

# TestLog0825

## 目录

- 1. 已测试通过的接口
  - 1.1. findEnrolledStudent
  - 1.2. findStudentById
  - 1.3. findStudentByTestId
  - 1.4. submitPlan
- 2. 未测试接口
- 3. approveEnrolledStudent
  - 3.1. testValidInput
    - 3.1.1. 输入字段读取错误
  - 3.2. testInvalidInput
    - 3.2.1. 未做出错处理
- 4. approvePlan
  - 4.1. testValidInput
    - 4.1.1. 类型转换错误
  - 4.2. testInvalidInput
    - 4.2.1. 验证方式错误
- 5. findEnrollmentResultById
  - 5.1. testValidInput
    - 5.1.1. 缺少返回值
    - 5.1.2. 数据格式错误
  - 5.2. testInvalidInput
    - 5.2.1. 缺少错误处理
- 6. rejectEnrolledStudent
  - 6.1. testInvalidInput
- 7. rejectPlan
  - 7.1. testValidInput
    - 7.1.1. 类型转换错误
  - 7.2. testInvalidInput
    - 7.2.1. 验证方式错误
- 8. submitVolunteer
  - 8.1. testValidInput
  - 8.2. testInvalidInput

- 9. findApprovedStudent
  - 9.1. testValidInput
    - 9.1.1. 文档错误

## 1. 已测试通过的接口

### 1.1. findEnrolledStudent

```
TestFindEnrolledStudent Input: {"universityId":"1"}
TestFindEnrolledStudent Output: [{"majors":[{"nclass":"经济与贸易类","classId":5,"regionId":1}]}]
TestFindEnrolledStudent Input: {"universityId":"0"}
TestFindEnrolledStudent Output: {"StatusCode":"403"}
```

### 1.2. findStudentById

```
[INFO] Running sources.TestFindStudentById
TestFindStudentById Input: {"id":"1"}
TestFindStudentById Output: {"gender":"男","englishScore":138,"regionId":5,"chineseScore":120}
TestFindStudentById Input: {"id":"0"}
TestFindStudentById Output: {"StatusCode":"402"}
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.153 s - in sources.TestFindStudentById
```

### 1.3. findStudentByTestid

```
[INFO] Running sources.TestFindStudentByTestid
TestFindStudentByTestid Input: {"name":"王一", "testId":"19875426874625"}
TestFindStudentByTestid Output: {"gender":"男","englishScore":138,"regionId":5,"chineseScore":120}
TestFindStudentByTestid Input: {"name":"王一", "testId":"0"}
TestFindStudentByTestid Output: {"StatusCode":"402"}
TestFindStudentByTestid Input: {"name":"王一", "testId":"12345824562584"}
TestFindStudentByTestid Output: {"StatusCode":"402"}
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.281 s - in sources.TestFindStudentByTestid
```

### 1.4. submitPlan

```
TestSubmitPlan Input: {"universityId":"1","number":"10","classId":"1","year":"2020","regionId":1}
TestSubmitPlan Output: {"StatusCode":200}
```

## 2. 未测试接口

1. login.java
2. enroll.java

## 3. apporveEnrolledStudent

### 3.1. testValidInput

#### 3.1.1. 输入字段读取错误

输入为 {studentId: 1}

读取为, `int id= Integer.parseInt(request.getParameter("studentid"));`

---

Test set: sources.TestApproveEnrolledStudent

---

Tests run: 4, Failures: 4, Errors: 0, Skipped: 0, Time elapsed: 0.001 s <<< FAILURE! - :  
sources.TestApproveEnrolledStudent.testValidInput Time elapsed: 0 s <<< FAILURE!  
java.lang.AssertionError:

Unexpected method call HttpServletRequest.getParameter("studentid"):  
HttpServletRequest.getParameter("studentId"): expected: 1, actual: 0  
at sources.TestApproveEnrolledStudent.testValidInput(TestApproveEnrolledStudent.java:

sources.TestApproveEnrolledStudent.testValidInput Time elapsed: 0 s <<< FAILURE!  
java.lang.AssertionError:

Expectation failure on verify:  
HttpServletRequest.getParameter("studentId"): expected: 1, actual: 0  
Unexpected method calls:  
HttpServletRequest.getParameter("studentid")  
at sources.TestApproveEnrolledStudent.tearDown(TestApproveEnrolledStudent.java:62)

sources.TestApproveEnrolledStudent.testInvalidInput Time elapsed: 0 s <<< FAILURE!  
java.lang.AssertionError:

Unexpected method call HttpServletRequest.getParameter("studentid"):  
HttpServletRequest.getParameter("studentId"): expected: 1, actual: 0  
at sources.TestApproveEnrolledStudent.testInvalidInput(TestApproveEnrolledStudent.ja:

sources.TestApproveEnrolledStudent.testInvalidInput Time elapsed: 0 s <<< FAILURE!  
java.lang.AssertionError:

Expectation failure on verify:  
HttpServletRequest.getParameter("studentId"): expected: 1, actual: 0  
Unexpected method calls:  
HttpServletRequest.getParameter("studentid")  
at sources.TestApproveEnrolledStudent.tearDown(TestApproveEnrolledStudent.java:62)

## 3.2. testInvalidInput

### 3.2.1. 未做出错处理

访问不存在的学生id时，应该返回错误码 `response.sendError(403, "提交错误，请刷新后重试")`；，参考 `findStudentById` 进行修改。

## 4. approvePlan

### 4.1. testValidInput

### 4.1.1. 类型转换错误

```
java.lang.ClassCastException: java.util.ArrayList cannot be cast to com.courseDesign.plan
    at com.courseDesign.servlet.approvePlan.doPost(approvePlan.java:37)
```

List 类型不能通过强制类型转换转为 plan 类型。

```
plan plan0 = (plan) baseDao.executeQuery(sql0,planid);
```

## 4.2. testInvalidInput

### 4.2.1. 验证方式错误

类型转换错误，以下错误处理的验证方式 `planid0 ==null` 也失效。

```
if(planid0 ==null){
    System.out.println("没进行批量退选");
}else {
    PrintWriter out=response.getWriter();
    for (int i = 0; i < planid0.size(); i++) {
        planid = Integer.parseInt(planid0.getString(i));
        String sql0="select * from plan where planid=?";
        plan plan0 = (plan) baseDao.executeQuery(sql0,planid);
        if(plan0==null){
            response.sendError(402, "提交失败，请刷新后重试");
        }
        else {
            String sql = "update plan set is_approved=1 where planid=?";
            baseDao.executeUpdate(sql, planid);
        }
    }
}
```

## 5. findEnrollmentResultById

### 5.1. testValidInput

#### 5.1.1. 缺少返回值

---

```
Test set: sources.TestFindEnrollmentResultById
```

---

```
Tests run: 3, Failures: 3, Errors: 0, Skipped: 0, Time elapsed: 4.858 s <<< FAILURE! - :
sources.TestFindEnrollmentResultById.testValidInput Time elapsed: 4.797 s <<< FAILURE
java.lang.AssertionError: expected:<1> but was:<null>
    at sources.TestFindEnrollmentResultById.testValidInput(TestFindEnrollmentResultById.
```

缺少接口文档中的返回值，且返回值key与期望输出不匹配。

[https://app.swaggerhub.com/apis-docs/CurrenWong/CourseDesign/0.0.2#/Student/post\\_findEnrollmentResultById\\_do](https://app.swaggerhub.com/apis-docs/CurrenWong/CourseDesign/0.0.2#/Student/post_findEnrollmentResultById_do)

期望输出为

```
{
  "id": 1,
  "universityId": 1,
  "universityName": "上海理工大学",
  "studentId": 1,
  "studentName": "王一",
  "year": 2020,
  "type": "统招",
  "classId": 5,
  "className": "经济与贸易类",
  "isApproved": 0
}
```

实际输出为

```
{
  "studentid":1,
  "universityid":1,
  "year":"2020-01-01",
  "class_id":5,
  "is_approved":1,
  "id":1,
  "type":"统招"
}
```

### 5.1.2. 数据格式错误

年份在数据库中存储为 2020，实际输出为 "2020-01-01"。

数据库里的 year 类型，按照 int 类型读取即可，不需要转为 Date

```
ues.setYear((Date) map.get("year"));
```

## 5.2. testInvalidInput

```
sources.TestFindEnrollmentResultById.testInvalidInput Time elapsed: 0 s <<< FAILURE!  
java.lang.AssertionError:
```

```
Unexpected method call HttpServletResponse.getWriter():  
HttpServletResponse.sendError(403 (int), "无录取结果"): expected: 1, actual: 0  
at sources.TestFindEnrollmentResultById.testInvalidInput(TestFindEnrollmentResultBy:
```

```
sources.TestFindEnrollmentResultById.testInvalidInput Time elapsed: 0 s <<< FAILURE!  
java.lang.AssertionError:
```

```
Expectation failure on verify:  
HttpServletResponse.sendError(403 (int), "无录取结果"): expected: 1, actual: 0  
Unexpected method calls:  
HttpServletResponse.getWriter()  
at sources.TestFindEnrollmentResultById.tearDown(TestFindEnrollmentResultById.java:6
```

### 5.2.1. 缺少错误处理

输入不匹配的 `studentid` 时没有返回错误码和错误信息。

参考 `findStudentById.java` 进行修改。

## 6. rejectEnrolledStudent

### 6.1. testInvalidInput

输入不匹配的 `studentId` 时没有返回错误码和错误信息。

参考 `findStudentById.java` 进行修改。

---

```
Test set: sources.TestRejectEnrolledStudent
```

---

```
Tests run: 2, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 0.425 s <<< FAILURE! - :  
sources.TestRejectEnrolledStudent.testInvalidInput Time elapsed: 0.152 s <<< FAILURE!  
java.lang.AssertionError:
```

```
Expectation failure on verify:  
HttpServletResponse.sendError(403 (int), "提交错误, 请刷新后重试"): expected: 1, actual  
at sources.TestRejectEnrolledStudent.tearDown(TestRejectEnrolledStudent.java:62)
```

## 7. rejectPlan

### 7.1. testValidInput

#### 7.1.1. 类型转换错误

同 approvePlan.java

List 类型不能通过强制类型转换转为 plan 类型。

```
plan plan0 = (plan) baseDao.executeQuery(sql0,planid);
```

### 7.2. testInvalidInput

#### 7.2.1. 验证方式错误

同 approvePlan.java

类型转换错误，以下错误处理的验证方式 planid0 ==null 也失效。

```
if(planid0 ==null){
    System.out.println("没进行批量退选");
}else {

    for (int i = 0; i < planid0.size(); i++) {
        planid = Integer.parseInt(planid0.getString(i));
        String sql0="select * from plan where planid=?";
        plan plan0 = (plan) baseDao.executeQuery(sql0,planid);
        if(plan0==null){
            response.sendError(403, "提交失败，请刷新后重试");
        }
        else {
            String sql = "update plan set is_approved=-1 where planid=?";
            baseDao.executeUpdate(sql, planid);
        }
    }
}
```

## 8. submitVolunteer

### 8.1. testValidInput



未按照接口文件读取输入

文档输入为

```
{
  "volunteerNo": 1,
  "batch": 1,
  "isAdjust": 1,
  "studentId": 1,
  "universityId": 1,
  "classId": 5,
  "type": "统招"
}
```

读取输入为

```
// 读取请求内容
int volunteerid= Integer.parseInt(request.getParameter("volunteerid").toString());
int volunteerno=Integer.parseInt(request.getParameter("volunteerno").toString());
int batch=Integer.parseInt(request.getParameter("batch").toString());
int is_adjust=Integer.parseInt(request.getParameter("is_adjust").toString());
int studentid=Integer.parseInt(request.getParameter("studentid").toString());
int universityid=Integer.parseInt(request.getParameter("universityid").toString());
int classid=Integer.parseInt(request.getParameter("classid").toString());
String type=request.getParameter("type").toString();
```

## 8.2. testInvalidInput

# 9. findApprovedStudent

## 9.1. testValidInput

```
java.lang.ClassCastException: com.alibaba.fastjson.JSONObject cannot be cast to com.alil
at sources.TestFindApprovedStudent.testValidInput(TestFindApprovedStudent.java:88)
```

### 9.1.1. 文档错误

文档中原本标注的输入描述为学校编号，字段应为 `universityId`，但是错写为了 `studentId`，因此引发了误解。

目前接口测试通过，但是测试逻辑是通过学生id查询的，而不是通过学校id查询的，需要修改为和 `findEnrolledStudent.java` 类似的功能，查找输入的 `universityId` 在录取表中 `is_approved` 字段为1的学生信息。

返回类型应为JSONArray。