

- 1 功能概述
- 2 使用演示
- 3 实现介绍
  - 3.1 PLVLoginHiClassActivity
  - 3.2 PLVHCRoleSelectFragment
  - 3.3 PLVHCTeacherLoginFragment
  - 3.4 PLVHCTeacherCompanyFragment
  - 3.5 PLVHCLessonSelectFragment
  - 3.6 PLVHCStudentLoginFragment
  - 3.7 PLVHCStudentVerifyFragment

## 1 功能概述

互动学堂场景登录功能的界面实现为 **PLVLoginHiClassActivity**

互动学堂场景下定义的 讲师登录、学生登录 的 共用界面。

讲师登录流程：

选择老师角色->输入账号、密码，登录验证->选择公司，登录验证->验证成功，获取课节列表页面->选择其中一个课节，进入上课页。

当讲师重复出现在多个公司时，则讲师需要选择公司，否则不需要。

学生使用课节号登录流程：

选择学生角色->输入课节号，课节验证->输入名称+/验证/学生码，登录验证->验证成功，进入上课页。

学生使用课程号登录流程：

选择学生角色->输入课程号，课程验证->输入名称+/验证/学生码，登录验证->验证成功，获取课节列表页面->选择其中一个课节，进入上课页。

## 2 使用演示

因为整个登录系统有多个流程，每个流程的UI交互都不一样，因此在demo中使用一个 Activity+ViewPager+多个Fragment实现。以下示例讲师登录验证的方式，代码如下：

```

//讲师登录验证示例
loginManager.loginHiClassTeacher(
    areaCode,
    account,
    password,
    null,
    null,
    new IPLVSceneLoginManager.OnLoginListener<PLVHCTeacherLoginVO>() {
        @Override
        public void onLoginSuccess(PLVHCTeacherLoginVO result) {
            if (result == null
                || result.getStatus() == null
                || loginViewModel == null
                || getContext() == null) {
                return;
            }

            PLVHCTeacherLoginAccountVO accountVO = new PLVHCTeacherLoginAccountVO();
            accountVO.setAreaCode(areaCode);
            accountVO.setAccount(account);
            accountVO.setPassword(password);
            loginViewModel.getTeacherLoginAccountLiveData().setValue(accountVO);

            saveLastLoginAccount(areaCode, account);
            saveLastLoginPassword(plvhcLoginTeacherRememberPasswordCb.isChecked(), password);
            saveAgreeContract(true);

            if (result.getStatus() == PLVHCTeacherLoginVO.STATUS_LOGIN_SELECT_COMPANY) {
                List<PLVHCLoginCompanyVO> loginCompanyVOList = new ArrayList<>();
                for (PLVHCTeacherLoginResultVO.CompanyVO companyVO : result.getCompanyList()) {
                    loginCompanyVOList.add(PLVHCLoginCompanyVO.fromLoginResult(companyVO));
                }
                loginViewModel.getCompanyListLiveData().setValue(loginCompanyVOList);
                KeyboardUtils.hideSoftInput(plvhcLoginTeacherPasswordEt);
                //进入选择公司页
                PLVHCLoginFragmentManager.getInstance().addLast(PLVHCLoginFragmentManager.FRAG_TEACHER_COMPANY);
            } else if (result.getStatus() == PLVHCTeacherLoginVO.STATUS_LOGIN_SUCCESS) {
                if (result.getSuccessData() != null) {
                    PLVHCLoginDataVO loginTokenVO = new PLVHCLoginDataVO();
                    loginTokenVO.setRole(PLVSocketUserConstant.USER_TYPE_TEACHER);
                    loginTokenVO.setToken(result.getSuccessData().getToken());
                    loginTokenVO.setNickname(result.getSuccessData().getNickname());
                    loginTokenVO.setViewerId(result.getSuccessData().getViewerId());
                    loginViewModel.setLoginDataVOWithSave(loginTokenVO, timestamp);

                    List<PLVHCLoginLessonVO> loginLessonVOList = listMap(result.getSuccessData().getLessonList(),
                        new PLVSugarUtil.Function<PLVHCTeacherLoginResultVO.DataVO.LessonVO, PLVHCLoginLessonVO>() {
                            @Override
                            public PLVHCLoginLessonVO apply(PLVHCTeacherLoginResultVO.DataVO.LessonVO lessonVO) {
                                return PLVHCLoginLessonVO.fromTeacherLoginResultLessonVO(lessonVO);
                            }
                        });
                    loginViewModel.getLoginLessonListLiveData().setValue(loginLessonVOList);
                }
            }
            KeyboardUtils.hideSoftInput(plvhcLoginTeacherPasswordEt);

```

```

        // 进入选择课节页
        PLVHCLoginFragmentManager.getInstance().addLast(PLVHCLoginFragmentManager.FRAG_LESSON_SELECT);
    } else {
        PLVHCToast.Builder.context(getContext())
            .setText("登录状态错误: " + result.getStatus())
            .build().show();
    }
}

@Override
public void onLoginFailed(String msg, Throwable throwable) {
    if (getContext() == null) {
        return;
    }
    PLVHCToast.Builder.context(getContext())
        .setText(msg)
        .build().show();
}
}
);

```

上述方法的具体用例可以在 `PLVHCTeacherLoginFragment` 类中找到的。

## 3 实现介绍

登录的整个流程，其实就是确认和获取用户类型、用户Id、用户昵称、登录token、课程号、课节Id的过程。

### 3.1 PLVLoginHiClassActivity

互动学堂登录页面Activity，布局里定义了一个ViewPager，用于添加登录流程中所出现的Fragment。

### 3.2 PLVHCRoleSelectFragment

互动学堂登录 - 角色选择Fragment，用于选择讲师/学生角色。

### 3.3 PLVHCTeacherLoginFragment

互动学堂登录 - 讲师登录Fragment，提供输入讲师的账号、密码和登录验证的UI交互。

### 3.4 PLVHCTeacherCompanyFragment

互动学堂登录 - 讲师公司选择Fragment，提供选择讲师所在公司的UI交互。当一个讲师绑定了多个公司时，登录验证成功后就会先出现这个Fragment。

### 3.5 PLVHCLessonSelectFragment

互动学堂登录 - 课节选择Fragment，该页面会展示一个可以上课的课节列表，可以从该列表选择一个课节进行上课。

### **3.6 PLVHCStudentLoginFragment**

互动学堂登录 - 学生登录Fragment，提供输入课程号/课节号和信息验证的UI交互。

### **3.7 PLVHCStudentVerifyFragment**

互动学堂登录 - 学生验证Fragemnt，提供输入学生名称+/验证/学生码和观看条件验证的UI交互。