import org.springframework.web.servlet.mvc.annotation.annotation.AnnotationMethodMapping;
6  import org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerMethod;
7
8  @Controller
9  @RequestMapping("/step1")
10 public class Step1Controller {
11
12     @GetMapping
13     public String index(Model model) {
14         model.addAttribute("title", "Step1Controller");
15         return "step1/index";
16     }
17 }
18
```

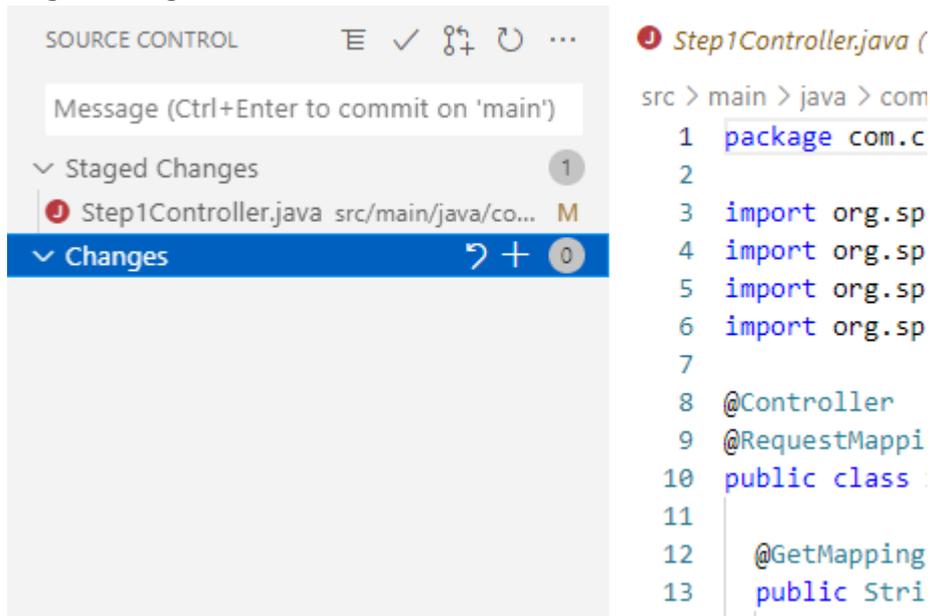
2. これまでに行ったファイルへの変更ないようが表示されています。変更内容に問題がないか今一度確認しましょう。



3. コミットするファイルを選択しましょう。コミットしたいファイルの横の+ボタンを押しましょう。



4. Staged Changesにコミットしたいファイルが含まれていることを確認しましょう。



SOURCE CONTROL

Message (Ctrl+Enter to commit on 'main')

Staged Changes 1

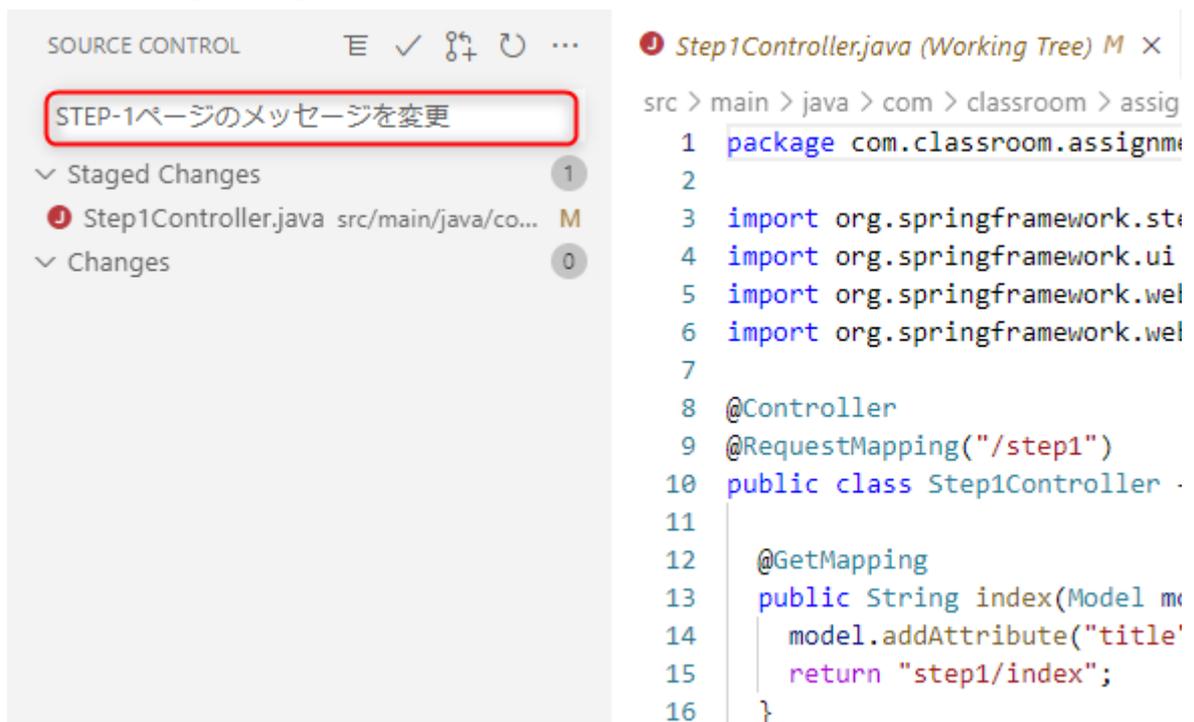
Step1Controller.java src/main/java/co... M

Changes 0

```
Step1Controller.java (
src > main > java > com
1 package com.c
2
3 import org.sp
4 import org.sp
5 import org.sp
6 import org.sp
7
8 @Controller
9 @RequestMappi
10 public class :
11
12 @GetMapping
13 public Stri
```

5. Message欄にCommit Message(コミットメッセージ)を入力しましょう。

- Commit Messageには、今回のコミットの追加・変更内容について記載します。
- 課題のテーマを記載しましょう



SOURCE CONTROL

STEP-1ページのメッセージを変更

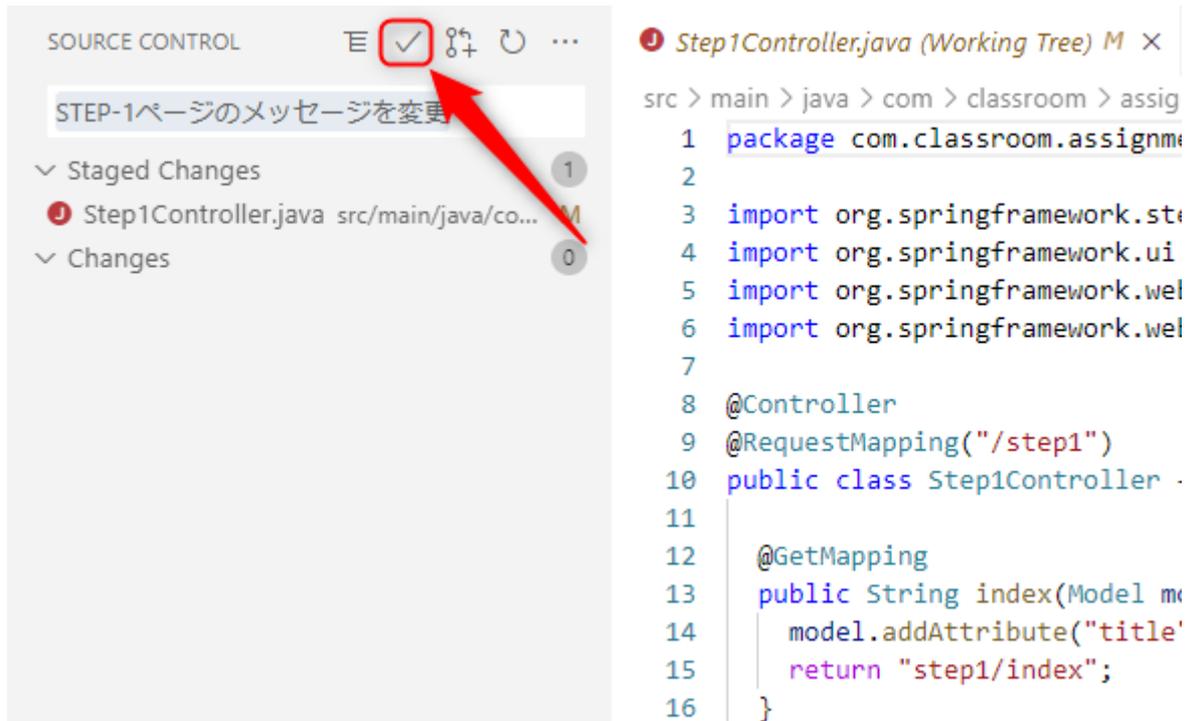
Staged Changes 1

Step1Controller.java src/main/java/co... M

Changes 0

```
Step1Controller.java (Working Tree) M X
src > main > java > com > classroom > assignm
1 package com.classroom.assignme
2
3 import org.springframework.ste
4 import org.springframework.ui
5 import org.springframework.wel
6 import org.springframework.wel
7
8 @Controller
9 @RequestMapping("/step1")
10 public class Step1Controller .
11
12 @GetMapping
13 public String index(Model mo
14     model.addAttribute("title'
15     return "step1/index";
16 }
```

6. チェックボタンを押してコミットしましょう。



SOURCE CONTROL

STEP-1ページのメッセージを変更

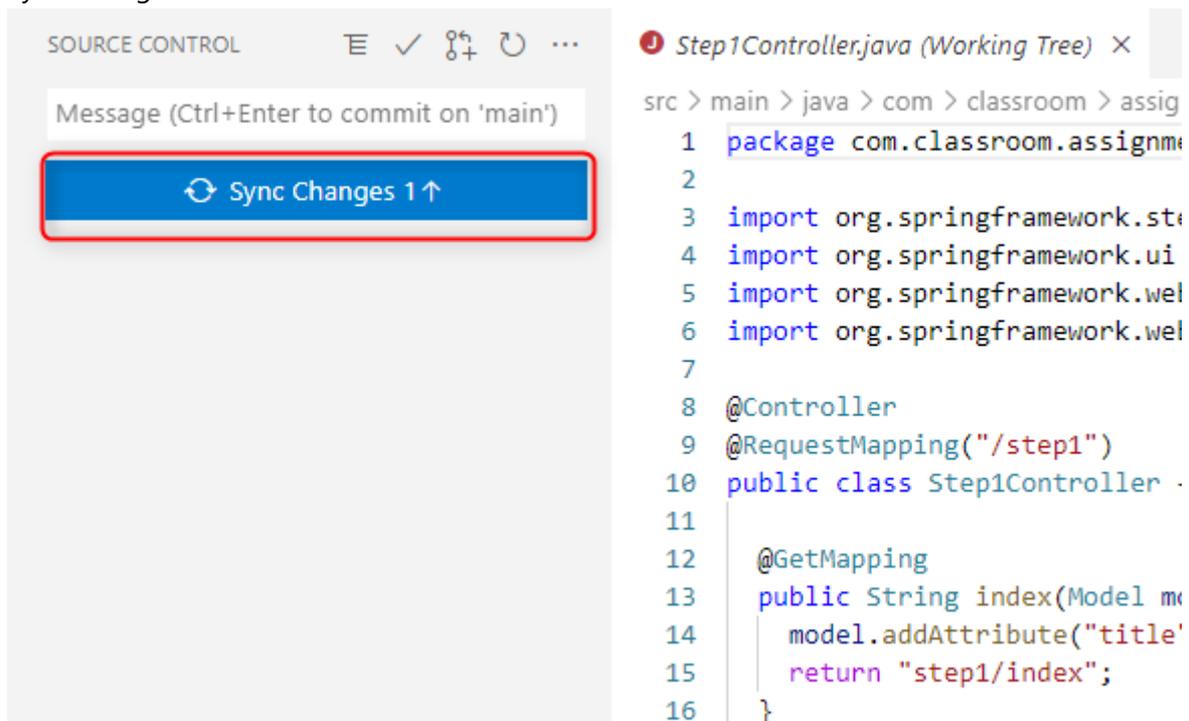
Staged Changes 1

Step1Controller.java src/main/java/co... 14

Changes 0

```
src > main > java > com > classroom > assign
1 package com.classroom.assignme
2
3 import org.springframework.ste
4 import org.springframework.ui
5 import org.springframework.wel
6 import org.springframework.wel
7
8 @Controller
9 @RequestMapping("/step1")
10 public class Step1Controller {
11
12     @GetMapping
13     public String index(Model mo
14         model.addAttribute("title'
15         return "step1/index";
16     }
```

7. Sync Changesボタンを押してコミットをプッシュしましょう。



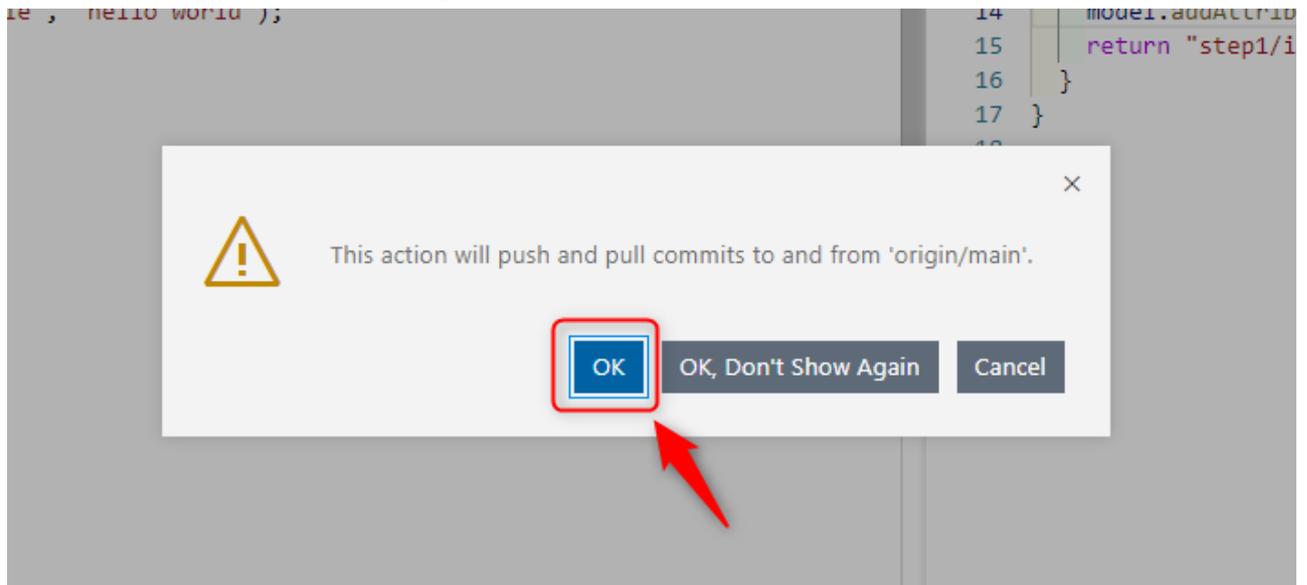
SOURCE CONTROL

Message (Ctrl+Enter to commit on 'main')

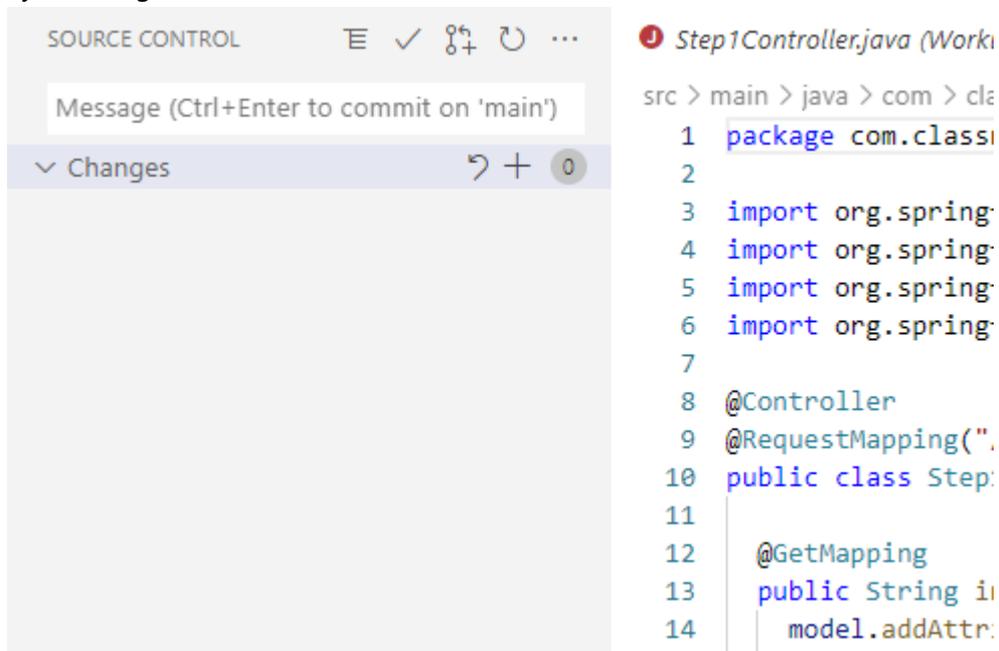
Sync Changes 1 ↑

```
src > main > java > com > classroom > assign
1 package com.classroom.assignme
2
3 import org.springframework.ste
4 import org.springframework.ui
5 import org.springframework.wel
6 import org.springframework.wel
7
8 @Controller
9 @RequestMapping("/step1")
10 public class Step1Controller {
11
12     @GetMapping
13     public String index(Model mo
14         model.addAttribute("title'
15         return "step1/index";
16     }
```

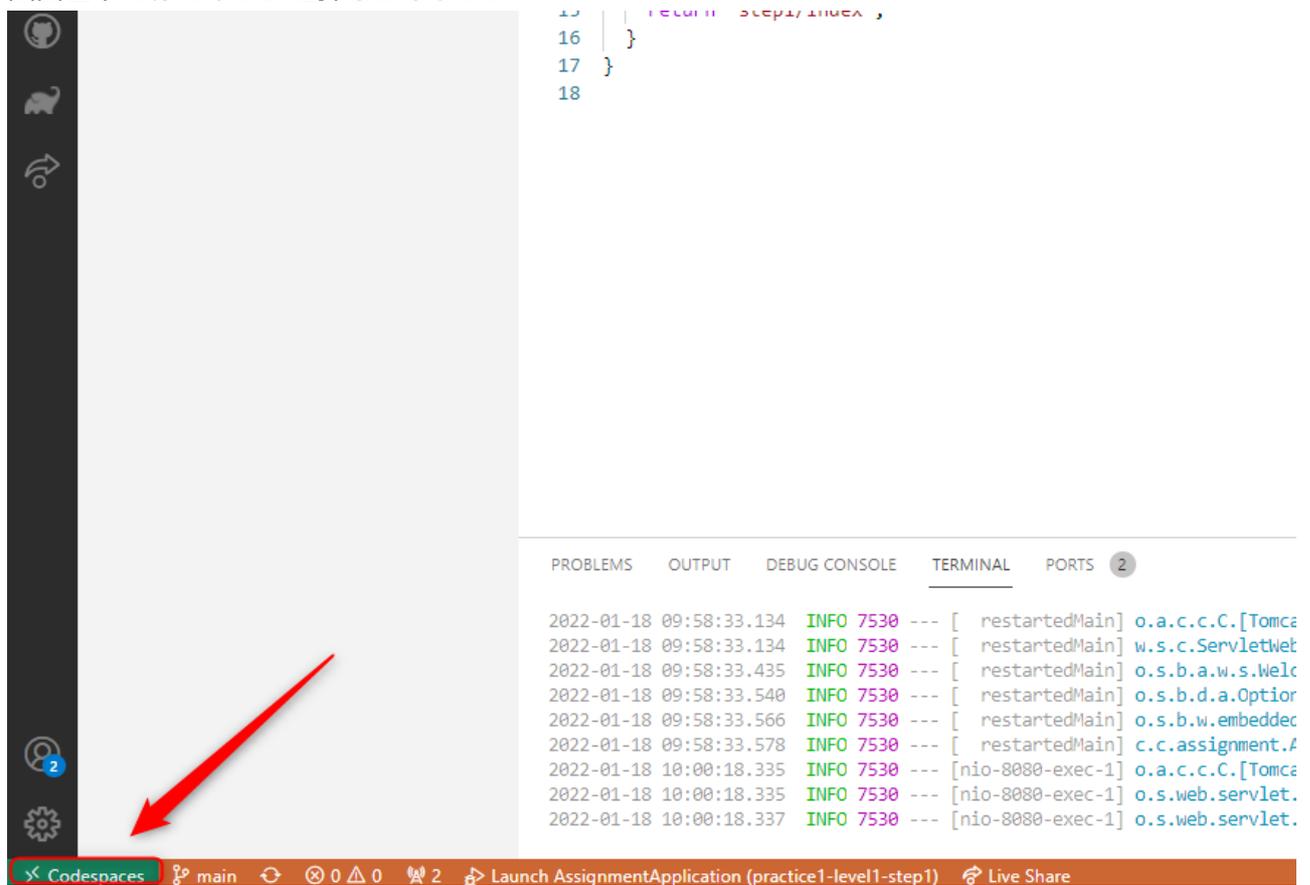
8. ダイアログが出ますので「OK」を押しましょう。



9. Sync Changesボタンが消えていることを確認してください。



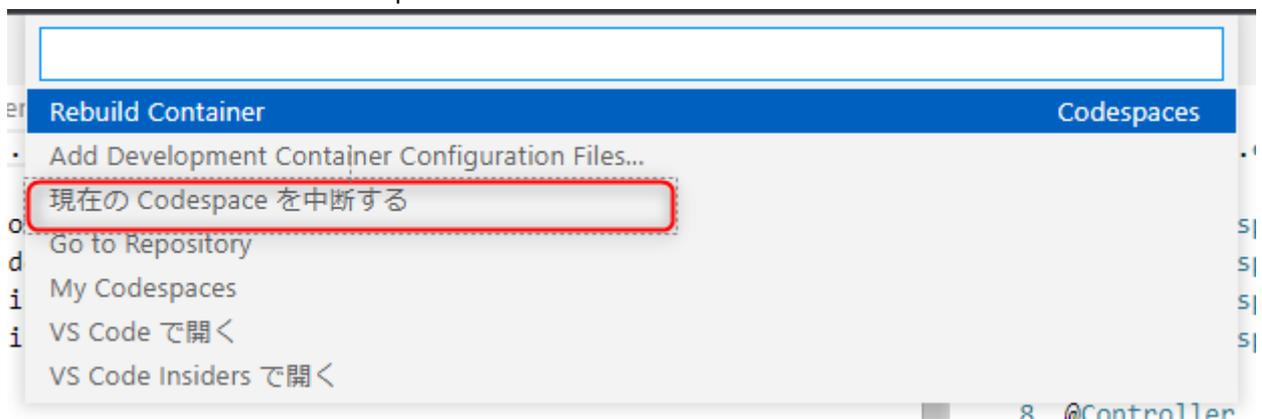
10. 画面左下の緑のボタンを押しましょう。



The screenshot shows the VS Code Codespace interface. On the left, a vertical sidebar contains several icons, with a green button at the bottom highlighted by a red arrow. The main area displays a code editor with lines 15-18 of code. Below the editor is a terminal window showing logs for a Java application. The status bar at the bottom indicates the current workspace and a 'Live Share' button.

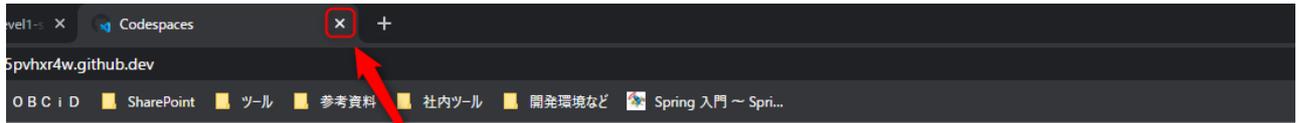
```
15 | | return step1Index ,  
16 | | }  
17 | }  
18 |  
  
2022-01-18 09:58:33.134 INFO 7530 --- [ restartedMain] o.a.c.c.C.[Tomca  
2022-01-18 09:58:33.134 INFO 7530 --- [ restartedMain] w.s.c.ServletWeb  
2022-01-18 09:58:33.435 INFO 7530 --- [ restartedMain] o.s.b.a.w.s.Welc  
2022-01-18 09:58:33.540 INFO 7530 --- [ restartedMain] o.s.b.d.a.Optior  
2022-01-18 09:58:33.566 INFO 7530 --- [ restartedMain] o.s.b.w.embeddec  
2022-01-18 09:58:33.578 INFO 7530 --- [ restartedMain] c.c.assignment.A  
2022-01-18 10:00:18.335 INFO 7530 --- [nio-8080-exec-1] o.a.c.c.C.[Tomca  
2022-01-18 10:00:18.335 INFO 7530 --- [nio-8080-exec-1] o.s.web.servlet.  
2022-01-18 10:00:18.337 INFO 7530 --- [nio-8080-exec-1] o.s.web.servlet.
```

11. 画面中央上部の「現在のCodespaceを中断する」を押します。



The screenshot shows the context menu for a Codespace in VS Code. The menu items are: 'Rebuild Container', 'Add Development Container Configuration Files...', '現在のCodespaceを中断する' (highlighted with a red dashed box), 'Go to Repository', 'My Codespaces', 'VS Code で開く', and 'VS Code Insiders で開く'. The 'Codespaces' header is visible in the top right of the menu.

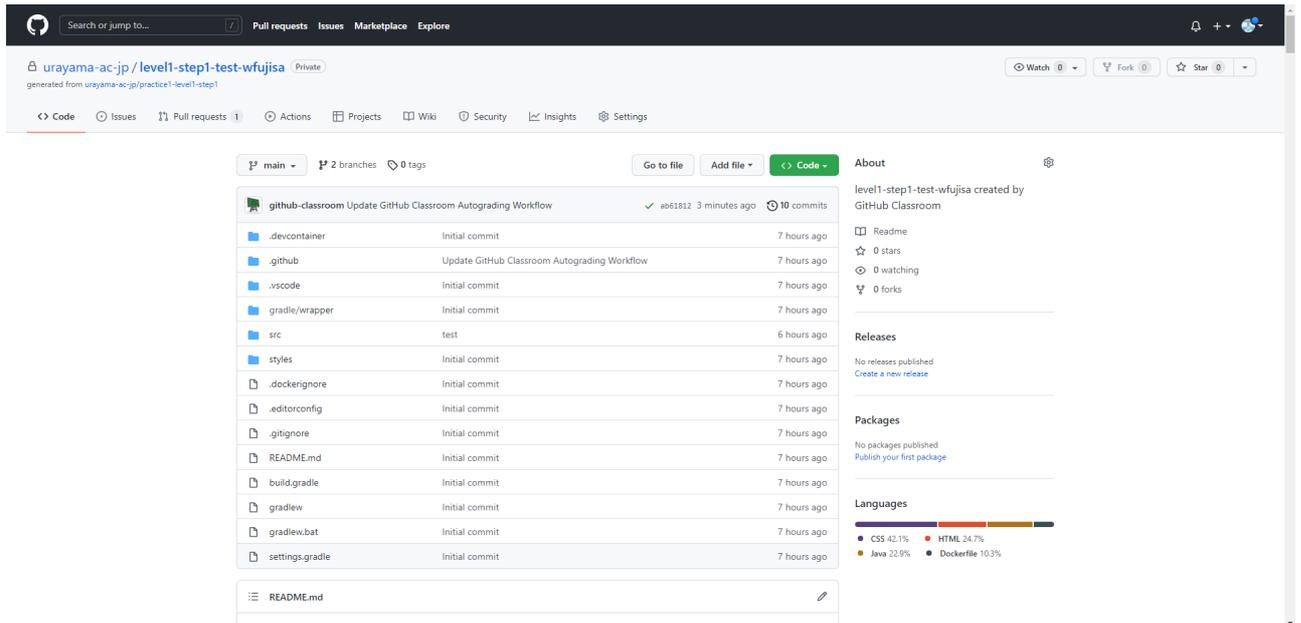
12. タブをxボタンで閉じましょう。



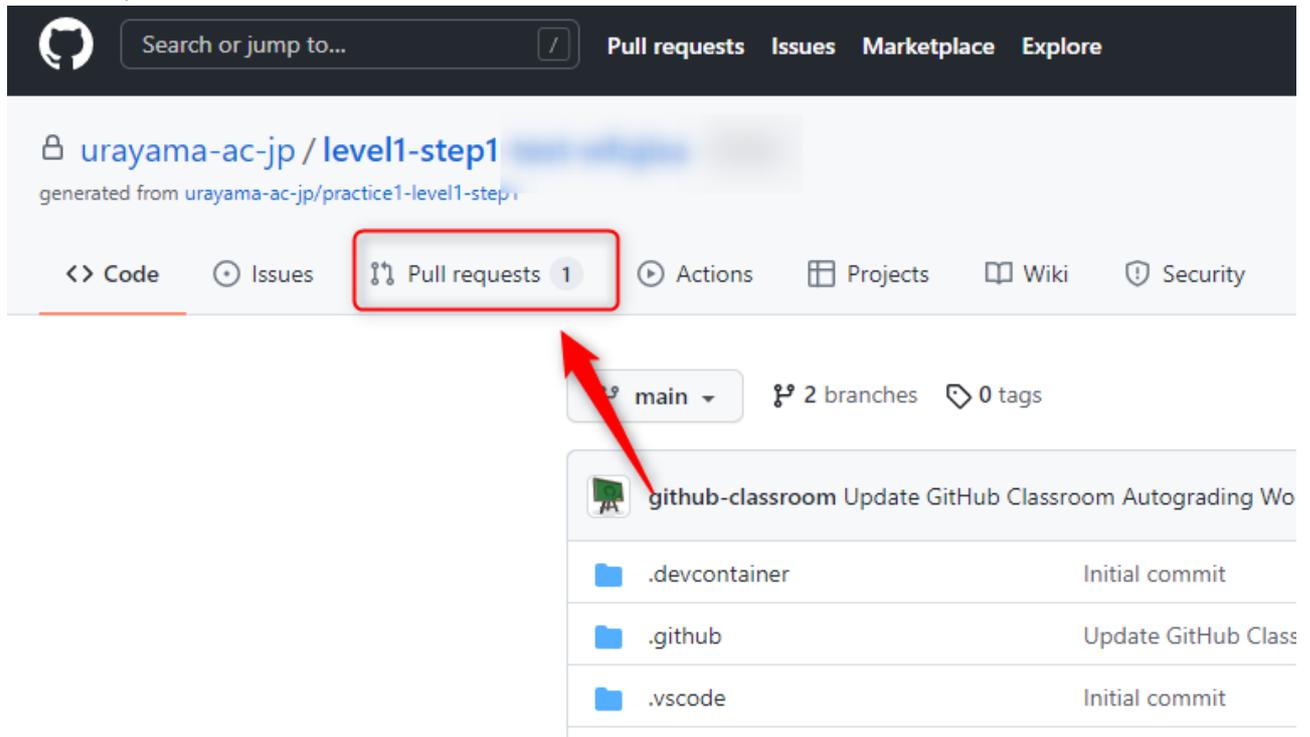
Codespace is stopped

Restart codespace

13. 画面コピーの画面に戻ります。



14. 「Pull requests」 ボタンを押します。



15. 講師からのフィードバックがある場合は、この画面に表示されます。

