

/\*\*  
 \* 移动开发学习系统启动页面  
 \*/

public class SplashActivity extends Activity {

@Override

protected void onCreate(Bundle savedInstanceState) {

//淡入淡出效果

overridePendingTransition(android.R.anim.*fade\_in*, android.R.anim.*fade\_out*);

super.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_splash*);

RelativeLayout rl\_splash = findViewById(R.id.*rl\_splash*);

//透明动画

AlphaAnimation alphaAnimation = new AlphaAnimation(0.3f, 1.0f);

alphaAnimation.setDuration(1000);

alphaAnimation.setAnimationListener(new Animation.AnimationListener() {

@Override

public void onAnimationStart(Animation animation) {

}

@Override

public void onAnimationEnd(Animation animation) {

startActivity(new Intent(SplashActivity.this, MainActivity.class));

finish();

overridePendingTransition(android.R.anim.*fade\_in*,android.R.anim.*fade\_out*);

}

@Override

public void onAnimationRepeat(Animation animation) {

}

});

rl\_splash.startAnimation(alphaAnimation);

}

}

/\*\*

\*移动开发学习系统布局文件

\*/

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:id="@+id/rl\_splash"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<TextView

android:id="@+id/tv\_splash"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:layout\_centerHorizontal="true"

android:layout\_marginBottom="30dp"

android:textSize="18sp"

android:textColor="@color/black"

android:text=" @string/welcome\_word

"/>

</RelativeLayout>

public abstract class AbstractSimpleActivity extends SupportActivity {  
  
 private Unbinder unbinder;  
 private static final String *TAG* = "AbstractSimpleActivity";  
  
 @Override  
 protected void onCreate(@Nullable Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(getLayoutId());  
 //沉浸式状态栏  
 ImmersionBar.*with*(this)  
 .statusBarView(findViewById(R.id.*status\_bar*))  
 .statusBarDarkFont(true)  
 .keyboardEnable(true)  
 .init();  
 unbinder = ButterKnife.*bind*(this);  
 Log.*d*(*TAG*, "onCreate: ");  
 onViewCreate();  
 initToolbar();  
 initView();  
 initEventAndData();  
 }  
  
 protected abstract int getLayoutId();  
 protected abstract void onViewCreate();  
 protected abstract void initToolbar();  
 protected abstract void initView();  
 protected abstract void initEventAndData();  
  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
 ImmersionBar.*with*(this).destroy();  
 if (unbinder != null && unbinder != Unbinder.*EMPTY*) {  
 unbinder.unbind();  
 unbinder = null;  
 }  
 }

}

/\*\*  
 \* 每个Activity都需要继承此类  
 \* @param<T>  
 \*/

public abstract class BaseActivity<T extends BasePresenter> extends AbstractSimpleActivity  
 implements HasSupportFragmentInjector, BaseView {  
 private static final String *TAG* = "BaseActivity";  
  
 @Inject  
 protected T mPresenter;  
 @Inject  
 DispatchingAndroidInjector<Fragment> dispatchingAndroidInjector;  
  
 @Override  
 protected void onCreate(@Nullable Bundle savedInstanceState) {  
 AndroidInjection.*inject*(this); //实现 Activity 的注入  
 super.onCreate(savedInstanceState);  
 Log.*d*(*TAG*, "onCreate: ");  
 }  
 /\*\*  
 \* 父类onCreate调用onViewCreate  
 \* 先执行父类onCreate，在执行子类实现的 onViewCreate  
 \* 最后执行子类的 onCreate  
 \*/@Override  
 protected void onViewCreate() {  
 Log.*d*(*TAG*, "onViewCreate: onViewCreate");  
 if (mPresenter != null) {  
 mPresenter.attachView(this);  
 }  
 }  
  
 @Override  
 public AndroidInjector<Fragment> supportFragmentInjector() {  
 return dispatchingAndroidInjector;  
 }  
  
 @Override  
 public void showErrorMessage(String message) {  
  
 }  
  
 @Override  
 public void showLoading() {  
  
 }  
  
 @Override  
 public void hideLoading() {  
  
 }  
  
 @Override  
 public void showError() {  
  
 }  
  
 @Override  
 public void showNoNetwork() {  
  
 }  
  
 @Override  
 public void showEmpty() {  
  
 }  
  
 @Override  
 public void showContent() {  
  
 }  
  
 @Override  
 public void handleLoginSuccess() {  
  
 }  
  
 @Override  
 public void handleLogoutSuccess() {  
  
 }  
}

/\*\*  
 \* View 基类接口  
 \*/public interface BaseView {  
 void showErrorMessage(String message);  
 void showLoading();  
 void hideLoading();  
 void showError();  
  
 void showNoNetwork();  
 void showEmpty();  
 void showContent();  
 void handleLoginSuccess();  
  
 void handleLogoutSuccess();  
}

/\*\*  
 \* Presenter 基类接口  
 \* @param <T>  
 \*/public interface BasePresenter<T extends BaseView> {  
  
 void attachView(T view);  
 void detachView();  
 void reload();  
 void registerEventBus();  
 void unregisterEventBus();  
  
 void setLoginStatus(boolean loginStatus);  
 boolean getLoginStatus();  
 String getLoginAccount();  
 void setLoginAccount(String account);  
  
}

/\*\*  
 \* 主页面  
 \*/public class MainActivity extends BaseActivity<MainPresenter> implements MainContract.View {  
  
 private static final String *TAG* = "MainActivity";  
 @BindView(R.id.*drawer\_layout*)  
 DrawerLayout drawer\_layout;  
 @BindView(R.id.*toolbar*)  
 Toolbar toolbar;  
 @BindView(R.id.*toolbar\_title*)  
 TextView toolbar\_title;  
 @BindView(R.id.*navigation\_view*)  
 NavigationView navigation\_view;  
 @BindView(R.id.*bottom\_navigation\_view*)  
 BottomNavigationView bottom\_navigation\_view;  
  
 private int currentFgIndex = 0;  
 private int lastFgIndex = -1;  
  
 private HomePageFragment homePageFragment;  
 private NavigationFragment navigationFragment;  
 private KnowledgeFragment knowledgeFragment;  
 private PublicNumFragment publicNumFragment;  
 private ProjectFragment projectFragment;  
 private TextView tv\_welcome;  
 private TextView tv\_login\_account;  
 private long clickTime;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 Log.*d*(*TAG*, "onCreate: ");  
 }  
  
 protected int getLayoutId() {  
 return R.layout.*activity\_main*;  
 }  
  
 /\*\*  
 \* 初始化 Toolbar  
 \*/@Override  
 protected void initToolbar() {  
 setSupportActionBar(toolbar);  
 ActionBar actionBar = getSupportActionBar();  
 if (actionBar != null) {  
 actionBar.setDisplayShowTitleEnabled(false);  
 toolbar\_title.setText(R.string.*home\_page*);  
 }

getWindow().setNavigationBarColor(getResources().getColor(R.colo r.*navigation\_bar*));

}  
  
 @Override  
 protected void initView() {  
 initDrawerLayout();  
 showFragment(currentFgIndex);  
 initNavigationView();  
 initBottomNavigationView();  
  
 }  
  
 /\*\*  
 \* 初始化底部导航栏  
 \*/private void initBottomNavigationView() {  
 bottom\_navigation\_view.setOnNavigationItemSelectedListener(new

BottomNavigationView.OnNavigationItemSelectedListener() {  
 @Override  
 public boolean onNavigationItemSelected(@NonNull MenuItem

menuItem) {  
 switch (menuItem.getItemId()) {  
 case R.id.*tab\_home\_pager*:  
 showFragment(Constants.*FRAGMENT\_HOME\_PAGE*);  
 break;  
 case R.id.*tab\_knowledge*:  
 showFragment(Constants.*FRAGMENT\_KNOWLEDGE*);  
 break;  
 case R.id.*tab\_navigation*:  
 showFragment(Constants.*FRAGMENT\_NAIVIGATION*);  
 break;  
 case R.id.*tab\_project*:  
 showFragment(Constants.*FRAGMENT\_PRPJECT*);  
 break;  
 case R.id.*tab\_public\_num*:  
 showFragment(Constants.*FRAGMENT\_PUBLIC\_NUM*);  
 break;  
 }  
 return true;  
 }  
 });  
 }  
  
 private void initNavigationView() {  
 navigation\_view.setNavigationItemSelectedListener(new

NavigationView.OnNavigationItemSelectedListener() {  
 @Override  
 public boolean onNavigationItemSelected(@NonNull MenuItem

menuItem) {  
 switch (menuItem.getItemId()) {  
 case R.id.*item\_login*:  
 startLoginActivity();  
 break;  
 case R.id.*item\_collect*:  
 if (mPresenter.getLoginStatus()) {   
 Intent intent = new

Intent(MainActivity.this, ItemActivity.class);  
 intent.putExtra("Type",

Constants.*TYPE\_COLLECT*);  
 startActivity(intent);  
 } else {  
 startLoginActivity(); //未登录跳转登录页  
 Toast.*makeText*(MainActivity.this,

getString(R.string.*please\_login*),  
 Toast.*LENGTH\_SHORT*).show();  
 }  
 break;  
 case R.id.*item\_about*:  
 Intent intentAbout = new

Intent(MainActivity.this, ItemActivity.class);  
 intentAbout.putExtra("Type",

Constants.*TYPE\_ABOUT*);  
 startActivity(intentAbout);  
 break;  
 case R.id.*item\_logout*:  
 mPresenter.logout();  
 break;  
 }  
 return true;  
 }  
 });  
 tv\_welcome = navigation\_view.getHeaderView(0)

.findViewById(R.id.*tv\_welcome*);  
tv\_login\_account = navigation\_view.getHeaderView(0)

.findViewById(R.id.*tv\_login\_account*);

if (mPresenter.getLoginStatus()) {  
 tv\_welcome.setVisibility(View.*VISIBLE*);  
 tv\_login\_account.setText(mPresenter.getLoginAccount());

navigation\_view.getMenu().findItem(R.id.*item\_login*)

.setVisible(false); navigation\_view.getMenu().findItem(R.id.*item\_logout*)

.setVisible(true);  
} else {  
 tv\_welcome.setVisibility(View.*INVISIBLE*);  
 tv\_login\_account.setText("");

navigation\_view.getMenu().findItem(R.id.*item\_login*)

.setVisible(true); navigation\_view.getMenu().findItem(R.id.*item\_logout*)

.setVisible(false);

}

}  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*toolbar\_menu*, menu);  
 return super.onCreateOptionsMenu(menu);  
 }  
  
 /\*\*  
 \* Toolbar 两个按钮  
 \* @param item  
 \* @return\*/@Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 Intent intent;  
 switch (item.getItemId()) {  
 case R.id.*action\_usage*:  
 intent=new Intent(MainActivity.this, ItemActivity.class);  
 intent.putExtra("Type", Constants.*TYPE\_WEBSITE*);  
 startActivity(intent);  
 break;  
 case R.id.*action\_search*:

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

startActivity(intent);  
 break;  
 default:  
 break;  
 }  
 return true;  
 }  
  
 /\*\*  
 \* 展示对应 Fragment  
 \* @paramindex  
 \*/private void showFragment(int index) {  
 currentFgIndex = index;  
 FragmentTransaction transaction = getSupportFragmentManager()

.beginTransaction();

hideFragment(transaction);  
 lastFgIndex = index;  
 switch (index) {  
 case Constants.*FRAGMENT\_HOME\_PAGE*:  
 toolbar\_title.setText(R.string.*home\_page*);  
 if (homePageFragment == null) {  
 homePageFragment = HomePageFragment.*newInstance*();  
 transaction.add(R.id.*fragment\_root*,

this.homePageFragment);  
 }  
 transaction.show(this.homePageFragment);  
 break;  
 case Constants.*FRAGMENT\_PUBLIC\_NUM*:  
 toolbar\_title.setText(R.string.*public\_num*);  
 if (publicNumFragment == null) {  
 publicNumFragment = PublicNumFragment.*newInstance*();  
 transaction.add(R.id.*fragment\_root*,

this.publicNumFragment);  
 }  
 transaction.show(this.publicNumFragment);  
 break;  
 case Constants.*FRAGMENT\_KNOWLEDGE*:  
 toolbar\_title.setText(R.string.*knowledge*);  
 if (knowledgeFragment == null) {  
 knowledgeFragment = KnowledgeFragment.*newInstance*();  
 transaction.add(R.id.*fragment\_root*, this.knowledgeFragment);  
 }  
 transaction.show(this.knowledgeFragment);  
 break;  
 case Constants.*FRAGMENT\_PRPJECT*:  
 toolbar\_title.setText(R.string.*project*);  
 if (projectFragment == null) {  
 projectFragment = ProjectFragment.*newInstance*();  
 transaction.add(R.id.*fragment\_root*,

this.projectFragment);  
 }  
 transaction.show(this.projectFragment);  
 break;  
 case Constants.*FRAGMENT\_NAIVIGATION*:  
 toolbar\_title.setText(R.string.*navigation*);  
 if (navigationFragment == null) {  
 navigationFragment =

NavigationFragment.*newInstance*();  
 transaction.add(R.id.*fragment\_root*,

this.navigationFragment);  
 }  
 transaction.show(this.navigationFragment);  
 break;  
 default:  
 break;  
 }  
 transaction.commit();  
 }  
  
 private void hideFragment(FragmentTransaction transaction) {  
 switch (lastFgIndex) {  
 case Constants.*FRAGMENT\_HOME\_PAGE*:  
 if (homePageFragment != null) {  
 transaction.hide(homePageFragment);  
 }  
 break;  
 case Constants.*FRAGMENT\_NAIVIGATION*:  
 if (navigationFragment != null) {  
 transaction.hide(navigationFragment);  
 }  
 break;  
 case Constants.*FRAGMENT\_KNOWLEDGE*:  
 if (knowledgeFragment != null) {  
 transaction.hide(knowledgeFragment);  
 }  
 break;  
 case Constants.*FRAGMENT\_PUBLIC\_NUM*:  
 if (publicNumFragment != null) {  
 transaction.hide(publicNumFragment);  
 }  
 break;  
 case Constants.*FRAGMENT\_PRPJECT*:  
 if (projectFragment != null) {  
 transaction.hide(projectFragment);  
 }  
 break;  
 default:  
 break;  
 }  
 }  
  
 private void initDrawerLayout() {  
 ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(this,

drawer\_layout, toolbar,

R.string.*open\_drawer*, R.string.*close\_drawer*);  
 toggle.syncState();  
 drawer\_layout.addDrawerListener(toggle);  
 }  
  
 private void startLoginActivity() {  
 Intent intent = new Intent(this, LoginActivity.class);  
 startActivity(intent);  
 }  
  
  
 @Override  
 protected void initEventAndData() {  
  
 }  
  
  
 @Override  
 public void showErrorMessage(String message) {  
  
 }  
  
 @Override  
 public void showLoading() {  
  
 }  
  
 @Override  
 public void hideLoading() {  
  
 }  
  
 @Override  
 public void showError() {  
  
 }  
  
 @Override  
 public void showNoNetwork() {  
  
 }  
  
 @Override  
 public void showEmpty() {  
  
 }  
  
 @Override  
 public void showContent() {  
  
 }  
  
 @Override  
 public void handleLoginSuccess() {  
 tv\_welcome.setVisibility(View.*VISIBLE*);  
 tv\_login\_account.setText(mPresenter.getLoginAccount());  
 //隐藏登录按钮

navigation\_view.getMenu().findItem(R.id.*item\_login*)

.setVisible(false);  
 //显示注销按钮

navigation\_view.getMenu().findItem(R.id.*item\_logout*)

.setVisible(true);  
 }  
  
 @Override  
 public void handleLogoutSuccess() {  
 tv\_welcome.setVisibility(View.*INVISIBLE*);  
 tv\_login\_account.setText("");

navigation\_view.getMenu().findItem(R.id.*item\_login*)

.setVisible(true);

navigation\_view.getMenu().findItem(R.id.*item\_logout*)

.setVisible(false);  
Toast.*makeText*(this,getString(R.string.*logout\_success*)

,Toast.*LENGTH\_SHORT*).show();

}  
 */*\*\*  
 \* 处理回退事件  
 \*/@Override  
 public void onBackPressedSupport() {  
 if (getSupportFragmentManager().getBackStackEntryCount() > 1) {  
 pop();  
 } else {  
 long currentTime = System.*currentTimeMillis*();  
 if((currentTime - clickTime) > Constants.*DOUBLE\_INTERVAL\_TIME*)

{  
 Toast.*makeText*(MainActivity.this,

getString(R.string.*double\_click\_exit\_toast*)

,Toast.*LENGTH\_SHORT*).show();  
 clickTime = System.*currentTimeMillis*();  
 } else {  
 finish();  
 }  
 }  
 }  
}

/\*\*

\* Toolbar 布局文件

\*/

<?xml version="1.0" encoding="utf-8"?>

<!--支持响应滚动手势的 bar-->

<android.support.design.widget.AppBarLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:elevation='0dp'>

<View

android:id="@+id/status\_bar"

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:background="@color/Grey250"/>

<android.support.v7.widget.Toolbar

android:id="@+id/toolbar"

style="@style/ToolBarStyle"

android:layout\_width="match\_parent"

android:layout\_height="?attr/actionBarSize"

android:background="@color/Grey250"

android:gravity="center\_vertical"

app:contentInsetStart="0dp"

app:contentInsetEnd="0dp"

app:contentInsetStartWithNavigation="0dp"

app:layout\_scrollFlags="scroll|enterAlways|snap">

<ImageView

android:id="@+id/toolbar\_back"

android:layout\_width="24dp"

android:layout\_height="24dp"

android:layout\_marginLeft="15dp"

android:layout\_marginRight="10dp"

android:src="@drawable/ic\_left\_black"

android:visibility="gone"/>

<TextView

android:id="@+id/toolbar\_title"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="start"

android:ellipsize="marquee"

android:singleLine="true"

android:textSize="16sp"

android:textStyle="bold"

android:textColor="@color/black"

/>

</android.support.v7.widget.Toolbar>

</android.support.design.widget.AppBarLayout>

/\*\*

\* 主页面布局文件

\*/

<?xml version="1.0" encoding="utf-8"?>

<android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/drawer\_layout"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

tools:context=".MainActivity">

<android.support.design.widget.CoordinatorLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<include layout="@layout/toolbar" />

<FrameLayout

android:id="@+id/fragment\_root"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior">

</FrameLayout>

<android.support.design.widget.BottomNavigationView

android:id="@+id/bottom\_navigation\_view"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="bottom"

android:background="@color/white"

app:elevation="10dp"

app:itemIconTint="@drawable/navigation\_bottom\_color"

app:itemTextColor="@drawable/navigation\_bottom\_color"

app:labelVisibilityMode="labeled"

app:menu="@menu/bottom\_navigation">

</android.support.design.widget.BottomNavigationView>

</android.support.design.widget.CoordinatorLayout>

<android.support.design.widget.NavigationView

android:id="@+id/navigation\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_gravity="start"

android:background="@color/main\_white"

app:headerLayout="@layout/navigation\_header"

app:insetForeground="@android:color/transparent"

app:itemIconTint="@color/navigation\_item\_ic\_color"

app:itemTextColor="@color/black"

app:menu="@menu/navigation\_view">

</android.support.design.widget.NavigationView>

</android.support.v4.widget.DrawerLayout>

public abstract class AbstractSimpleFragment extends SupportFragment {

private Unbinder unbinder;

private static final String TAG = "AbstractSimpleFragment";

@Override

public void onAttach(Activity activity) {

Log.d(TAG, "onAttach: ");

super.onAttach(activity);

}

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

View view = inflater.inflate(getLayoutId(),container,false);

Log.d(TAG, "onCreateView1: ");

unbinder = ButterKnife.bind(this,view);

initView();

Log.d(TAG, "onCreateView: ");

return view;

}

@Override

public void onLazyInitView(@Nullable Bundle savedInstanceState) {

super.onLazyInitView(savedInstanceState);

initData();

}

@Override

public void onDestroyView() {

super.onDestroyView();

if (unbinder != null && unbinder != Unbinder.EMPTY ) {

unbinder.unbind();

unbinder = null;

}

}

/\*\*

\*Fragment 销毁时进行内存泄漏检测

\*/

@Override

public void onDestroy() {

super.onDestroy();

RefWatcher refWatcher = MobileLearnApp.getRefWatcher(\_mActivity);

refWatcher.watch(this);

}

protected abstract int getLayoutId();

protected abstract void initView();

protected abstract void initData();

}

/\*\*

\* 每个Fragment都需要继承此类

\* @param <T>

\*/

public abstract class BaseFragment<T extends BasePresenter> extends AbstractSimpleFragment implements BaseView {

private static final String TAG = "BaseFragment";

@Inject

protected T presenter;

private MultipleStatusView custom\_multiple\_view;

private LoadingDialog dialog;

@Override

public void onAttach(Activity activity) {

Log.d(TAG, "onAttach: ");

AndroidSupportInjection.inject(this);

//必须在 onAttach 中完成

super.onAttach(activity);

}

@Override

public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {

Log.d(TAG, "onViewCreated: ");

super.onViewCreated(view, savedInstanceState);

custom\_multiple\_view = view.findViewById(R.id.custom\_multiple\_view);

if (presenter != null) {

presenter.attachView(this);

}

if (custom\_multiple\_view != null) {

custom\_multiple\_view.setOnRetryClickListener(v -> presenter.reload());

}

}

@Override

public void onDetach() {

super.onDetach();

if (presenter != null) {

presenter = null;

}

}

@Override

public void onDestroyView() {

super.onDestroyView();

if (presenter != null) {

presenter.detachView();

}

}

@Override

public void showErrorMessage(String message) {

Toast.makeText(\_mActivity, message, Toast.LENGTH\_SHORT).show();

}

/\*\*

\* 弹出加载框

\*/

@Override

public void showLoading() {

System.out.println("showdialog");

if (custom\_multiple\_view == null) {

return;

} else {

System.out.println("showdialog");

if (dialog == null) {

dialog = new LoadingDialog(\_mActivity, getString(R.string.loading\_data),

R.mipmap.ic\_dialog\_loading);

}

dialog.show();

}

}

/\*\*

\* 隐藏加载框

\*/

@Override

public void hideLoading() {

if (dialog != null) {

System.out.println("hidedialog");

dialog.dismiss();

dialog = null;

}

}

@Override

public void showError() {

if (custom\_multiple\_view == null) return;

custom\_multiple\_view.showError();

}

@Override

public void showNoNetwork() {

if (custom\_multiple\_view == null) return;

custom\_multiple\_view.showNoNetwork();

}

@Override

public void showEmpty() {

if (custom\_multiple\_view == null) return;

custom\_multiple\_view.showEmpty();

}

@Override

public void showContent() {

if (custom\_multiple\_view == null) {

return;

} else {

custom\_multiple\_view.showContent();

}

}

@Override

public void handleLoginSuccess() {

}

@Override

public void handleLogoutSuccess() {

}

}

public interface CollectContract {

interface View extends BaseView {

void showCollectSuccess(int position);

void showCancelCollect(int position);

}

interface Presenter<V extends View> extends BasePresenter<V> {

void addCollectArticle(int position, int id);

void cancelAddCollectArticle(int position, int id);

}

}

/\*\*

\* 关于收藏相关

\*/

public class CollectPresenter<V extends CollectContract.View> extends Presenter<V>implements CollectContract.Presenter<V> {

@Override

public void addCollectArticle(int position, int id) {

addSubscribe(dataManager.addArticleCollect(id)

.compose(RxJavaUtil.SchedulerTransformer()) //调度器

.filter(articleListData -> mView != null) //添加过滤

.subscribeWith(new BaseObserver<ArticleListData>(mView,

MobileLearnApp.getContext().getString

(R.string.collect\_failed),false) {

@Override

public void success(ArticleListData articleListData) {

mView.showCollectSuccess(position);

}

}));

}

@Override

public void cancelAddCollectArticle(int position, int id) {

addSubscribe(dataManager.cancelAddArticleCollect(id)

.compose(RxJavaUtil.SchedulerTransformer())

.filter(articleListData -> mView != null)

.subscribeWith(new BaseObserver<ArticleListData>(mView,

MobileLearnApp.getContext().getString

(R.string.failed\_to\_cancel\_collect)) {

@Override

public void success(ArticleListData articleListData) {

mView.showCancelCollect(position);

}

}));

}

}

/\*\*

\* 在线学习模块相关调用

\*/

Public class HomepagePresenter extends CollectPresenter<HomepageContract

.View implements HomepageContract.Presenter {

private int currentPage;

private boolean isRefresh;

@Inject

HomepagePresenter() {

}

@Override

public void getArticleData(boolean isShowStatusView) {

addSubscribe(dataManager.getArticleList(currentPage)

.compose(RxJavaUtil.SchedulerTransformer())

.filter(articleListData -> mView != null)

.subscribeWith(new BaseObserver<ArticleListData>(mView, MobileLearnApp .getContext().getString(R.string.failed\_to\_get\_article), isShowStatusView) {

@Override

public void success(ArticleListData articleListData) {

mView.showList(articleListData, isRefresh);

}

}));

}

/\*\*

\* 获取首页的 Banner 数据

\* @param isShowStatusView

\*/

@Override

public void getBannerData(boolean isShowStatusView) {

addSubscribe(dataManager.getBannerData()

.compose(RxJavaUtil.SchedulerTransformer())

.filter(articleBannerData -> mView != null)

.subscribeWith(new BaseObserver<List<BannerData>>(mView,

MobileLearnApp.getContext().getString(R.string.failed\_to\_get\_banner),isShowStatusView) {

@Override

public void success(List<BannerData> bannerData) {

mView.showBannerData(bannerData);

}

}));

}

/\*\*

\* zip() 一个函数组合多个 Observable 发射的数据集合

\* 可将多个请求结合

\* lambda 接收两个参数

\* subscribeWith 一般使用 Rx 请求接口的这种情况会用

\* @param isShowStatusView

\*/

@Override

public void getHomepageData(boolean isShowStatusView) {

getBannerData(isShowStatusView);

addSubscribe(Observable.zip(dataManager.getUpArticles(), dataManager.getArticleList(0),

(upArticlesResponse, articleListResponse) -> {

articleListResponse.getData().getDatas().addAll(0,

upArticlesResponse.getData());

return articleListResponse;

})

.compose(RxJavaUtil.SchedulerTransformer())

.filter(articleListData -> mView != null)

.subscribeWith(new BaseObserver<ArticleListData>(mView, MobileLearnApp.getContext().getString(R.string.failed\_to\_get\_article), isShowStatusView) {

@Override

public void success(ArticleListData articleListData) {

mView.showList(articleListData, isRefresh);

}

}));

}

/\*\*

\* 下拉刷新

\* @param isShowStatusView

\*/

@Override

public void refreshLayout(boolean isShowStatusView) {

isRefresh = true;

currentPage = 0;

getHomepageData(isShowStatusView);

}

/\*\*

\* 下拉加载更多

\*/

@Override

public void loadMore() {

isRefresh = false;

currentPage++;

getArticleData(false);

}

/\*\*

\* 注销成功后刷新数据

\* @param logoutEvent

\*/

@Subscriber()

public void logoutSuccessEvent(LogoutEvent logoutEvent) {

getHomepageData(false);

}

/\*\*

\* 登录成功后刷新数据

\* @param loginEvent

\*/

@Subscriber()

public void loginSuccessEvent(LoginEvent loginEvent) {

getHomepageData(false);

}

@Override

public void registerEventBus() {

EventBus.getDefault().register(this);

}

@Override

public void unregisterEventBus() {

EventBus.getDefault().unregister(this);

}

}

/\*\*

\* 在线学习模块 Fragment

\*/

public class HomePageFragment extends BaseFragment<HomepagePresenter> implements HomepageContract.View {

private static final String TAG = "HomePageFragment";

@BindView(R.id.hp\_recycler\_view)

RecyclerView hpRecyclerView;

@BindView(R.id.hp\_refresh\_layout)

SmartRefreshLayout hpSmartRefreshLayout;

private Banner banner;

private ArticleListAdapter adapter;

public static HomePageFragment newInstance() {

HomePageFragment homePageFragment = new HomePageFragment();

return homePageFragment;

}

@Override

protected int getLayoutId() {

return R.layout.fragment\_homepage;

}

@Override

protected void initView() {

List<ArticleItemData> articleList = new ArrayList<>();

Adapter = new ArticleListAdapter

(R.layout.item\_article.articleList)

adapter.setOnItemClickListener(new BaseQuickAdapter.OnItemClickListener() {

@Override

public void onItemClick(BaseQuickAdapter adapter, View view, int position) {

openArticleDetail(view, position);

}

});

adapter.setOnItemChildClickListener(new BaseQuickAdapter.OnItemChildClickListener() {

@Override

public void onItemChildClick(BaseQuickAdapter adapter, View view, int position) {

childOnClick(view, position);

}

});

hpRecyclerView.setLayoutManager(new LinearLayoutManager(\_mActivity));

DividerItemDecorationdivider = newDividerItemDecoration

(\_mActivity,DividerItemDecoration.VERTICAL);

divider.setDrawable(ContextCompat.getDrawable(\_mActivity,R.drawable.shape\_hp\_rv\_line));

hpRecyclerView.addItemDecoration(divider);

hpRecyclerView.setHasFixedSize(true);

//添加头布局 Banner 支持轮播效果

LinearLayout mGroup = (LinearLayout) getLayoutInflater()

.inflate(R.layout.head\_banner, null);

banner = mGroup.findViewById(R.id.head\_banner);

mGroup.removeView(banner);

adapter.setHeaderView(banner);

hpRecyclerView.setAdapter(adapter);

}

private void childOnClick(View view, int position) {

switch (view.getId()) {

case R.id.iv\_article\_like:

collectArticle(position);

break;

default:

break;

}

}

/\*\*

\* 收藏文章

\*

\* @param position 传入点击的位置

\*/

private void collectArticle(int position) {

if (presenter.getLoginStatus()) {

if (adapter.getData().get(position).isCollect()) {

presenter.cancelAddCollectArticle(position,adapter.getData().get(position).getId());

} else {

presenter.addCollectArticle(position,adapter.getData().get(position).getId());

}

} else {

Intent intent = new Intent(\_mActivity, LoginActivity.class);

startActivity(intent);

}

}

/\*\*

\* 打开文章详情

\* @param view

\* @param position

\*/

private void openArticleDetail(View view, int position) {

Intent intent = new Intent

(\_mActivity, ArticleDetailActivity.class);

intent.putExtra(Constants.ARTICLE\_LINK,adapter.getData().get(position).getLink()); intent.putExtra(Constants.ARTICLE\_TITLE,adapter.getData().get(position).getTitle());

intent.putExtra(Constants.ARTICLE\_ID,adapter.getData().get(position).getId()); intent.putExtra(Constants.IS\_COLLECTED,adapter.getData().get(position).isCollect());

intent.putExtra(Constants.IS\_SHOW\_COLLECT\_ICON,true);

intent.putExtra(Constants.ARTICLE\_ITEM\_POSITION,position);

intent.putExtra(Constants.EVENT\_BUS\_TAG,"homepage");

startActivity(intent);

}

@Override

protected void initData() {

initRefreshLayout();

presenter.refreshLayout(true);

}

private void initRefreshLayout() {

//下拉刷新

hpSmartRefreshLayout.setOnRefreshListener(new OnRefreshListener() {

@Override

public void onRefresh(@NonNull RefreshLayout refreshLayout) {

presenter.refreshLayout(false);

refreshLayout.finishRefresh();

}

});

//上拉加载更多

hpSmartRefreshLayout.setOnLoadMoreListener(new OnLoadMoreListener() {

@Override

public void onLoadMore(@NonNull RefreshLayout refreshLayout) {

presenter.loadMore();

refreshLayout.finishLoadMore();

}

});

}

/\*\*

\* 展示文章列表

\* @param listData 文章列表 list

\* @param isRefresh 是否刷新

\*/

@Override

public void showList(ArticleListData listData, boolean isRefresh) {

if (isRefresh) {

adapter.replaceData(listData.getDatas());

} else {

adapter.addData(listData.getDatas());

}

}

/\*\*

\* 展示轮播图

\* @param bannerDataList 轮播图 list

\*/

@Override

public void showBannerData(List<BannerData> bannerDataList) {

ArrayList<String> imageList = new ArrayList<>();

ArrayList<String> titleList = new ArrayList<>();

ArrayList<String> urlList = new ArrayList<>();

ArrayList<Integer> articleIdList = new ArrayList<>();

for (BannerData bannerData : bannerDataList) {

imageList.add(bannerData.getImagePath());

titleList.add(bannerData.getTitle());

urlList.add(bannerData.getUrl());

articleIdList.add(bannerData.getId());

}

//显示圆形指示器和标题

banner.setBannerStyle(BannerConfig.CIRCLE\_INDICATOR\_TITLE\_INSIDE);

banner.setImageLoader(new GlideImageLoader());

banner.setImages(imageList);

banner.setBannerAnimation(Transformer.ZoomOutSlide);

banner.setBannerTitles(titleList);

banner.isAutoPlay(true);

banner.setDelayTime(2800);

banner.setIndicatorGravity(BannerConfig.CENTER);

banner.start();

banner.setOnBannerListener(new OnBannerListener() {

@Override

public void OnBannerClick(int position) {

Intent intent = new Intent

(\_mActivity, ArticleDetailActivity.class);

intent.putExtra(Constants.ARTICLE\_LINK,urlList.get

(posiion));

intent.putExtra(Constants.ARTICLE\_TITLE,titleList.get

(position));

intent.putExtra(Constants.ARTICLE\_ID,articleIdList.get

(position));

intent.putExtra(Constants.IS\_COLLECTED,false);

intent.putExtra(Constants.IS\_SHOW\_COLLECT\_ICON,false);

intent.putExtra(Constants.ARTICLE\_ITEM\_POSITION,-1);

intent.putExtra(Constants.EVENT\_BUS\_TAG,

Constants.TAG\_DEFAULT);

startActivity(intent);

}

});

}

/\*\*

\* 收藏成功

\* @param position 收藏位置

\*/

@Override

public void showCollectSuccess(int position) {

adapter.getData().get(position).setCollect(true);

adapter.setData(position,adapter.getData().get(position));

Toast.makeText(\_mActivity,R.string.collect\_success,Toast.LENGTH\_SHORT).show();

}

/\*\*

\* 取消收藏

\* @param position 取消位置

\*/

@Override

public void showCancelCollect(int position) {

adapter.getData().get(position).setCollect(false);

adapter.setData(position,adapter.getData().get(position));

Toast.makeText(\_mActivity,R.string.cancel\_collect,Toast.LENGTH\_SHORT).show();

}

}

/\*\*

\* 在线学习模块布局文件

\*/

<com.classic.common.MultipleStatusView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/custom\_multiple\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<com.scwang.smartrefresh.layout.SmartRefreshLayout

android:id="@+id/hp\_refresh\_layout"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

app:srlAccentColor="?attr/colorPrimary"

app:srlEnablePreviewInEditMode="true"

app:srlPrimaryColor="@color/colorAccent">

<com.scwang.smartrefresh.header.MaterialHeader

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

</com.scwang.smartrefresh.header.MaterialHeader>

<android.support.v7.widget.RecyclerView

android:id="@+id/hp\_recycler\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</android.support.v7.widget.RecyclerView>

<com.scwang.smartrefresh.header.MaterialHeader

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

</com.scwang.smartrefresh.header.MaterialHeader>

</com.scwang.smartrefresh.layout.SmartRefreshLayout>

</com.classic.common.MultipleStatusView>

/\*\*

\* 学习文章详情页

\*/

public class ArticleDetailActivity extends BaseActivity

<ArticleDetailPresenter> implements ArticleDetailContract.View {

private String title;

private String articleLink;

private int articleId;

private boolean isCollected;

private boolean isShowCollectIcon;

private int articleItemPosition;

private String eventBusTag;

@BindView(R.id.toolbar)

Toolbar toolbar;

@BindView(R.id.toolbar\_title)

TextView toolbarTitle;

@BindView(R.id.toolbar\_back)

ImageView toolbar\_back;

@BindView(R.id.cl\_content)

CoordinatorLayout cl\_content;

private AgentWeb agentWeb;

private MenuItem collectItem;

@Override

protected int getLayoutId() {

return R.layout.activity\_article\_detail;

}

@Override

protected void initToolbar() {

getIntentData();

getWindow().setNavigationBarColor(getResources()

.getColor(R.color.white\_smoke));

toolbar.setNavigationIcon(R.drawable.ic\_arrow\_back\_black\_24dp);

setSupportActionBar(toolbar);

ActionBar actionBar = getSupportActionBar();

if (actionBar != null) {

//左上角图标左边加上返回图标

actionBar.setDisplayHomeAsUpEnabled(false);

toolbar\_back.setVisibility(View.VISIBLE);

actionBar.setDisplayShowTitleEnabled(false);

toolbarTitle.setText(Html.fromHtml(title));

toolbarTitle.setSelected(true);

}

toolbar.setNavigationOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

onBackPressedSupport();

}

});

toolbar\_back.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

finish();

}

});

}

private void getIntentData() {

Bundle bundle = getIntent().getExtras();

title = bundle.getString(Constants.ARTICLE\_TITLE);

articleLink = bundle.getString(Constants.ARTICLE\_LINK);

articleId = bundle.getInt(Constants.ARTICLE\_ID);

isCollected = bundle.getBoolean(Constants.IS\_COLLECTED);

isShowCollectIcon = bundle.getBoolean

(Constants.IS\_SHOW\_COLLECT\_ICON);

articleItemPosition = bundle.getInt

(Constants.ARTICLE\_ITEM\_POSITION, -1);

eventBusTag = bundle.getString(Constants.EVENT\_BUS\_TAG);

}

@Override

protected void initView() {

}

@Override

protected void initEventAndData() {

//集合了影响浏览器的事件到来时的回调方法

WebChromeClient webChromeClient = new WebChromeClient() {

@Override

public void onReceivedTitle(WebView view, String title) {

super.onReceivedTitle(view, title);

// toolbar.setTitle(Html.fromHtml(title));

}

};

CoordinatorLayout.LayoutParams layoutParams = new

CoordinatorLayout.LayoutParams(-1, -1);

layoutParams.setBehavior(new AppBarLayout.ScrollingViewBehavior());

NestedScrollAgentWebView nestedScrollAgentWebView = new

NestedScrollAgentWebView(this);

//AgentWeb 是对 WebView 的封装

agentWeb = AgentWeb.with(this)

.setAgentWebParent(cl\_content, layoutParams)

.useDefaultIndicator()

.setWebView(nestedScrollAgentWebView)

.setWebChromeClient(webChromeClient)

.setMainFrameErrorView(R.layout.agentweb\_error\_page, -1)

.setOpenOtherPageWays(DefaultWebClient.OpenOtherPageWays.ASK)

.createAgentWeb()

.ready()

.go(articleLink);

}

/\*\*

\* 创建 ToolBar 上的菜单

\*

\* @param menu 菜单样式

\* @return

\*/

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.article\_detail\_menu, menu);

collectItem = menu.findItem(R.id.item\_collect);

collectItem.setVisible(isShowCollectIcon);

collectItem.setIcon(isCollected ? R.drawable.ic\_collect :

R.drawable.ic\_collect\_not);

return super.onCreateOptionsMenu(menu);

}

/\*\*

\* 可以让打开的菜单同时显示图标和文字

\*

\* @param featureId

\* @param menu

\* @return

\*/

@Override

public boolean onMenuOpened(int featureId, Menu menu) {

if (menu != null) {

if ("MenuBuilder".equalsIgnoreCase(menu.getClass().getSimpleName())) {

try {

@SuppressLint("PrivateApi")

Method method = menu.getClass()

.getDeclaredMethod("setOptionalIconsVisible",

Boolean.TYPE);

method.setAccessible(true);

method.invoke(menu, true);

} catch (Exception e) {

e.printStackTrace();

}

}

}

return super.onMenuOpened(featureId, menu);

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

switch (item.getItemId()) {

case R.id.item\_collect:

dealCollect();

break;

case R.id.item\_share:

mPresenter.checkSharePermission(new RxPermissions(this));

break;

case R.id.item\_open\_browser:

Intent intent = new Intent(Intent.ACTION\_VIEW,

Uri.parse(articleLink));

startActivity(intent);

break;

default:

break;

}

return super.onOptionsItemSelected(item);

}

private void dealCollect() {

if (mPresenter.getLoginStatus()) {

if (isCollected) {

mPresenter.cancelAddCollectArticle(articleItemPosition,

articleId);

} else {

mPresenter.addCollectArticle(articleItemPosition, articleId);

}

} else {

Intent intent = new Intent(this, LoginActivity.class);

startActivity(intent);

}

}

/\*\*

\* 分享学习资源

\*/

@Override

public void shareArticle() {

Intent intent = new Intent(Intent.ACTION\_SEND);

intent.putExtra(Intent.EXTRA\_TEXT, getString(R.string.share\_type\_url, "MobileLearn",

title, articleLink));

intent.setType("text/plain");

startActivity(intent);

}

@Override

public void shareFailed() {

Toast.makeText(this, R.string.failed\_to\_share\_failed,

Toast.LENGTH\_SHORT).show();

}

@Override

public void showCollectSuccess(int position) {

isCollected = true;

collectItem.setIcon(R.drawable.ic\_collect);

if (position < 0) {

Toast.makeText(this, R.string.collect\_success,

Toast.LENGTH\_SHORT).show();

} else {

EventBus.getDefault().post(new CollectEvent(false, position), eventBusTag);

}

}

@Override

public void showCancelCollect(int position) {

isCollected = false;

collectItem.setIcon(R.drawable.ic\_collect\_not);

if (position < 0) {

Toast.makeText(this, R.string.collect\_failed,

Toast.LENGTH\_SHORT).show();

} else {

EventBus.getDefault().post(new CollectEvent(true, position), eventBusTag);

}

}

@Override

public void onBackPressedSupport() {

if (!agentWeb.back()) {

super.onBackPressedSupport();

}

}

@Override

protected void onResume() {

agentWeb.getWebLifeCycle().onResume();

super.onResume();

}

@Override

protected void onPause() {

agentWeb.getWebLifeCycle().onPause();

super.onPause();

}

@Override

protected void onDestroy() {

agentWeb.getWebLifeCycle().onDestroy();

super.onDestroy();

}

}

/\*\*

\* 学习资源详情页布局文件

\*/

<?xml version="1.0" encoding="utf-8"?>

<android.support.design.widget.CoordinatorLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/cl\_content"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/white"

tools:context=".activity.main.ui.ArticleDetailActivity">

<include layout="@layout/toolbar"/>

</android.support.design.widget.CoordinatorLayout>

/\*\*

\* 学习导航相关调用

\*/

public class NavigationPresenter extends Presenter

<NavigationContract.View> implements NavigationContract.Presenter {

@Inject

public NavigationPresenter() {

}

@Override

public void getNavigationData() {

addSubscribe(dataManager.getNavigationData()

.compose(RxJavaUtil.SchedulerTransformer())

.filter(navigationListData -> mView != null)

.subscribeWith(new

BaseObserver<List<NavigationData>>(mView,

MobileLearnApp.getContext().getString

(R.string.failed\_to\_get\_navigation\_data), true) {

@Override

public void success(List<NavigationData>

navigationData) {

mView.showNavigationData(navigationData);

}

}));

}

@Override

public void reload() {

getNavigationData();

}

}

/\*\*

\* 学习导航页面

\*/

public class NavigationFragment extends BaseFragment

<NavigationPresenter> implements NavigationContract.View {

@BindView(R.id.rv\_navigation)

RecyclerView rv\_navigation;

@BindView(R.id.vtl\_navigation)

VerticalTabLayout vtl\_navigation;

private NavigationAdapter adapter;

private boolean scroll;

private int index;

private boolean isClickTab;

private LinearLayoutManager mLinearLayoutManager;

public static NavigationFragment newInstance() {

return new NavigationFragment();

}

@Override

protected int getLayoutId() {

return R.layout.fragment\_navigation;

}

@Override

protected void initView() {

List<NavigationData> navigationDataList = new ArrayList<>();

adapter = new NavigationAdapter(R.layout.item\_navigation,

navigationDataList);

DividerItemDecoration divider = new DividerItemDecoration

(\_mActivity, DividerItemDecoration.VERTICAL);

divider.setDrawable(ContextCompat.getDrawable

(\_mActivity, R.drawable.shape\_hp\_rv\_line));

mLinearLayoutManager = new LinearLayoutManager(\_mActivity);

rv\_navigation.setLayoutManager(mLinearLayoutManager);

rv\_navigation.setHasFixedSize(true);

rv\_navigation.setAdapter(adapter);

}

@Override

protected void initData() {

presenter.getNavigationData();

}

/\*\*

\* 展示学习导航资源

\* @param navigationData 导航数据

\*/

@Override

public void showNavigationData(List<NavigationData> navigationData) {

vtl\_navigation.setTabAdapter(new TabAdapter() {

@Override

public int getCount() {

return navigationData == null ? 0 : navigationData.size();

}

@Override

public ITabView.TabBadge getBadge(int i) {

return null;

}

@Override

public ITabView.TabIcon getIcon(int i) {

return null;

}

@Override

public ITabView.TabTitle getTitle(int i) {

return new TabView.TabTitle.Builder()

.setContent(navigationData.get(i).getName())

.setTextColor(ContextCompat.getColor(\_mActivity, R.color.Grey1000),

ContextCompat.getColor(\_mActivity,

R.color.bottom\_bar))

.build();

}

@Override

public int getBackground(int i) {

return -1;

}

});

adapter.replaceData(navigationData);

rv\_navigation.addOnScrollListener(new RecyclerView.OnScrollListener() {

@Override

public void onScrollStateChanged(RecyclerView recyclerView, int newState) {

super.onScrollStateChanged(recyclerView, newState);

if (scroll && (newState ==

RecyclerView.SCROLL\_STATE\_IDLE)) {

scrollRecyclerView();

}

rightLinkageLeft(newState);

}

@Override

public void onScrolled(RecyclerView recyclerView, int dx, int dy) {

super.onScrolled(recyclerView, dx, dy);

if (scroll) {

scrollRecyclerView();

}

}

});

vtl\_navigation.addOnTabSelectedListener(new VerticalTabLayout.OnTabSelectedListener() {

@Override

public void onTabSelected(TabView tabView, int i) {

isClickTab = true;

selectTag(i);

}

@Override

public void onTabReselected(TabView tabView, int i) {

}

});

}

/\*\*

\* 滑动 RecyclerView

\*/

private void scrollRecyclerView() {

scroll = false;

int indexDistance = index –

mLinearLayoutManager.findFirstVisibleItemPosition();

if (indexDistance >= 0 && indexDistance <

rv\_navigation.getChildCount()) {

int top = rv\_navigation.getChildAt(indexDistance).getTop();

rv\_navigation.smoothScrollBy(0, top);

}

}

private void rightLinkageLeft(int newState) {

if (newState == RecyclerView.SCROLL\_STATE\_IDLE) {

if (isClickTab) {

isClickTab = false;

return;

}

int firstPosition = mLinearLayoutManager

.findFirstVisibleItemPosition();

if (index != firstPosition) {

index = firstPosition;

setChecked(index);

}

}

}

private void setChecked(int position) {

if (isClickTab) {

isClickTab = false;

} else {

if (vtl\_navigation == null) {

return;

}

vtl\_navigation.setTabSelected(index);

}

index = position;

}

private void selectTag(int i) {

index = i;

rv\_navigation.stopScroll();

smoothScrollToPosition(i);

}

private void smoothScrollToPosition(int currentPosition) {

int firstPosition = mLinearLayoutManager

.findFirstVisibleItemPosition();

int lastPosition = mLinearLayoutManager

.findLastVisibleItemPosition();

if (currentPosition <= firstPosition) {

rv\_navigation.smoothScrollToPosition(currentPosition);

} else if (currentPosition <= lastPosition) {

int top = rv\_navigation.getChildAt(currentPosition –

firstPosition).getTop();

rv\_navigation.smoothScrollBy(0, top);

} else {

rv\_navigation.smoothScrollToPosition(currentPosition);

scroll = true;

}

}

}

/\*\*

\* 学习导航布局文件

\*/

<?xml version="1.0" encoding="utf-8"?>

<com.classic.common.MultipleStatusView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/custom\_multiple\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/white">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="horizontal">

<q.rorbin.verticaltablayout.VerticalTabLayout

android:id="@+id/vtl\_navigation"

android:layout\_width="112dp"

android:layout\_height="match\_parent"

android:background="@color/main\_white"

app:indicator\_color="@color/main\_white"

app:tab\_height="45dp"

app:tab\_margin="15dp"

app:tab\_mode="scrollable">

</q.rorbin.verticaltablayout.VerticalTabLayout>

<View

android:layout\_width="0.5dp"

android:layout\_height="match\_parent"

android:background="@color/Grey300" />

<android.support.v7.widget.RecyclerView

android:id="@+id/rv\_navigation"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</android.support.v7.widget.RecyclerView>

</LinearLayout>

</com.classic.common.MultipleStatusView>

/\*\*

\* 学习导航学习内容适配器

\*/

public class NavigationAdapter extends BaseQuickAdapter

<NavigationData, BaseViewHolder> {

public NavigationAdapter(int layoutResId, @Nullable

List<NavigationData> data) {

super(layoutResId, data);

}

@Override

protected void convert(BaseViewHolder helper, NavigationData item) {

if (!TextUtils.isEmpty(item.getName())) {

helper.setText(R.id.item\_navigation\_tv, item.getName());

}

TagFlowLayout mTagFlowLayout = helper.getView

(R.id.item\_navigation\_flow\_layout);

List<ArticleItemData> mArticles = item.getArticles();

mTagFlowLayout.setAdapter(new TagAdapter<ArticleItemData>(mArticles) {

@Override

public View getView(FlowLayout parent, int position,

ArticleItemData articleItemData) {

TextView tv = (TextView) LayoutInflater

.from(parent.getContext())

.inflate(R.layout.tag\_flow\_layout\_tv,

mTagFlowLayout, false);

if (articleItemData == null) {

return null;

}

String name = articleItemData.getTitle();

tv.setText(name);

tv.setTextColor(getRandomColor());

mTagFlowLayout.setOnTagClickListener((view, position1,

parent1) -> {

Intent intent = new Intent(parent.getContext(),

ArticleDetailActivity.class); intent.putExtra(Constants.ARTICLE\_LINK,mArticles

.get(position).getLink()); intent.putExtra(Constants.ARTICLE\_TITLE,mArticles

.get(position).getTitle());

intent.putExtra(Constants.ARTICLE\_ID,mArticles

.get(position).getId());

intent.putExtra(Constants.IS\_COLLECTED,mArticles

.get(position).isCollect());

intent.putExtra(Constants.EVENT\_BUS\_TAG,

"flow\_layout");

intent.putExtra(Constants.IS\_SHOW\_COLLECT\_ICON,true);

intent.putExtra(Constants.ARTICLE\_ITEM\_POSITION,-1);

parent.getContext().startActivity(intent);

return true;

});

return tv;

}

});

}

/\*\*

\* 获取随机颜色

\* @return

\*/

public static int getRandomColor() {

Random random = new Random();

//0-190, 如果颜色值过大,就越接近白色,就看不清了,所以需要限定范围

int red;

int green;

int blue;

red = random.nextInt(190);

green = random.nextInt(190);

blue = random.nextInt(190);

//使用rgb混合生成一种新的颜色,Color.rgb生成的是一个int数

return Color.rgb(red, green, blue);

}

}

/\*\*

\* 学习导航是适配器中布局文件

\*/

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@color/main\_white"

android:padding="12dp"

android:orientation="vertical">

<TextView

android:id="@+id/item\_navigation\_tv"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textColor="@color/black"

android:textSize="16sp"

tools:text="@string/app\_name" />

<com.zhy.view.flowlayout.TagFlowLayout

android:id="@+id/item\_navigation\_flow\_layout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@color/main\_white"

android:padding="12dp"

app:max\_select="0" />

</LinearLayout>

public interface NavigationContract {

interface View extends BaseView {

void showNavigationData(List<NavigationData> navigationData);

}

interface Presenter extends BasePresenter<View> {

void getNavigationData();

}

}

/\*\*

\* 知识体系相关调用

\*/

public class KnowledgePresenter extends Presenter

<KnowledgeContract.View> implements KnowledgeContract.Presenter {

@Inject

public KnowledgePresenter() {

}

@Override

public void getKnowledgeSystemData(boolean isShowViewStatus) {

addSubscribe(dataManager.getKnowledgeData()

.compose(RxJavaUtil.SchedulerTransformer())

.filter(listBaseResponse -> mView != null)

.subscribeWith(new BaseObserver<List<KnowledgeSystemData>>(mView,

MobileLearnApp.getContext().getString

(R.string.failed\_to\_get\_knowledge),

isShowViewStatus) {

//知识体系数据请求成功后回调函数

@Override

public void success(List<KnowledgeSystemData>

knowledgeSystemData) {

mView.showKnowledgeSystemData(knowledgeSystemData);

}

}));

}

/\*\*

\* 重新加载

\*/

@Override

public void reload() {

getKnowledgeSystemData(false);

}

}

/\*\*

\* 知识体系主页面

\*/

public class KnowledgeFragment extends BaseFragment

<KnowledgePresenter> implements KnowledgeContract.View {

@BindView(R.id.rv\_knowledge)

RecyclerView rv\_knowledge;

@BindView(R.id.srl\_knowledge)

SmartRefreshLayout srl\_knowledge;

private KnowledgeAdapter mAdapter;

public static KnowledgeFragment newInstance() {

KnowledgeFragment knowledgeFragment = new KnowledgeFragment();

return knowledgeFragment;

}

@Override

protected int getLayoutId() {

return R.layout.fragment\_knowledge;

}

@Override

protected void initView() {

ArrayList<KnowledgeSystemData> knowledgeSystemData = new

ArrayList<>();

mAdapter = new KnowledgeAdapter(R.layout.item\_knowledge,

knowledgeSystemData);

mAdapter.setOnItemClickListener((adapter, view, position) ->

startKnowledgeDetail(view, position));

DividerItemDecoration divider = new

DividerItemDecoration(\_mActivity,DividerItemDecoration.VERTICAL);

divider.setDrawable(ContextCompat.getDrawable(\_mActivity,R.drawable.shape\_hp\_rv\_line));

rv\_knowledge.addItemDecoration(divider);

rv\_knowledge.setHasFixedSize(true);

rv\_knowledge.setLayoutManager(new LinearLayoutManager(\_mActivity));

rv\_knowledge.setAdapter(mAdapter);

srl\_knowledge.setOnRefreshListener(refreshLayout -> {

presenter.getKnowledgeSystemData(false);

refreshLayout.finishRefresh();

});

}

@Override

protected void initData() {

presenter.getKnowledgeSystemData(true);

}

@Override

public void showKnowledgeSystemData(List<KnowledgeSystemData>

knowledgeSystems) {

mAdapter.replaceData(knowledgeSystems);

}

/\*\*

\* 开启知识体系详情页

\* @param view

\* @param position

\*/

public void startKnowledgeDetail(View view, int position) {

Intent intent = new Intent(\_mActivity,

KnowledgeDetailActivity.class);

intent.putExtra(Constants.KNOWLEDGE\_DATA, mAdapter.getData().get(position));

startActivity(intent);

}

}

/\*\*

\* 知识体系主页面布局文件

\*/

<?xml version="1.0" encoding="utf-8"?>

<com.classic.common.MultipleStatusView

xmlns:android="http://schemas.android.com/apk/res/android"

android:id="@+id/custom\_multiple\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<com.scwang.smartrefresh.layout.SmartRefreshLayout

android:id="@+id/srl\_knowledge"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<com.scwang.smartrefresh.header.MaterialHeader

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

<android.support.v7.widget.RecyclerView

android:id="@+id/rv\_knowledge"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/white" />

<com.scwang.smartrefresh.header.MaterialHeader

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

</com.scwang.smartrefresh.layout.SmartRefreshLayout>

</com.classic.common.MultipleStatusView>

/\*\*

\* 知识体系详情页

\*/

public class KnowledgeDetailActivity extends BaseActivity

<KnowledgeDetailPresenter> implements KnowledgeDetailContract.View {

@BindView(R.id.toolbar)

Toolbar toolbar;

@BindView(R.id.toolbar\_title)

TextView toolbar\_title;

@BindView(R.id.tb\_knowledge\_detail)

TabLayout tb\_knowledge\_detail;

@BindView(R.id.vp\_knowledge\_detail)

ViewPager vp\_knowledge\_detail;

@BindView(R.id.toolbar\_back)

ImageView toolbar\_back;

private SparseArray<KnowledgeArticleFragment>

knowledgeArticleFragmentArray= new SparseArray<>();

private List<KnowledgeSystemData> dataChildren;

@Override

protected int getLayoutId() {

return R.layout.activity\_knowledge\_detail;

}

@Override

protected void initToolbar() {

setSupportActionBar(toolbar);

ActionBar actionBar = getSupportActionBar();

if (actionBar != null) {

actionBar.setDisplayShowTitleEnabled(false);

toolbar\_back.setVisibility(View.VISIBLE);

}

toolbar\_back.setOnClickListener(view -> finish());

getWindow().setNavigationBarColor(getResources().getColor(R.color.white\_smoke));

}

@Override

public void onBackPressedSupport() {

super.onBackPressedSupport();

}

@Override

protected void initView() {

}

/\*\*

\* 初始化点击事件和数据

\*/

@Override

protected void initEventAndData() {

KnowledgeSystemData knowledgeSystemData =

(KnowledgeSystemData) getIntent().getSerializableExtra(Constants.KNOWLEDGE\_DATA);

toolbar\_title.setText(knowledgeSystemData.getName());

dataChildren = knowledgeSystemData.getChildren();

vp\_knowledge\_detail.setAdapter(new FragmentStatePagerAdapter(getSupportFragmentManager()) {

@Override

public Fragment getItem(int position) {

KnowledgeArticleFragment knowledgeArticleFragment =

knowledgeArticleFragmentArray.get(position);

if (knowledgeArticleFragment == null) {

Bundle bundle = new Bundle();

bundle.putInt("KnowledgeDataId", dataChildren.get(position).getId());

knowledgeArticleFragment =

KnowledgeArticleFragment.newInstance(bundle);

nowledgeArticleFragmentArray.put(position,knowledgeArticleFragment);

return knowledgeArticleFragment;

} else {

return knowledgeArticleFragment;

}

}

@Override

public int getCount() {

return dataChildren.size();

}

@Nullable

@Override

public CharSequence getPageTitle(int position) {

return Html.fromHtml(dataChildren

.get(position).getName());

}

@Override

public void destroyItem(@NonNull ViewGroup container, int position, @NonNull Object object) {

System.out.println("destory");

}

});

vp\_knowledge\_detail.addOnPageChangeListener(new TabLayout.TabLayoutOnPageChangeListener(tb\_knowledge\_detail));

tb\_knowledge\_detail.setupWithViewPager(vp\_knowledge\_detail);

tb\_knowledge\_detail.addOnTabSelectedListener(new TabLayout.ViewPagerOnTabSelectedListener(vp\_knowledge\_detail));

tb\_knowledge\_detail.addOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {

@Override

public void onTabSelected(TabLayout.Tab tab) {

vp\_knowledge\_detail.setCurrentItem(tab.getPosition()

,false);

}

@Override

public void onTabUnselected(TabLayout.Tab tab) {

}

@Override

public void onTabReselected(TabLayout.Tab tab) {

}

});

}

@Override

public void onDestroy() {

if (knowledgeArticleFragmentArray != null) {

knowledgeArticleFragmentArray.clear();

knowledgeArticleFragmentArray = null;

}

if (dataChildren != null) {

dataChildren.clear();

dataChildren = null;

}

super.onDestroy();

}

}

/\*\*

\* 知识体系详情页布局文件

\*/

<?xml version="1.0" encoding="utf-8"?>

<android.support.design.widget.CoordinatorLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<include layout="@layout/toolbar" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/white"

android:orientation="vertical"

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior">

<android.support.design.widget.TabLayout

android:id="@+id/tb\_knowledge\_detail"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@color/white"

app:tabIndicatorColor="@color/black"

app:tabIndicatorHeight="2dp"

app:tabMode="scrollable"

app:tabSelectedTextColor="@color/black"

app:tabTextColor="@color/Grey700" />

<android.support.v4.view.ViewPager

android:id="@+id/vp\_knowledge\_detail"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</android.support.v4.view.ViewPager>

</LinearLayout>

</android.support.design.widget.CoordinatorLayout>

/\*\*

\* 公众号主页面

\*/

public class PublicNumFragment extends BaseFragment

<PublicNumPresenter> implements PublicNumContract.View {

@BindView(R.id.tb\_public\_num)

TabLayout tb\_public\_num;

@BindView(R.id.vp\_public\_num)

ViewPager vp\_public\_num;

private List<PublicNumData> mPublicNumList;

private SparseArray<PersonalPublicNumFragment> personalFragment = new SparseArray<>();

private PersonalPublicNumFragment currentFragment;

public static PublicNumFragment newInstance() {

return new PublicNumFragment();

}

@Override

protected int getLayoutId() {

return R.layout.fragment\_public\_num;

}

@Override

protected void initView() {

}

@Override

protected void initData() {

presenter.getPublicNumData();

}

/\*\*

\* 展示公众号列表

\* @param publicNumData 列表数据

\*/

@Override

public void showPublicNumData(List<PublicNumData> publicNumData) {

mPublicNumList = publicNumData;

vp\_public\_num.setAdapter(new FragmentStatePagerAdapter(getChildFragmentManager()) {

@Override

public Fragment getItem(int position) {

PersonalPublicNumFragment personalPublicNumFragment =

personalFragment.get(position);

if (personalPublicNumFragment == null) {

Bundle bundle = new Bundle();

bundle.putInt("publicNumId",

publicNumData.get(position).getId());

personalPublicNumFragment =

PersonalPublicNumFragment.newInstance(bundle);

personalFragment.put(position,personalPublicNumFragment);

return personalPublicNumFragment;

} else {

return personalPublicNumFragment;

}

}

@Override

public int getCount() {

return mPublicNumList == null ? 0 : mPublicNumList.size();

}

@Nullable

@Override

public CharSequence getPageTitle(int position) {

return Html.fromHtml(mPublicNumList.get(position).getName());

}

//不添加会内存溢出

@Override

public void destroyItem(@NonNull ViewGroup container, int position, @NonNull Object object) {

}

});

vp\_public\_num.addOnPageChangeListener(new TabLayout.TabLayoutOnPageChangeListener(tb\_public\_num));

tb\_public\_num.setupWithViewPager(vp\_public\_num);

tb\_public\_num.addOnTabSelectedListener(new TabLayout.ViewPagerOnTabSelectedListener(vp\_public\_num));

tb\_public\_num.addOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {

@Override

public void onTabSelected(TabLayout.Tab tab) {

vp\_public\_num.setCurrentItem(tab.getPosition(), false);

}

@Override

public void onTabUnselected(TabLayout.Tab tab) {

}

@Override

public void onTabReselected(TabLayout.Tab tab) {

}

});

}

@Override

public void onDestroyView() {

super.onDestroyView();

if (mPublicNumList != null) {

mPublicNumList.clear();

mPublicNumList = null;

}

if (personalFragment != null) {

personalFragment.clear();

personalFragment = null;

}

}

}

/\*\*

\*公众号主页面布局

\*/

<?xml version="1.0" encoding="utf-8"?>

<com.classic.common.MultipleStatusView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/custom\_multiple\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/main\_white">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<android.support.design.widget.TabLayout

android:id="@+id/tb\_public\_num"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@color/main\_white"

app:tabTextColor="@color/bottom\_bar"

app:tabIndicatorColor="@color/Grey1000"

app:tabIndicatorHeight="2dp"

app:tabSelectedTextColor="@color/Grey900"

app:tabMode="scrollable"

/>

<android.support.v4.view.ViewPager

android:id="@+id/vp\_public\_num"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</android.support.v4.view.ViewPager>

</LinearLayout>

</com.classic.common.MultipleStatusView>

/\*\*

\* 个人公众号页面

\*/

public class PersonalPublicNumFragment extends BaseFragment

<PersonalPublicNumPresenter>

implements PersonalPublicNumContract.View {

@BindView(R.id.srl\_personal\_public\_num)

SmartRefreshLayout srl\_personal\_public\_num;

@BindView(R.id.rv\_public\_num)

RecyclerView rv\_public\_num;

private PersonalPublicNumAdapter mAdapter;

public static PersonalPublicNumFragment newInstance(Bundle bundle) {

PersonalPublicNumFragment personalPublicNumFragment = new

PersonalPublicNumFragment();

personalPublicNumFragment.setArguments(bundle);

return personalPublicNumFragment;

}

@Override

protected int getLayoutId() {

return R.layout.fragment\_personal\_public\_num;

}

@Override

protected void initView() {

ArrayList<ArticleItemData> articleItemData = new ArrayList<>();

mAdapter = new PersonalPublicNumAdapter(R.layout.item\_article,

articleItemData);

mAdapter.setOnItemClickListener((adapter, view, position) ->

startArticleDetail(view，position));

mAdapter.setOnItemChildClickListener((adapter, view, position) ->

childOnClick(view，position));

DividerItemDecoration divider = new

DividerItemDecoration(\_mActivity,

DividerItemDecoration.VERTICAL);

divider.setDrawable(ContextCompat.getDrawable(\_mActivity, R.drawable.shape\_hp\_rv\_line));

rv\_public\_num.addItemDecoration(divider);

rv\_public\_num.setLayoutManager(new LinearLayoutManager(\_mActivity));

//item 固定宽高，避免重复计算大小

rv\_public\_num.setHasFixedSize(true); rv\_public\_num.setAdapter(mAdapter);

}

@Override

protected void initData() {

int publicNumId = getArguments().getInt("publicNumId");

presenter.refreshLayout(publicNumId, true);

srl\_personal\_public\_num.setOnRefreshListener(refreshLayout -> {

presenter.refreshLayout(publicNumId, false);

srl\_personal\_public\_num.finishRefresh();

});

srl\_personal\_public\_num.setOnLoadMoreListener(refreshLayout -> {

presenter.loadMore();

srl\_personal\_public\_num.finishLoadMore();

});

}

/\*\*

\* 打开学习资源详情

\* @param view

\* @param position

\*/

private void startArticleDetail(View view, int position) {

Intent intent = new Intent(\_mActivity,

ArticleDetailActivity.class);

intent.putExtra(Constants.ARTICLE\_LINK, mAdapter.getData().get(position).getLink());

intent.putExtra(Constants.ARTICLE\_TITLE, mAdapter.getData().get(position).getTitle());

intent.putExtra(Constants.ARTICLE\_ID, mAdapter.getData().get(position).getId());

intent.putExtra(Constants.IS\_COLLECTED, mAdapter.getData().get(position).isCollect());

intent.putExtra(Constants.IS\_SHOW\_COLLECT\_ICON, true);

intent.putExtra(Constants.ARTICLE\_ITEM\_POSITION, position);

intent.putExtra(Constants.EVENT\_BUS\_TAG, "publicNum");

startActivity(intent);

}

private void childOnClick(View view, int position) {

if (presenter.getLoginStatus()) {

if (mAdapter.getData().get(position).isCollect()) {

presenter.cancelAddCollectArticle(position,

mAdapter.getData().get(position).getId());

} else {

presenter.addCollectArticle(position, mAdapter.getData().get(position).getId());

}

} else {

Intent intent = new Intent(\_mActivity, LoginActivity.class);

startActivity(intent);

}

}

/\*\*

\* 展示公众号数据

\* @param articleListData 列表数据

\* @param isRefresh 是否刷新

\*/

@Override

public void showPersonalPublicNum(ArticleListData articleListData, boolean isRefresh) {

if (isRefresh) {

mAdapter.replaceData(articleListData.getDatas());

} else {

mAdapter.addData(articleListData.getDatas());

}

}

@Override

public void showCollectSuccess(int position) {

mAdapter.getData().get(position).setCollect(true);

mAdapter.setData(position, mAdapter.getData().get(position));

}

@Override

public void showCancelCollect(int position) {

mAdapter.getData().get(position).setCollect(false);

mAdapter.setData(position, mAdapter.getData().get(position));

}

}

/\*\*

\* 个人公众号布局

\*/

<?xml version="1.0" encoding="utf-8"?>

<com.classic.common.MultipleStatusView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/custom\_multiple\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<com.scwang.smartrefresh.layout.SmartRefreshLayout

android:id="@+id/srl\_personal\_public\_num"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

app:srlAccentColor="?attr/colorPrimary"

app:srlEnablePreviewInEditMode="true"

app:srlPrimaryColor="?attr/colorPrimary">

<com.scwang.smartrefresh.header.MaterialHeader

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

<android.support.v7.widget.RecyclerView

android:id="@+id/rv\_public\_num"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/white"

android:visibility="visible" />

<com.scwang.smartrefresh.header.MaterialHeader

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

</com.scwang.smartrefresh.layout.SmartRefreshLayout>

</com.classic.common.MultipleStatusView>

/\*\*

\* 个人公众号适配器

\*/

public class PersonalPublicNumAdapter extends BaseQuickAdapter

<ArticleItemData, BaseViewHolder> {

public PersonalPublicNumAdapter(int layoutResId, @Nullable

List<ArticleItemData> data) {

super(layoutResId, data);

}

@Override

protected void convert(BaseViewHolder helper, ArticleItemData item) {

helper.setText(R.id.tv\_article\_title, item.getTitle()).setText(R.id.tv\_article\_author,

"作者：" + item.getAuthor())

.setImageResource(R.id.iv\_article\_like, item.isCollect() ? R.drawable.ic\_collect :

R.drawable.ic\_collect\_not);

if (!TextUtils.isEmpty(item.getNiceDate())) {

helper.setText(R.id.tv\_article\_public\_date, item.getNiceDate());

}

if (!TextUtils.isEmpty(item.getChapterName())) {

String classifyName = item.getSuperChapterName() + " / " + item.getChapterName();

helper.setText(R.id.tv\_article\_chapterName, classifyName);

}

//是否置顶

helper.getView(R.id.iv\_article\_top).setVisibility(item.getType() == 1 ? View.VISIBLE :View.GONE);

//是否是最新

helper.getView(R.id.iv\_article\_fresh).setVisibility(item.isFresh() ? View.VISIBLE :

View.GONE);

if (item.getTags().size() > 0) {

helper.setText(R.id.tv\_article\_tag, item.getTags().get(0).getName()).

getView(R.id.tv\_article\_tag).setVisibility(View.VISIBLE);

} else {

helper.getView(R.id.tv\_article\_tag).setVisibility(View.GONE);

}

if (!TextUtils.isEmpty(item.getEnvelopePic())) {

helper.getView(R.id.iv\_article\_thumbnail).setVisibility(View.VISIBLE);

GlideUtil.download(mContext, item.getEnvelopePic(),

helper.getView(R.id.iv\_article\_thumbnail));

} else {

helper.getView(R.id.iv\_article\_thumbnail).setVisibility(View.GONE);

}

helper.addOnClickListener(R.id.iv\_article\_like);

}

}

/\*\*

\* 个人公众号适配器布局

\*/

<?xml version="1.0" encoding="utf-8"?>

<android.support.v7.widget.CardView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/item\_search\_pager\_group"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:clickable="true"

android:focusable="true"

android:foreground="?android:attr/selectableItemBackground"

app:cardBackgroundColor="@color/main\_white"

app:cardCornerRadius="0dp"

android:orientation="vertical"

app:cardElevation="1dp">

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:background="@color/white"

android:paddingLeft="12dp">

<ImageView

android:id="@+id/iv\_article\_thumbnail"

android:layout\_width="130dp"

android:layout\_height="match\_parent"

android:layout\_gravity="center\_vertical"

android:layout\_marginRight="10dp"

android:layout\_marginTop="10dp"

android:layout\_marginBottom="10dp"

android:scaleType="centerCrop"

android:visibility="gone"

tools:visibility="visible" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:paddingTop="10dp"

android:paddingRight="14dp"

android:paddingBottom="10dp">

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="20dp"

>

<ImageView

android:id="@+id/iv\_article\_top"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:layout\_marginEnd="5dp"

android:src="@drawable/ic\_article\_top"

android:visibility="gone"

tools:visibility="visible"

/>

<ImageView

android:id="@+id/iv\_article\_fresh"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:layout\_marginEnd="5dp"

android:layout\_toEndOf="@id/iv\_article\_top"

android:layout\_toRightOf="@id/iv\_article\_top"

android:textColor="@color/red"

android:visibility="gone"

android:src="@drawable/ic\_article\_new"

tools:visibility="visible" />

<TextView

android:id="@+id/tv\_article\_tag"

android:layout\_width="wrap\_content"

android:layout\_height="15dp"

android:layout\_centerVertical="true"

android:layout\_marginEnd="5dp"

android:layout\_marginRight="5dp"

android:layout\_toEndOf="@id/iv\_article\_fresh"

android:layout\_toRightOf="@id/iv\_article\_fresh"

android:background="@drawable/background\_tag"

android:paddingLeft="4dp"

android:paddingTop="1dp"

android:paddingRight="4dp"

android:paddingBottom="2dp"

android:textColor="@color/colorPrimary"

android:textSize="10sp"

android:visibility="gone"

tools:text="wanandroid"

tools:visibility="visible" />

<TextView

android:id="@+id/tv\_article\_author"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:layout\_toEndOf="@id/tv\_article\_tag"

android:layout\_toRightOf="@id/tv\_article\_tag"

android:gravity="center"

android:textColor="@color/Grey700"

android:textSize="11sp"

tools:text="author" />

/>

<TextView

android:id="@+id/tv\_article\_public\_date"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentRight="true"

android:layout\_centerVertical="true"

android:textColor="@color/Grey500"

android:textSize="12sp" />

</RelativeLayout>

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="8dp">

<TextView

android:id="@+id/tv\_article\_title"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:ellipsize="end"

android:gravity="top|start"

android:lineSpacingExtra="2dp"

android:maxLines="2"

android:textColor="@color/black"

android:textSize="16sp"

tools:text="WanAndroid" />

</RelativeLayout>

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:gravity="center\_vertical">

<ImageView

android:id="@+id/iv\_article\_zan"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/ic\_zan"

android:visibility="gone"

/>

<ImageView

android:id="@+id/iv\_article\_like"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:contentDescription="@null"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" />

<TextView

android:id="@+id/tv\_article\_chapterName"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:layout\_alignParentRight="true"

android:gravity="center\_vertical"

android:textColor="@color/Grey700"

android:textSize="12sp"/>

</RelativeLayout>

</LinearLayout>

</LinearLayout>

</android.support.v7.widget.CardView>

/\*\*

\* 公众号相关调用

\*/

public class PublicNumPresenter extends Presenter

<PublicNumContract.View> implements PublicNumContract.Presenter {

@Inject

public PublicNumPresenter() {

}

@Override

public void getPublicNumData() {

addSubscribe(dataManager.getPublicNumData()

.compose(RxJavaUtil.SchedulerTransformer())

.filter(publicNumList -> mView != null)

.subscribeWith(new BaseObserver<List<PublicNumData>>(mView,

MobileLearnApp.getContext().getString(R.string.failed\_to\_get\_public\_num),true) {

@Override

public void success(List<PublicNumData> publicNumData) {

mView.showPublicNumData(publicNumData);

}

}));

}

}

/\*\*

\* 个人公众号相关调用

\*/

public class PersonalPublicNumPresenter extends

CollectPresenter<PersonalPublicNumContract.View>

implements PersonalPublicNumContract.Presenter {

private int currentPage = 1;

private boolean isRefresh = true;

private int id;

@Inject

public PersonalPublicNumPresenter() {

}

@Override

public void getPersonalPublicNum(boolean viewStatus) {

addSubscribe(dataManager.getPersonalPublicNumData(id, currentPage)

.compose(RxJavaUtil.SchedulerTransformer())

.filter(articleListDataBaseResponse -> mView != null)

.subscribeWith(new BaseObserver<ArticleListData>(mView,

MobileLearnApp.getContext().getString(R.string.failed\_to

\_get\_personal\_public\_num),viewStatus) {

@Override

public void success(ArticleListData listData) {

if(viewStatus && currentPage == 1 &&

listData.getDatas().size() < 1) {

mView.showEmpty();

} else {

mView.showPersonalPublicNum(listData,

isRefresh);

}

}

}));

}

/\*\*

\* 加载更多

\*/

@Override

public void loadMore() {

currentPage++;

isRefresh = false;

getPersonalPublicNum(false);

}

/\*\*

\* 刷新列表

\* @param id id

\* @param viewStatus 是否刷新

\*/

@Override

public void refreshLayout(int id, boolean viewStatus) {

currentPage = 1;

isRefresh = true;

this.id = id;

getPersonalPublicNum(viewStatus);

}

@Override

public void registerEventBus() {

EventBus.getDefault().register(this);

}

@Override

public void unregisterEventBus() {

EventBus.getDefault().unregister(this);

}

@Subscriber(tag = "publicNum")

public void getData(CollectEvent collectEvent){

if (mView == null) {

return;

}

if (collectEvent.isCancel()) {

mView.showCancelCollect(collectEvent.getArticlePosition());

} else {

mView.showCollectSuccess(collectEvent.getArticlePosition());

}

}

}

public interface PersonalPublicNumContract {

interface View extends CollectContract.View {

void showPersonalPublicNum(ArticleListData articleListData,

boolean isRefresh);

}

interface Presenter extends BasePresenter<View> {

void getPersonalPublicNum(boolean viewStatus);

void loadMore();

void refreshLayout(int id, boolean viewStatus);

}

}

/\*\*

\* 项目模块主页面

\*/

public class ProjectFragment extends BaseFragment<ProjectPresenter>

implements ProjectContract.View {

@BindView(R.id.tb\_project)

TabLayout tb\_project;

@BindView(R.id.vp\_project)

ViewPager vp\_project;

private SparseArray<ProjectListFragment> fragmentSparseArray = new

SparseArray<>();

private List<ProjectData> mProjectTreeData;

public static ProjectFragment newInstance() {

return new ProjectFragment();

}

@Override

protected int getLayoutId() {

return R.layout.fragment\_project;

}

@Override

protected void initView() {

}

@Override

protected void initData() {

presenter.getProjectData();

}

/\*\*

\* 加载项目列表

\* @param projectData 列表数据

\*/

@Override

public void showProjectData(List<ProjectData> projectData) {

mProjectTreeData = projectData;

vp\_project.setAdapter(new

FragmentStatePagerAdapter(getChildFragmentManager()) {

@Override

public Fragment getItem(int position) {

ProjectListFragment projectListFragment =

fragmentSparseArray.get(position);

if (projectListFragment == null) {

Bundle bundle = new Bundle();

bundle.putInt("projectId",

mProjectTreeData.get(position).getId());

projectListFragment =

ProjectListFragment.newInstance(bundle);

fragmentSparseArray.put(position, projectListFragment);

return projectListFragment;

} else {

return projectListFragment;

}

}

@Override

public int getCount() {

return projectData.size();

}

@Nullable

@Override

public CharSequence getPageTitle(int position) {

return Html.fromHtml(mProjectTreeData.get(position).getName());

}

@Override

public void destroyItem(@NonNull ViewGroup container, int

position, @NonNull Object object) {

}

});

vp\_project.addOnPageChangeListener(new TabLayout.TabLayoutOnPageChangeListener(tb\_project));

tb\_project.setupWithViewPager(vp\_project);

tb\_project.addOnTabSelectedListener(new TabLayout.ViewPagerOnTabSelectedListener(vp\_project));

tb\_project.addOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {

@Override

public void onTabSelected(TabLayout.Tab tab) {

vp\_project.setCurrentItem(tab.getPosition(), false);

}

@Override

public void onTabUnselected(TabLayout.Tab tab) {

}

@Override

public void onTabReselected(TabLayout.Tab tab) {

}

});

}

@Override

public void onDestroyView() {

if (fragmentSparseArray != null) {

fragmentSparseArray.clear();

fragmentSparseArray = null;

}

if (mProjectTreeData != null) {

mProjectTreeData.clear();

mProjectTreeData = null;

}

super.onDestroyView();

}

}

/\*\*

\* 项目模块主页面布局

\*/

<?xml version="1.0" encoding="utf-8"?>

<com.classic.common.MultipleStatusView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/custom\_multiple\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/main\_white">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<android.support.design.widget.TabLayout

android:id="@+id/tb\_project"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@color/white"

app:tabGravity="center"

app:tabIndicatorColor="@color/Grey1000"

app:tabIndicatorHeight="2dp"

app:tabMode="scrollable"

app:tabSelectedTextColor="@color/Grey900"

app:tabTextColor="@color/Grey700">

</android.support.design.widget.TabLayout>

<android.support.v4.view.ViewPager

android:id="@+id/vp\_project"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</android.support.v4.view.ViewPager>

</LinearLayout>

</com.classic.common.MultipleStatusView>

/\*\*

\* 项目详情页

\*/

public class ProjectListFragment extends BaseFragment

<ProjectListFragmentPresenter> implements

ProjectListFragmentContract.View {

@BindView(R.id.srl\_project\_list)

SmartRefreshLayout srl\_project\_list;

@BindView(R.id.rv\_project\_list)

RecyclerView rv\_project\_list;

private ProjectListAdapter mAdapter;

public static ProjectListFragment newInstance(Bundle bundle) {

ProjectListFragment projectListFragment = new

ProjectListFragment();

projectListFragment.setArguments(bundle);

return projectListFragment;

}

@Override

protected int getLayoutId() {

return R.layout.fragment\_project\_list;

}

@Override

protected void initView() {

List<ArticleItemData> mArticleList = new ArrayList<>();

mAdapter = new ProjectListAdapter(R.layout.item\_project\_list,

mArticleList);

mAdapter.setOnItemClickListener((adapter, view, position) ->

openArticleDetail(view, position));

mAdapter.setOnItemChildClickListener((adapter, view, position) -> clickChild(view, position));

DividerItemDecoration divider = new

DividerItemDecoration(\_mActivity, DividerItemDecoration.VERTICAL);

divider.setDrawable(ContextCompat.getDrawable(\_mActivity,

R.drawable.shape\_hp\_rv\_line));

rv\_project\_list.addItemDecoration(divider);

rv\_project\_list.setLayoutManager(new LinearLayoutManager(\_mActivity));

rv\_project\_list.setHasFixedSize(true);

rv\_project\_list.setAdapter(mAdapter);

}

@Override

protected void initData() {

assert getArguments() != null;

int projectId = getArguments().getInt("projectId");

srl\_project\_list.setOnRefreshListener(refreshLayout -> {

presenter.refreshLayout(projectId, false);

refreshLayout.finishRefresh();

});

srl\_project\_list.setOnLoadMoreListener(refreshLayout -> {

presenter.loadMore();

refreshLayout.finishLoadMore();

});

presenter.refreshLayout(projectId,true);

}

private void openArticleDetail(View view,int position){

Intent intent = new Intent(\_mActivity,

ArticleDetailActivity.class);

intent.putExtra(Constants.ARTICLE\_LINK,mAdapter.getData().get(position).getLink());

intent.putExtra(Constants.ARTICLE\_TITLE,mAdapter.getData().get(position).getTitle());

intent.putExtra(Constants.ARTICLE\_ID,mAdapter.getData().get(position).getId());

intent.putExtra(Constants.IS\_COLLECTED,mAdapter.getData().get(position).isCollect());

intent.putExtra(Constants.IS\_SHOW\_COLLECT\_ICON,true);

intent.putExtra(Constants.ARTICLE\_ITEM\_POSITION,position);

intent.putExtra(Constants.EVENT\_BUS\_TAG,"project");

startActivity(intent);

}

private void clickChild(View view,int position){

if (presenter.getLoginStatus()) {

if (mAdapter.getData().get(position).isCollect()) {

presenter.cancelAddCollectArticle(position,mAdapter.getData().get(position).getId());

} else {

presenter.addCollectArticle(position,mAdapter.getData().get(position).getId());

}

} else {

Intent intent = new Intent(\_mActivity, LoginActivity.class);

startActivity(intent);

}

}

@Override

public void showProjectListData(ArticleListData articleListData,

boolean isRefresh) {

if (isRefresh) {

mAdapter.replaceData(articleListData.getDatas());

} else {

mAdapter.addData(articleListData.getDatas());

}

}

@Override

public void showCollectSuccess(int position) {

mAdapter.getData().get(position).setCollect(true);

mAdapter.setData(position, mAdapter.getData().get(position));

}

@Override

public void showCancelCollect(int position) {

mAdapter.getData().get(position).setCollect(false);

mAdapter.setData(position, mAdapter.getData().get(position));

}

}

/\*\*

\*项目详情页布局

\*/

<?xml version="1.0" encoding="utf-8"?>

<com.classic.common.MultipleStatusView xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/custom\_multiple\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<com.scwang.smartrefresh.layout.SmartRefreshLayout

android:id="@+id/srl\_project\_list"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

app:srlAccentColor="?attr/colorPrimary"

app:srlEnablePreviewInEditMode="true"

app:srlPrimaryColor="?attr/colorPrimary">

<com.scwang.smartrefresh.header.MaterialHeader

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

<android.support.v7.widget.RecyclerView

android:id="@+id/rv\_project\_list"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/white"

android:visibility="visible" />

<com.scwang.smartrefresh.header.MaterialHeader

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

</com.scwang.smartrefresh.layout.SmartRefreshLayout>

</com.classic.common.MultipleStatusView>

/\*\*

\* 项目列表适配器

\*/

public class ProjectListAdapter extends BaseQuickAdapter

<ArticleItemData, BaseViewHolder> {

public ProjectListAdapter(int layoutResId, @Nullable

List<ArticleItemData> data) {

super(layoutResId, data);

}

@Override

protected void convert(BaseViewHolder helper, ArticleItemData item) {

helper.setText(R.id.item\_project\_list\_title\_tv, Html.fromHtml(item.getTitle()))

.setText(R.id.item\_project\_list\_author\_tv, item.getAuthor())

.setImageResource(R.id.item\_project\_list\_like\_iv, item.isCollect() ? R.drawable.ic\_collect :

R.drawable.ic\_collect\_not);

if (!TextUtils.isEmpty(item.getDesc())) {

helper.setText(R.id.item\_project\_list\_content\_tv, Html.fromHtml(item.getDesc()));

}

if (!TextUtils.isEmpty(item.getNiceDate())) {

helper.setText(R.id.item\_project\_list\_time\_tv, item.getNiceDate());

}

if (!TextUtils.isEmpty(item.getEnvelopePic())) {

GlideUtil.download(mContext, item.getEnvelopePic(),

helper.getView(R.id.item\_project\_list\_iv));

}

helper.addOnClickListener(R.id.item\_project\_list\_like\_iv);

}

}

/\*\*

\* 项目列表适配器布局

\*/

<?xml version="1.0" encoding="utf-8"?>

<android.support.v7.widget.CardView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/item\_search\_pager\_group"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="0dp"

android:layout\_marginRight="0dp"

android:clickable="true"

android:focusable="true"

android:foreground="?android:attr/selectableItemBackground"

app:cardBackgroundColor="@color/white"

app:cardCornerRadius="0dp"

app:cardElevation="1dp">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="180dp"

android:orientation="horizontal"

android:paddingLeft="14dp"

android:paddingTop="10dp"

android:paddingRight="14dp"

android:paddingBottom="10dp">

<ImageView

android:id="@+id/item\_project\_list\_iv"

android:layout\_width="0dp"

android:layout\_height="match\_parent"

android:layout\_weight="2"

android:contentDescription="@null"

app:srcCompat="@mipmap/ic\_launcher" />

<RelativeLayout

android:layout\_width="0dp"

android:layout\_height="match\_parent"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="5dp"

android:layout\_weight="5">

<TextView

android:id="@+id/item\_project\_list\_title\_tv"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="start"

android:ellipsize="end"

android:maxLines="1"

android:text="@string/app\_name"

android:textColor="@color/black"

android:textSize="16sp" />

<TextView

android:id="@+id/item\_project\_list\_content\_tv"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@id/item\_project\_list\_title\_tv"

android:layout\_marginTop="10dp"

android:ellipsize="end"

android:maxLines="4"

android:text="@string/app\_name"

android:textColor="@color/Grey800"

android:textSize="14sp" />

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:orientation="horizontal">

<ImageView

android:id="@+id/item\_project\_list\_like\_iv"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/ic\_collect\_not" />

<TextView

android:id="@+id/item\_project\_list\_time\_tv"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_vertical"

android:layout\_marginStart="10dp"

android:textColor="@color/Grey700"

android:textSize="12sp"

tools:text="time" />

<RelativeLayout

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

android:gravity="center\_vertical">

<TextView

android:id="@+id/item\_project\_list\_author\_tv"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentEnd="true"

android:text="@string/app\_name"

android:textColor="@color/Grey700"

android:textSize="12sp" />

</RelativeLayout>

</LinearLayout>

</RelativeLayout>

</LinearLayout>

</android.support.v7.widget.CardView>

/\*\*

\* 项目相关调用

\*/

public class ProjectPresenter extends Presenter

<ProjectContract.View> implements ProjectContract.Presenter {

@Inject

public ProjectPresenter() {

}

@Override

public void getProjectData() {

addSubscribe(dataManager.getProjectData()

.compose(RxJavaUtil.SchedulerTransformer())

.filter(projectTreeDataList -> mView != null)

.subscribeWith(new BaseObserver<List<ProjectData>>(mView,

MobileLearnApp.getContext().getString(R.string.failed\_to\_get\_project), true) {

@Override

public void success(List<ProjectData> projectData) {

mView.showProjectData(projectData);

}

}));

}

@Override

public void reload() {

getProjectData();

}

}

/\*\*

\* 项目列表相关调用

\*/

public class ProjectListFragmentPresenter extends CollectPresenter

<ProjectListFragmentContract.View> implements

ProjectListFragmentContract.Presenter {

private int currentPage = 1;

private boolean isRefresh = true;

private int projectId;

@Inject

public ProjectListFragmentPresenter() {

}

@Override

public void refreshLayout(int projectId, boolean isShowStatusView) {

isRefresh = true;

currentPage = 1;

this.projectId = projectId;

getProjectListData(isShowStatusView);

}

@Override

public void getProjectListData(boolean isShowStatusView) {

addSubscribe(dataManager.getProjectListData(currentPage, projectId)

.compose(RxJavaUtil.SchedulerTransformer())

.filter(articleListData -> mView != null)

.subscribeWith(new BaseObserver<ArticleListData>(mView,

MobileLearnApp.getContext().getString(R.string.failed\_to\_get\_project),isShowStatusView) {

@Override

public void success(ArticleListData articleListData) {

if(isShowStatusView && currentPage == 1 &&

articleListData.getDatas().size() < 1) {

mView.showEmpty();

} else {

mView.showProjectListData(articleListData,

isRefresh);

}

}

}));

}

@Override

public void reload() {

refreshLayout(projectId, true);

}

@Override

public void loadMore() {

isRefresh = false;

currentPage++;

getProjectListData(false);

}

@Override

public void registerEventBus() {

EventBus.getDefault().register(this);

}

@Override

public void unregisterEventBus() {

EventBus.getDefault().unregister(this);

}

@Subscriber(tag = "project")

public void collectEvent(CollectEvent collectEvent) {

if (mView == null) return;

if (collectEvent.isCancel()) {

mView.showCancelCollect(collectEvent.getArticlePosition());

} else {

mView.showCollectSuccess(collectEvent.getArticlePosition());

}

}

}

/\*\*

\* 我的收藏主页面

\*/

public class CollectFragment extends BaseFragment

<CollectListPresenter> implements CollectListContract.View {

@BindView(R.id.srl\_collect\_list)

SmartRefreshLayout srl\_collect\_list;

@BindView(R.id.rv\_collect\_list)

RecyclerView rv\_collect\_list;

private List<ArticleItemData> mCollectList;

private CollectListAdapter mAdapter;

public static CollectFragment newInstance(){

return new CollectFragment();

}

@Override

protected int getLayoutId() {

return R.layout.fragment\_collect\_list;

}

@Override

protected void initView() {

mCollectList = new ArrayList<>();

mAdapter = new CollectListAdapter(R.layout.item\_article,

mCollectList);

mAdapter.setOnItemClickListener((adapter, view, position) -> startArticleDetail(position));

mAdapter.setOnItemChildClickListener((adapter, view, position) -> collectArticle(position));

DividerItemDecoration divider = new

DividerItemDecoration(\_mActivity,DividerItemDecoration.VERTICAL);

divider.setDrawable(ContextCompat.getDrawable(\_mActivity,R.drawable.shape\_hp\_rv\_line));

rv\_collect\_list.addItemDecoration(divider);

rv\_collect\_list.setLayoutManager(new

LinearLayoutManager(\_mActivity));

rv\_collect\_list.setHasFixedSize(true);

rv\_collect\_list.setAdapter(mAdapter);

}

@Override

protected void initData() {

srl\_collect\_list.setOnRefreshListener(refreshLayout -> {

presenter.getCollectArticle(false);

refreshLayout.finishRefresh();

});

srl\_collect\_list.setOnLoadMoreListener(refreshLayout -> {

presenter.loadMore();

refreshLayout.finishLoadMore();

});

presenter.getCollectArticle(true);

}

private void startArticleDetail(int position){

Intent intent = new Intent(\_mActivity,

ArticleDetailActivity.class);

intent.putExtra(Constants.ARTICLE\_LINK,mAdapter.getData().get(position).getLink());

intent.putExtra(Constants.ARTICLE\_TITLE,mAdapter.getData().get(position).getTitle());

intent.putExtra(Constants.ARTICLE\_ID,mAdapter.getData().get(position).getOriginId());

intent.putExtra(Constants.IS\_COLLECTED,true);

intent.putExtra(Constants.IS\_SHOW\_COLLECT\_ICON,true);

intent.putExtra(Constants.ARTICLE\_ITEM\_POSITION,position);

intent.putExtra(Constants.EVENT\_BUS\_TAG,"collectList");

startActivity(intent);

}

private void collectArticle(int position){

presenter.cancelCollectInCollectPage(position, mAdapter.getData().get(position).getId(),

mAdapter.getData().get(position).getOriginId());

}

/\*\*

\* 展示收藏列表

\* @param articleListData 列表数据

\* @param isRefresh 是否刷新

\*/

@Override

public void showCollectList(ArticleListData articleListData, boolean isRefresh) {

if (isRefresh) {

mAdapter.replaceData(articleListData.getDatas());

} else {

mAdapter.addData(articleListData.getDatas());

}

}

@Override

public void showCollectSuccess(int position) {

Toast.makeText(\_mActivity, getString(R.string.collect\_success),Toast.LENGTH\_SHORT).show();

}

@Override

public void showCancelCollect(int position) {

mAdapter.remove(position);

Toast.makeText(\_mActivity, getString(R.string.cancel\_collect),Toast.LENGTH\_SHORT).show();

}

}

/\*\*

\*我的收藏主页面布局

\*/

<?xml version="1.0" encoding="utf-8"?>

<com.classic.common.MultipleStatusView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/custom\_multiple\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<com.scwang.smartrefresh.layout.SmartRefreshLayout

android:id="@+id/srl\_collect\_list"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

app:srlAccentColor="?attr/colorPrimary"

app:srlEnablePreviewInEditMode="true"

app:srlPrimaryColor="?attr/colorPrimary">

<com.scwang.smartrefresh.header.MaterialHeader

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

<android.support.v7.widget.RecyclerView

android:id="@+id/rv\_collect\_list"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/main\_white"

android:visibility="visible" />

<com.scwang.smartrefresh.header.MaterialHeader

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

</com.scwang.smartrefresh.layout.SmartRefreshLayout>

</com.classic.common.MultipleStatusView>

/\*\*

\* 登录主页面

\*/

public class LoginFragment extends BaseFragment<LoginFragmentPresenter> implements LoginFragmentContract.View {

@BindView(R.id.et\_username)

EditText et\_username;

@BindView(R.id.et\_password)

EditText et\_password;

@BindView(R.id.btn\_login)

TextView btn\_login;

@BindView(R.id.tv\_sign\_up)

TextView tv\_sign\_up;

private LoadingDialog dialog;

public static LoginFragment newInstance(){

return new LoginFragment();

}

@Override

public void loginSuccess() {

Toast.makeText(\_mActivity,getString(R.string.login\_success),Toast.LENGTH\_SHORT).show();

\_mActivity.finish();

}

@Override

public void registerSuccess(RegisterEvent registerEvent) {

et\_username.setText(registerEvent.getUsername());

et\_password.setText(registerEvent.getPassword());

}

@Override

protected int getLayoutId() {

return R.layout.fragment\_login;

}

@Override

protected void initView() {

btn\_login.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

login();

}

});

tv\_sign\_up.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

register();

}

});

}

/\*\*

\* 登录

\*/

private void login() {

String username = et\_username.getText().toString();

String password = et\_password.getText().toString();

if (TextUtils.isEmpty(username)) {

Toast.makeText(\_mActivity, getString(R.string.username\_not\_empty),Toast.LENGTH\_SHORT)

.show();

} else if (TextUtils.isEmpty(password)) {

Toast.makeText(\_mActivity, getString(R.string.password\_not\_empty),Toast.LENGTH\_SHORT)

.show();

} else {

presenter.login(username, password);

}

}

/\*\*

\* 注册

\*/

private void register(){

RegisterFragment registerFragment =

findFragment(RegisterFragment.class);

if (registerFragment == null) {

registerFragment = RegisterFragment.newInstance();

}

getSupportDelegate().showHideFragment(registerFragment, LoginFragment.this);

}

@Override

public void showLoading() {

if (dialog == null) {

dialog = new LoadingDialog(\_mActivity,

getString(R.string.logining), R.mipmap.ic\_dialog\_loading);

}

dialog.show();

}

@Override

public void hideLoading() {

if (dialog != null) {

dialog.dismiss();

dialog = null;

}

}

@Override

protected void initData() {

}

}

/\*\*

\* 登录主页面布局

\*/

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/login\_scroll\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/main\_white"

android:orientation="vertical"

android:scrollbars="none"

android:visibility="visible"

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="18dp"

android:orientation="vertical">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="22dp"

android:text="@string/tv\_login\_title"

android:textColor="@color/Grey980"

android:textSize="28sp" />

<android.support.design.widget.TextInputLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="40dp"

android:layout\_marginTop="35dp"

android:layout\_marginRight="40dp"

android:layout\_marginBottom="8dp"

android:focusable="true"

android:focusableInTouchMode="true">

<EditText

android:id="@+id/et\_username"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/username"

android:inputType="textEmailAddress"

android:textColor="@color/Grey980"

android:textSize="15sp" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="40dp"

android:layout\_marginTop="15dp"

android:layout\_marginRight="40dp"

android:layout\_marginBottom="12dp"

android:focusable="true"

android:focusableInTouchMode="true">

<EditText

android:id="@+id/et\_password"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/password"

android:inputType="textPassword"

android:textColor="@color/Grey980"

android:textSize="15sp" />

</android.support.design.widget.TextInputLayout>

<TextView

android:id="@+id/tv\_sign\_up"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="end"

android:layout\_marginRight="40dp"

android:text="@string/register"

android:textColor="#999999"

android:textSize="15sp" />

<TextView

android:id="@+id/btn\_login"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="40dp"

android:layout\_marginTop="40dp"

android:layout\_marginRight="40dp"

android:background="@drawable/btn\_login\_bg"

android:gravity="center"

android:padding="12dp"

android:text="@string/login"

android:textColor="@color/white"

android:textSize="16sp" />

</LinearLayout>

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/login\_protocol"

android:layout\_alignParentBottom="true"

android:layout\_centerHorizontal="true"

android:layout\_marginBottom="30dp"

android:textColor="#999999" />

</RelativeLayout>

</LinearLayout>

/\*\*

\* 登录页相关调用

\*/

public class LoginFragmentPresenter extends

Presenter<LoginFragmentContract.View>implements LoginFragmentContract.Presenter {

@Inject

public LoginFragmentPresenter() {

}

@Override

public void login(String username, String password) {

addSubscribe(dataManager.login(username, password)

.compose(RxJavaUtil.SchedulerTransformer())

.filter(loginData -> mView != null)

.subscribeWith(new BaseObserver<LoginData>(mView,

MobileLearnApp.getContext().getString(R.string.failed\_to\_login), true) {

@Override

public void success(LoginData loginData) {

setLoginStatus(true);

setLoginAccount(loginData.getUsername());

EventBus.getDefault().post(new LoginEvent());

mView.loginSuccess();

}

}));

}

@Override

public void registerEventBus() {

EventBus.getDefault().register(this);

}

@Override

public void unregisterEventBus() {

EventBus.getDefault().unregister(this);

}

@Subscriber()

public void RegisterSuccessEvent(RegisterEvent registerEvent) {

mView.registerSuccess(registerEvent);

}

}

public interface LoginFragmentContract {

interface View extends BaseView {

void loginSuccess();

void registerSuccess(RegisterEvent registerEvent);

}

interface Presenter extends BasePresenter<View> {

void login(String username,String password);

}

}

/\*\*

\* 注册主页面

\*/

public class RegisterFragment extends BaseFragment

<RegisterPresenter> implements RegisterContract.View {

@BindView(R.id.register\_username)

EditText register\_username;

@BindView(R.id.register\_password)

EditText register\_password;

@BindView(R.id.confirm\_password)

EditText confirm\_password;

@BindView(R.id.btn\_register)

TextView btn\_register;

@BindView(R.id.tv\_login)

TextView tv\_login;

private AlertDialog dialog;

public static RegisterFragment newInstance(){

return new RegisterFragment();

}

@Override

protected int getLayoutId() {

return R.layout.fragment\_register;

}

@Override

protected void initView() {

btn\_register.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

register();

}

});

tv\_login.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

goLogin();

}

});

}

/\*\*

\* 注册

\*/

private void register() {

String username = register\_username.getText().toString();

String password = register\_password.getText().toString();

String password2 = confirm\_password.getText().toString();

if (TextUtils.isEmpty(username)) {

Toast.makeText(\_mActivity, getString(R.string.username\_not\_empty),Toast.LENGTH\_SHORT)

.show();

} else if (TextUtils.isEmpty(password)) {

Toast.makeText(\_mActivity, getString(R.string.password\_not\_empty),Toast.LENGTH\_SHORT)

.show();

} else if (TextUtils.isEmpty(password2)) {

Toast.makeText(\_mActivity, getString(R.string.confirm\_password\_not\_empty),Toast.LENGTH\_SHORT).show();

} else if (!TextUtils.equals(password, password2)) {

Toast.makeText(\_mActivity, getString(R.string.twice\_password\_not\_same),Toast.LENGTH\_SHORT).show();

} else {

presenter.register(username, password, password2);

}

}

@Override

protected void initData() {

}

@Override

public void registerSuccess() {

Toast.makeText(\_mActivity,getString(R.string.register\_success),Toast.LENGTH\_SHORT).show();

goLogin();

}

private void goLogin() {

LoginFragment loginFragment = findFragment(LoginFragment.class);

if (loginFragment == null) {

loginFragment = LoginFragment.newInstance();

}

getSupportDelegate().showHideFragment(loginFragment, RegisterFragment.this);

}

@Override

public void showLoading() {

if (dialog == null) {

View view = LayoutInflater.from(\_mActivity)

.inflate(R.layout.login\_progress\_bar, null, false);

TextView loadingText = view.findViewById(R.id.loading\_text);

loadingText.setText(getResources().getString(R.string.logining));

dialog = new AlertDialog.Builder(\_mActivity)

.setView(view).create();

Objects.requireNonNull(dialog.getWindow()).setBackgroundDrawableResource(android.R.color.transparent);

}

dialog.show();

}

@Override

public void hideLoading() {

if (dialog != null) {

dialog.dismiss();

dialog = null;

}

}

}

/\*\*

\* 注册主页面布局

\*/

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/login\_scroll\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/main\_white"

android:orientation="vertical"

android:scrollbars="none"

android:visibility="visible"

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="18dp"

android:orientation="vertical">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="22dp"

android:text="@string/tv\_login\_title"

android:textColor="@color/Grey980"

android:textSize="28sp" />

<android.support.design.widget.TextInputLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="40dp"

android:layout\_marginTop="35dp"

android:layout\_marginRight="40dp"

android:layout\_marginBottom="8dp"

android:focusable="true"

android:focusableInTouchMode="true">

<EditText

android:id="@+id/register\_username"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/username"

android:inputType="textEmailAddress"

android:textColor="@color/Grey980"

android:textSize="15sp" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="40dp"

android:layout\_marginTop="15dp"

android:layout\_marginRight="40dp"

android:layout\_marginBottom="12dp"

android:focusable="true"

android:focusableInTouchMode="true">

<EditText

android:id="@+id/register\_password"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/password"

android:inputType="textPassword"

android:textColor="@color/Grey980"

android:textSize="15sp" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="40dp"

android:layout\_marginTop="15dp"

android:layout\_marginRight="40dp"

android:layout\_marginBottom="12dp"

android:focusable="true"

android:focusableInTouchMode="true">

<EditText

android:id="@+id/confirm\_password"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/confirm\_password"

android:inputType="textPassword"

android:textColor="@color/Grey980"

android:textSize="15sp" />

</android.support.design.widget.TextInputLayout>

<TextView

android:id="@+id/tv\_login"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="end"

android:layout\_marginRight="40dp"

android:text="@string/back\_login"

android:textColor="#999999"

android:textSize="15sp" />

<TextView

android:id="@+id/btn\_register"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="40dp"

android:layout\_marginTop="40dp"

android:layout\_marginRight="40dp"

android:background="@drawable/btn\_login\_bg"

android:gravity="center"

android:padding="12dp"

android:text="@string/confirm\_register"

android:textColor="@color/white"

android:textSize="16sp" />

</LinearLayout>

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/register\_protocol"

android:layout\_alignParentBottom="true"

android:layout\_centerHorizontal="true"

android:layout\_marginBottom="30dp"

android:textColor="#999999" />

</RelativeLayout>

</LinearLayout>

/\*\*

\* 注册主页面相关调用

\*/

public class RegisterPresenter extends Presenter<RegisterContract.View> implements RegisterContract.Presenter {

@Inject

public RegisterPresenter() {

}

@Override

public void register(String username, String password,String

confirmPassword) {

addSubscribe(dataManager.register(username,password,confirmPassword)

.compose(RxJavaUtil.SchedulerTransformer())

.filter(loginDataBaseResponse -> mView != null)

.subscribeWith(new BaseObserver<LoginData>(mView,

MobileLearnApp.getContext().getString(R.string.failed\_to\_register), true) {

@Override

public void success(LoginData loginData) {

EventBus.getDefault().post(new RegisterEvent(loginData.getUsername(),loginData.getPassword()));

mView.registerSuccess();

}

}));

}

}

public interface RegisterContract {

interface View extends BaseView {

void registerSuccess();

}

interface Presenter extends BasePresenter<View>{

void register(String username,String password,String

confirmPassword);

}

}

/\*\*

\* 自定义 Application

\*/

public class MobileLearnApp extends Application implements

HasActivityInjector{

private static Context context;

@Inject

DispatchingAndroidInjector<Activity> dispatchingAndroidInjector;

@Inject

public DataManager mDataManager;

private RefWatcher refWatcher;

@Override

public void onCreate() {

super.onCreate();

context = getApplicationContext();

DaggerAppComponent.builder()

.appModule(new AppModule(this))

.httpModule(new HttpModule())

.build().inject(this);

refWatcher = setLeakCanary();

}

/\*\*

\* 内存泄漏检测工具

\* @return 如果当前进程是给LeakCanary堆分析则返回

\*/

private RefWatcher setLeakCanary() {

if (LeakCanary.isInAnalyzerProcess(this)) {

return RefWatcher.DISABLED;

}

return LeakCanary.install(this);

}

public static RefWatcher getRefWatcher(Context context){

MobileLearnApp mobileLearnApp = (MobileLearnApp)

context.getApplicationContext();

return mobileLearnApp.refWatcher;

}

@Override

public AndroidInjector<Activity> activityInjector() {

return dispatchingAndroidInjector;

}

public static Context getContext(){

return context;

}

}

/\*\*

\* Module 告诉 Component 可以从这个类中获取依赖对象，生成各种类

\* Provides 用来生成类的实例

\* Name 存在相同依赖，告知 Component 到底提供哪一个依赖对象

\* Singleton 局部单例，可以保证注入的相应类中是单例

\* 需要参数的依赖对象，也要提供参数的方法

\*/

@Module

public class AppModule {

private final MobileLearnApp application;

public AppModule(MobileLearnApp application){

this.application = application;

}

@Provides

@Singleton

MobileLearnApp provideApplicationContext(){

return application;

}

@Provides

@Singleton

HttpHelper provideHttpHelper(HttpHelperImpl httpHelperImpl){

return httpHelperImpl;

}

@Provides

@Singleton

PreHelper providePreferenceHelper(PreHelperImpl preHelper){

return preHelper;

}

@Provides

@Singleton

DbHelper provideDbHelper(DbHelperImpl dbHelper){

return dbHelper;

}

@Provides

@Singleton

DataManager provideDataManager(HttpHelper httpHelper,PreHelper

preHelper,DbHelper dbHelper){

return new DataManager(httpHelper,preHelper,dbHelper);

}

}

@Module

public class HttpModule {

@Singleton

@Provides

ApiService provideApiService(Retrofit retrofit) {

return retrofit.create(ApiService.class);

}

@Singleton

@Provides

Retrofit provideRetrofit(Retrofit.Builder builder, OkHttpClient

client) {

return createRetrofit(builder, client, ApiService.BASE\_URL);

}

@Singleton

@Provides

Retrofit.Builder provideRetrofitBuilder() {

return new Retrofit.Builder();

}

//OkHttpClient 创建

@Singleton

@Provides

OkHttpClient.Builder provideOkHttpBuilder() {

return new OkHttpClient.Builder();

}

@Singleton

@Provides

OkHttpClient provideOkHttpClient(OkHttpClient.Builder builder) {

if (BuildConfig.DEBUG) {

//消息拦截器，可看到接口返回的所有内容 BASIC 级别是请求/响应行

HttpLoggingInterceptor loggingInterceptor = new

HttpLoggingInterceptor();

loggingInterceptor.setLevel(HttpLoggingInterceptor.Level.BASIC);

builder.addInterceptor(loggingInterceptor);

}

//设置网络缓存

File cacheFile = new File(CacheUtils.getCacheDir());

System.out.println("HttpModule 设置缓存路径" + cacheFile);

Cache cache = new Cache(cacheFile, 1024 \* 1024 \* 100);//100M

builder.addNetworkInterceptor(new NetCacheInterceptor());

builder.cache(cache);

//设置超时时间

builder.connectTimeout(8, TimeUnit.SECONDS);

builder.readTimeout(15, TimeUnit.SECONDS);

builder.writeTimeout(15, TimeUnit.SECONDS);

//设置重新连接

builder.retryOnConnectionFailure(true);

//持久化 Cookie

builder.cookieJar(new PersistentCookieJar(new SetCookieCache(),

new SharedPrefsCookiePersistor(MobileLearnApp.getContext())));

return builder.build();

}

private Retrofit createRetrofit(Retrofit.Builder builder,

OkHttpClient client, String baseUrl) {

return builder

.baseUrl(baseUrl)

.client(client)

.addCallAdapterFactory(RxJava2CallAdapterFactory.create())

.addConverterFactory(GsonConverterFactory.create()) //把retrofit的请求转化为RxJava

// 的Observable

.build();

}

}

public class HttpHelperImpl implements HttpHelper {

private ApiService mApiService;

@Inject

HttpHelperImpl(ApiService apiService) {

mApiService = apiService;

}

@Override

public Observable<BaseResponse<ArticleListData>>

getArticleList(int num) {

return mApiService.getArticleList(num);

}

@Override

public Observable<BaseResponse<ArticleListData>>

addArticleCollect(int id) {

return mApiService.addArticleCollect(id);

}

@Override

public Observable<BaseResponse<ArticleListData>>

cancelAddArticleCollect(int id) {

return mApiService.cancelAddArticleCollect(id);

}

@Override

public Observable<BaseResponse<List<ArticleItemData>>>

getUpArticles() {

return mApiService.getUpArticles();

}

@Override

public Observable<BaseResponse<List<BannerData>>> getBannerData() {

return mApiService.getBannerData();

}

@Override

public Observable<BaseResponse<List<PublicNumData>>>

getPublicNumData() {

return mApiService.getPublicNumData();

}

@Override

public Observable<BaseResponse<ArticleListData>>

getPersonalPublicNumData(int id,int page) {

return mApiService.getPersonalPublicNumData(id,page);

}

@Override

public Observable<BaseResponse<List<KnowledgeSystemData>>>

getKnowledgeData() {

return mApiService.getKnowledgeData();

}

@Override

public Observable<BaseResponse<List<ProjectData>>> getProjectData() {

return mApiService.getProjectData();

}

@Override

public Observable<BaseResponse<ArticleListData>>

getProjectListData(int page, int projectId) {

return mApiService.getProjectListData(page,projectId);

}

@Override

public Observable<BaseResponse<List<NavigationData>>>

getNavigationData() {

return mApiService.getNavigationData();

}

@Override

public Observable<BaseResponse<LoginData>>

login(String username, String password) {

return mApiService.login(username,password);

}

@Override

public Observable<BaseResponse<LoginData>>

register(String username, String password,String confirmPassword) {

return mApiService.register(username,password,confirmPassword);

}

@Override

public Observable<BaseResponse<LoginData>> logout() {

return mApiService.logout();

}

@Override

public Observable<BaseResponse<ArticleListData>>

cancelCollectInCollectPage(int id,

int originId) {

return mApiService.cancelCollectInCollectPage(id,originId);

}

@Override

public Observable<BaseResponse<ArticleListData>>

getCollectList(int page) {

return mApiService.getCollectList(page);

}

@Override

public Observable<BaseResponse<List<WebsiteData>>> getWebsiteList() {

return mApiService.getWebsiteList();

}

@Override

public Observable<BaseResponse<List<HotSearchData>>> getSearchData() {

return mApiService.getSearchData();

}

@Override

public Observable<BaseResponse<ArticleListData>>

getSearchResultList(int pageNum, String key) {

return mApiService.getSearchResultList(pageNum,key);

}

/\*\*

\* 数据库实现类

\*/

public class DbHelperImpl implements DbHelper {

private final DaoMaster daoMaster;

private final DaoSession daoSession;

private String data;

private HistoryData historyData;

private List<HistoryData> historyDataList;

private static final int HISTORY\_LIST\_SIZE = 10;

@Inject

public DbHelperImpl() {

DaoMaster.DevOpenHelper devOpenHelper = new DaoMaster

.DevOpenHelper(MobileLearnApp.getContext(),Constants.DB\_NAM);

SQLiteDatabase database = devOpenHelper

.getWritableDatabase();

daoMaster = new DaoMaster(database);

daoSession = daoMaster.newSession();

}

/\*\*

\* 添加搜索历史

\* @param data

\* @return

\*/

@Override

public List<HistoryData> addHistoryData(String data) {

this.data = data;

getHistoryList();

createHistoryData();

if (historyDataForward()) {

return historyDataList;

}

if (historyDataList.size() < HISTORY\_LIST\_SIZE) {

getHistoryDataDao().insert(historyData);

} else {

historyDataList.remove(0);

historyDataList.add(historyData);

getHistoryDataDao().deleteAll();

getHistoryDataDao().insertInTx(historyDataList);

}

return historyDataList;

}

/\*\*

\* 清空搜索历史

\*/

@Override

public void clearHistoryData() {

daoSession.getHistoryDataDao().deleteAll();

}

/\*\*

\* 删除某一条搜索历史

\* @param id

\*/

@Override

public void deleteHistoryDataId(Long id) {

daoSession.getHistoryDataDao().deleteByKey(id);

}

/\*\*

\* 加载搜索历史

\* @return

\*/

@Override

public List<HistoryData> loadAllHistoryData() {

return daoSession.getHistoryDataDao().loadAll();

}

/\*\*

\* 获取历史搜索列表

\*/

private void getHistoryList() {

historyDataList = getHistoryDataDao().loadAll();

}

private void createHistoryData() {

historyData = new HistoryData();

historyData.setDate(System.currentTimeMillis());

historyData.setData(data);

}

private HistoryDataDao getHistoryDataDao() {

return daoSession.getHistoryDataDao();

}

/\*\*

\* 历史数据前移

\*

\* @return 返回true表示查询的数据已存在

\*/

private boolean historyDataForward() {

//重复搜索时进行历史记录前移

Iterator<HistoryData> iterator = historyDataList.iterator();

//不要在foreach循环中进行元素的remove、add操作，使用Iterator模式

while (iterator.hasNext()) {

HistoryData historyData1 = iterator.next();

if (historyData1.getData().equals(data)) {

historyDataList.remove(historyData1);

historyDataList.add(historyData);

getHistoryDataDao().deleteAll();

getHistoryDataDao().insertInTx(historyDataList);

return true;

}

}

return false;

}

}

@Singleton

@Component(modules = {

AndroidInjectionModule.class, //支持四大组件都可以被绑定

AndroidSupportInjectionModule.class,

AllActivityModule.class,

AllFragmentModule.class,

AppModule.class,

HttpModule.class

})

public interface AppComponent {

void inject(MobileLearnApp mobileLearnApp);

}

/\*\*

\* 系统中用到的颜色代码

\*/

<?xml version="1.0" encoding="utf-8"?>

<resources>

<color name="colorPrimary">#008577</color>

<color name="colorPrimaryDark">#00574B</color>

<color name="colorAccent">#28BFA0</color>

<color name="lightWhite">#cccccc</color>

<!--<color name="bottom\_select\_green">#08C261</color>-->

<color name="bottom\_select\_green">#28BFA0</color>

<color name="white">#FFFFFF</color>

<color name="white\_smoke">#F7F7F7</color>

<color name="Grey50">#FAFAFA</color>

<color name="Grey100">#F5F5F5</color>

<color name="Grey200">#EEEEEE</color>

<color name="Grey250">#EDEDED</color>

<color name="Grey280">#E2E2E2</color>

<color name="Grey300">#E0E0E0</color>

<color name="Grey350">#D9D9D9</color>

<color name="Grey400">#bdbdbd</color>

<color name="Grey500">#9e9e9e</color>

<color name="Grey600">#757575</color>

<color name="Grey700">#616161</color>

<color name="Grey750">#363636</color>

<color name="Grey800">#3D3D3D</color>

<color name="Grey900">#212121</color>

<color name="Grey950">#181818</color>

<color name="Grey980">#333333</color>

<color name="Grey1000">#1A1A1A</color>

<color name="black">#19191B</color>

<color name="black\_pure">#000000</color>

<color name="navigation\_item\_ic\_color">#999999</color>

<color name="red">#f44336</color>

<color name="navigation\_bar">#F0F0F0</color>

<color name="bottom\_bar">#999999</color>

<color name="bottom\_bar\_bg">#FFFFFF</color>

<color name="main\_white">#FDFDFD</color>

<color name="green">#28BFA0</color>

</resources>

/\*\*

\* SharedPreferences 实现类

\*/

public class PreHelperImpl implements PreHelper {

private final SharedPreferences sp;

@Inject

PreHelperImpl(){

sp = MobileLearnApp.getContext().getSharedPreferences(Constants.MY\_SP, Context.MODE\_PRIVATE);

}

@Override

public void setLoginStatus(boolean isLogin) {

sp.edit().putBoolean(Constants.LOGIN\_STATUS,isLogin).apply();

}

@Override

public boolean getLoginStatus() {

return sp.getBoolean(Constants.LOGIN\_STATUS,false);

}

@Override

public void setLoginAccount(String account) {

sp.edit().putString(Constants.ACCOUNT,account).apply();

}

@Override

public String getLoginAccount() {

return sp.getString(Constants.ACCOUNT,"");

}

@Override

public void setNightMode(boolean isNightMode) {

}

@Override

public boolean isNightMode() {

return false;

}

}

public interface PreHelper {

void setLoginStatus(boolean isLogin);

boolean getLoginStatus();

void setLoginAccount(String account);

String getLoginAccount();

void setNightMode(boolean isNightMode);

boolean isNightMode();

}

//系统底部按钮布局

<?xml version="1.0" encoding="utf-8"?>

<menu xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto">

<item

android:id="@+id/tab\_home\_pager"

android:enabled="true"

android:icon="@drawable/ic\_home\_pager"

android:title="@string/home\_page"

app:showAsAction="ifRoom"> <!--如果有足够的空间会显示-->

</item>

<item

android:id="@+id/tab\_navigation"

android:enabled="true"

android:title="@string/navigation"

android:icon="@drawable/ic\_navigation"

app:showAsAction="ifRoom">

</item>

<item

android:id="@+id/tab\_knowledge"

android:enabled="true"

android:title="@string/knowledge"

android:icon="@drawable/ic\_knowledge"

app:showAsAction="ifRoom">

</item>

<item

android:id="@+id/tab\_public\_num"

android:enabled="true"

android:title="@string/public\_num"

android:icon="@drawable/ic\_public\_num"

app:showAsAction="ifRoom">

</item>

<item

android:id="@+id/tab\_project"

android:enabled="true"

android:title="@string/project"

android:icon="@drawable/ic\_project"

app:showAsAction="ifRoom">

</item>

</menu>

/\*\*

\* 系统清单文件

\*/

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.sunshinexu.mobilelearn">

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

<uses-permission

android:name="android.permission.ACCESS\_NETWORK\_STATE" />

<uses-permission

android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />

<application

android:name=".app.MobileLearnApp"

android:allowBackup="true"

android:icon="@drawable/ic\_logo1"

android:label="@string/app\_name"

android:supportsRtl="true"

android:theme="@style/AppTheme">

<activity

android:name=".activity.main.ui.SplashActivity"

android:theme="@style/SplashTheme">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<action android:name="android.intent.action.VIEW" />

<category

android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<activity android:name=".MainActivity" />

<activity android:name=".activity.main.ui.LoginActivity"/>

<activity

android:name=".activity.main.ui.ArticleDetailActivity" />

<activity

android:name=".activity.main.ui.KnowledgeDetailActivity"/>

<activity android:name=".activity.main.ui.ItemActivity"/>

<activity android:name=".activity.main.ui.SearchActivity"/>

</application>

</manifest>