

E-Learning System 设计文档

Assignment 5



小组成员	
17301088	包天活
17301101	刘彬

一、Unit Testing

本项目使用用户邮箱作为账号，所以在用户登录时需要对用户输入的邮箱进行邮箱格式的有效性的检测。

1、 Junit 配置

Junit 为安卓的项目自动携带的测试支持。查看“build.gradle”文件，发现该依赖已经自动添加。

```
testImplementation 'junit:junit:4.12'
```

2、 待测试类和待测试方法

```
public class RegexUtil {  
    /**  
     * 正则表达式:验证手机号  
     */  
    private static final String REGEX_MOBILE = "^(0|86|17951)?(13[0-9]|15[012356789]|17[678]|18[0-9]|14[57])[0-9]{8}$";  
  
    /**  
     * 正则表达式:验证邮箱  
     */  
    private static final String REGEX_EMAIL = "^[a-zA-Z0-9-]+(\\.[a-zA-Z0-9-]+)*@[a-zA-Z0-9-]+(\\.[a-zA-Z0-9-]+)+$";  
  
    public static boolean isPhone(String value) { return value.matches(REGEX_MOBILE); }  
  
    public static boolean isEmail(String value) { return value.matches(REGEX_EMAIL); }  
}
```

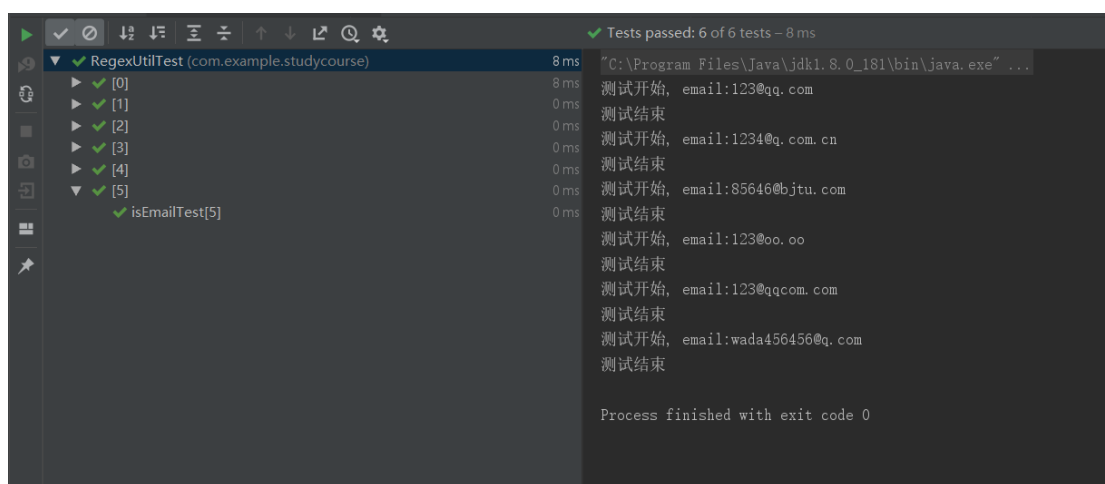
3、 编写测试类

```
@RunWith(Parameterized.class)  
public class RegexUtilTest {  
  
    private String mail;  
  
    public RegexUtilTest(String mail) { this.mail = mail; }  
  
    @Parameterized.Parameters  
    public static Collection emails() { ... }  
  
    @Before  
    public void before() {  
        System.out.println("测试开始, email:" + mail);  
    }  
  
    @After  
    public void after() {  
        System.out.println("测试结束");  
    }  
  
    @Test  
    public void isEmailTest() {  
        assertTrue(RegexUtil.isEmail(mail));  
    }  
}
```

4、 添加测试参数，进行参数化测试

```
@Parameterized.Parameters
public static Collection emails(){
    return Arrays.asList(new String[]{
        "123@qq.com",
        "1234@q.com.cn",
        "85646@bjtu.com",
        "123@oo.oo",
        "123@qqcom.com",
        "wada456456@q.com"
    });
}
```

5、 测试结果



Tests passed: 6 of 6 tests - 8 ms

Test Case	Duration	Status
RegexUtilTest (com.example.studycourse)	8 ms	Passed
[0]	8 ms	Passed
[1]	0 ms	Passed
[2]	0 ms	Passed
[3]	0 ms	Passed
[4]	0 ms	Passed
[5]	0 ms	Passed
isEmailTest[5]	0 ms	Passed

Test Log:

- 测试开始, email:123@qq.com
- 测试结束
- 测试开始, email:1234@q.com.cn
- 测试结束
- 测试开始, email:85646@bjtu.com
- 测试结束
- 测试开始, email:123@oo.oo
- 测试结束
- 测试开始, email:123@qqcom.com
- 测试结束
- 测试开始, email:wada456456@q.com
- 测试结束

Process finished with exit code 0

二、 Android Testing

测试单个应用内的用户交互有助于确保用户在与应用交互时不会遇到意外结果或体验不佳。如果需要验证应用的界面是否正常运行，可以创建界面测试。

本项目对登录模块进行测试。

1、 添加依赖

```
androidTestImplementation 'androidx.test.espresso:espresso-intents:3.1.0'
```

2、 新建测试类

```

@RunWith(AndroidJUnit4.class)
@LargeTest
public class LoginActivityTest {

    private String mail = "123@qq.com";
    private String password = "123456";

    @Rule
    public IntentsTestRule<LoginActivity> intentsTestRule = new IntentsTestRule<>(LoginActivity.class);

    @Test
    public void testLogin(){
        //输入邮箱和密码
        onView(withId(R.id.et_username))
            .perform(typeText(mail), closeSoftKeyboard());
        onView(withId(R.id.et_password))
            .perform(typeText(password), closeSoftKeyboard());

        //点击登录
        onView(withId(R.id.bt_login))
            .perform(click());
    }
}

```

3、 运行结果（详情见附件“AutoTest.mp4”）

