

Eclipse에서 Git사용하기.

-GitHub에 올려져있는 프로젝트 가져오기

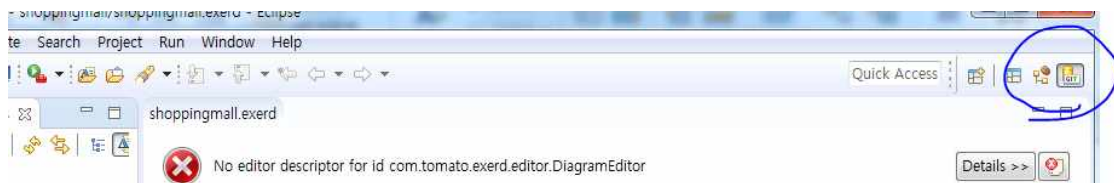
◆ 사전 준비 조건

- 1) 이클립스 neon3설치 (2017년도 06-13기준) =>egit이 플러그인 되어 있음
- 2) GitHub에 가입하여 새로운 프로젝트를 생성해 원격저장소를 만든 뒤 해당 URL정보를 복사해 둔다.

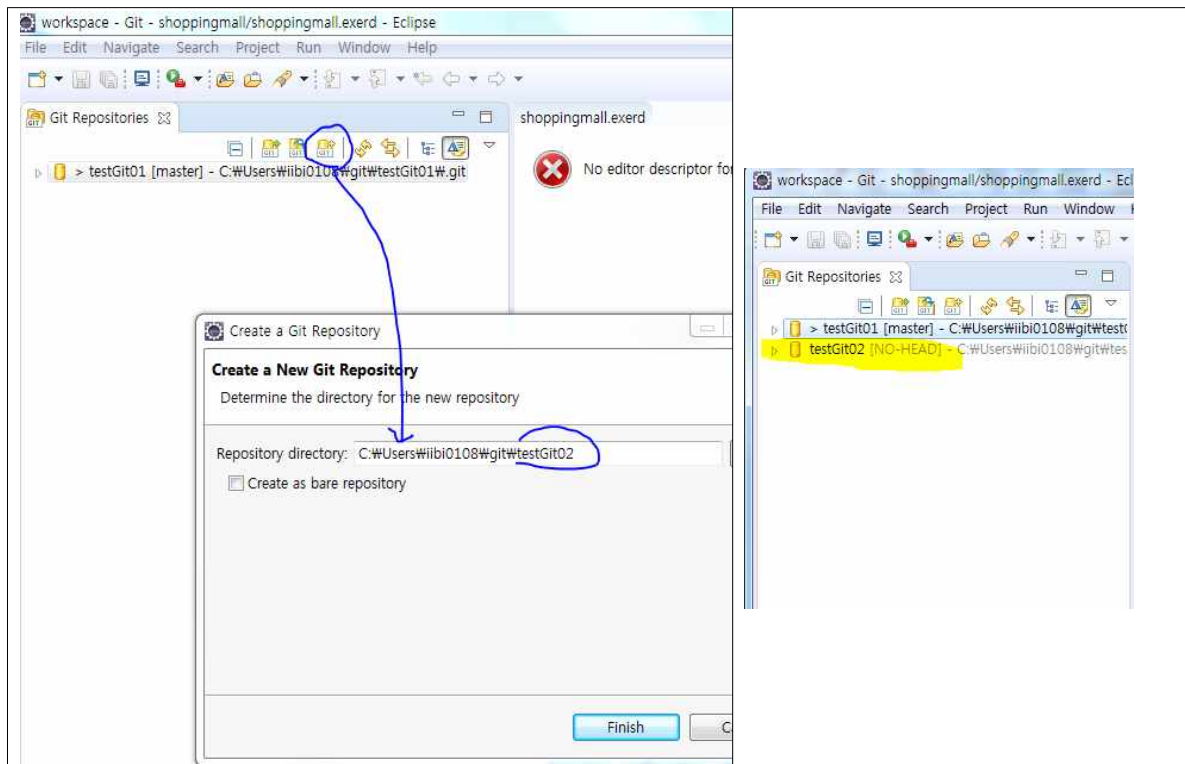
[예]

<https://github.com/swanbaek/IIBI02>

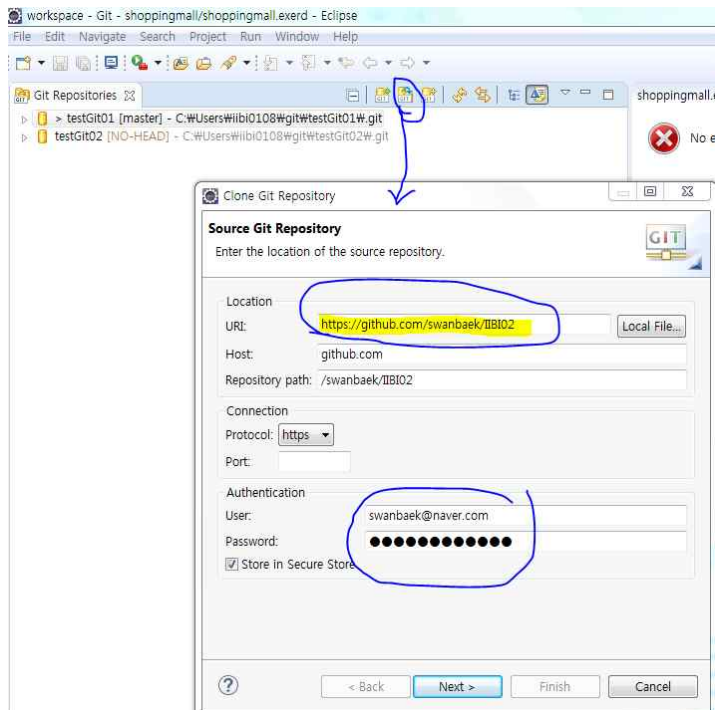
1. 이클립스에서 Git Perspective를 선택한다.



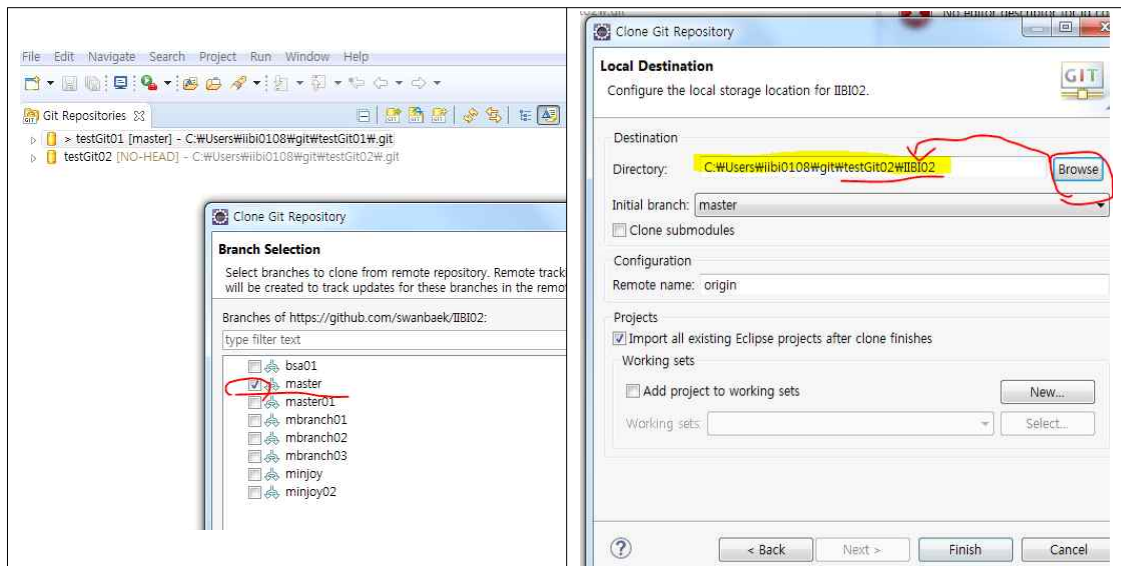
2. 로컬저장소 만들기



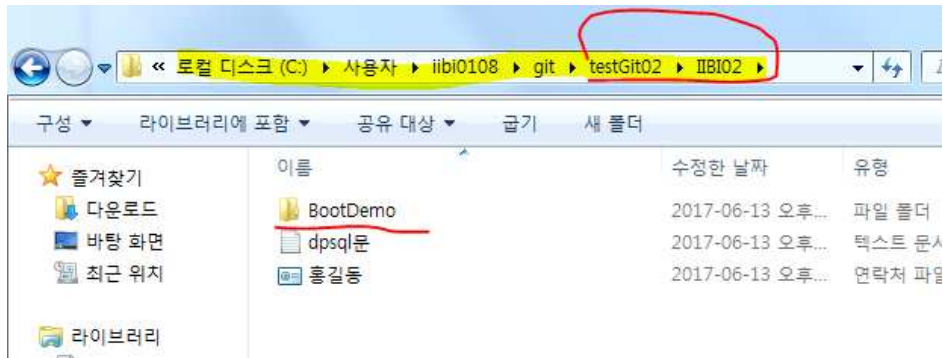
3. GitHub의 프로젝트 clone해서 가져오기



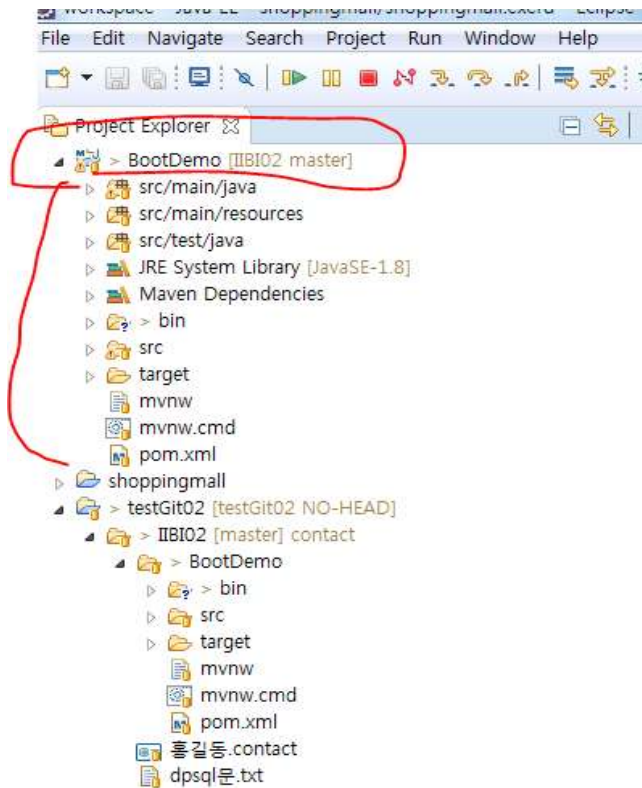
branch 선택-> mater로 선택함



4. 윈도우 탐색기에서 아래 경로를 찾아가보면 clone 되어 깃헙 원격저장소에 있던 프로젝트가 로컬 저장소에 있는 것을 확인할 수 있음



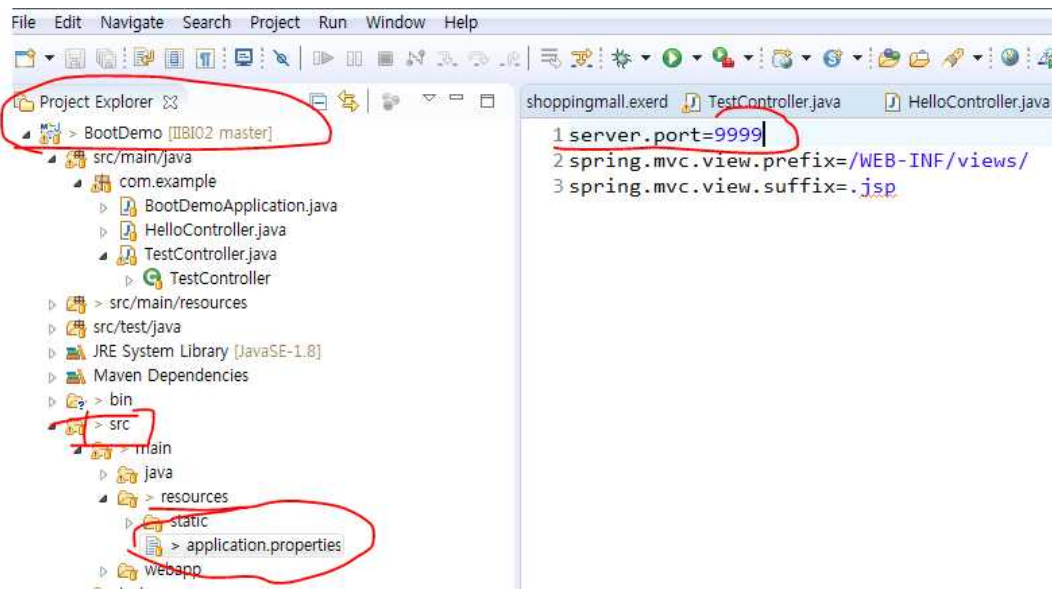
5. Git perspective에서 testGit02아래 Working Tree 오른 마우스를 클릭하여 Import Projects를 선택해 가져올 프로젝트를 선택하면 자바 perspective에 아래와 같이 프로젝트가 import된 것을 확인할 수 있음



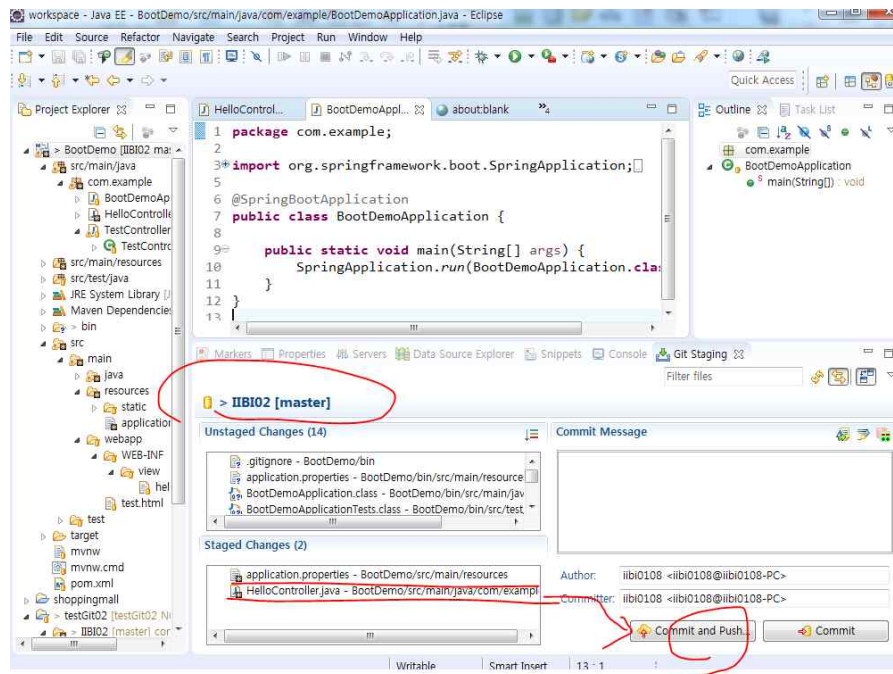
6. import한 프로젝트의 내용을 수정하여 로컬 저장소에 commit한 뒤 원격 저장소에도 반영해보자.

BootDemo/src/main/resource/application.properties파일을 열어

포트번호가 8080 되어 있던 것을 9999로 수정하자. (그 외에 HelloController.java파일 내용도 개인적으로 수정하였음)

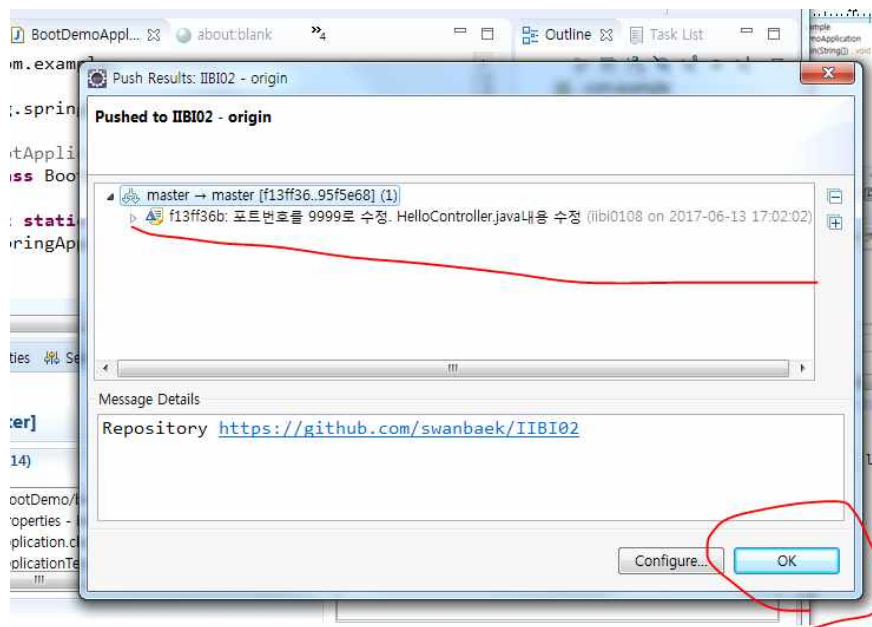


7. 프로젝트 익스플로러에서 BootDemo프로젝트 오른 마우스를 누르면 Team을 선택하고 Commit을 해보자. 그러면 아래와 같이 git staging이 나온다.



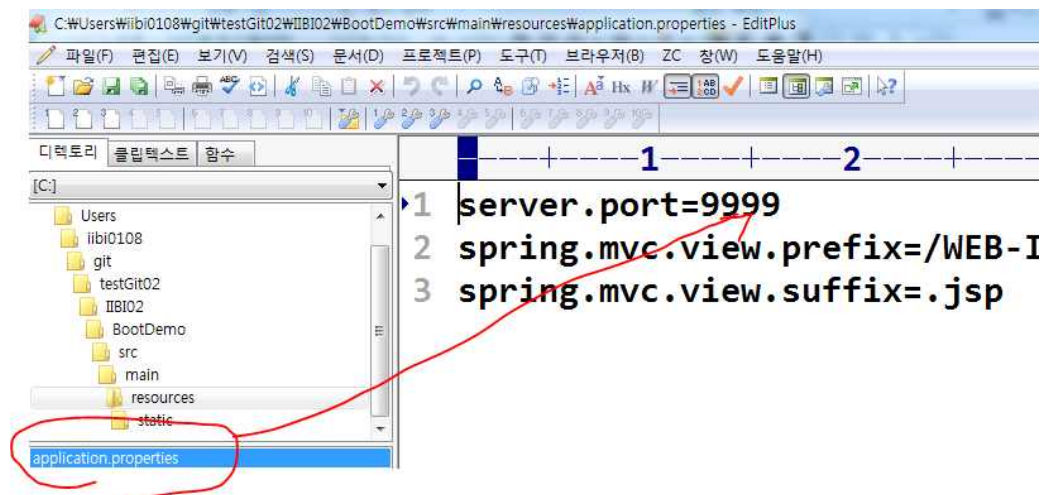
Commit and Push를 누른다. 이때 Commit Message를 입력하라고 나오면 간단하게 변경된

이력 정보를 써놓자. => 포트번호를 9999로 수정. HelloController.java내용 수정
식으로 입력한 뒤 다시 Commit and Push를 누른다.

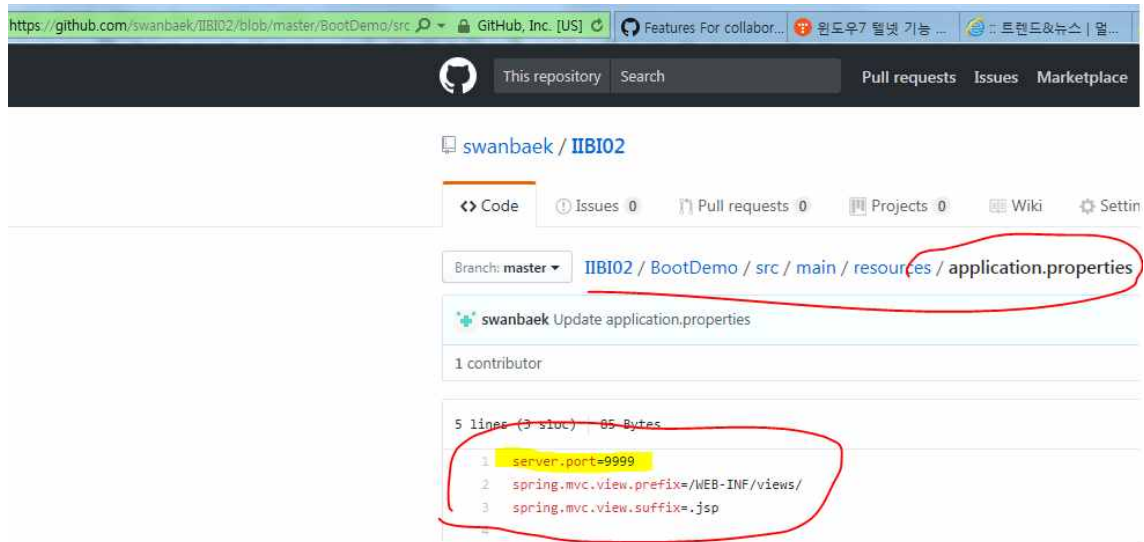


OK버튼을 누르자.

8. 그러면 윈도우 탐색기로 로컬 저장소에 가서 변경되었는지 확인하자.

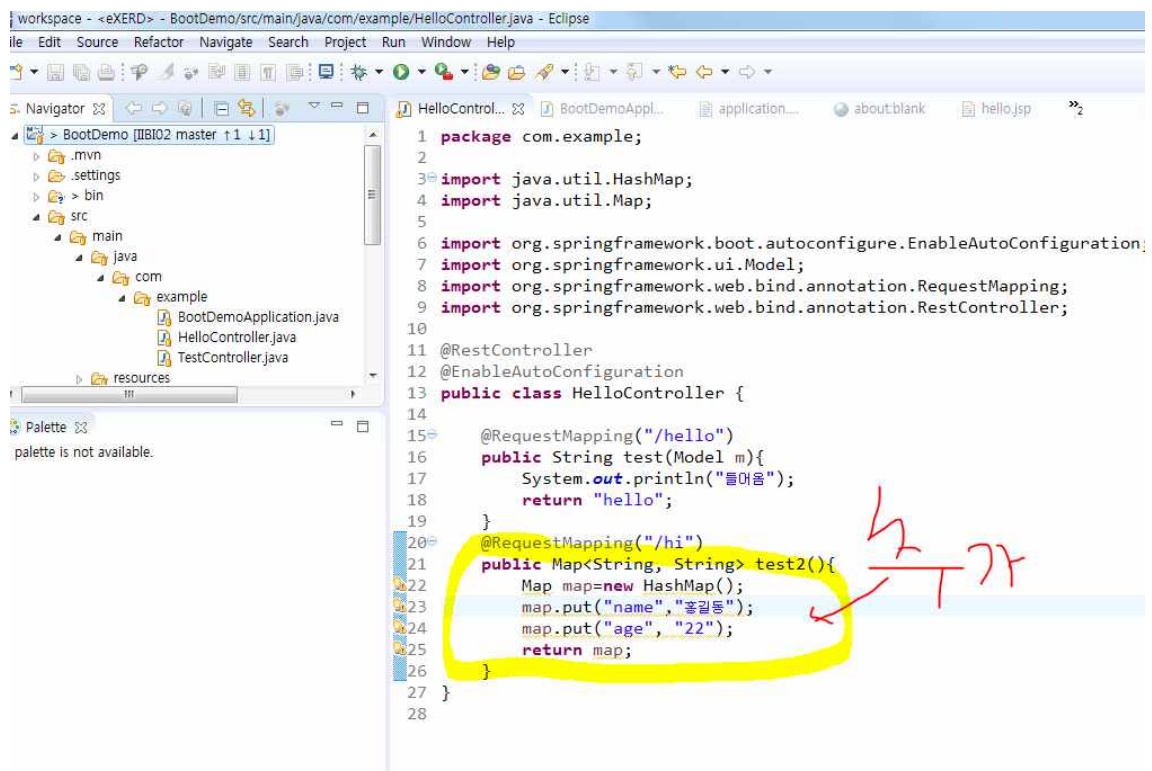


9. 또한 원격 저장소에 가서도 확인해보면 변경된 것을 볼 수 있다.



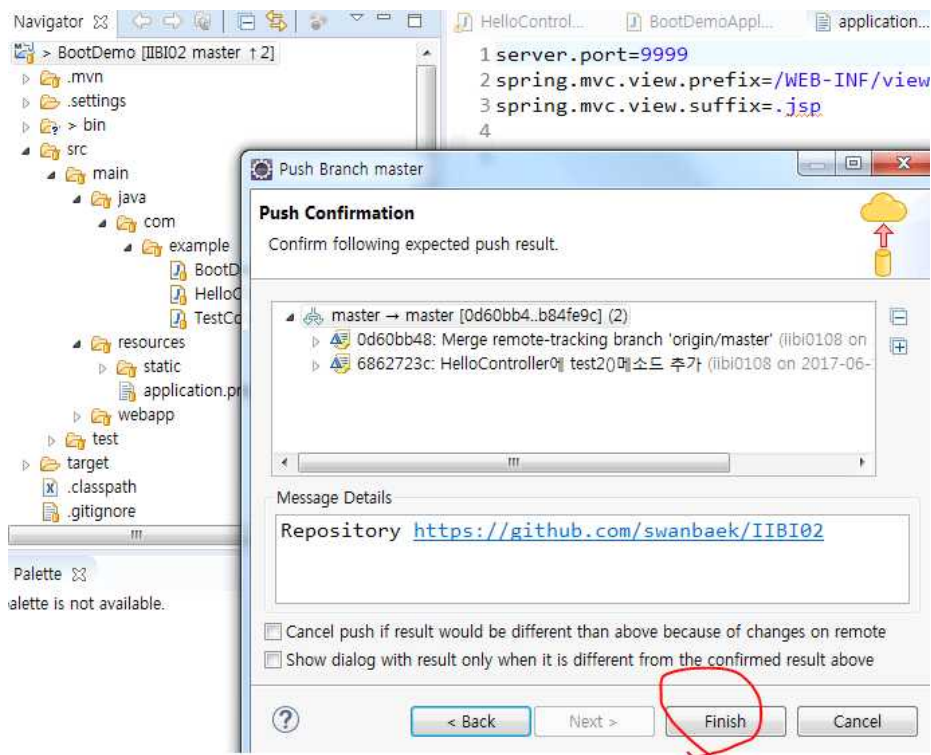
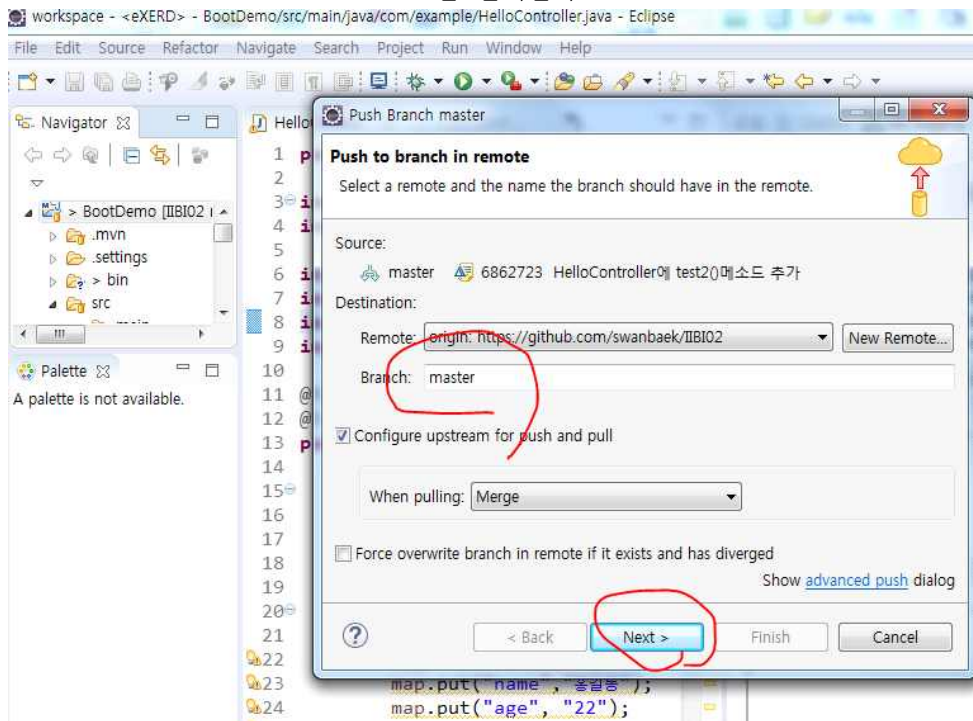
10. 자바 프로젝트에 소스를 추가 또는 수정하여 로컬에서 작업한 내역을 원격 저장소로 올려 보자.

in my case => HelloController에 아래 메소드를 추가함



11. 자바 프로젝트 익스플로러에서 BootDemo프로젝트 우클릭

=> Team=> Push branch Master를 선택한다.



finish마친 뒤 깃헙 원격 저장소에서 변경되었는지 확인하자.

swanbaek / IIBIO2

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights

Branch: master IIBIO2 / BootDemo / src / main / java / com / example / HelloController.java

Find file Copy path

IIBIO2 HelloController에 test2()메소드 추가

6862723 13 minutes ago

1 contributor

28 lines (23 sloc) 652 Bytes

Raw Blame History

```
1 package com.example;
2
3 import java.util.HashMap;
4 import java.util.Map;
5
6 import org.springframework.boot.autoconfigure.EnableAutoConfiguration;
7 import org.springframework.ui.Model;
8 import org.springframework.web.bind.annotation.RequestMapping;
9 import org.springframework.web.bind.annotation.RestController;
10
11 @RestController
12 @EnableAutoConfiguration
13 public class HelloController {
14
15     @RequestMapping("/hello")
16     public String test(Model m){
17         System.out.println("들아름");
18         return "hello";
19     }
20
21     @RequestMapping("/hi")
22     public Map<String, String> test2(){
23         Map map=new HashMap();
24         map.put("name","홍길동");
25         map.put("age","22");
26         return map;
27     }
28 }
```