

測試驅動程式開發實作

徐効群

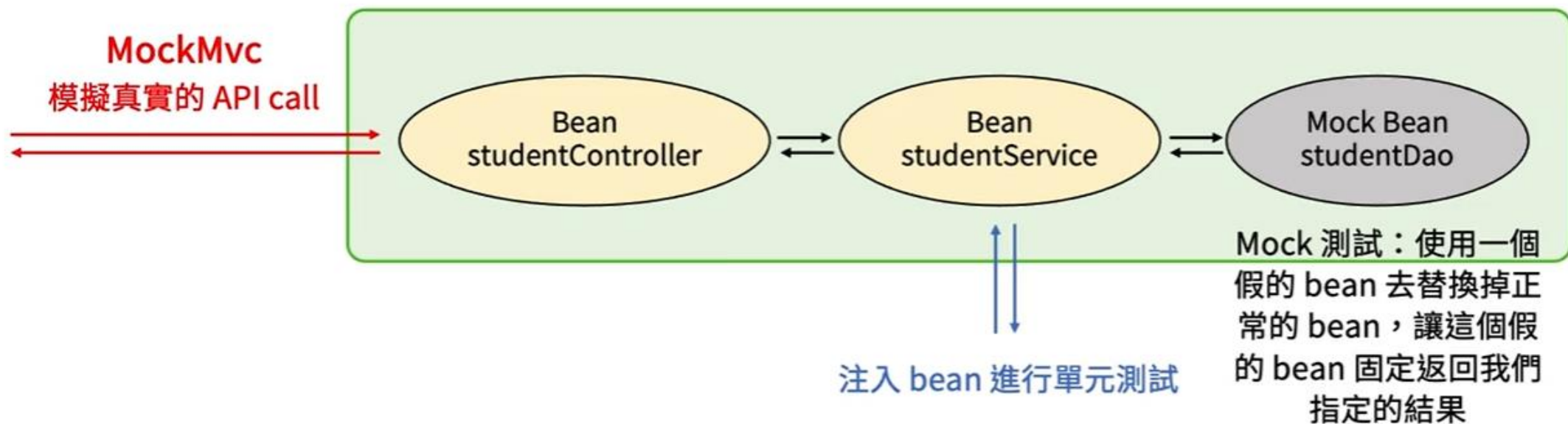
Jack Hsu

jack23h67@hotmail.com

範例下載連結

- ▶ <https://github.com/jack23h67/demoForTest02>

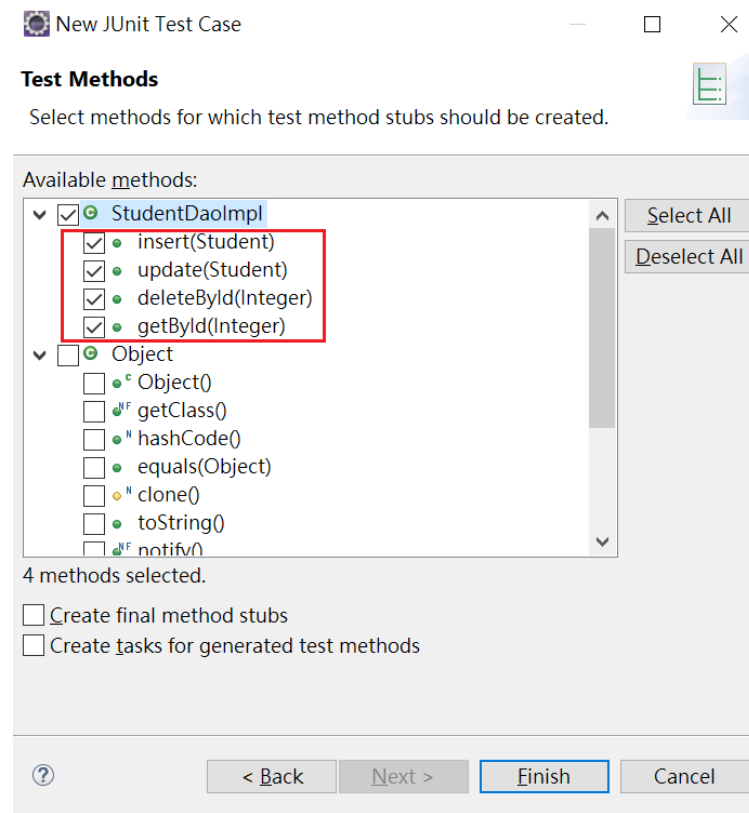
測試程式架構圖



虛擬資料庫 h2-pom.xml

```
28<dependency>
29    <groupId>org.springframework.boot</groupId>
30    <artifactId>spring-boot-starter-validation</artifactId>
31</dependency>
32<dependency>
33    <groupId>org.springframework.boot</groupId>
34    <artifactId>spring-boot-starter-jdbc</artifactId>
35</dependency>
36<!-- https://mvnrepository.com/artifact/com.h2database/h2 -->
37<dependency>
38    <groupId>com.h2database</groupId>
39    <artifactId>h2</artifactId>
40    <version>1.4.200</version>
41</dependency>
42<!-- https://mvnrepository.com/artifact/org.junit.jupiter/junit-jupiter-api -->
43<dependency>
44    <groupId>org.junit.jupiter</groupId>
45    <artifactId>junit-jupiter-api</artifactId>
46    <version>5.8.2</version>
47    <scope>test</scope>
48</dependency>
```

Dao層建立測試程式(1)



Dao層建立測試程式(2)

@Transactional

```
StudentDaoImplTest.java
7
8 import static org.junit.jupiter.api.Assertions.*;
9
10 import com.example.demo.model.Student;
11
12 @SpringBootTest
13 public class StudentDaoImplTest {
14
15     @Autowired
16     private StudentDao studentDao;
17
18     @Transactional
19     @Test
20     public void getById() {
21         Student student = studentDao.getById(3);
22         assertNotNull(student);
23         assertEquals("Judy", student.getName());
24         assertEquals(100.0, student.getScore());
25         assertTrue(student.isGraduate());
26         assertNotNull(student.getCreateDate());
27     }
28
29     @Transactional
30     @Test
31     public void deleteById() {
32         studentDao.deleteById(3);
33         Student student = studentDao.getById(3);
34         assertNull(student);
35     }
36 }
```

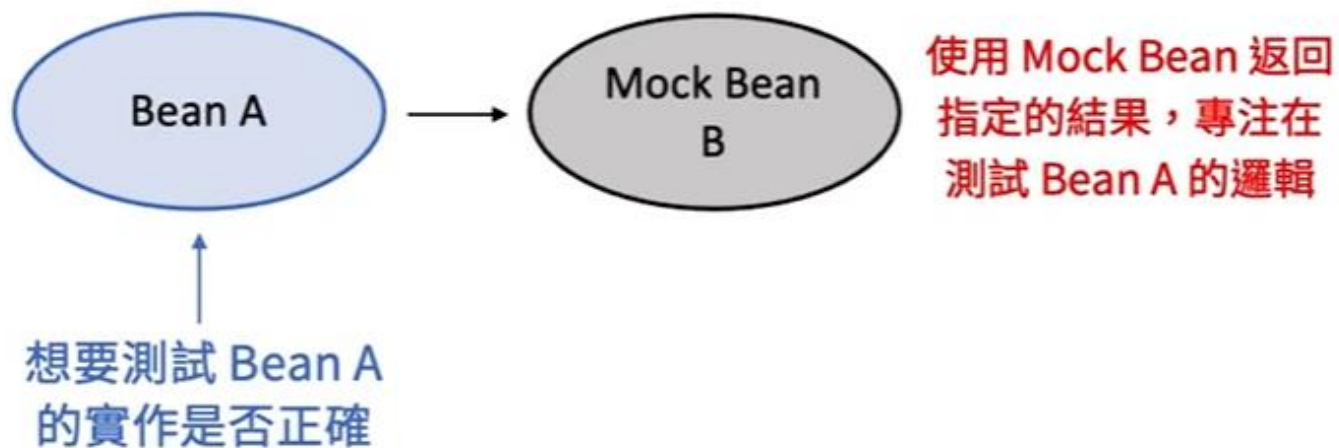
Dao層

測試檔加上註解-@Transactional

- ▶ @Transactional：在測試檔的測試方法上，加上此註解後，在執行完此測試方法後，spring boot就會rollback整個方法所執行過的資料庫操作
- ▶ 用法：可以加在方法上、也可以加在class上
- ▶ 用途：在單元測試結束後，會rollback所有資料庫操作，將數據恢復原狀
- ▶ @Transactional註解在測試檔與正式程式中，用途不同

Service層Mock測試

- ▶ 目的：避免為了測試某一個單元測試，而去建立了整個bean的依賴性
- ▶ 作法：創造一個假的bean，去替換掉Spring容器中原有的Bean



Service層建立測試程式(1)

New JUnit Test Case

JUnit Test Case
Select the name of the new JUnit test case. You have the options to specify

☐ New JUnit 3 test ☐ New JUnit 4 test ☒ New JUnit Jupiter test

Source folder:

Package:

Name:

Superclass:

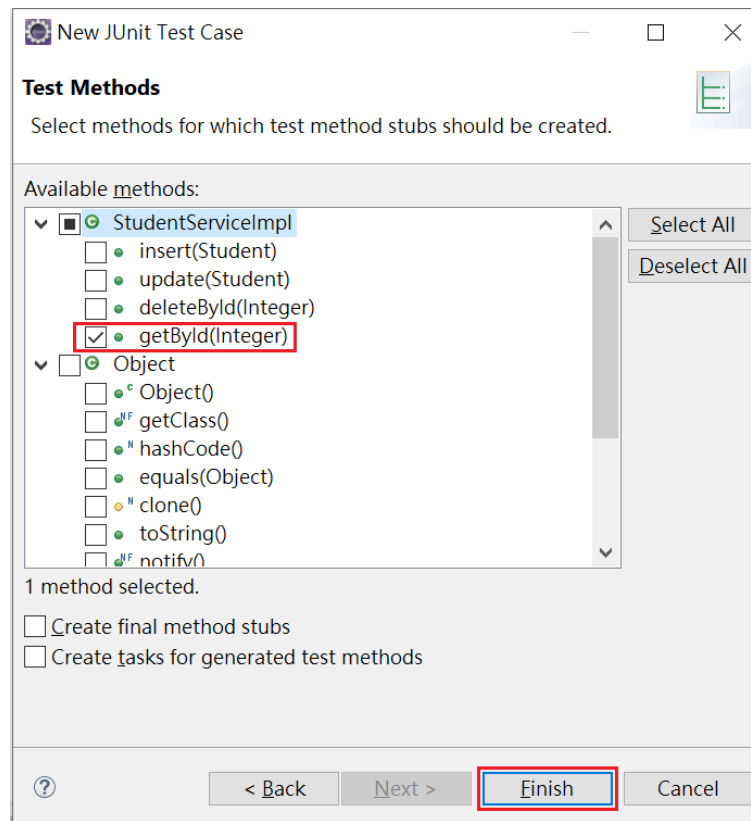
Which method stubs would you like to create?

☐ setUpBeforeClass() ☐ tearDownAfterClass()
☐ setUp() ☐ tearDown()
☐ constructor

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

Class under test:

Service層建立測試程式(2)



Service層建立測試程式(3)

```
StudentServiceImplMockTest.java
16 @SpringBootTest
17 public class StudentServiceImplMockTest {
18
19     @Autowired
20     private StudentService studentService;
21
22     // @SpyBean // 仍舊是正常的Bean，沒有定義的方法，預設返回null
23     // @MockBean // 產生一個假的Bean，只替換其中幾個方法，沒有定義的方法，預設返回真實的方法
24     private StudentDao studentDao;
25
26     @Test
27     public void testGetById() {
28         Student mockst = new Student();
29         mockst.setId(100);
30         mockst.setName("I'm mock");
31
32         Mockito.when(studentDao.getById(Mockito.any())).thenReturn(mockst);
33         // Mockito.doReturn(mockst).when(studentDao.getById(Mockito.any()));
34
35         // Mockito.when(studentDao.insert(Mockito.any())).thenThrow(new RuntimeException());
36         // Mockito.doThrow(new RuntimeException()).when(studentDao.insert(Mockito.any()));
37
38         // Mockito.verify(studentDao, Mockito.times(2)).getById(Mockito.any());
39
40         Student st = studentService.getById(3);
41         assertNotNull(st);
42         assertEquals(100, st.getId());
43         assertEquals("I'm mock", st.getName());
44     }
45
46 }
```

@MockBean 、 @SpyBean

Mockito when() thenReturn()

```
StudentServiceImplMockTest.java
16 @SpringBootTest
17 public class StudentServiceImplMockTest {
18
19     @Autowired
20     private StudentService studentService;
21
22     // @SpyBean // 仍舊是正常的Bean，沒有定義的方法，預設返回null
23     // @MockBean // 產生一個假的Bean，只替換其中幾個方法，沒有定義的方法，預設返回真實的方法
24     private StudentDao studentDao;
25
26     @Test
27     public void testGetById() {
28         Student mockst = new Student();
29         mockst.setId(100);
30         mockst.setName("I'm mock");
31
32         Mockito.when(studentDao.getById(Mockito.any())).thenReturn(mockst);
33         // Mockito.doReturn(mockst).when(studentDao.getById(Mockito.any()));
34
35         // Mockito.when(studentDao.insert(Mockito.any())).thenThrow(new RuntimeException());
36         // Mockito.doThrow(new RuntimeException()).when(studentDao.insert(Mockito.any()));
37
38         // Mockito.verify(studentDao, Mockito.times(2)).getById(Mockito.any());
39
40         Student st = studentService.getById(3);
41         assertNotNull(st);
42         assertEquals(100, st.getId());
43         assertEquals("I'm mock", st.getName());
44     }
45
46 }
```

Controller層建立測試程式(1)-pom.xml

```
demoForTest02/pom.xml x StudentControllerTest.java
39      <groupId>org.springframework</groupId>
40      <version>1.4.200</version>
41    </dependency>
42    <!-- https://mvnrepository.com/artifact/org.junit.jupiter/junit-jupiter-api -->
43    <dependency>
44      <groupId>org.junit.jupiter</groupId>
45      <artifactId>junit-jupiter-api</artifactId>
46      <version>5.8.2</version>
47      <scope>test</scope>
48    </dependency>
49    <dependency>
50      <groupId>io.rest-assured</groupId>
51      <artifactId>spring-mock-mvc</artifactId>
52      <version>3.0.0</version>
53      <scope>test</scope>
54    </dependency>
```

Controller層建立測試程式(2)-測試程式 MockMvc、perform()、andExpect()

```
demoForTest02/pom.xml StudentControllerTest.java
1 package com.example.demo.controller;
2
3 import org.junit.jupiter.api.Test;
4 import org.springframework.beans.factory.annotation.Autowired;
5 import org.springframework.boot.test.autoconfigure.web.servlet.AutoConfigureMockMvc;
6 import org.springframework.boot.test.context.SpringBootTest;
7 import org.springframework.test.web.servlet.MockMvc;
8 import org.springframework.test.web.servlet.RequestBuilder;
9 import org.springframework.test.web.servlet.request.MockMvcRequestBuilders;
10 import org.springframework.test.web.servlet.result.MockMvcResultMatchers;
11
12 @SpringBootTest
13 @AutoConfigureMockMvc
14 public class StudentControllerTest {
15
16     @Autowired
17     private MockMvc mockMvc;
18
19     @Test
20     public void testGetById() throws Exception {
21         RequestBuilder requestBuilder = MockMvcRequestBuilders.get("/students/3");
22
23         mockMvc.perform((org.springframework.test.web.servlet.RequestBuilder) requestBuilder)
24             .andExpect(MockMvcResultMatchers.status().is(200));
25     }
26 }
27
28 }
29
```

測試驅動開發應用-Lab

- ▶ 完成service層增刪改查的操作測試檔
- ▶ 查找一下在Controller層中的andExpect()的其他用法
- ▶ 進階作業：完成Controller層的增刪改操作測試檔

參考資源

- ▶ <https://kucw.github.io/blog/2020/2/spring-unit-test-mockito/>
- ▶ <https://hahow.in/courses/5fe22e7fe810e10fc483dd78/main>

thanks for listening