链滴

Mockito 简单使用

作者：MingGH

原文链接：https://ld246.com/article/1632661328393

来源网站：链滴

1. 一些关于Mockito的资料

Mockito官方文档

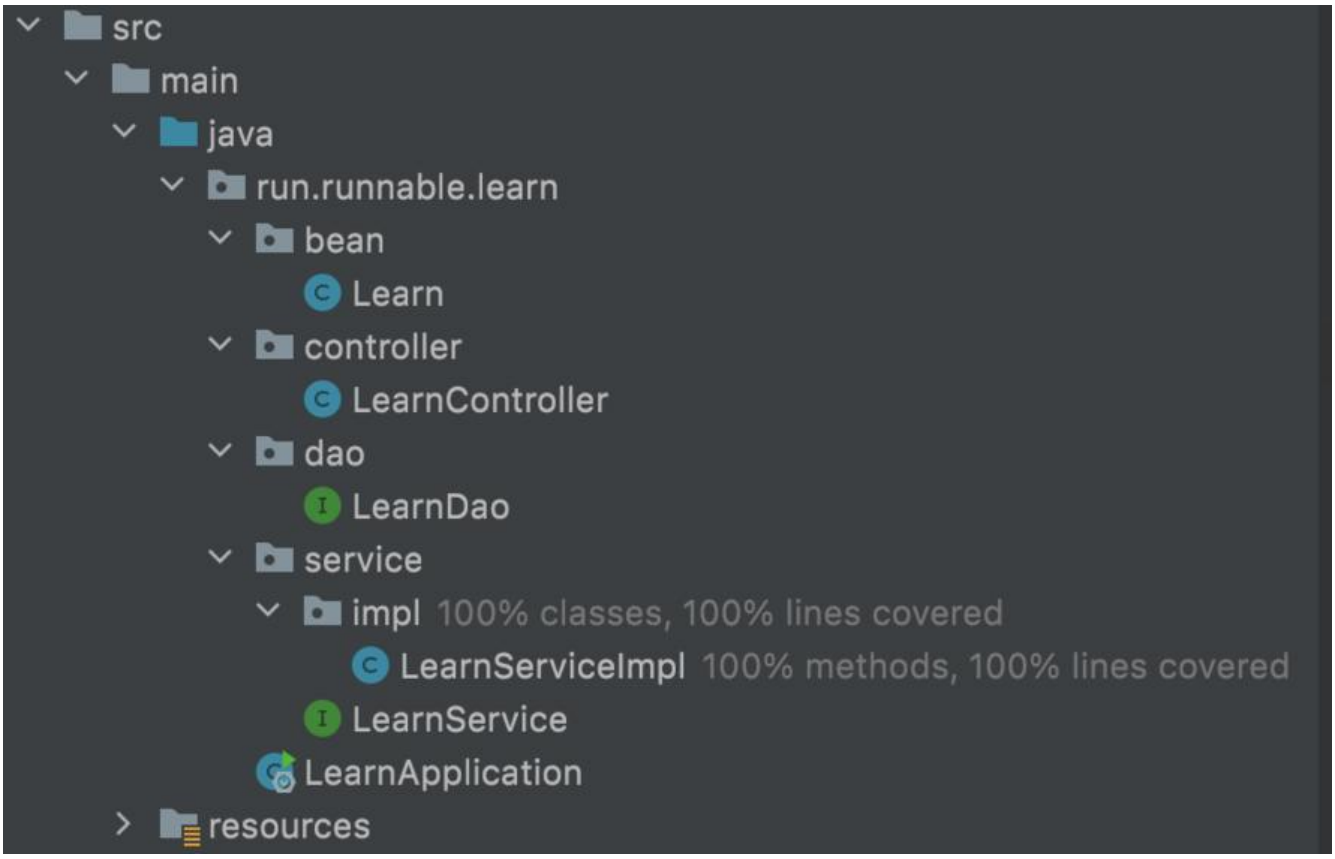
掘金 | Mockito 的最佳实践

2. 搭建一个学习Mockito 的环境

新建一个SpringBoot项目，然后引入依赖

```
<dependency>
        <groupId>org.mockito</groupId>
        <artifactId>mockito-core</artifactId>
        <version>3.12.4</version>
        <scope>test</scope>
    </dependency>

    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <scope>test</scope>
    </dependency>

    <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter</artifactId>
        <scope>test</scope>
    </dependency>

    <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter-engine</artifactId>
        <scope>test</scope>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>

    <dependency>
        <groupId>com.alibaba</groupId>
        <artifactId>fastjson</artifactId>
        <version>1.2.70</version>
    </dependency>
```

创建通常项目使用类



Learn

```
public class Learn {

    private Integer id;
    private String title;

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    public Learn() {
    }

    public Learn(Integer id, String title) {
        this.id = id;
        this.title = title;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
```

```
    }

    @Override
    public String toString() {
        return "Learn{" +
                "id=" + id +
                ", title='" + title + '\'' +
                '}';
    }
}
```

LearnController

```
package run.runnable.learn.controller;

import com.alibaba.fastjson.JSONObject;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.ResponseBody;
import run.runnable.learn.bean.Learn;
import run.runnable.learn.service.LearnService;

@Controller
public class LearnController {

    @Autowired
    private LearnService learnService;

    @ResponseBody
    public Learn getById(Integer id){
        return learnService.getById(id);
    }

}
```

LearnDao

```
package run.runnable.learn.dao;

import run.runnable.learn.bean.Learn;


public interface LearnDao {

    Learn getById(Integer id);

    int save(Learn learn);

    int update(Integer id, Learn learn);

}
```

LearnServiceImpl

```
package run.runnable.learn.service.impl;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import run.runnable.learn.bean.Learn;
import run.runnable.learn.dao.LearnDao;
import run.runnable.learn.service.LearnService;

@Service
public class LearnServiceImpl implements LearnService {

    @Autowired
    private LearnDao learnDao;

    @Override
    public Learn getById(Integer id) {
        if (id==null){
            throw new NullPointerException();
        }
        return learnDao.getById(id);
    }

    @Override
    public int save(Learn learn) {
        if (learn==null){
            return 0;
        }
        return learnDao.save(learn);
    }

    @Override
    public int update(Integer id, Learn learn) {
        Learn findLearn = learnDao.getById(id);
        findLearn.setTitle(learn.getTitle());
        return learnDao.update(id,findLearn);
    }
}
```

LearnService

```
package run.runnable.learn.service;

import run.runnable.learn.bean.Learn;

public interface LearnService {

    Learn getById(Integer id);

    int save(Learn learn);
```

```
    int update(Integer id,Learn learn);

}
```

3. 针对LearnServiceImpl单元测试

```
package run.runnable.learn.service.impl;

import org.junit.Assert;
import org.junit.jupiter.api.Assertions;
import org.junit.jupiter.api.DisplayName;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.extension.ExtendWith;
import org.mockito.InjectMocks;
import org.mockito.Mock;
import org.mockito.Mockito;
import org.mockito.junit.jupiter.MockitoExtension;
import run.runnable.learn.bean.Learn;
import run.runnable.learn.dao.LearnDao;

@ExtendWith(MockitoExtension.class)
public class LearnServiceImplTest  {

    @Mock
    private LearnDao learnDao;
    @InjectMocks
    private LearnServiceImpl learnService;

    @Test
    void getById(){
        Mockito.when(learnDao.getById(Mockito.any())).thenReturn(new Learn(1,"Math"));
        Learn learn = learnService.getById(1);
        Assert.assertEquals(new Integer(1),learn.getId());
    }

    @Test
    @DisplayName("捕获异常")
    public void getByIdWithNull(){
        Assertions.assertThrows(NullPointerException.class,()->{
            learnService.getById(null);
        });
    }

    @Test
    void save(){
        Mockito.when(learnDao.save(Mockito.any())).thenReturn(1);
        int saveCount = learnService.save(new Learn());
        Assert.assertEquals(new Integer(1),new Integer(saveCount));
    }

    @Test
    void saveNull(){
        int saveCount = learnService.save(null);
```

```
        Assert.assertEquals(new Integer(0),new Integer(saveCount));
    }

    @Test
    void update(){
        Mockito.when(learnDao.update(Mockito.anyInt(),Mockito.any())).thenReturn(1);
        Mockito.when(learnDao.getById(Mockito.any())).thenReturn(new Learn(1,"Math"));
        int update = learnService.update(1, new Learn(1, "2"));
        Assert.assertEquals(new Integer(1),new Integer(update));
    }

}
```