

得明健身 APP 软件代码（前后 30 页）

版本号：V1.0

```
package com.tm.dmkeep.ui.activitys;

import android.app.Activity;
import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;
import android.text.Editable;
import android.text.SpannableString;
import android.text.SpannableStringBuilder;
import android.text.Spanned;
import android.text.TextWatcher;
import android.text.method.LinkMovementMethod;
import android.text.style.ClickableSpan;
import android.text.style.ForegroundColorSpan;
import android.view.View;
import android.view.Window;
import androidx.annotation.NonNull;
import androidx.lifecycle.Observer;
import com.lzy.okgo.model.Response;
import com.mob.MobSDK;
import com.mob.pushsdk.MobPush;
import com.mob.secverify.CustomUIRegister;
import com.mob.secverify.CustomViewClickListener;
import com.mob.secverify.SecVerify;
import com.mob.secverify.VerifyCallback;
import com.mob.secverify.common.exception.VerifyException;
import com.mob.secverify.datatype.VerifyResult;
import com.netting.baselibrary.base.BaseActivity;
import com.netting.baselibrary.okgoutils.BaseHttpConfig;
import com.netting.baselibrary.okgoutils.HttpLibResultModel;
import com.netting.baselibrary.okgoutils.OkGoCallback;
import com.netting.baselibrary.ui.activitys.WebActivity;
import com.netting.baselibrary.ui.dialog.LoadingDialog;
import com.netting.baselibrary.utils.ActivityUtils;
import com.netting.baselibrary.utils.LiveDataBus;
import com.netting.baselibrary.utils.LogUtils;
import com.netting.baselibrary.utils.MmkvManager;
import com.netting.baselibrary.utils.ToastUtils;
import com.tencent.connect.common.Constants;
import com.tencent.tauth.Tencent;
import com.tm.dmkeep.R;
import com.tm.dmkeep.databinding.ActivityLoginRegisterBinding;
import com.tm.dmkeep.http.HttpConfig;
import com.tm.dmkeep.http.HttpResult;
import com.tm.dmkeep.http.mall.MallRequestManager;
import com.tm.dmkeep.http.user.UserRequestManager;
import com.tm.dmkeep.http.user.UserViewModel;
import com.tm.dmkeep.http.user.entity.MobPhoneInfo;
import com.tm.dmkeep.http.user.entity.UserInfo;
import com.tm.dmkeep.managers.BaseUiListener;
import com.tm.dmkeep.managers.LoginManager;
```

```
import com.tm.dmkeep.ui.dialogs.DmDialog;
import com.tm.dmkeep.utils.CustomizeUtils;
import com.tm.dmkeep.utils.VerifyUtils;
import static com.tm.dmkeep.http.HttpConfig.getUserInfo;
import static com.tm.dmkeep.utils.CustomizeUtils.customizeUi;

/**
 * 时间：2022/4/26 11:29
 * 作者：庞宇锋
 * 功能：账号登录注册
 */
public class LoginRegisterActivity extends BaseActivity<ActivityLoginRegisterBinding> {

    @Override
    protected int getContentViewId() {
        return R.layout.activity_login_register;
    }

    private LoadingDialog loadingDialog;

    @Override
    protected void initView() {
        String userName = MmkvManager.getSettings().getString("Account", "");
        String userPassword = MmkvManager.getSettings().getString("PassWord", "");
        binding.llEdit.etLoginUserName.setText(userName);
        binding.llEdit.etLoginUserPassword.setText(userPassword);
    }

    /**
     * 登录
     */
    binding.tvLoginRegister.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            loginPassword();
        }
    });

    String tr = "我已阅读并同意";
    String tr1 = "使用条款";
    String tr2 = "和";
    String tr3 = "隐私政策";
    String text = tr + tr1 + tr2 + tr3;
    SpannableString textSpanned1 = new SpannableString(text);
    SpannableStringBuilder spannableStringBuilder = new SpannableStringBuilder();
    spannableStringBuilder.append(textSpanned1);
    ClickableSpan clickableSpan1 = new ClickableSpan() {
        @Override
        public void onClick(@NonNull View widget) {
            Bundle bundle = new Bundle();
            bundle.putString(WebActivity.TITLE_WEB, "");
            bundle.putString(WebActivity.URL_WEB, HttpConfig.registerPro);
            ActivityUtils.startActivityForBundleData(LoginRegisterActivity.this, WebActivity.class, bundle);
        }
    };

    ClickableSpan clickableSpan2 = new ClickableSpan() {
        @Override
```

```

        public void onClick(@NonNull View widget) {
            Bundle bundle = new Bundle();
            bundle.putString(WebActivity.TITLE_WEB, "");
            bundle.putString(WebActivity.URL_WEB, HttpConfig.privacyPro);
            ActivityUtils.startActivityForBundleData(LoginRegisterActivity.this, WebActivity.class, bundle);
        }
    };

    int str1 = tr.length();
    int end1 = tr.length() + tr1.length();
    int str2 = tr.length() + tr1.length() + tr2.length();
    int end2 = str2 + tr3.length();

    spannableStringBuilder.setSpan(clickableSpan1, str1, end1, Spanned.SPAN_EXCLUSIVE_INCLUSIVE);
    spannableStringBuilder.setSpan(clickableSpan2, str2, end2, Spanned.SPAN_EXCLUSIVE_INCLUSIVE);
    spannableStringBuilder.setSpan(new ForegroundColorSpan(Color.parseColor("#2859FF")), str1, end1,
Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
    spannableStringBuilder.setSpan(new ForegroundColorSpan(Color.parseColor("#2859FF")), str2, end2,
Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);

    binding.tvLoginXy.setMovementMethod(LinkMovementMethod.getInstance());
    binding.tvLoginXy.setText(spannableStringBuilder);
    binding.boxXy.setChecked(true);
    binding.ivFin.setOnClickListener(v -> {
        finish();
    });

    binding.llEdit.etLoginUserName.addTextChangedListener(new TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence s, int start, int count, int after) {
        }
        @Override
        public void onTextChanged(CharSequence s, int start, int before, int count) {
            showLoginButton();
        }
        @Override
        public void afterTextChanged(Editable s) {
        }
    });

    binding.llEdit.etLoginUserPassword.addTextChangedListener(new TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence s, int start, int count, int after) {
        }
        @Override
        public void onTextChanged(CharSequence s, int start, int before, int count) {
            showLoginButton();
        }
        @Override
        public void afterTextChanged(Editable s) {
        }
    });

    binding.llEdit.etLoginUserName.setOnFocusChangeListener(new View.OnFocusChangeListener() {
        @Override
        public void onFocusChange(View v, boolean hasFocus) {

```

```

        showEtTitle("0");
    }
});
binding.llEdit.etLoginUserPassword.setOnFocusChangeListener(new View.OnFocusChangeListener() {
    @Override
    public void onFocusChange(View v, boolean hasFocus) {
        showEtTitle("1");
    }
});
//忘记密码
binding.tvLoginForget.setOnClickListener(v -> {
    ActivityUtils.startActivity(LoginRegisterActivity.this, ForgetPassword1Activity.class);
});
binding.ivLoginQq.setOnClickListener(v -> {
    showLoadDialog();
    LoginManager.QQLogin(LoginRegisterActivity.this);
});
binding.ivLoginWx.setOnClickListener(v -> {
    showLoadDialog();
    LoginManager.WxLogin();
});
binding.ivLoginMobile.setOnClickListener(v -> {
    LoginManager.phoneLogin(LoginRegisterActivity.this);
});
showLoginButton();
}
public void showLoadDialog() {
    loadingDialog = new LoadingDialog(LoginRegisterActivity.this);
    loadingDialog.timingShow();
    loadingDialog.setContentText("正在登录");
}
/**
 * 登录是否可点击
 */
public void showLoginButton() {
    if (binding.llEdit.etLoginUserName.getText().toString().length() > 0 && binding.llEdit.etLoginUserPassword.getText().toString().length() >=
3) {
        binding.tvLoginRegister.setBackgroundResource(R.drawable.bg_login_yes);
    } else {
        binding.tvLoginRegister.setBackgroundResource(R.drawable.bg_login_no);
    }
}
public void showEtTitle(String type) {
    binding.llEdit.tvTitleAccount.setVisibility(View.INVISIBLE);
    binding.llEdit.tvTitlePassword.setVisibility(View.INVISIBLE);
    LogUtils.e("***** + type");
    switch (type) {
        case "0":
            binding.llEdit.tvTitleAccount.setVisibility(View.VISIBLE);
            LogUtils.e("账号获取焦点");

```

```

        break;
    case "1":
        LogUtils.e("密码获取焦点");
        binding.llEdit.tvTitlePassword.setVisibility(View.VISIBLE);
        break;
    }
}

@Override
protected void initData() {
    LiveDataBus.get().with(UserViewModel.class.getName(), UserViewModel.class).observe(this, new Observer<UserViewModel>() {
        @Override
        public void onChanged(UserViewModel userViewModel) {
            LogUtils.e("用户登录信息" + LoginManager.isLogin());
            LogUtils.e("用户登录信息*****" + userViewModel.getUserInfo());
            switch (userViewModel.getResponse_type()) {
                case 5101://未注册用户
                    if (loadingDialog != null) {
                        loadingDialog.dismiss();
                    }
                    loadingDialog = null;
                    UserViewModel.getInstance().setResponse_type(0);
                    break;
                default:
                    if (LoginManager.isLogin()) {
                        ActivityUtils.startActivityAndFinish(LoginRegisterActivity.this, MainActivity.class);
                        finish();
                    }
                    break;
            }
        }
    });
}

/**
 * 账号密码登录
 */
public void loginPassword() {
    if (!binding.boxXy.isChecked()) {
        ToastUtils.showToastShort(getApplicationContext(), getResources().getString(R.string.appPrivateText));
        return;
    }
    String userName = binding.llEdit.etLoginUserName.getText().toString().trim();
    if (!VerifyUtils.isUserName(userName)) {
        ToastUtils.showToastLong(getApplicationContext(), "请输入论坛昵称/手机号");
        return;
    }
    String userPassword = binding.llEdit.etLoginUserPassword.getText().toString().trim();
    if (!VerifyUtils.isUserPassword(userPassword)) {
        ToastUtils.showToastLong(getApplicationContext(), "请输入大于 3 位密码");
        return;
    }
}

```

```

        UserRequestManager.getInstance().userLogin(LoginRegisterActivity.this, userName, userPassword, new HttpResult<UserInfo>() {
            @Override
            public void success(UserInfo obj) {
                if (obj.isReg()) {
                    String msg = binding.llEdit.etLoginUserName.getText().toString();
                    new DmDialog.Builder(LoginRegisterActivity.this)
                        .setContent("是否注册用户为「" + msg + "」的账号？")
                        .setCallBack(new DmDialog.CallBack() {
                            @Override
                            public void cancel() {
                            }
                            @Override
                            public void next() {
                                UserRequestManager.getInstance().setConfigStatus(obj.getUserId(), new
                                OkGoCallback<HttpLibResultModel<Boolean>>() {
                                    @Override
                                    public void onSur(Response<HttpLibResultModel<Boolean>> response) {
                                        MmkvManager.getSettings().putString("Account", userName);
                                        MmkvManager.getSettings().putString("PassWord", userPassword);
                                        BaseHttpConfig.getInstance().getHttpHeaders().put(UserViewModel.TOKEN,
                                        obj.getHeaderToken());

                                        UserViewModel.getInstance().getUserInfo().setToken(obj.getToken());
                                        UserRequestManager.getInstance().getUserInfo(String.valueOf(3));
                                    }
                                    @Override
                                    public void onErr(String msg) {
                                    }
                                });
                            }
                        }).getDmDialog().show();
                } else {
                    BaseHttpConfig.getInstance().getHttpHeaders().put(UserViewModel.TOKEN, obj.getHeaderToken());
                    UserViewModel.getInstance().getUserInfo().setToken(obj.getToken());
                    UserRequestManager.getInstance().getUserInfo(String.valueOf(3));
                    MmkvManager.getSettings().putString("Account", userName);
                    MmkvManager.getSettings().putString("PassWord", userPassword);
                }
            }
            @Override
            public void err(String msg) {
            }
        });
    }
    /**
     * 注册账号
     */
    public void userRegister() {
        String userName = binding.llEdit.etLoginUserName.getText().toString().trim();
        String userPassword = binding.llEdit.etLoginUserPassword.getText().toString().trim();
    }

```

```
@Override
protected void onDestroy() {
    super.onDestroy();
    if (loadingDialog != null) {
        loadingDialog.dismiss();
    }
    loadingDialog = null;
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    LogUtils.e("-->onActivityResult " + requestCode + " resultCode=" + resultCode);
    switch (requestCode) {
        case Constants.REQUEST_LOGIN:
        case Constants.REQUEST_APPBAR: {
            Tencent.onActivityResultData(requestCode, resultCode, data, new BaseUiListener());
        }
        break;
        default:
            break;
    }
    super.onActivityResult(requestCode, resultCode, data);
}

}

package com.tm.dmkeep.ui.activitys;
import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.os.Build;
import android.provider.Settings;
import android.util.Log;
import android.view.KeyEvent;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.lifecycle.Observer;
import androidx.viewpager.widget.ViewPager;
import com.flyco.tablayout.listener.CustomTabEntity;
import com.flyco.tablayout.listener.OnTabSelectListener;
import com.google.gson.Gson;
import com.lzy.okgo.model.Response;
import com.mob.pushsdk.MobPush;
import com.mob.pushsdk.MobPushCustomMessage;
import com.mob.pushsdk.MobPushNotifyMessage;
import com.mob.pushsdk.MobPushReceiver;
import com.netting.baselibrary.base.BaseActivity;
import com.netting.baselibrary.okgoutils.HttpLibResultModel;
import com.netting.baselibrary.okgoutils.OkGoCallback;
import com.netting.baselibrary.utils.CopyLinkTextHelper;
import com.netting.baselibrary.utils.LiveDataBus;
import com.netting.baselibrary.utils.LogUtils;
import com.netting.baselibrary.utils.Super;
```



```
import com.netting.baselibrary.utils.ToastUtils;
import com.netting.baselibrary.viewmodel.SetupViewModel;
import com.tm.dmkeep.R;
import com.tm.dmkeep.databinding.ActivityMainBinding;
import com.tm.dmkeep.http.HttpResult;
import com.tm.dmkeep.http.app.AppInfo;
import com.tm.dmkeep.http.app.AppManager;
import com.tm.dmkeep.http.home.HomeRequestManager;
import com.tm.dmkeep.http.home.HomeViewModel;
import com.tm.dmkeep.http.home.entity.HomeInfo;
import com.tm.dmkeep.http.home.entity.PushMessageInfo;
import com.tm.dmkeep.http.live.LiveRequestManager;
import com.tm.dmkeep.http.user.UserViewModel;
import com.tm.dmkeep.managers.LoginManager;
import com.tm.dmkeep.service.DownloadService;
import com.tm.dmkeep.ui.adapters.ViewPagerAdapter;
import com.tm.dmkeep.ui.dialogs.DmDialog;
import com.tm.dmkeep.ui.dialogs.LiveDialog;
import com.tm.dmkeep.ui.fragments.CommunityFragment;
import com.tm.dmkeep.ui.fragments.CourseFragment;
import com.tm.dmkeep.ui.fragments.HomeFragment;
import com.tm.dmkeep.ui.fragments.MyFragment;
import com.tm.dmkeep.ui.fragments.ShoppingFragment;
import com.tm.dmkeep.ui.views.live.CustomLiveShareView;
import com.tm.dmkeep.websocket.WebClient;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import static com.netting.baselibrary.utils.ContextUtil.showToastShort;

/**
 * 首页
 */
public class MainActivity extends BaseActivity<ActivityMainBinding> {
    private List<Fragment> fragmentList = new ArrayList<>();
    private List<CustomTabEntity> customTabEntityList = new ArrayList<>();
    private ViewPagerAdapter viewPagerAdapter;
    public static final String SHOW_INDEX = "show_index";

    @Override
    protected int getContentViewId() {
        return R.layout.activity_main;
    }

    @Override
    protected void initView() {
        initFragment();
        int index = getIntent().getIntExtra(SHOW_INDEX, 0);
        binding.vpViewpager.setCurrentItem(index, true);
    }

    private boolean is = false;

    @Override
    protected void initData() {
```

```
/**
 * 检查版本更新
 */
AppManager.getInstance().getAppVersion(new OkGoCallback<HttpLibResultModel<AppInfo>>() {
    @Override
    public void onSur(Response<HttpLibResultModel<AppInfo>> response) {
        showNoticeDialog(response.body().getData());
    }
    @Override
    public void onErr(String msg) {
    }
});

//获取缓存信息
HomeViewModel.getInstance().getHomeInfo();

//请求首页数据
HomeRequestManager.getInstance().getHomeList(new HttpResult<HomeInfo>() {
    @Override
    public void success(HomelInfo obj) {
        LiveDataBus.get().with(HomeViewModel.class.getName()).postValue(HomeViewModel.getInstance());
    }
    @Override
    public void err(String msg) {
    }
});

LiveDataBus.get().with(UserViewModel.class.getName(), UserViewModel.class).observe(this, new Observer<UserViewModel>() {
    @Override
    public void onChanged(UserViewModel userViewModel) {
        if (LoginManager.isLogin()) {
            //直播连接
            LiveRequestManager.getInstance().initLive(MainActivity.this);
            MobPush.setAlias(userViewModel.getUserInfo().getUserid());
            //websocket
            WebClient.initWebSocket(MainActivity.this, 7918);
        }
    }
});

//监听消息推送
MobPush.addPushReceiver(new MobPushReceiver() {
    //自定义消息
    @Override
    public void onCustomMessageReceive(Context context, MobPushCustomMessage message) {
        //接收到自定义消息（透传消息）
        String id = message.getMessageId();//获取任务 ID
        String content = message.getContent();//获取推送内容
        LogUtils.e("消息 id" + id + "消息内容" + content);
    }
    //通知消息
    @Override
    public void onNotifyMessageReceive(Context context, MobPushNotifyMessage message) {
        String notifyId = message.getMobNotifyId();//获取消息 ID
    }
});
```

```

String id = message.getMessageId();//获取任务 ID
String title = message.getTitle();//获取推送标题
String content = message.getContent();//获取推送内容
HashMap map = message.getExtrasMap();
if (map.size() > 0 && map.get("pushData") != null) {
    PushMessageInfo pushMessageInfo = new Gson().fromJson(map.get("pushData").toString(), PushMessageInfo.class);
    switch (pushMessageInfo.getType()) {
        case "1"://开始直播
        case "2"://结束直播
            HomeRequestManager.getInstance().getClassInfo(null);
            break;
    }
}
LogUtils.e("收到通知" + map.get("pushData"));
LogUtils.e("通知 id : " + notifyId + "***消息 id : " + id + "***消息标题:" + title + "****消息内容:" + content);
}

@Override
public void onNotifyMessageOpenedReceive(Context context, MobPushNotifyMessage message) {
    //通知被点击事件
    String notifyId = message.getMobNotifyId();//获取消息 ID
    String id = message.getMessageId();//获取任务 ID
    String title = message.getTitle();//获取推送标题
    String content = message.getContent();//获取推送内容
    HashMap map = message.getExtrasMap();
    if (map.size() > 0 && map.get("pushData") != null) {
        PushMessageInfo pushMessageInfo = new Gson().fromJson(map.get("pushData").toString(), PushMessageInfo.class);
        switch (pushMessageInfo.getType()) {
            case "1"://开始直播
                if (pushMessageInfo.getAnchorId().equals(UserViewModel.getInstance().getUserInfo().getUserid())) {
                    LiveRequestManager.getInstance().onStartLive(MainActivity.this, "", pushMessageInfo.getLiveId());
                } else {
                    LiveRequestManager.getInstance().onViewLive(MainActivity.this, pushMessageInfo.getLiveId());
                }
            case "2"://结束直播
                binding.tabLayout.setCurrentTab(1);
                break;
        }
    }
    LogUtils.e("点击消息" + map.get("pushData"));
    LogUtils.e("点击了通知事假, 通知 id : " + notifyId + "/n 消息 id : " + id + "/n 消息标题:" + title + "/n 消息内容:" + content);
}

@Override
public void onTagsCallback(Context context, String[] strings, int i, int i1) {
}

@Override
public void onAliasCallback(Context context, String s, int i, int i1) {
    LogUtils.e("***" + s + "i" + i);
}
});

```

```
MobPush.getAlias();

//websocket
WebClient.initWebSocket(this, 7918);
LoginManager.preVerify();
}

/**
 * 初始化页面
 */
public void initFragment() {
    fragmentList.add(new HomeFragment());
    fragmentList.add(new CommunityFragment());
    fragmentList.add(new CourseFragment());
    fragmentList.add(new ShoppingFragment());
    fragmentList.add(new MyFragment());
    for (int i = 0; i < fragmentList.size(); i++) {
        int finalI = i;
        customTabEntityList.add(new CustomTabEntity() {
            @Override
            public String getTabTitle() {
                String title = "";
                switch (finalI) {
                    case 0:
                        title = "首页";
                        break;
                    case 1:
                        title = "社区";
                        break;
                    case 2:
                        title = "课程";
                        break;
                    case 3:
                        title = "商城";
                        break;
                    case 4:
                        title = "我的";
                        break;
                }
                return title;
            }
        });
    }
    @Override
    public int getTabSelectedIcon() {
        int res = 0;
        switch (finalI) {
            case 0:
                res = R.mipmap.icon_home_yes;
                break;
            case 1:
                res = R.mipmap.icon_home_community_yes;
                break;
            case 2:
                res = R.mipmap.icon_home_course_yes;
                break;
            case 3:
                res = R.mipmap.icon_home_shop_yes;
                break;
            case 4:
                res = R.mipmap.icon_home_my_yes;
                break;
        }
        return res;
    }
}
```

```

        res = R.mipmap.icon_course_yes;
        break;
    case 3:
        res = R.mipmap.icon_shopping_yes;
        break;
    case 4:
        res = R.mipmap.icon_my_yes;
        break;
    }
    return res;
}

@Override
public int getTabUnselectedIcon() {
    int res = 0;
    switch (finalI) {
        case 0:
            res = R.mipmap.icon_home_no;
            break;
        case 1:
            res = R.mipmap.icon_home_community_no;
            break;
        case 2:
            res = R.mipmap.icon_course_no;
            break;
        case 3:
            res = R.mipmap.icon_shopping_no;
            break;
        case 4:
            res = R.mipmap.icon_my_no;
            break;
    }
    return res;
}

});
}

binding.tabLayout.setTabData((ArrayList<CustomTabEntity>) customTabEntityList);
binding.tabLayout.setOnTabSelectListener(new OnTabSelectListener() {
    @Override
    public void onTabSelect(int position) {
        binding.vpViewPager.setCurrentItem(position, false);
    }
    @Override
    public void onTabReselect(int position) {
    }
});

viewPagerAdapter = new ViewPagerAdapter(getSupportFragmentManager());
viewPagerAdapter.setFragmentList(fragmentList);
binding.vpViewPager.addOnPageChangeListener(new ViewPager.OnPageChangeListener() {
    @Override
    public void onPageScrolled(int position, float positionOffset, int positionOffsetPixels) {

```

```

    }

    @Override
    public void onPageSelected(int position) {
        binding.tabLay.setCurrentTab(position);
    }

    @Override
    public void onPageScrollStateChanged(int state) {
    }

});

binding.vpViewpager.setAdapter(viewPagerAdapter);
binding.vpViewpager.setCurrentItem(0, false);
binding.vpViewpager.setOffscreenPageLimit(4);

LiveDataBus.get().with(SetupViewModel.class.getName(), SetupViewModel.class).observe(this, new Observer<SetupViewModel>() {

    @Override
    public void onChanged(SetupViewModel setupViewModel) {
        if (setupViewModel.isToken()) {
            LoginManager.logout();
            if (dmDialog1 == null) {
                dmDialog1 = new DmDialog.Builder(MainActivity.this)
                    .setLeftText("")
                    .setRightText("")
                    .setContent(setupViewModel.getTokenMessage())
                    .setCallBack(new DmDialog.CallBack() {
                        @Override
                        public void cancel() {
                            LoginManager.login(MainActivity.this);
                        }
                        @Override
                        public void next() {
                            LoginManager.login(MainActivity.this);
                        }
                    })
                    .getDmDialog();
            }
            dmDialog1.show();
        }
    }

});

}

@Override
protected void onDestroy() {
    super.onDestroy();
    if (dmDialog1 != null) {
        dmDialog1.dismiss();
        dmDialog1 = null;
    }
}

DmDialog dmDialog1;

@Override
protected void onResume() {
    super.onResume();

```

```

getWindow().getDecorView().post(new Runnable() {
    @Override
    public void run() {
        String copyText = CopyLinkTextHelper.getInstance(getApplicationContext()).getCopyString();
        if (copyText != null && !copyText.equals("")) {
            if (copyText.contains(CustomLiveShareView.content) || copyText.contains(CustomLiveShareView.endText)) {
                int start = 0;
                start = copyText.indexOf("就等你来^^^") + 7;
                String id = copyText.substring(start, copyText.length() - CustomLiveShareView.endText.length());
                LogUtils.e("获取到直播信息 " + id);
                new LiveDialog.Builder(MainActivity.this)
                    .setId(id)
                    .getDmDialog().show();
            }
        }
    }
});
}

public int getPosition(String str, int ciShu) {
    int number = 0;
    char arr[] = str.toCharArray();
    for (int i = 0; i < arr.length; i++) {
        if (arr[i] == '【') {
            number++;
        }
        if (number == ciShu) {
            return i + 1;
        }
    }
    return 1;
}

/**
 * 显示更新对话框
 *
 * @param version_info
 */
DmDialog dmDialog;

private void showNoticeDialog(AppInfo appInfo) {
    int code = Integer.valueOf(appInfo.getVersionCode());
    int mycode = Super.getVersionCode(getApplicationContext());
    appInfo.setForceUpdate(0);
    if (code <= mycode) {
        return;
    }
    if (dmDialog == null) {
        String left = "";
        if (appInfo.getForceUpdate() != 1) {
            left = "以后更新";
        } else {
            left = "";
        }
    }
}

```

```

    }

    dmDialog = new DmDialog.Builder(MainActivity.this)
        .setContent(applInfo.getVersionInfo())
        .setTitle("有新版本")
        .setLeftText(left)
        .setRightText("立即更新")
        .setCallBack(new DmDialog.CallBack() {
            @Override
            public void cancel() {
            }
            @Override
            public void next() {
                apkurl = applInfo.getDownloadUrl();
                ToastUtils.showToastLong(getApplicationContext(), "正在后台下载，请稍后！");
                UpdateManager(apkurl);
            }
        })
        .getDmDialog();

    if (applInfo.getForceUpdate() != 1) {
        dmDialog.setCancelable(false);
    } else {
        dmDialog.setCancelable(true);
    }

    dmDialog.show();
} else {
    dmDialog.show();
}
}

private String apkurl = "";
//判断版本号，并进行更新
public void UpdateManager(String apkurl) {
    if (apkurl == null || apkurl.equals("")) {
        showToastShort("未获取到下载地址，请从官网下载");
        return;
    }

    if (Build.VERSION.SDK_INT >= 26) {
        boolean b = getPackageManager().canRequestPackageInstalls();
        if (b && apkurl != "") {
            Intent intent = new Intent(MainActivity.this, DownloadService.class);
            intent.putExtra("LoadUrl", apkurl);
            startService(intent);
        } else {
            Uri packageURI = Uri.parse("package:" + MainActivity.this.getPackageName());
            Intent intent = new Intent(Settings.ACTION_MANAGE_UNKNOWN_APP_SOURCES, packageURI);
            startActivityForResult(intent, INSTALL_PERMISS_CODE);
        }
    } else {
        Intent intent = new Intent(getApplicationContext(), DownloadService.class);
        intent.putExtra("LoadUrl", apkurl);
        startService(intent);
    }
}

```



```
}

private final int INSTALL_PERMISS_CODE = 100;

@Override

protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {

    super.onActivityResult(requestCode, resultCode, data);

    if (resultCode == RESULT_OK && requestCode == INSTALL_PERMISS_CODE) { //判断是否已获取权限

        Log.i("asdf", "已获取权限");

        Intent intent = new Intent(MainActivity.this, DownLoadService.class);

        intent.putExtra("LoadUrl", apkurl);

        startService(intent);

    }

}

private long exitTime = 2000;

@Override

public boolean onKeyDown(int keyCode, KeyEvent event) {

    if (keyCode == KeyEvent.KEYCODE_BACK) {

        if (System.currentTimeMillis() - exitTime > 2000) {

            ToastUtils.showToastShort(MainActivity.this, "再按一次退出 APP");

            exitTime = System.currentTimeMillis();

            return false;

        } else {

            return super.onKeyDown(keyCode, event);

        }

    }

    return super.onKeyDown(keyCode, event);

}

}

package com.tm.dmkeep.ui.fragments;

import android.content.Context;

import android.os.Bundle;

import android.view.LayoutInflater;

import android.view.View;

import androidx.annotation.NonNull;

import androidx.lifecycle.Observer;

import com.chad.library.adapter.base.BaseQuickAdapter;

import com.chad.library.adapter.base.listener.OnItemChildClickListener;

import com.chad.library.adapter.base.listener.OnItemClickListener;

import com.makeramen.roundedimageview.RoundedImageView;

import com.netting.baselibrary.base.BaseBindFragment;

import com.netting.baselibrary.utils.ActivityUtils;

import com.netting.baselibrary.utils.LiveDataBus;

import com.scwang.smart.refresh.layout.api.RefreshLayout;

import com.scwang.smart.refresh.layout.listener.OnRefreshListener;

import com.tencent.bugly.beta.Beta;

import com.tencent.bugly.beta.UpgradeInfo;

import com.tm.dmkeep.R;

import com.tm.dmkeep.databinding.FragmentHomeBinding;

import com.tm.dmkeep.http.HttpResult;

import com.tm.dmkeep.http.home.MessageViewModel;

import com.tm.dmkeep.http.home.HomeRequestManager;
```

```
import com.tm.dmkeep.http.home.HomeViewModel;
import com.tm.dmkeep.http.home.entity.EssayInfo;
import com.tm.dmkeep.http.home.entity.HomeInfo;
import com.tm.dmkeep.http.home.entity.MessageInfo;
import com.tm.dmkeep.http.live.StartLiveActivityManager;
import com.tm.dmkeep.ui.activities.EssayDetailsActivity;
import com.tm.dmkeep.ui.activities.EssayListActivity;
import com.tm.dmkeep.ui.activities.EveryDayInfoActivity;
import com.tm.dmkeep.ui.activities.EveryDayListActivity;
import com.tm.dmkeep.ui.activities.LiveListActivity;
import com.tm.dmkeep.ui.activities.MessageActivity;
import com.tm.dmkeep.ui.activities.QuestionDetailsActivity;
import com.tm.dmkeep.ui.activities.QuestionListActivity;
import com.tm.dmkeep.ui.activities.RecruitStudentsActivity;
import com.tm.dmkeep.ui.activities.SearchActivity;
import com.tm.dmkeep.ui.activities.SIDetailsActivity;
import com.tm.dmkeep.ui.activities.SIListActivity;
import com.tm.dmkeep.ui.adapters.EveryDayAdapter;
import com.tm.dmkeep.ui.adapters.HomeVideoAdapter;
import com.tm.dmkeep.ui.adapters.QuestionAdapter;
import com.tm.dmkeep.ui.adapters.RecommendAdapter;
import com.tm.dmkeep.utils.ImageLoadUtils;
import com.zhouwei.mzbanner.holder.MZHolderCreator;
import com.zhouwei.mzbanner.holder.MZViewHolder;

/**
 * 时间：2022/4/11 15:57
 * 作者：庞宇锋
 * 功能：首页
 */
public class HomeFragment extends BaseBindFragment<FragmentHomeBinding> {
    private EveryDayAdapter everyDayAdapter;//每日
    private HomeVideoAdapter homeVideoAdapter;//沙龙
    private RecommendAdapter recommendAdapter;//推荐
    private QuestionAdapter questionAdapter;//榜单
    private String TAB_1_ID = "1485077";//直播
    private String TAB_2_ID = "1484883";//答疑
    private String TAB_3_ID = "142572";//招生
    private String TAB_4_ID = "1410633";//视频
    private String TAB_5_ID = "1129081";//分享

    @Override
    public int getLayout() {
        return R.layout.fragment_home;
    }

    @Override
    protected void initView() {
        //直播
        binding.tvTab1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Bundle bundle = new Bundle();
```

```
        ActivityUtils.startActivityForBundleData(getActivity(), LiveListActivity.class, bundle);
    }
});
//答疑
binding.tvTab2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Bundle bundle = new Bundle();
        bundle.putString(QuestionListActivity.BOARD_Id, String.valueOf(TAB_2_ID));
        ActivityUtils.startActivityForBundleData(getActivity(), QuestionListActivity.class, bundle);
    }
});
//招生
binding.tvTab3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        ActivityUtils.startActivity(getActivity(), RecruitStudentsActivity.class);
    }
});
//视频
binding.tvTab4.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Bundle bundle = new Bundle();
        bundle.putString(EveryDayListActivity.BOARD_Id, String.valueOf(TAB_4_ID));
        bundle.putString(EveryDayListActivity.BOARD_TITLE, "视频");
        ActivityUtils.startActivityForBundleData(getActivity(), EveryDayListActivity.class, bundle);
    }
});
//分享
binding.tvTab5.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Bundle bundle = new Bundle();
        bundle.putString(EssayListActivity.BOARD_Id, String.valueOf(TAB_5_ID));
        bundle.putString(EssayListActivity.BOARD_TITLE, "分享");
        ActivityUtils.startActivityForBundleData(getActivity(), EssayListActivity.class, bundle);
    }
});
binding.refLay.setOnRefreshListener(new OnRefreshListener() {
    @Override
    public void onRefresh(@NonNull RefreshLayout refreshLayout) {
        getData();
    }
});
binding.llLayTitle1.tvTitleItemName.setText("每日跟操练");
binding.llLayTitle1.tvItemGd.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (HomeViewModel.getInstance().getEveryDayInfoList().size() > 0) {
```

```

        long board_id = HomeViewModel.getInstance().getEveryDayInfoList().get(0).getBoardid();
        Bundle bundle = new Bundle();
        bundle.putString(EveryDayListActivity.BOARD_Id, String.valueOf(board_id));
        ActivityUtils.startActivityForBundleData(getActivity(), EveryDayListActivity.class, bundle);
    }
}

});

everyDayAdapter = new EveryDayAdapter(R.layout.item_home_daily, HomeViewModel.getInstance().getEveryDayInfoList());
everyDayAdapter.setOnItemClickListner(new OnItemClickListener() {
    @Override
    public void onItemClick(@NonNull BaseQuickAdapter<?, ?> adapter, @NonNull View view, int position) {
        HomeViewModel.getInstance().setEveryDay(everyDayAdapter.getItem(position));
        ActivityUtils.startActivity(getActivity(), EveryDayInfoActivity.class);
    }
});

binding.rvListEveryDay.setAdapter(everyDayAdapter);
binding.llLayTitle2.tvTitleItemName.setText("精选答疑榜单");
binding.llLayTitle2.tvItemGd.setVisibility(View.GONE);

questionAdapter = new QuestionAdapter(R.layout.item_question_lay, HomeViewModel.getInstance().getQuestionInfoList());
binding.rvListQuestion.setAdapter(questionAdapter);
questionAdapter.setOnItemClickListner(new OnItemClickListener() {
    @Override
    public void onItemClick(@NonNull BaseQuickAdapter adapter, @NonNull View view, int position) {
        HomeViewModel.getInstance().setQuestionEveryDay(null);
        switch (view.getId()) {
            case R.id.iv_title:
                long board_id = questionAdapter.getData().get(position).getId();
                Bundle bundle = new Bundle();
                bundle.putString(QuestionListActivity.BOARD_Id, String.valueOf(board_id));
                bundle.putInt(QuestionListActivity.INDEX, position);
                ActivityUtils.startActivityForBundleData(getActivity(), QuestionListActivity.class, bundle);
                break;
            case R.id.tv_qu_1:
                HomeViewModel.getInstance().setChildrenListBean(questionAdapter.getData().get(position).getChildrenList().get(0));
                ActivityUtils.startActivity(getActivity(), QuestionDetailsActivity.class);
                break;
            case R.id.tv_qu_2:
                HomeViewModel.getInstance().setChildrenListBean(questionAdapter.getData().get(position).getChildrenList().get(1));
                ActivityUtils.startActivity(getActivity(), QuestionDetailsActivity.class);
                break;
            case R.id.tv_qu_3:
                HomeViewModel.getInstance().setChildrenListBean(questionAdapter.getData().get(position).getChildrenList().get(2));
                ActivityUtils.startActivity(getActivity(), QuestionDetailsActivity.class);
                break;
            case R.id.tv_qu_4:
                HomeViewModel.getInstance().setChildrenListBean(questionAdapter.getData().get(position).getChildrenList().get(3));
                ActivityUtils.startActivity(getActivity(), QuestionDetailsActivity.class);
                break;
            case R.id.tv_qu_5:
                HomeViewModel.getInstance().setChildrenListBean(questionAdapter.getData().get(position).getChildrenList().get(4));

```

```

        ActivityUtils.startActivity(getActivity(), QuestionDetailsActivity.class);
        break;
    }
}

});

homeVideoAdapter = new HomeVideoAdapter(R.layout.item_video_lay, HomeViewModel.getInstance().getVideoInfoList());
binding.rvListSl.setAdapter(homeVideoAdapter);
homeVideoAdapter.setOnItemClickListener(new OnItemClickListener() {
    @Override
    public void onItemClick(@NonNull BaseQuickAdapter<?, ?> adapter, @NonNull View view, int position) {
        HomeViewModel.getInstance().setEveryDay(homeVideoAdapter.getItem(position));
        ActivityUtils.startActivity(getActivity(), SldetailsActivity.class);
    }
});

binding.llLayTitle3.tvTitleItemName.setText("沙龙");
binding.llLayTitle3.tvItemGd.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (HomeViewModel.getInstance().getVideoInfoList().size() > 0) {
            long board_id = HomeViewModel.getInstance().getVideoInfoList().get(0).getBoardid();
            Bundle bundle = new Bundle();
            bundle.putString(SlListActivity.BOARD_Id, String.valueOf(board_id));
            ActivityUtils.startActivityForBundleData(getActivity(), SlListActivity.class, bundle);
        }
    }
});

binding.llLayTitle4.tvTitleItemName.setText("为您推荐");
binding.llLayTitle4.tvItemGd.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (HomeViewModel.getInstance().getRecommendList().size() > 0) {
            long board_id = HomeViewModel.getInstance().getRecommendList().get(0).getBoardid();
            Bundle bundle = new Bundle();
            bundle.putString(EssayListActivity.BOARD_Id, String.valueOf(board_id));
            ActivityUtils.startActivityForBundleData(getActivity(), EssayListActivity.class, bundle);
        }
    }
});

recommendAdapter = new RecommendAdapter(HomeViewModel.getInstance().getRecommendList());
binding.rvListTj.setAdapter(recommendAdapter);
recommendAdapter.setOnItemClickListener(new OnItemClickListener() {
    @Override
    public void onItemClick(@NonNull BaseQuickAdapter<?, ?> adapter, @NonNull View view, int position) {
        HomeViewModel.getInstance().setEssayInfo(recommendAdapter.getItem(position));
        ActivityUtils.startActivity(getActivity(), EssayDetailsActivity.class);
    }
});

binding.ivHomeSearch.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

```

```

        ActivityUtils.startActivity(getActivity(), SearchActivity.class);
    }
});
binding.ivHomeMsg.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        ActivityUtils.startActivity(getActivity(), MessageActivity.class);
    }
});
binding.vpBanner.setIndicatorVisible(false);
binding.ivHomeAds.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (essayInfo != null && essayInfo.getId() != 1485101) { //广告
            StartLiveActivityManager.startLive(getActivity(), essayInfo.getTitle(), essayInfo.getType(),
                essayInfo.getAnchorId(), String.valueOf(essayInfo.getLiveId()), String.valueOf(essayInfo.getId()));
        }
    }
});
}

private void loadUpgradeInfo() {
    /***** 获取升级信息 *****/
    UpgradeInfo upgradeInfo = Beta.getUpgradeInfo();
    StringBuilder info = new StringBuilder();
    info.append("id: ").append(upgradeInfo.id).append("\n");
    info.append("标题: ").append(upgradeInfo.title).append("\n");
    info.append("升级说明: ").append(upgradeInfo.newFeature).append("\n");
    info.append("versionCode: ").append(upgradeInfo.versionCode).append("\n");
    info.append("versionName: ").append(upgradeInfo.versionName).append("\n");
    info.append("发布时间: ").append(upgradeInfo.publishTime).append("\n");
    info.append("安装包 Md5: ").append(upgradeInfo.apkMd5).append("\n");
    info.append("安装包下载地址: ").append(upgradeInfo.apkUrl).append("\n");
    info.append("安装包大小: ").append(upgradeInfo.fileSize).append("\n");
    info.append("弹窗间隔 ( ms ) : ").append(upgradeInfo.popInterval).append("\n");
    info.append("弹窗次数: ").append(upgradeInfo.popTimes).append("\n");
    info.append("发布类型 ( 0:测试 1:正式 ) : ").append(upgradeInfo.publishType).append("\n");
    info.append("弹窗类型 ( 1:建议 2:强制 3:手工 ) : ").append(upgradeInfo.upgradeType).append("\n");
    info.append("图片地址 : ").append(upgradeInfo.imageUrl);
}

/**
 * 请求数据
 */
private void getData() {
    HomeRequestManager.getInstance().getHomeList(new HttpResult<HomeInfo>() {
        @Override
        public void success(HomeInfo obj) {
        }
        @Override
        public void err(String msg) {
        }
    })
}

```

```

    });

    HomeRequestManager.getInstance().getUnReadMessageCount(new HttpResult<MessageInfo>() {

        @Override

        public void success(MessageInfo obj) {

        }

        @Override

        public void err(String msg) {

        }

    });

}

public class BannerViewHolder implements MZViewHolder<EssayInfo> {

    private RoundedImageView mImageView;

    @Override

    public View onCreateView(Context context) {

        // 返回页面布局

        View view = LayoutInflater.from(context).inflate(R.layout.item_banner, null);

        mImageView = (RoundedImageView) view.findViewById(R.id.iv_image_banner);

        return view;

    }

    @Override

    public void onBind(Context context, int position, EssayInfo data) {

        ImageLoadUtils.load(context, data.getImgs(), mImageView);

        mImageView.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View view) {

                HomeViewModel.getInstance().setEssayInfo(data);

                ActivityUtils.startActivity(getActivity(), EssayDetailsActivity.class);

            }

        });

    }

}

private EssayInfo essayInfo;

@Override

protected void initData() {

    LiveDataBus.get().with(HomeViewModel.class.getName(), HomeViewModel.class).observe(this, new Observer<HomeViewModel>() {

        @Override

        public void onChanged(HomeViewModel homeViewModel) {

            if (homeViewModel.getAdvert() != null && homeViewModel.getAdvert().size() > 0) {

                essayInfo = homeViewModel.getAdvert().get(0);

                ImageLoadUtils.load(getContext(), homeViewModel.getAdvert().get(0).getImgs(), binding.ivHomeAds);

                binding.ivHomeAds.setVisibility(View.VISIBLE);

                if (essayInfo != null && essayInfo.getId() != 1485101) {

                    binding.ivHomeLiveStates.setVisibility(View.VISIBLE);

                    binding.tvHomeLiveTimer.setVisibility(View.GONE);

                    switch (essayInfo.getType()) {

                        case "0"://未开始

                            binding.ivHomeLiveStates.setImageResource(R.mipmap.icon_live_sta_0);

                            binding.tvHomeLiveTimer.setText("开播时间：" + essayInfo.getCreatedate());

                            break;

                        case "1"://直播中

```

```

        binding.ivHomeLiveStates.setImageResource(R.mipmap.icon_live_sta_1);
        binding.tvHomeLiveTimer.setText("开播时间：" + essayInfo.getCreatedate());
        break;
    case "2"://结束
        binding.ivHomeLiveStates.setImageResource(R.mipmap.icon_live_sta_2);
        binding.tvHomeLiveTimer.setText("开播时间：" + essayInfo.getCreatedate());
        break;
    }
    } else {
        binding.ivHomeLiveStates.setVisibility(View.GONE);
        binding.tvHomeLiveTimer.setVisibility(View.GONE);
    }
    } else {
        binding.ivHomeAds.setVisibility(View.GONE);
    }
    everyDayAdapter.notifyDataSetChanged();
    homeVideoAdapter.notifyDataSetChanged();
    recommendAdapter.notifyDataSetChanged();
    questionAdapter.notifyDataSetChanged();
    binding.vpBanner.setPages(HomeViewModel.getInstance().getBannerInfoList(), new MZHolderCreator<BannerViewHolder>() {
        @Override
        public BannerViewHolder createViewHolder() {
            return new BannerViewHolder();
        }
    });
}

LiveDataBus.get().with(MessageViewModel.class.getName(), MessageViewModel.class).observe(this, new
Observer<MessageViewModel>() {
    @Override
    public void onChanged(MessageViewModel messageViewModel) {
        if (messageViewModel.getCount() > 0) {
            binding.ivHomeMsg.setIsMsg(true);
        } else {
            binding.ivHomeMsg.setIsMsg(false);
        }
    }
});
}

@Override
public void onResume() {
    super.onResume();
    binding.vpBanner.start();
}

@Override
public void onPause() {
    super.onPause();
    binding.vpBanner.pause();
}
}
}

```



```
package com.tm.dmkeep.ui.fragments;

import android.view.View;

import android.widget.LinearLayout;

import androidx.annotation.NonNull;

import androidx.fragment.app.Fragment;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import androidx.viewpager.widget.ViewPager;

import com.chad.library.adapter.base.BaseQuickAdapter;

import com.chad.library.adapter.base.listener.OnItemClickListener;

import com.netting.baselibrary.base.BaseBindFragment;

import com.tm.dmkeep.R;

import com.tm.dmkeep.databinding.FragmentCommunityBinding;

import com.tm.dmkeep.http.HttpResultTotal;

import com.tm.dmkeep.http.circle.CircleHomeViewModel;

import com.tm.dmkeep.http.circle.CircleRequestManager;

import com.tm.dmkeep.http.circle.entity.TopicClassInfo;

import com.tm.dmkeep.http.mall.entity.MallTypeInfo;

import com.tm.dmkeep.ui.adapters.ShoppingTitleAdapter;

import com.tm.dmkeep.ui.adapters.TopicAdapter;

import com.tm.dmkeep.ui.adapters.ViewPagerAdapter;

import java.util.ArrayList;

import java.util.List;

/**
 * 时间 : 2022/6/15 09:21
 * 作者 : 庞宇锋
 * 功能 : 社区
 */

public class CommunityFragment extends BaseBindFragment<FragmentCommunityBinding> {

    private ArrayList<Fragment> commFragmentListList = new ArrayList<>();

    private ShoppingTitleAdapter shoppingTitleAdapter;

    private ViewPagerAdapter viewPagerAdapter;

    private List<MallTypeInfo> commTypeInfoList = new ArrayList<>();

    @Override

    public int getLayout() {

        return R.layout.fragment_community;

    }

    @Override

    protected void initView() {

        MallTypeInfo mallTypeInfo = new MallTypeInfo();

        mallTypeInfo.setName("精选");

        mallTypeInfo.setld("0");

        commTypeInfoList.add(mallTypeInfo);

        MallTypeInfo mallTypeInfo2 = new MallTypeInfo();

        mallTypeInfo2.setName("圈子");

        mallTypeInfo2.setld("1");

        commTypeInfoList.add(mallTypeInfo2);

        shoppingTitleAdapter = new ShoppingTitleAdapter(R.layout.item_tab_shopping, commTypeInfoList);

        shoppingTitleAdapter.setOnItemClickListener(new.OnItemClickListener() {

            @Override
```

```

        public void onItemClick(@NonNull BaseQuickAdapter<?, ?> adapter, @NonNull View view, int position) {
            binding.vpShopping.setCurrentItem(position);
        }
    });
    binding.lvTitleComLay.setAdapter(shoppingTitleAdapter);
    for (int i = 0; i < commTypeInfoList.size(); i++) {
        switch (i) {
            case 0://精选
                CommunityCurationFragment communityCurationFragment = new CommunityCurationFragment();
                commFragmentManagerList.add(commFragmentManagerList.add(communityCurationFragment));
                break;
            case 1://圈子
                CircleHomeFragment circleHomeFragment = new CircleHomeFragment();
                commFragmentManagerList.add(circleHomeFragment);
                break;
        }
    }
    viewPagerAdapter = new ViewPagerAdapter(getChildFragmentManager());
    viewPagerAdapter.setFragmentList(commFragmentManagerList);
    binding.vpShopping.setAdapter(viewPagerAdapter);
    binding.vpShopping.addOnPageChangeListener(new ViewPager.OnPageChangeListener() {
        @Override
        public void onPageScrolled(int position, float positionOffset, int positionOffsetPixels) {
        }
        @Override
        public void onPageSelected(int position) {
            showTitleTab(position);
        }
        @Override
        public void onPageScrollStateChanged(int state) {
        }
    });
}

private int selectIndex = 0;
public void showTitleTab(int index) {
    binding.lvTitleComLay.smoothScrollToPosition(index);
    shoppingTitleAdapter.setIndex(index);
    selectIndex = index;
}

@Override
protected void initData() {
}
}

package com.tm.dmkeep.ui.fragments;
import android.os.Bundle;
import android.view.View;
import androidx.annotation.NonNull;
import androidx.databinding.DataBindingUtil;
import androidx.lifecycle.Observer;
import androidx.recyclerview.widget.GridLayoutManager;

```

```
import com.chad.library.adapter.base.BaseQuickAdapter;
import com.chad.library.adapter.base.listener.OnItemClickListener;
import com.chad.library.adapter.base.listener.OnItemLongClickListener;
import com.lzy.okgo.model.Response;
import com.netting.baselibrary.base.BaseBindFragment;
import com.netting.baselibrary.okgoutils.HttpLibResultModel;
import com.netting.baselibrary.okgoutils.OkGoCallback;
import com.netting.baselibrary.utils.ActivityUtils;
import com.netting.baselibrary.utils.LiveDataBus;
import com.netting.baselibrary.utils.LogUtils;
import com.netting.baselibrary.utils.ToastUtils;
import com.scwang.smart.refresh.layout.api.RefreshLayout;
import com.scwang.smart.refresh.layout.listener.OnLoadMoreListener;
import com.scwang.smart.refresh.layout.listener.OnRefreshListener;
import com.tm.dmkeep.R;
import com.tm.dmkeep.databinding.FragmentCourseBinding;
import com.tm.dmkeep.databinding.TitleCourseListBinding;
import com.tm.dmkeep.http.HttpConfig;
import com.tm.dmkeep.http.HttpResult;
import com.tm.dmkeep.http.circle.CircleRequestManager;
import com.tm.dmkeep.http.course.CourseRequestManager;
import com.tm.dmkeep.http.course.entity.CourseClassInfo;
import com.tm.dmkeep.http.home.HomeRequestManager;
import com.tm.dmkeep.http.home.HomeViewModel;
import com.tm.dmkeep.http.home.entity.CourseInfo;
import com.tm.dmkeep.http.home.entity.EssayInfo;
import com.tm.dmkeep.http.live.LiveRequestManager;
import com.tm.dmkeep.http.live.LiveStartViewModel;
import com.tm.dmkeep.http.live.entity.LiveInfo;
import com.tm.dmkeep.http.user.UserViewModel;
import com.tm.dmkeep.managers.LoginManager;
import com.tm.dmkeep.http.live.StartLiveActivityManager;
import com.tm.dmkeep.ui.activitys.ClassDetailsInfoActivity;
import com.tm.dmkeep.ui.activitys.CoursePayDetailsActivity;
import com.tm.dmkeep.ui.activitys.EveryDayInfoActivity;
import com.tm.dmkeep.ui.adapters.CourseClassListAdapter;
import com.tm.dmkeep.ui.adapters.CourseFragmentListAdapter;
import com.tm.dmkeep.ui.adapters.CourseLiveAdapter;
import com.tm.dmkeep.ui.adapters.EveryDayRecommendAdapter;
import com.tm.dmkeep.ui.dialogs.DmDialog;
import com.tm.dmkeep.ui.dialogs.PostsTypeDialog;
import java.util.ArrayList;
import java.util.List;

/**
 * 时间 : 2022/4/12 15:21
 * 作者 : 庞宇锋
 * 功能 : 课程
 */
public class CourseFragment extends BaseBindFragment<FragmentCourseBinding> {
    private CourseFragmentListAdapter courseFragmentListAdapter;
```

```

private TitleCourseListBinding titleCourseListBinding;
private CourseClassListAdapter courseClassListAdapter;
private List<CourseClassInfo> courseClassInfoList = new ArrayList<>();
/**
 * 公开课
 */
private EveryDayRecommendAdapter everyDayRecommendAdapter;
private List<EssayInfo> essayInfoList = new ArrayList<>();
private CourseLiveAdapter courseLiveAdapter;
@Override
public int getLayout() {
    return R.layout.fragment_course;
}
private int page = 1;
@Override
protected void initView() {
    titleCourseListBinding = DataBindingUtil.inflate(getLayoutInflater(), R.layout.title_course_list, null, false);
    courseFragmentManager = new CourseFragmentManager(R.layout.item_course_fragment_list_lay, courseInfoList);
    courseFragmentManager.setHeaderView(titleCourseListBinding.getRoot());
    courseFragmentManager.setOnItemClickListener(new OnItemClickListener() {
        @Override
        public void onItemClick(@NonNull BaseQuickAdapter<?, ?> adapter, @NonNull View view, int position) {
            if (LoginManager.isLogin(getActivity())) {
                CourseInfo courseInfo = courseFragmentManager.getData().get(position);
                LogUtils.e("请求的 id" + courseInfo.getId());
                Bundle bundle = new Bundle();
                bundle.putString("ID", courseInfo.getId());
                bundle.putString("BoardId", courseInfo.getForid());
                ActivityUtils.startActivityForBundleData(getActivity(), CoursePayDetailsActivity.class, bundle);
            }
        }
    });
    binding.lvCourseList.setAdapter(courseFragmentManager);
    binding.refCourseRefresh.setOnLoadMoreListener(new OnLoadMoreListener() {
        @Override
        public void onLoadMore(@NonNull RefreshLayout refreshLayout) {
            getData(false);
        }
    });
    binding.refCourseRefresh.setOnRefreshListener(new OnRefreshListener() {
        @Override
        public void onRefresh(@NonNull RefreshLayout refreshLayout) {
            getData(true);
        }
    });
    courseClassListAdapter = new CourseClassListAdapter(R.layout.item_coures_item, courseClassInfoList);
    titleCourseListBinding.lvCourseClassList.setLayoutManager(new GridLayoutManager(getActivity(), 2));
    titleCourseListBinding.lvCourseClassList.setAdapter(courseClassListAdapter);
    courseClassListAdapter.setOnItemClickListener(new OnItemClickListener() {
        @Override

```

```

        public void onItemClickListener(@NonNull BaseQuickAdapter<?, ?> adapter, @NonNull View view, int position) {
            String id = courseClassListAdapter.getItem(position).getId();
            String topicid = courseClassListAdapter.getItem(position).getTopicid();
            String title = courseClassListAdapter.getItem(position).getName();
            if (topicid == null || topicid.equals("") || topicid.equals("0")) {
                ToastUtils.showToastShort(getContext(), "敬请期待");
            } else {
                Bundle bundle = new Bundle();
                bundle.putString("Url", HttpConfig.webDetails + topicid + "&type=share2");
                bundle.putString("title", title);
                ActivityUtils.startActivityForBundleData(getActivity(), ClassDetailsInfoActivity.class, bundle);
            }
        }
//    }
    });
    everyDayRecommendAdapter = new EveryDayRecommendAdapter(R.layout.item_every_day_recommend, essayInfoList);
    titleCourseListBinding.lvPublicCourseList.setAdapter(everyDayRecommendAdapter);
    everyDayRecommendAdapter.setOnItemClickListener(new OnItemClickListener() {
        @Override
        public void onItemClick(@NonNull BaseQuickAdapter<?, ?> adapter, @NonNull View view, int position) {
            HomeViewModel.getInstance().setEveryDay(everyDayRecommendAdapter.getItem(position));
            ActivityUtils.startActivity(getActivity(), EveryDayInfoActivity.class);
        }
    });
    /**
     * 今日直播
     */
    courseLiveAdapter = new CourseLiveAdapter(R.layout.item_live_lay, LiveStartViewModel.getInstance().getLiveInfoList());
    titleCourseListBinding.lvPublicCourseLiveList.setAdapter(courseLiveAdapter);
    courseLiveAdapter.setOnItemClickListener(new OnItemClickListener() {
        @Override
        public void onItemClick(@NonNull BaseQuickAdapter<?, ?> adapter, @NonNull View view, int position) {
            LiveInfo liveInfo = courseLiveAdapter.getItem(position);
            if (liveInfo.getClassid().equals(HttpConfig.PUBLIC_LIVE_ID)) {
                StartLiveActivityManager.startLive(getActivity(), liveInfo.getTitle(), liveInfo.getTyp(), liveInfo.getAnchorid(), liveInfo.getLiveid(),
                liveInfo.getId());
            } else if (liveInfo.isOpen()) {
                StartLiveActivityManager.startLive(getActivity(), liveInfo.getTitle(), liveInfo.getTyp(), liveInfo.getAnchorid(), liveInfo.getLiveid(),
                liveInfo.getId());
            } else {
                ToastUtils.showToastShort(getContext(), "无观看权限，如果已经报班，请刷新后重试！");
            }
        }
    }
    });
    //长按删除课程
    courseLiveAdapter.setOnItemLongClickListener(new OnItemLongClickListener() {
        @Override
        public boolean onItemLongClick(@NonNull BaseQuickAdapter adapter, @NonNull View view, int position) {
            LiveInfo liveInfo = courseLiveAdapter.getItem(position);
            showDialog(liveInfo);
        }
    });

```

```

        return true;
    }

});

getData(true);
}

/**
 * 删除，新建课程
 */

public void showDialog(LiveInfo liveInfo) {
    if (!LoginManager.isLogin()) {
        return;
    }

    if (!UserViewModel.getInstance().getUserInfo().getUserid().equals(liveInfo.getAnchorid())) {
        return;
    }

    if (liveInfo.getType().equals("0")){//未开始
        createDialog("1", "删除课程", liveInfo);
    } else if (liveInfo.getType().equals("2")){//直播结束
        createDialog("0", "复制课程", liveInfo);
    }
}

/**
 * 创建不同类型弹出框
 */

public void createDialog(String id, String name, LiveInfo liveInfo) {
    PostsTypeDialog postsTypeDialog = new PostsTypeDialog
        .Builder(getActivity())
        .addItem(id, name)
        .getPostsTypeDialog(new PostsTypeDialog.PostsCallBack() {
            @Override
            public void onItemTypeClick(int p, PostsTypeDialog.PostsTypeInfo postsTypeInfo) {
                switch (postsTypeInfo.getTypeId()) {
                    case "0"://复制课程
                        LiveRequestManager.getInstance().copyLive(liveInfo.getClassid(), liveInfo.getBoardid(),
                            liveInfo.getimgs(), liveInfo.getTitle(),
                            liveInfo.getCreatedate(),
                            new OkGoCallback<HttpLibResultModel<String>>() {
                                @Override
                                public void onSur(Response<HttpLibResultModel<String>> response) {
                                    ToastUtils.showToastShort(getContext(), "复制成功");
                                }
                            }
                        );
                        /**
                         * 获取今日直播
                         */
                        LiveRequestManager.getInstance().getTodayTopicList(null);
                    }
                }
            }
            @Override
            public void onErr(String msg) {
            }
        });
    break;
}

```

```

        case "1"://删除课程
            DmDialog dmDialog1 = new DmDialog.Builder(getActivity()).setContent("删除当前课程吗 ? ")
                .setCallBack(new DmDialog.CallBack() {
                    @Override
                    public void cancel() {
                    }
                    @Override
                    public void next() {
                        CircleRequestManager.getInstance().deleteTopic(liveInfo.getId(), new
                            HttpResult<String>() {
                                @Override
                                public void success(String obj) {
                                    ToastUtils.showToastShort(getActivity(), "删除成功");
                                    LiveRequestManager.getInstance().getTodayTopicList(null);
                                }
                                @Override
                                public void err(String msg) {
                                }
                            })
                        });
                    })
                .getDmDialog();
            dmDialog1.show();
            break;
        }
    }
    });
    postsTypeDialog.show();
}

@Override
protected void initData() {
    /**
     * 获取今日直播列表
     */
    LiveRequestManager.getInstance().getTodayTopicList(new HttpResult<List<LiveInfo>>() {
        @Override
        public void success(List<LiveInfo> obj) {
        }
        @Override
        public void err(String msg) {
        }
    })
    });
    /**
     * 监听今日直播
     */
    LiveDataBus.get().with(LiveStartViewModel.class.getName(), LiveStartViewModel.class).observe(this, new Observer<LiveStartViewModel>() {
        @Override
        public void onChanged(LiveStartViewModel liveStartViewModel) {
            courseLiveAdapter.notifyDataSetChanged();
            titleCourseListBinding.tvTabDay.setText("今日直播(" + liveStartViewModel.getLiveInfoList().size() + ")");
        }
    });
}

```

```

    }
});
LiveDataBus.get().with(UserViewModel.class.getName(), UserViewModel.class).observe(this, new Observer<UserViewModel>() {
    @Override
    public void onChanged(UserViewModel userViewModel) {
        getData(true);
    }
});
/**
 * 获取四班制列表
 */
CourseRequestManager.getInstance().getCourseClassList(new HttpStatus<List<CourseClassInfo>>>() {
    @Override
    public void success(List<CourseClassInfo> obj) {
        courseClassInfoList.clear();
        courseClassInfoList.addAll(obj);
        courseClassListAdapter.notifyDataSetChanged();
    }
    @Override
    public void err(String msg) {
    }
});
/**
 * 获取推荐列表
 */
HomeRequestManager.getInstance().getRecommendList(String.valueOf("1485155"), String.valueOf("1562985"), 10, new
HttpStatus<List<EssayInfo>>>() {
    @Override
    public void success(List<EssayInfo> obj) {
        essayInfoList.addAll(obj);
        everyDayRecommendAdapter.notifyDataSetChanged();
    }
    @Override
    public void err(String msg) {
    }
});
}
private List<CourseInfo> courseInfoList = new ArrayList<>();
/**
 * 请求课程列表
 */
public void getData(boolean is) {
    if (is) {
        page = 1;
    }
    CourseRequestManager.getInstance().getCourseList("511100", String.valueOf(page), "20", new HttpStatus<List<CourseInfo>>>() {
        @Override
        public void success(List<CourseInfo> obj) {
            if (is) {
                courseInfoList.clear();
            }
        }
    });
}

```



```

        }
        page++;
        courseInfoList.addAll(obj);
        binding.refCourseRefresh.finishRefresh();
        binding.refCourseRefresh.finishLoadMore();
        courseFragmentManager.notifyDataSetChanged();
    }
    @Override
    public void err(String msg) {
        binding.refCourseRefresh.finishRefresh();
        binding.refCourseRefresh.finishLoadMore();
    }
});
/**
 * 获取推荐列表
 */
HomeRequestManager.getInstance().getRecommendList(String.valueOf("1485155"), String.valueOf("1562985"), 10, new
HttpResult<List<EssayInfo>>() {
    @Override
    public void success(List<EssayInfo> obj) {
        essayInfoList.clear();
        essayInfoList.addAll(obj);
        everyDayRecommendAdapter.notifyDataSetChanged();
    }
    @Override
    public void err(String msg) {
    }
});
package com.tm.dmkeep.ui.fragments;
import android.content.Context;
import android.view.Gravity;
import android.view.View;
import androidx.annotation.MainThread;
import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.lifecycle.Observer;
import androidx.viewpager.widget.ViewPager;
import com.chad.library.adapter.base.BaseQuickAdapter;
import com.chad.library.adapter.base.listener.OnItemClickListener;
import com.netting.baselibrary.base.BaseBindFragment;
import com.netting.baselibrary.utils.ActivityUtils;
import com.netting.baselibrary.utils.LiveDataBus;
import com.tm.dmkeep.R;
import com.tm.dmkeep.databinding.FragmentShoppingBinding;
import com.tm.dmkeep.http.mall.MallRequestManager;
import com.tm.dmkeep.http.mall.MallViewModel;
import com.tm.dmkeep.managers.LoginManager;
import com.tm.dmkeep.ui.activitys.MainActivity;
import com.tm.dmkeep.ui.activitys.ShoppingCarActivity;
import com.tm.dmkeep.ui.adapters.ShoppingTitleAdapter;

```

```
import com.tm.dmkeep.ui.adapters.ViewPagerAdapter;
import java.util.ArrayList;
import java.util.List;
import io.reactivex.Observable;
import io.reactivex.ObservableEmitter;
import io.reactivex.ObservableOnSubscribe;
import io.reactivex.disposables.Disposable;
import io.reactivex.schedulers.Schedulers;
import q.rorbin.badgeview.QBadgeView;
import static com.tm.dmkeep.utils.Message.QBadge;

/**
 * 时间：2022/4/12 15:19
 * 作者：庞宇锋
 * 功能：商城
 */
public class ShoppingFragment extends BaseBindFragment<FragmentShoppingBinding> {
    private ArrayList<Fragment> mallFragmentList = new ArrayList<>();
    private ShoppingTitleAdapter shoppingTitleAdapter;
    private ViewPagerAdapter viewPagerAdapter;

    @Override
    public int getLayout() {
        return R.layout.fragment_shopping;
    }

    @Override
    protected void initView() {
        MallRequestManager.getInstance().getShoppingCarList();
        MallRequestManager.getInstance().getMallTypeList();
        shoppingTitleAdapter = new ShoppingTitleAdapter(R.layout.item_tab_shopping, MallViewModel.getInstance().getMallTypeInfoList());
        binding.lvTitleLay.setAdapter(shoppingTitleAdapter);
        binding.vpShopping.addOnPageChangeListener(new ViewPager.OnPageChangeListener() {
            @Override
            public void onPageScrolled(int position, float positionOffset, int positionOffsetPixels) {
            }
            @Override
            public void onPageSelected(int position) {
                showTitleTab(position);
            }
            @Override
            public void onPageScrollStateChanged(int state) {
            }
        });
        shoppingTitleAdapter.setOnItemClickListener(new OnItemClickListener() {
            @Override
            public void onItemClick(@NonNull BaseQuickAdapter<?, ?> adapter, @NonNull View view, int position) {
                binding.vpShopping.setCurrentItem(position);
            }
        });
        binding.ivShopping.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
```

```

        if (LoginManager.isLogin(getActivity())) {
            ActivityUtils.startActivity(getActivity(), ShoppingCarActivity.class);
        }
    }
});
initFragment();
}

private int selectIndex = 0;
public void showTitleTab(int index) {
    binding.lvTitleLay.smoothScrollToPosition(index);
    shoppingTitleAdapter.setIndex(index);
    selectIndex = index;
}

public void initFragment() {
    if (mallFragmentListList.size() > 0) {
        return;
    }
    if (MallViewModel.getInstance().getMallTypeInfoList() == null || MallViewModel.getInstance().getMallTypeInfoList().size() <= 0) {
        return;
    }
    mallFragmentListList.clear();
    for (int i = 0; i < MallViewModel.getInstance().getMallTypeInfoList().size(); i++) {
        String id = MallViewModel.getInstance().getMallTypeInfoList().get(i).getId();
        MallFragmentList mallFragmentList = new MallFragmentList();
        mallFragmentList.setCid(id);
        mallFragmentListList.add(mallFragmentList);
    }
    viewPagerAdapter = new ViewPagerAdapter(getChildFragmentManager());
    viewPagerAdapter.setFragmentList(mallFragmentListList);
    binding.vpShopping.setAdapter(viewPagerAdapter);
    binding.vpShopping.setOffscreenPageLimit(mallFragmentListList.size());
    binding.vpShopping.setCurrentItem(selectIndex);
}

private QBadgeView qBadgeView;
@Override
protected void initData() {
    LiveDataBus.get().with(MallViewModel.class.getName(), MallViewModel.class).observe(this, new Observer<MallViewModel>() {
        @Override
        public void onChanged(MallViewModel mallViewModel) {
            initFragment();
            shoppingTitleAdapter.notifyDataSetChanged();
            if (mallViewModel.getShoppingList() != null) {
                qBadgeView = QBadge(getContext(), binding.ivShopping, mallViewModel.getShoppingList().size(), qBadgeView);
            } else {
                qBadgeView = QBadge(getContext(), binding.ivShopping, 0, qBadgeView);
            }
        }
    });
}
}
}

```

```
package com.tm.dmkeep.ui.fragments;

import android.graphics.Color;
import android.os.Bundle;
import android.text.Spannable;
import android.text.SpannableString;
import android.text.SpannableStringBuilder;
import android.text.Spanned;
import android.text.style.AbsoluteSizeSpan;
import android.text.style.ForegroundColorSpan;
import android.view.View;
import android.widget.AdapterView;
import androidx.annotation.NonNull;
import androidx.databinding.DataBindingUtil;
import androidx.lifecycle.Observer;
import androidx.recyclerview.widget.GridLayoutManager;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import com.chad.library.adapter.base.BaseQuickAdapter;
import com.chad.library.adapter.base.listener.OnItemChildClickListener;
import com.chad.library.adapter.base.listener.OnItemClickListener;
import com.google.gson.Gson;
import com.netting.baselibrary.base.BaseBindFragment;
import com.netting.baselibrary.ui.activities.WebActivity;
import com.netting.baselibrary.utils.ActivityUtils;
import com.netting.baselibrary.utils.LiveDataBus;
import com.netting.baselibrary.utils.LogUtils;
import com.netting.baselibrary.utils.ToastUtils;
import com.scwang.smart.refresh.layout.api.RefreshLayout;
import com.scwang.smart.refresh.layout.listener.OnRefreshListener;
import com.tencent.mm.opensdk.constants.Build;
import com.tencent.mm.opensdk.modelbiz.WXOpenCustomerServiceChat;
import com.tencent.mm.opensdk.openapi.IWXAPI;
import com.tencent.mm.opensdk.openapi.WXAPIFactory;
import com.tm.dmkeep.R;
import com.tm.dmkeep.databinding.FragmentMyBinding;
import com.tm.dmkeep.databinding.MyHomeTitleBinding;
import com.tm.dmkeep.http.HttpConfig;
import com.tm.dmkeep.http.HttpResult;
import com.tm.dmkeep.http.HttpResultTotal;
import com.tm.dmkeep.http.circle.CircleRequestManager;
import com.tm.dmkeep.http.circle.entity.CircleInfo;
import com.tm.dmkeep.http.home.HomeRequestManager;
import com.tm.dmkeep.http.home.MyDetailsViewModel;
import com.tm.dmkeep.http.home.entity.CourseInfo;
import com.tm.dmkeep.http.home.entity.CourseItemInfo;
import com.tm.dmkeep.http.home.entity.EssayInfo;
import com.tm.dmkeep.http.live.CourseLiveViewModel;
import com.tm.dmkeep.http.user.UserRequestManager;
import com.tm.dmkeep.http.user.UserViewModel;
import com.tm.dmkeep.http.user.entity.DeskTopInfo;
```

```
import com.tm.dmkeep.http.user.entity.UserInfo;
import com.tm.dmkeep.managers.LoginManager;
import com.tm.dmkeep.http.live.StartLiveActivityManager;
import com.tm.dmkeep.ui.activitys.AnswerListActivity;
import com.tm.dmkeep.ui.activitys.CircleActivity;
import com.tm.dmkeep.ui.activitys.CircleDetailsActivity;
import com.tm.dmkeep.ui.activitys.ClassListActivity;
import com.tm.dmkeep.ui.activitys.CourseOrderActivity;
import com.tm.dmkeep.ui.activitys.EtDiaryActivity;
import com.tm.dmkeep.ui.activitys.InstructorDetailsActivity;
import com.tm.dmkeep.ui.activitys.MyClassDetailsActivity;
import com.tm.dmkeep.ui.activitys.MyCommentsListActivity;
import com.tm.dmkeep.ui.activitys.MyDetailsActivity;
import com.tm.dmkeep.ui.activitys.MyEssayActivity;
import com.tm.dmkeep.ui.activitys.OrderListActivity;
import com.tm.dmkeep.ui.activitys.SetUpActivity;
import com.tm.dmkeep.ui.activitys.ShoppingCarActivity;
import com.tm.dmkeep.ui.activitys.VipPayActivity;
import com.tm.dmkeep.ui.adapters.CircleHomeAdapter;
import com.tm.dmkeep.ui.adapters.CourseAdapter;
import com.tm.dmkeep.ui.adapters.DesktopAdapter;
import com.tm.dmkeep.ui.adapters.InstructorAdapter;
import com.tm.dmkeep.ui.adapters.RecommendAdapter;
import com.tm.dmkeep.ui.adapters.SpinnerAdapter;
import com.tm.dmkeep.ui.views.CustomSpinner;
import com.tm.dmkeep.utils.ImageLoadUtils;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

/**
 * 时间 : 2022/4/11 15:58
 * 作者 : 庞宇锋
 * 功能 : 个人中心
 */
public class MyFragment extends BaseBindFragment<FragmentMyBinding> {
    private MyHomeTitleBinding myHomeTitleBinding;
    private RecommendAdapter recommendAdapter;
    private List<EssayInfo> essayInfoList = new ArrayList<>();
    private InstructorAdapter instructorAdapter;
    private CourseAdapter courseAdapter;
    private SpinnerAdapter spinnerAdapter;
    private CircleHomeAdapter circleHomeAdapter;
    private DesktopAdapter desktopAdapter;

    @Override
    public int getLayout() {
        return R.layout.fragment_my;
    }

    @Override
    protected void initView() {
```

```
myHomeTitleBinding = DataBindingUtil.inflate(getLayoutInflater(), R.layout.my_home_title, null, false);
recommendAdapter = new RecommendAdapter(essayInfoList);
recommendAdapter.addHeaderView(myHomeTitleBinding.getRoot());
binding.rvMyList.setAdapter(recommendAdapter);

//教练
instructorAdapter = new InstructorAdapter(R.layout.item_instructor);
instructorAdapter.setOnItemClickListener(new OnItemClickListener() {
    @Override
    public void onItemClick(@NonNull BaseQuickAdapter<?, ?> adapter, @NonNull View view, int position) {
        MyDetailsViewModel.getInstance().setInstructorInfo(instructorAdapter.getItem(position));
        ActivityUtils.startActivity(getActivity(), InstructorDetailsActivity.class);
    }
});

myHomeTitleBinding.lvMember.setAdapter(instructorAdapter);

//设置
myHomeTitleBinding.ivMySetup.setOnClickListener(v -> {
    ActivityUtils.startActivity(getActivity(), SetUpActivity.class);
});

//客服
myHomeTitleBinding.ivMyKefu.setOnClickListener(view -> {
    String appId = "wx12dcf36bffa9702"; // 微信开放平台, 填移动应用(App)的 AppId
    WXAPI api = WXAPIFactory.createWXAPI(getActivity(), appId);
    // 判断当前版本是否支持拉起客服会话
    if (api.getWXAppSupportAPI() >= Build.SUPPORT_OPEN_CUSTOMER_SERVICE_CHAT) {
        WXOpenCustomerServiceChat.Req req = new WXOpenCustomerServiceChat.Req();
        req.corpId = "ww7e9cd8c327873ad"; // 微信客服后台, 企业 ID
        req.url = "https://work.weixin.qq.com/kfid/kfcdb94bd7c0bf77bcd"; // 客服 URL
        api.sendReq(req);
    }
});

binding.refLay.setOnRefreshListener(new OnRefreshListener() {
    @Override
    public void onRefresh(@NonNull RefreshLayout refreshLayout) {
        HomeRequestManager.getInstance().getClassInfo(new HttpResult<CourseInfo>()) {
            @Override
            public void success(CourseInfo obj) {
                binding.refLay.finishRefresh();
            }
            @Override
            public void err(String msg) {
                binding.refLay.finishRefresh();
            }
        }
    }
});

UserRequestManager.getInstance().getUserInfoNotify();

//写日记
myHomeTitleBinding.tvMyTab1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
```

```
        if (LoginManager.isLogin() && LoginManager.isRegisterClass(getActivity(), "1")) {
            Bundle intent = new Bundle();
            intent.putString(EtDiaryActivity.TYPE, "1");
            ActivityUtils.startActivityForBundleData(getActivity(), EtDiaryActivity.class, intent);
        } else {
            LoginManager.login(getActivity());
        }
    }
});
//圈子
myHomeTitleBinding.tvQz.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
    }
});
//月总结
myHomeTitleBinding.tvMyTab2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (LoginManager.isLogin() && LoginManager.isRegisterClass(getActivity(), "2")) {
            String grade = UserViewModel.getInstance().getUserInfo().getGrade();
            Bundle intent = new Bundle();
            intent.putString(EtDiaryActivity.TYPE, "2");
            ActivityUtils.startActivityForBundleData(getActivity(), EtDiaryActivity.class, intent);
        } else {
            LoginManager.login(getActivity());
        }
    }
});
//月考核
myHomeTitleBinding.tvMyTab3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (LoginManager.isLogin() && LoginManager.isRegisterClass(getActivity(), "3")) {
            Bundle intent = new Bundle();
            intent.putString(EtDiaryActivity.TYPE, "3");
            ActivityUtils.startActivityForBundleData(getActivity(), EtDiaryActivity.class, intent);
        } else {
            LoginManager.login(getActivity());
        }
    }
});
//答题
myHomeTitleBinding.tvMyTab4.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (LoginManager.isLogin()) {
            Bundle intent = new Bundle();
            intent.putString(EtDiaryActivity.TYPE, "3");
            ActivityUtils.startActivityForBundleData(getActivity(), AnswerListActivity.class, intent);
        }
    }
});
```

```

        } else {
            LoginManager.login(getActivity());
        }
    }
}

});

//学员所在组
spinnerAdapter = new SpinnerAdapter(getContext());
myHomeTitleBinding.spClass.setAdapter(spinnerAdapter);
if (UserViewModel.getInstance().getUserInfo() != null) {
    myHomeTitleBinding.spClass.setSelection(UserViewModel.getInstance().getUserInfo().getSelectClassIndex(), false);
}
myHomeTitleBinding.spClass.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
    @Override
    public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
        UserInfo userInfo = UserViewModel.getInstance().getUserInfo();
        userInfo.setSelectClassIndex(i);
        UserViewModel.getInstance().setUserInfo(userInfo);
        HomeRequestManager.getInstance().getClassInfo(null);
    }
    @Override
    public void onNothingSelected(AdapterView<?> adapterView) {
    }
});

myHomeTitleBinding.spClass.setSpinnerEventsListener(new CustomSpinner.OnSpinnerEventsListener() {
    @Override
    public void onSpinnerOpened() {
        myHomeTitleBinding.ivSp.setImageResource(R.mipmap.icon_zk);
    }
    @Override
    public void onSpinnerClosed() {
        myHomeTitleBinding.ivSp.setImageResource(R.mipmap.icon_sq);
    }
});

//班级详情
myHomeTitleBinding.ivMyClassDetails.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (LoginManager.isLogin() && MyDetailsViewModel.getInstance().getCourseInfo() != null &&
MyDetailsViewModel.getInstance().getCourseInfo().getId() != null) {
            ActivityUtils.startActivity(getActivity(), MyClassDetailsActivity.class);
        } else {
            LoginManager.login(getActivity());
        }
    }
});

//个人中心
myHomeTitleBinding.llUserLay.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (LoginManager.isLogin()) {

```



```

        ActivityUtils.startActivity(getActivity(), MyDetailsActivity.class);
    } else {
        LoginManager.login(getActivity());
    }
}

});

myHomeTitleBinding.tvCourseBm.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
    }
});

//排行榜
myHomeTitleBinding.tvLook.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        ActivityUtils.startActivity(getActivity(), ClassListActivity.class);
    }
});

//课程适配器
courseAdapter = new CourseAdapter(R.layout.item_course_lay);
courseAdapter.setOnItemClickListner(new OnItemClickListener() {
    @Override
    public void onItemClick(@NonNull BaseQuickAdapter<?, ?> adapter, @NonNull View view, int position) {
        CourseInfo courseInfo = MyDetailsViewModel.getInstance().getCourseInfo();
        CourseLiveViewModel.getInstance().setCourseInfo(courseInfo);
        CourseItemInfo courseItemInfo = courseAdapter.getItem(position);
        StartLiveActivityManager.startLive(getActivity(), courseItemInfo);
    }
});

myHomeTitleBinding.lvCourseList.setAdapter(courseAdapter);
setUiUserInfo();

LiveDataBus.get().with(UserViewModel.class.getName(), UserViewModel.class).observe(this, new Observer<UserViewModel>() {
    @Override
    public void onChanged(UserViewModel userViewModel) {
        setUiUserInfo();
    }
});

circleHomeAdapter = new CircleHomeAdapter(R.layout.item_circle_home_lay, circleList);
circleHomeAdapter.setOnItemClickListner(new OnItemClickListener() {
    @Override
    public void onItemClick(@NonNull BaseQuickAdapter adapter, @NonNull View view, int position) {
        switch (view.getId()) {
            case R.id.iv_circle_inlet:
                if (LoginManager.isLogin(activity)) {
                    ActivityUtils.startActivity(getActivity(), CircleActivity.class);
                }
                break;
            case R.id.lay_2:
                String id = circleHomeAdapter.getItem(position).getTopicId();
                Bundle bundle = new Bundle();

```

```
        bundle.putString(CircleDetailsActivity.CIR_ID, id);
        ActivityUtils.startActivityForBundleData(getActivity(), CircleDetailsActivity.class, bundle);
        break;
    }
}

});

myHomeTitleBinding.lvCircleList.setAdapter(circleHomeAdapter);

desktopAdapter = new DesktopAdapter(R.layout.item_desk_top, UserViewModel.getInstance().getDesktopInfoList());
desktopAdapter.setOnItemClickListener(new OnItemClickListener() {

    @Override

    public void onItemClick(@NonNull BaseQuickAdapter<?, ?> adapter, @NonNull View view, int position) {

        DesktopInfo desktopInfo = desktopAdapter.getItem(position);
        switch (desktopInfo.getText()) {

            case "我贴":
                if (LoginManager.isLogin(getActivity())) {
                    Bundle bundle = new Bundle();
                    bundle.putString(MyEssayActivity.TYPE, "0");
                    ActivityUtils.startActivityForBundleData(getActivity(), MyEssayActivity.class, bundle);
                }
                break;
            case "评论":
                if (LoginManager.isLogin(getActivity())) {
                    ActivityUtils.startActivity(getActivity(), MyCommentsListActivity.class);
                }
                break;
            case "收藏":
                if (LoginManager.isLogin(getActivity())) {
                    Bundle sc = new Bundle();
                    sc.putString(MyEssayActivity.TYPE, "1");
                    ActivityUtils.startActivityForBundleData(getActivity(), MyEssayActivity.class, sc);
                }
                break;
            case "点赞":
                if (LoginManager.isLogin(getActivity())) {
                    Bundle dz = new Bundle();
                    dz.putString(MyEssayActivity.TYPE, "2");
                    ActivityUtils.startActivityForBundleData(getActivity(), MyEssayActivity.class, dz);
                }
                break;
            case "商城订单"://我的订单
                ActivityUtils.startActivity(getActivity(), OrderListActivity.class);
                break;
            case "购物车"://购物车
                if (LoginManager.isLogin(getActivity())) {
                    ActivityUtils.startActivity(getActivity(), ShoppingCarActivity.class);
                }
                break;
            case "课程订单":
                ActivityUtils.startActivity(getActivity(), CourseOrderActivity.class);
                break;
        }
    }
});
```

```
        case "考级中心"://考级中心
            Bundle bundle = new Bundle();
            bundle.putString(WebActivity.TITLE_WEB, "考级中心");
            bundle.putString(WebActivity.URL_WEB, HttpConfig.kjzxUrl);
            ActivityUtils.startActivityForBundleData(getActivity(), WebActivity.class, bundle);
            break;
        default:
            ToastUtils.showToastShort(getContext(), "开发中，敬请期待！");
            break;
    }
}

});

myHomeTitleBinding.lvZmList.setLayoutManager(new GridLayoutManager(getContext(), 4));
myHomeTitleBinding.lvZmList.setAdapter(desktopAdapter);
myHomeTitleBinding.tvMyHomeWt.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Bundle bundle = new Bundle();
        bundle.putString(MyEssayActivity.TYPE, "0");
        ActivityUtils.startActivityForBundleData(getActivity(), MyEssayActivity.class, bundle);
    }
});

myHomeTitleBinding.tvMyHomeSc.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Bundle bundle = new Bundle();
        bundle.putString(MyEssayActivity.TYPE, "1");
        ActivityUtils.startActivityForBundleData(getActivity(), MyEssayActivity.class, bundle);
    }
});

myHomeTitleBinding.tvMyHomeDz.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Bundle bundle = new Bundle();
        bundle.putString(MyEssayActivity.TYPE, "2");
        ActivityUtils.startActivityForBundleData(getActivity(), MyEssayActivity.class, bundle);
    }
});

myHomeTitleBinding.tvMyHomePl.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        ActivityUtils.startActivity(getActivity(), MyCommentsListActivity.class);
    }
});

myHomeTitleBinding.tvJoinVip.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Bundle bundle = new Bundle();
        bundle.putString("TYPE", "0");
        ActivityUtils.startActivityForBundleData(getActivity(), VipPayActivity.class, bundle);
    }
});
```

```
    }

    });

    myHomeTitleBinding.tvUserVip.setOnClickListener(new View.OnClickListener() {

        @Override

        public void onClick(View v) {

            Bundle bundle = new Bundle();

            bundle.putString("TYPE", "1");

            ActivityUtils.startActivityForBundleData(getActivity(), VipPayActivity.class, bundle);

        }

    });

    getList();

}

private List<CircleInfo> circleList = new ArrayList();

/**
 * 请求日记数据
 */

public void getList() {

    circleList.add(null);

    String json = "";

    Map map = new HashMap();

    map.put("type", "1");

    map.put("page", String.valueOf(1));

    map.put("limit", String.valueOf(10));

    json = new Gson().toJson(map);

    CircleRequestManager.getInstance().getTopicInfo(json, new HttpResultTotal<List<CircleInfo>>>() {

        @Override

        public void success(List<CircleInfo> obj, String total) {

            circleList.clear();

            circleList.add(null);

            circleList.addAll(obj);

            circleHomeAdapter.notifyDataSetChanged();

            binding.refLay.finishRefresh();

            binding.refLay.finishLoadMore();

        }

        @Override

        public void err(String msg) {

            binding.refLay.finishRefresh();

            binding.refLay.finishLoadMore();

        }

    });

}

public void setUiUserInfo() {

    if (LoginManager.isLogin()) {

        UserInfo userInfo = UserModel.getInstance().getUserInfo();

        ImageLoadUtils.loadHead(getContext(), userInfo.getImgs(), myHomeTitleBinding.ivMyUserHead);

        myHomeTitleBinding.tvUserName.setText(userInfo.getUsername());

        myHomeTitleBinding.tvCourseBm.setVisibility(View.GONE);

        myHomeTitleBinding.tvUserZu.setVisibility(View.GONE);

        myHomeTitleBinding.tvJy1.setVisibility(View.VISIBLE);

        myHomeTitleBinding.tvJy.setVisibility(View.VISIBLE);

    }

}
```

```
myHomeTitleBinding.llKcLay.setVisibility(View.VISIBLE);
myHomeTitleBinding.llRj.setVisibility(View.VISIBLE);
//是否是开通了 vip
if (userInfo.isMember()) {
    myHomeTitleBinding.tvJoinVip.setVisibility(View.GONE);
    myHomeTitleBinding.tvUserVip.setVisibility(View.VISIBLE);
    myHomeTitleBinding.llVipNoLay.setVisibility(View.GONE);
    myHomeTitleBinding.llVipYesLay.setVisibility(View.VISIBLE);
} else {
    //展示基础班报名入口
    myHomeTitleBinding.ivBgVip.setImageResource(R.mipmap.icon_home_vip_jcb);
    myHomeTitleBinding.tvJoinVip.setVisibility(View.VISIBLE);
    myHomeTitleBinding.tvUserVip.setVisibility(View.GONE);
    myHomeTitleBinding.llVipNoLay.setVisibility(View.VISIBLE);
    myHomeTitleBinding.llVipYesLay.setVisibility(View.GONE);
}
myHomeTitleBinding.tvUserZu.setText(userInfo.getPlatname());
myHomeTitleBinding.tvUserClass.setVisibility(View.VISIBLE);
switch (userInfo.getGrade()) {
    case "3*://
        myHomeTitleBinding.tvUserClass.setImageResource(R.mipmap.icon_class_1);
        break;
    case "4*://
        myHomeTitleBinding.tvUserClass.setImageResource(R.mipmap.icon_class_2);
        break;
    case "5*://
        myHomeTitleBinding.tvUserClass.setImageResource(R.mipmap.icon_class_3);
        break;
    case "6*://
        myHomeTitleBinding.tvUserClass.setImageResource(R.mipmap.icon_class_4);
        break;
    default:
        myHomeTitleBinding.tvUserClass.setVisibility(View.GONE);
        break;
}
if (MyDetailsViewModel.getInstance().getCourseInfo() != null &&
    MyDetailsViewModel.getInstance().getCourseInfo().getUserList() != null
) {
    myHomeTitleBinding.llClassLay.setVisibility(View.VISIBLE);
    myHomeTitleBinding.llClassNullLay.setVisibility(View.GONE);
    instructorAdapter.setList(MyDetailsViewModel.getInstance().getCourseInfo().getUserList());
    instructorAdapter.notifyDataSetChanged();
    courseAdapter.setList(MyDetailsViewModel.getInstance().getCourseInfo().getCourseList());
    courseAdapter.notifyDataSetChanged();
    moveToPosition(myHomeTitleBinding.lvCourseList, MyDetailsViewModel.getInstance().getShowIndex());
    setTimerText();
    spinnerAdapter.sPNotifyDataSetChanged();
    LogUtils.e("获取到了" + UserViewModel.getInstance().getUserInfo().getSelectClassIndex());
    myHomeTitleBinding.spClass.setSelection(UserViewModel.getInstance().getUserInfo().getSelectClassIndex(), false);
    myHomeTitleBinding.tvCourseTimer.setVisibility(View.VISIBLE);
}
```

```

        } else {
            instructorAdapter.notifyDataSetChanged();
            myHomeTitleBinding.tvCourseTimer.setVisibility(View.INVISIBLE);
            myHomeTitleBinding.llClassLay.setVisibility(View.GONE);
            myHomeTitleBinding.llClassNullLay.setVisibility(View.VISIBLE);
        }
        if (userInfo.getClassList() == null || userInfo.getClassList().size() <= 0) {
            myHomeTitleBinding.tvCourseTimer.setVisibility(View.INVISIBLE);
            myHomeTitleBinding.llClassLay.setVisibility(View.GONE);
            myHomeTitleBinding.llClassNullLay.setVisibility(View.VISIBLE);
            myHomeTitleBinding.ivSp.setVisibility(View.GONE);
        } else {
            myHomeTitleBinding.ivSp.setVisibility(View.VISIBLE);
        }
    } else {
        myHomeTitleBinding.tvUserName.setText("未登录");
        myHomeTitleBinding.tvCourseBm.setVisibility(View.VISIBLE);
        myHomeTitleBinding.ivMyUserHead.setImageResource(R.mipmap.icon_default_head);
        myHomeTitleBinding.tvUserZu.setVisibility(View.GONE);
        myHomeTitleBinding.tvJy1.setVisibility(View.GONE);
        myHomeTitleBinding.tvJy.setVisibility(View.GONE);
        myHomeTitleBinding.tvJoinVip.setVisibility(View.GONE);
        myHomeTitleBinding.tvUserVip.setVisibility(View.GONE);
        myHomeTitleBinding.llKcLay.setVisibility(View.VISIBLE);
        myHomeTitleBinding.llRj.setVisibility(View.GONE);
        myHomeTitleBinding.tvUserClass.setVisibility(View.GONE);
        myHomeTitleBinding.tvCourseTimer.setVisibility(View.INVISIBLE);
        myHomeTitleBinding.llClassLay.setVisibility(View.GONE);
        myHomeTitleBinding.llClassNullLay.setVisibility(View.VISIBLE);
        myHomeTitleBinding.ivSp.setVisibility(View.GONE);
        myHomeTitleBinding.tvCourseBm.setVisibility(View.GONE);
        myHomeTitleBinding.llVipNoLay.setVisibility(View.VISIBLE);
        myHomeTitleBinding.llVipYesLay.setVisibility(View.GONE);
        myHomeTitleBinding.ivBgVip.setImageResource(R.mipmap.icon_home_vip);
    }
}

public void moveToPosition(RecyclerView recyclerView, int position) {
    LinearLayoutManager mLayoutManager =
        (LinearLayoutManager) recyclerView.getLayoutManager();
    mLayoutManager.scrollToPositionWithOffset(position, 0);
}

@Override
protected void initData() {
    HomeRequestManager.getInstance().getClassInfo(null);
    /**
     * 监听数据变化
     */
    LiveDataBus.get().with(MyDetailsViewModel.class.getName(), MyDetailsViewModel.class).observe(getViewLifecycleOwner(), new
    Observer<MyDetailsViewModel>() {
        @Override

```

```

        public void onChanged(MyDetailsViewModel myDetailsViewModel) {
            setUiUserInfo();
            binding.refLay.finishRefresh();
            binding.refLay.finishLoadMore();
        }
    });
    /**
     * 监听用户登录状态
     */
    LiveDataBus.get().with(UserViewModel.class.getName(), UserViewModel.class).observe(getViewLifecycleOwner(), new
    Observer<UserViewModel>() {
        @Override
        public void onChanged(UserViewModel userViewModel) {
            HomeRequestManager.getInstance().getClassInfo(null);
            UserViewModel.getInstance().getDeskTopInfoList();
            desktopAdapter.notifyDataSetChanged();
            setUiUserInfo();
        }
    });
}
/**
 * 设置时间
 */
public void setTimerText() {
    CourseInfo courseInfo = MyDetailsViewModel.getInstance().getCourseInfo();
    myHomeTitleBinding.tvCourseName.setText(courseInfo.getName());
    myHomeTitleBinding.tvCourseBm.setVisibility(View.GONE);
    myHomeTitleBinding.tvCourseTimer.setVisibility(View.VISIBLE);
    if (Integer.parseInt(courseInfo.getStartDays()) > 0) {
        String text1 = "已进行";
        String text2 = String.valueOf(courseInfo.getStartDays());
        String text3 = "天 距毕业还有";
        String text4 = String.valueOf(courseInfo.getEndDays());
        String text5 = "天";
        String text = text1 + text2 + text3 + text4 + text5;
        SpannableString textSpanned1 = new SpannableString(text);
        SpannableStringBuilder spannableStringBuilder = new SpannableStringBuilder();
        spannableStringBuilder.append(textSpanned1);
        int left1 = text1.length();
        int end1 = left1 + text2.length();
        int left2 = end1 + text3.length();
        int end2 = left2 + text4.length();
        AbsoluteSizeSpan absoluteSizeSpan = new AbsoluteSizeSpan(40);
        spannableStringBuilder.setSpan(absoluteSizeSpan, left1, end1, Spannable.SPAN_EXCLUSIVE_INCLUSIVE);
        AbsoluteSizeSpan absoluteSizeSpan2 = new AbsoluteSizeSpan(40);
        spannableStringBuilder.setSpan(absoluteSizeSpan2, left2, end2, Spannable.SPAN_EXCLUSIVE_INCLUSIVE);
        spannableStringBuilder.setSpan(new ForegroundColorSpan(Color.parseColor("#AD0003")), left1, end1,
        Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
        spannableStringBuilder.setSpan(new ForegroundColorSpan(Color.parseColor("#AD0003")), left2, end2,
        Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
    }
}

```

```

        myHomeTitleBinding.tvCourseTimer.setText(spannableStringBuilder);
    } else if (Integer.parseInt(courseInfo.getStartDays()) == 0) {
        String text = "今日开班";
        SpannableString textSpanned1 = new SpannableString(text);
        SpannableStringBuilder spannableStringBuilder = new SpannableStringBuilder();
        spannableStringBuilder.append(textSpanned1);
        spannableStringBuilder.setSpan(new ForegroundColorSpan(Color.parseColor("#AD0003")), 0, text.length(),
Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
        myHomeTitleBinding.tvCourseTimer.setText(spannableStringBuilder);
    } else {
        String text1 = "距离开班还有";
        String text2 = String.valueOf(courseInfo.getStartDays()).replace("-", "");
        int left1 = text1.length();
        int end1 = left1 + text2.length();
        SpannableString textSpanned1 = new SpannableString(text1 + text2 + "天");
        SpannableStringBuilder spannableStringBuilder = new SpannableStringBuilder();
        spannableStringBuilder.append(textSpanned1);
        AbsoluteSizeSpan absoluteSizeSpan = new AbsoluteSizeSpan(40);
        spannableStringBuilder.setSpan(absoluteSizeSpan, left1, end1, Spannable.SPAN_EXCLUSIVE_INCLUSIVE);
        spannableStringBuilder.setSpan(new ForegroundColorSpan(Color.parseColor("#AD0003")), left1, end1,
Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
        myHomeTitleBinding.tvCourseTimer.setText(spannableStringBuilder);
    }
    if (courseInfo.getYourRank() != null && courseInfo.getYourRank().length() > 0) {
        myHomeTitleBinding.llHomePm.setVisibility(View.VISIBLE);
        if (courseInfo.getYourRank().equals("1")) {
            String text1 = "恭喜您获得" + courseInfo.getName() + "第";
            String text2 = "一";
            String text3 = "名，继续保持！";
            String text = text1 + text2 + text3;
            SpannableString textSpanned1 = new SpannableString(text);
            SpannableStringBuilder spannableStringBuilder = new SpannableStringBuilder();
            spannableStringBuilder.append(textSpanned1);
            int left1 = text1.length();
            int end1 = left1 + text2.length();
            AbsoluteSizeSpan absoluteSizeSpan = new AbsoluteSizeSpan(40);
            spannableStringBuilder.setSpan(absoluteSizeSpan, left1, end1, Spannable.SPAN_EXCLUSIVE_INCLUSIVE);
            spannableStringBuilder.setSpan(new ForegroundColorSpan(Color.parseColor("#AD0003")), left1, end1,
Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
            myHomeTitleBinding.tvRanking.setText(spannableStringBuilder);
        } else {
            //展示排名
            String text1 = "您在本班排名第 ";
            String text2 = String.valueOf(courseInfo.getYourRank());
            String text3 = "，同班的 ";
            String text4 = courseInfo.getFirstUser();
            String text5 = " 考核优异！";
            String text = text1 + text2 + text3 + text4 + text5;
            SpannableString textSpanned1 = new SpannableString(text);
            SpannableStringBuilder spannableStringBuilder = new SpannableStringBuilder();

```



```

        spannableStringBuilder.append(textSpanned1);

        int left1 = text1.length();
        int end1 = left1 + text2.length();
        int left2 = end1 + text3.length();
        int end2 = left2 + text4.length();

        AbsoluteSizeSpan absoluteSizeSpan = new AbsoluteSizeSpan(40);
        spannableStringBuilder.setSpan(absoluteSizeSpan, left1, end1, Spannable.SPAN_EXCLUSIVE_INCLUSIVE);

        AbsoluteSizeSpan absoluteSizeSpan2 = new AbsoluteSizeSpan(40);
        spannableStringBuilder.setSpan(absoluteSizeSpan2, left2, end2, Spannable.SPAN_EXCLUSIVE_INCLUSIVE);

        spannableStringBuilder.setSpan(new ForegroundColorSpan(Color.parseColor("#AD0003")), left1, end1,
Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);

        spannableStringBuilder.setSpan(new ForegroundColorSpan(Color.parseColor("#AD0003")), left2, end2,
Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);

        myHomeTitleBinding.tvRanking.setText(spannableStringBuilder);
    }
} else {
    myHomeTitleBinding.llHomePm.setVisibility(View.GONE);
}
}
}

package com.tm.dmkeep.ui.activities;
import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;
import android.text.InputType;
import android.util.ArrayMap;
import android.view.View;
import android.widget.ImageView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.GridLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import com.aliyun.roompaas.base.util.FileUtil;
import com.blankj.utilcode.util.ConvertUtils;
import com.chad.library.adapter.base.BaseQuickAdapter;
import com.chad.library.adapter.base.listener.OnItemChildClickListener;
import com.chad.library.adapter.base.listener.OnItemClickListener;
import com.google.gson.Gson;
import com.icetec.silicompressor.FileUtils;
import android.content.Intent;
import android.net.Uri;
import android.os.Build;
import android.provider.Settings;
import android.util.Log;
import android.view.View;
import androidx.annotation.Nullable;
import com.lzy.okgo.model.Response;
import com.netting.baselibrary.base.BaseActivity;
import com.netting.baselibrary.okgoutils.HttpLibResultModel;
import com.netting.baselibrary.okgoutils.OkGoCallback;
import com.netting.baselibrary.utils.Super;

```

```
import com.tm.dmkeep.R;
import com.tm.dmkeep.databinding.ActivityAppVersionBinding;
import com.tm.dmkeep.http.app.AppInfo;
import com.tm.dmkeep.http.app.AppManager;
import com.tm.dmkeep.service.DownloadService;
import static com.netting.baselibrary.utils.ContextUtil.showToastShort;

/**
 * 时间 : 2022/9/19 10:23
 * 作者 : 庞宇锋
 * 功能 : App 版本页面
 */
public class AppVersionActivity extends BaseActivity<ActivityAppVersionBinding> {

    @Override
    protected int getContentViewId() {
        return R.layout.activity_app_version;
    }

    @Override
    protected void initView() {
        binding.llTitle.ivFin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });
        binding.llTitle.tvTitle.setText("版本号");
        binding.tvAppVersion.setText("version " + Super.getVersionName(getApplicationContext()));
    }

    /**
     * 检查版本更新
     */
    AppManager.getInstance().getAppVersion(new OkGoCallback<HttpLibResultModel<AppInfo>>() {
        @Override
        public void onSur(Response<HttpLibResultModel<AppInfo>> response) {
            showNoticeDialog(response.body().getData());
        }
        @Override
        public void onErr(String msg) {
        }
    });
}

@Override
protected void initData() {
}

private void showNoticeDialog(AppInfo appInfo) {
    int code = Integer.valueOf(appInfo.getVersionCode());
    int mycode = Super.getVersionCode(getApplicationContext());
    appInfo.setForceUpdate(0);
    if (code <= mycode) {
        binding.tvUpdateVersion.setText("已是最新版本");
        return;
    }
}
```

```

        binding.tvUpdateVersion.setText("更新");
        apkurl = appInfo.getDownloadUrl();
        binding.tvUpdateVersion.setOnClickListener(v -> {
            UpdateManager(apkurl);
        });
    }
    private String apkurl = "";
    //判断版本号，并进行更新
    public void UpdateManager(String apkurl) {
        if (apkurl == null || apkurl.equals("")) {
            showToastShort("未获取到下载地址，请从官网下载");
            return;
        }
        if (Build.VERSION.SDK_INT >= 26) {
            boolean b = getPackageManager().canRequestPackageInstalls();
            if (b && apkurl != "") {
                Intent intent = new Intent(AppVersionActivity.this, DownloadService.class);
                intent.putExtra("LoadUrl", apkurl);
                startService(intent);
            } else {
                Uri packageURI = Uri.parse("package:" + AppVersionActivity.this.getPackageName());
                Intent intent = new Intent(Settings.ACTION_MANAGE_UNKNOWN_APP_SOURCES, packageURI);
                startActivityForResult(intent, INSTALL_PERMISS_CODE);
            }
        } else {
            Intent intent = new Intent(getApplicationContext(), DownloadService.class);
            intent.putExtra("LoadUrl", apkurl);
            startService(intent);
        }
    }
    private final int INSTALL_PERMISS_CODE = 100;
    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (resultCode == RESULT_OK && requestCode == INSTALL_PERMISS_CODE) { //判断是否已获取权限
            Log.i("asdf", "已获取权限");
            Intent intent = new Intent(AppVersionActivity.this, DownloadService.class);
            intent.putExtra("LoadUrl", apkurl);
            startService(intent);
        }
    }
}
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.tm.dmkeep">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.RECORD_AUDIO" />
    <uses-permission android:name="android.permission.CAMERA" />

```

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />

<uses-permission
    android:name="android.permission.READ_LOGS"
    tools:ignore="ProtectedPermissions" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.REQUEST_INSTALL_PACKAGES" />

<queries>
    <!-- 腾讯微信 -->
    <package android:name="com.tencent.mm" />
    <!-- 阿里支付 -->
    <package android:name="com.eg.android.AlipayGphone" />
</queries>

<application
    android:name=".base.MyBaseApplication"
    android:allowBackup="true"
    android:icon="@mipmap/logo"
    android:label="@string/app_name"
    android:resizeableActivity="true"
    android:roundIcon="@mipmap/logo"
    android:supportRtl="true"

    android:theme="@style/AppTheme"
    android:usesCleartextTraffic="true"
    tools:replace="android:allowBackup">
    <activity
        android:name=".ui.activitys.StartActivity"
        android:theme="@style/LaunchTheme">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity
        android:name=".ui.activitys.MainActivity"
        android:launchMode="singleTask"
        android:theme="@style/LaunchTheme"> </activity>
```

```
<!--设置-->
<activity
    android:name=".ui.activitys.SetupActivity"
    android:screenOrientation="portrait" />
<!--app 版本号-->
<activity
    android:name=".ui.activitys.AppVersionActivity"
    android:screenOrientation="portrait" />
<!--社区动态发布-->
<activity
    android:name=".ui.activitys.CommunityPostsActivity"
    android:screenOrientation="portrait" />
<!--课程支付页-->
<activity
    android:name=".ui.activitys.CoursePayActivity"
    android:screenOrientation="portrait" />
<!--钱包列表-->
<activity
    android:name=".ui.activitys.PurseListActivity"
    android:screenOrientation="portrait" />
<!--豆豆列表-->
<activity
    android:name=".ui.activitys.PurseDListActivity"
    android:screenOrientation="portrait" />
<!--vip 开通页面-->
<activity
    android:name=".ui.activitys.VipPayActivity"
    android:screenOrientation="portrait" />
<!--我的评论列表-->
<activity
    android:name=".ui.activitys.MyCommentsListActivity"
    android:screenOrientation="portrait" />
<!--我贴，收藏，点赞-->
<activity
    android:name=".ui.activitys.MyEssayActivity"
    android:screenOrientation="portrait" />
<!--答题列表-->
<activity
    android:name=".ui.activitys.AnswerListActivity"
    android:screenOrientation="portrait" />
<!--答题详情-->
<activity
    android:name=".ui.activitys.AnswerDetailsActivity"
    android:screenOrientation="portrait" />
<!--答题结果-->
<activity
    android:name=".ui.activitys.AnswerResultsActivity"
    android:screenOrientation="portrait" />
<!--教练详情-->
<activity
```

```
        android:name=".ui.activitys.InstructorDetailsActivity"
        android:screenOrientation="portrait" />
<!--学员详情-->
<activity
    android:name=".ui.activitys.StudentsInfoActivity"
    android:screenOrientation="portrait" />
<!--班级详情-->
<activity
    android:name=".ui.activitys.ClassDetailsInfoActivity"
    android:screenOrientation="portrait" />
<!--课程订单详情-->
<activity
    android:name=".ui.activitys.CourseOrderDetailsActivity"
    android:screenOrientation="portrait" />
<activity
    android:name="com.tencent.bugly.beta.ui.BetaActivity"
    android:configChanges="keyboardHidden|orientation|screenSize|locale"
    android:theme="@android:style/Theme.Translucent" />

<service android:name=".service.DownloadService" />
<provider
    android:name=".utils.BuglyFileProvider"
    android:authorities="${applicationId}.fileProvider"
    android:exported="false"
    android:grantUriPermissions="true"
    tools:replace="name,authorities,exported,grantUriPermissions">
    <meta-data
        android:name="android.support.FILE_PROVIDER_PATHS"
        android:resource="@xml/provider_paths"
        tools:replace="name,resource" />
</provider>

<!--学员排行榜-->
<activity android:name=".ui.activitys.ClassListActivity" />
<!--专项动作详情-->
<activity
    android:name=".ui.activitys.ActionsDetailsActivity"
    android:configChanges="orientation|screenSize|keyboardHidden"

    android:screenOrientation="portrait" />
<!--个人中心详情-->
<activity android:name=".ui.activitys.MyDetailsActivity" />
<!--注册用户-->
<activity android:name=".ui.activitys.RegisterActivity" />
<!--登录页面-->
<activity
    android:name=".ui.activitys.LoginActivity"
    android:launchMode="singleTask"
    android:screenOrientation="portrait">
```

```
</activity>
<!--写日记-->
<activity
    android:name=".ui.activitys.EtDiaryActivity"
    android:screenOrientation="portrait" />
<!--话题详情-->
<activity
    android:name=".ui.activitys.TopicInfoActivity"
    android:screenOrientation="portrait" />
<!--圈子-->
<activity
    android:name=".ui.activitys.CircleActivity"
    android:screenOrientation="portrait" />
<activity
    android:name=".ui.activitys.VideoActivity"
    android:configChanges="orientation|screenSize|keyboardHidden"
    android:screenOrientation="portrait" />
<!--搜索页面-->
<activity android:name=".ui.activitys.SearchActivity" />
<!--忘记密码第一步-->
<activity android:name=".ui.activitys.ForgetPassword1Activity"></activity>
<!--忘记密码第二步-->
<activity android:name=".ui.activitys.ForgetPassword2Activity"></activity>
<!--忘记密码第三步-->
<activity android:name=".ui.activitys.ForgetPassword3Activity"></activity>
<!--文章列表-->
<activity android:name=".ui.activitys.EssayListActivity" />
<!--发布问题-->
<activity android:name=".ui.activitys.ReleaseIssueActivity" />
<!--文章详情-->
<activity android:name=".ui.activitys.EssayDetailsActivity" />
<!--日记详情-->
<activity
    android:name=".ui.activitys.CircleDetailsActivity"
    android:screenOrientation="portrait" />
<!--消息列表-->
<activity
    android:name=".ui.activitys.MessageActivity"
    android:screenOrientation="portrait" />
<!--每日跟操练列表-->
<activity
    android:name=".ui.activitys.EveryDayListActivity"
    android:screenOrientation="portrait" />
<!--直播列表-->
<activity
    android:name=".ui.activitys.LiveListActivity"
    android:screenOrientation="portrait" />
<!--每日操练详情-->
<activity
```

```
        android:name=".ui.activitys.EveryDayInfoActivity"
        android:configChanges="orientation|screenSize|keyboardHidden"
        android:screenOrientation="portrait" />
<!--购物车提交订单-->
<activity
    android:name=".ui.activitys.OrdersUpdateActivity"
    android:launchMode="singleTask"
    android:screenOrientation="portrait" />
<activity android:name=".ui.activitys.SlListActivity" />
<!--收货地址列表-->
<activity
    android:name=".ui.activitys.AddressListActivity"
    android:screenOrientation="portrait" />
<!--新建直播-->
<activity
    android:name=".ui.activitys.LiveNewActivity"
    android:screenOrientation="portrait" />
<!--地址新增，修改-->
<activity
    android:name=".ui.activitys.AddressNewActivity"
    android:screenOrientation="portrait" />
<!--订单列表-->
<activity
    android:name=".ui.activitys.OrderListActivity"
    android:screenOrientation="portrait" />
<!--课程订单列表-->
<activity
    android:name=".ui.activitys.CourseOrderActivity"
    android:screenOrientation="portrait" />
<!--订单详情-->
<activity
    android:name=".ui.activitys.OrderDetailsActivity"
    android:launchMode="singleTask"
    android:screenOrientation="portrait" />
<!--物流信息-->
<activity
    android:name=".ui.activitys.LogisticsDetailsActivity"
    android:screenOrientation="portrait" />
<!--支付信息-->
<activity
    android:name=".ui.activitys.PayActivity"
    android:launchMode="singleTask"
    android:screenOrientation="portrait" />
<!--支付结果-->
<activity
    android:name=".ui.activitys.PayResultsActivity"
    android:launchMode="singleTask"
    android:screenOrientation="portrait" />
<activity
    android:name=".ui.activitys.SlDetailsActivity"
```



```
        android:configChanges="orientation|screenSize|keyboardHidden"
        android:screenOrientation="portrait" />
<!-- 答疑列表 -->
<activity android:name=".ui.activitys.QuestionListActivity" />
<!-- 答疑详情 -->
<activity android:name=".ui.activitys.QuestionDetailsActivity" />
<!-- 招生 -->
<activity android:name=".ui.activitys.RecrutStudentsActivity" />
<!-- 我的圈子 -->
<activity
    android:name=".ui.activitys.CircleMyActivity"
    android:screenOrientation="portrait" />
<!-- 课程详情 -->
<activity
    android:name=".ui.activitys.CourseDetailsActivity"
    android:configChanges="orientation|screenSize|keyboardHidden"
    android:supportsPictureInPicture="true"
    android:screenOrientation="portrait" />
<!-- 课程详情分类列表 -->
<activity
    android:name=".ui.activitys.CourseClassDetailsActivity"
    android:screenOrientation="portrait" />
<!-- 班级详情 -->
<activity android:name=".ui.activitys.MyClassDetailsActivity" />
<!-- 商城详情 -->
<activity
    android:name=".ui.activitys.MallDetailsActivity"
    android:screenOrientation="portrait" />
<!-- 购物车 -->
<activity
    android:name=".ui.activitys.ShoppingCarActivity"
    android:screenOrientation="portrait" />
<!-- 视频详情 -->
<activity
    android:name=".ui.activitys.VideoDetailsActivity"
    android:configChanges="orientation|screenSize|keyboardHidden"
    android:screenOrientation="portrait" />
<!-- 直播结束 -->
<activity
    android:name=".ui.activitys.LiveEndActivity"
    android:screenOrientation="portrait" />
<!-- 课程详情支付页面 -->
<activity
    android:name=".ui.activitys.CoursePayDetailsActivity"
    android:configChanges="orientation|screenSize|keyboardHidden"
    android:screenOrientation="portrait" />
<!-- QQ -->
<activity
    android:name="com.tencent.tauth.AuthActivity"
    android:launchMode="singleTask"
```

```
        android:noHistory="true">
        <intent-filter>
            <action android:name="android.intent.action.VIEW" />

            <category android:name="android.intent.category.DEFAULT" />
            <category android:name="android.intent.category.BROWSABLE" />

            <data android:scheme="tencent101999197" />
        </intent-filter>
    </activity>
    <activity
        android:name="com.tencent.connect.common.AssistActivity"
        android:configChanges="orientation|keyboardHidden"
        android:screenOrientation="behind"
        android:theme="@android:style/Theme.Translucent.NoTitleBar" />
    <activity
        android:name=".wxapi.WXEntryActivity"
        android:exported="true"
        android:label="@string/app_name"
        android:launchMode="singleTask"
        android:taskAffinity="com.tm.dmkeep"
        android:theme="@android:style/Theme.Translucent.NoTitleBar"
        tools:ignore="InnerclassSeparator" />
    <activity
        android:name=".wxapi.WXPayEntryActivity"
        android:exported="true"
        android:launchMode="singleTask" />

    <meta-data
        android:name="WX_APP_ID"
        android:value="wxb12dcf36bffa9702">

    </meta-data>

    <provider
        android:name="androidx.core.content.FileProvider"
        android:authorities="com.tm.dmkeep.fileprovider"
        android:exported="false"
        android:grantUriPermissions="true">
        <meta-data
            android:name="android.support.FILE_PROVIDER_PATHS"
            android:resource="@xml/file_paths" />
    </provider>
    <uses-library
        android:name="org.apache.http.legacy"
        android:required="false" />
    <!--适配华为 ( huawei ) 刘海屏-->
    <meta-data
        android:name="android.notch_support"
        android:value="true" />
    <!--适配小米 ( xiaomi ) 刘海屏-->
```

```
<meta-data
    android:name="notch.config"
    android:value="portrait|landscape" />
<!--登录注册-->
<activity
    android:name=".ui.activitys.LoginRegisterActivity"
    android:screenOrientation="portrait" />
<!--绑定手机号-->
<activity
    android:name=".ui.activitys.PhoneBindActivity"
    android:screenOrientation="portrait" />
<!--      修改用户名-->
<activity
    android:name=".ui.activitys.EtUserNameActivity"
    android:screenOrientation="portrait" />
<!-- 修改密码-->
<activity
    android:name=".ui.activitys.RevisePasswordActivity"
    android:screenOrientation="portrait" />
<!-- 绑定手机号-->
<activity
    android:name=".ui.activitys.PhoneBindCodeActivity"
    android:screenOrientation="portrait" />
<!--秒验一键登录-->
<!--      <activity-->
<!--      android:name="com.mob.tools.MobUIShell"-->
<!--      android:configChanges="keyboardHidden|orientation|screenSize"-->
<!--      android:launchMode="standard"-->
<!--      android:theme="@android:style/Theme.Translucent.NoTitleBar" /> -->
</application>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<!--访问网络连接,可能产生 GPRS 流量-->
<uses-permission android:name="android.permission.INTERNET" />
<!--获取网络信息状态,如当前的网络连接是否有效-->
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<!--改变 WiFi 状态-->
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<!--获取当前 WiFi 接入的状态以及 WLAN 热点的信息-->
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<!--允许程序写入外部存储,如 SD 卡上写文件-->
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
</manifest>
plugins {
    id 'com.android.application'
}
apply from: 'tinker-support.gradle'
android {
    compileSdkVersion 30
    buildToolsVersion "30.0.3"
    defaultConfig {
```

```

        applicationId "com.tm.dmkeep"
        minSdkVersion 21
        targetSdkVersion 30
        versionCode 15
        versionName "1.5"
        multiDexEnabled true
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
    dataBinding {
        enabled = true
    }
    dexOptions {
        jumboMode = true
        javaMaxHeapSize "4g"
    }
    // APK 签名
    signingConfigs {
        release {
        }
    }
    buildTypes {
        release {
            signingConfig signingConfigs.release
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
            ndk {
                abiFilters 'armeabi','armeabi-v7a','mips','arm64-v8a'
            }
        }
        debug {
            signingConfig signingConfigs.release
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
            ndk {
                abiFilters 'armeabi','armeabi-v7a','mips','arm64-v8a'
            }
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
    android.applicationVariants.all {
        variant ->
        variant.outputs.all {
            //在这里修改 apk 文件名
            outputFileName
            =
            "android_dmKeep_v${variant.versionCode}_${variant.versionName}_${variant.name}_${releaseTime()}.apk"
        }
    }
}

```

```
}

def releaseTime() {
    new Date().format("MMdd", TimeZone.getTimeZone("GMT+08:00"))
}

// 调用 MobTech SDK
apply plugin: 'com.mob.sdk'

MobSDK {
    appKey "35bba3bffd7b7"
    MobPush {}
    SecVerify {}
}

dependencies {
    implementation fileTree(include: ['*.jar', '*.aar'], dir: 'libs')
    implementation 'androidx.appcompat:appcompat:1.3.1'
    implementation 'com.google.android.material:material:1.1.0'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
    api project(':baselibrary')
    implementation 'com.tencent.tauth:qqopensdk:3.52.0'
    //dialogplus
    implementation 'com.orhanobut:dialogplus:1.11@aar'
    //阿里云直播
    def LATEST_VERSION = '2.0.0.20221010001'
    api "com.aliyun.roompaas.biz:liveroom:$LATEST_VERSION"
    api "com.aliyun.roompaas.biz:rtc:$LATEST_VERSION"
    api "com.aliyun.roompaas.biz:beauty_pro:$LATEST_VERSION"
    //    bugly 热更新
    api "com.android.support:multidex:2.0.0" // 多 dex 配置
    api 'com.tencent.bugly:crashreport_upgrade:1.5.23'
    api 'com.tencent.tinker:tinker-android-lib:1.9.14.24'
    //socket
    implementation 'org.java-websocket:Java-WebSocket:1.4.0'
    //点赞效果
    api 'com.sackcentury:shinebutton:1.0.0'
    //视频压缩
    implementation 'com.icetec:silicompressor:2.2.1'
}
```