# Day03 商家列表

**商家页面**

1.跳转到商家页面

view.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intent = new Intent(mContext, BusinessActivity.class);  
 intent.putExtra("seller", mSeller); //序列化  
 mContext.startActivity(intent);  
 }  
});

2.挂载3个子页面并且添加指示器

private void initFragments() {  
 mFragmentList.add(new GoodsFragment());  
 mFragmentList.add(new CommentsFragment());  
 mFragmentList.add(new SellerFragment());  
  
 mPagerAdapter = new BusinessFragmentPagerAdapter(getFragmentManager());  
 mPagerAdapter.setFragmentList(mFragmentList);  
 mVp.setAdapter(mPagerAdapter);  
  
 mTabs.setupWithViewPager(mVp);  
}

3.商品界面接口

@GET("business")  
Call<ResponseInfo> getBusinessInfo(@Query("sellerId") int sellerId);

4.创建GoodsFragmentPresenter

public class GoodsFragmentPresenter extends NetPresenter {  
 private GoodsFragment mGoodsFragment;  
 private List<GoodsTypeInfo> mGoodsTypeInfoList;  
  
 public GoodsFragmentPresenter(GoodsFragment goodsFragment) {  
 this.mGoodsFragment = goodsFragment;  
 }  
  
 public void getBusinessInfo(int sellerId){  
 Call<ResponseInfo> businessCall = mTakeoutService.getBusinessInfo(sellerId);  
 businessCall.enqueue(mCallback);  
 }

5.解析数据

Gson gson = new Gson();  
try {  
 JSONObject jsonObj = new JSONObject(jsonData);  
 String typeListStr = jsonObj.getString("list"); //注意多嵌套了一层  
 mGoodsTypeInfoList = gson.fromJson(typeListStr, new TypeToken<List<GoodsTypeInfo>>() {  
 }.getType());  
 Log.e("business", "一共有" + mGoodsTypeInfoList.size() + "种商品类别");  
  
 mGoodsFragment.onGoodsTypeSuccess(mGoodsTypeInfoList);

6.刷新adapter实现左侧列表

public void onGoodsTypeSuccess(List<GoodsTypeInfo> goodsTypeInfoList) {  
 mGoodsTypeRvAdapter.setGoodsTypeInfoList(goodsTypeInfoList);  
}

**右侧列表实现**

1.双向绑定javabean

for (int i = 0; i < mGoodsTypeInfoList.size(); i++) {  
 GoodsTypeInfo goodsTypeInfo = mGoodsTypeInfoList.get(i);  
 //查找当前类别有多少个已点商品  
 List<GoodsInfo> aTypeGoodsList = goodsTypeInfo.getList(); //一个类别的商品  
 for (int j = 0; j < aTypeGoodsList.size(); j++) {  
 **GoodsInfo goodsInfo = aTypeGoodsList.get(j);  
 goodsInfo.setTypeId(goodsTypeInfo.getId());  
 goodsInfo.setTypeName(goodsTypeInfo.getName());**  
 }  
 mAllTypeGoodsList.addAll(aTypeGoodsList);  
}

mGoodsFragment.onGoodsSuccess(mAllTypeGoodsList);

2.右侧Adapter使用双向绑定效果

@Override  
public View getHeaderView(int position, View convertView, ViewGroup parent) {  
 GoodsInfo goodsInfo = mGoodsInfoList.get(position); //馒头  
 View headerView = LayoutInflater.from(mContext).inflate(R.layout.item\_type\_header,parent,fals e);  
 ((TextView) headerView).setText(goodsInfo.getTypeName());  
 return headerView;  
}  
  
@Override  
public long getHeaderId(int position) {  
 GoodsInfo goodsInfo = mGoodsInfoList.get(position); //馒头  
 return goodsInfo.getTypeId();  
}

3.右侧商品item赋值

public void setData(GoodsInfo goodsInfo) {  
 this.mGoodsInfo = goodsInfo;  
 mTvName.setText(goodsInfo.getName());  
 Picasso.with(mContext).load(goodsInfo.getIcon()).into(mIvIcon);  
 mTvZucheng.setText(goodsInfo.getForm());  
 mTvYueshaoshounum.setText("月售" + goodsInfo.getMonthSaleNum() + "份");  
 mTvNewprice.setText(PriceFormater.format(goodsInfo.getNewPrice()));  
 if(goodsInfo.getOldPrice() > 0){  
 mTvOldprice.setVisibility(View.VISIBLE);  
 //画线效果  
 mTvOldprice.getPaint().setFlags(Paint.STRIKE\_THRU\_TEXT\_FLAG);  
 }else{  
 mTvOldprice.setVisibility(View.GONE);  
 }  
 mTvOldprice.setText(PriceFormater.format(goodsInfo.getOldPrice()));  
 mTvCount.setText(goodsInfo.getCount() + "");  
 if(goodsInfo.getCount() > 0){  
 mTvCount.setVisibility(View.VISIBLE);  
 mIbMinus.setVisibility(View.VISIBLE);  
 }else{  
 //INVIABLE和GONE的区别  
 mTvCount.setVisibility(View.INVISIBLE);  
 mIbMinus.setVisibility(View.INVISIBLE);  
 }  
}

**双列表视图联动效果(点击左侧，右侧跳选）**

1.左侧选中效果

public int selectIndex = 0;  
class ViewHolder extends RecyclerView.ViewHolder {  
 private View mView;  
 @Bind(R.id.tvCount)  
 TextView mTvCount;  
 @Bind(R.id.type)  
 TextView mType;  
 private int mPosition;  
  
 ViewHolder(View view) {  
 super(view);  
 ButterKnife.bind(this, view);  
 this.mView = view;  
 view.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 selectIndex = mPosition;  
 notifyDataSetChanged();

}

public void setData(GoodsTypeInfo goodsTypeInfo, int position) {  
 this.mPosition = position;  
 //1.对不同的position做不同的ui处理  
 if(position == selectIndex){  
 //白底黑字  
 mView.setBackgroundColor(Color.WHITE);  
 mType.setTextColor(Color.BLACK);  
 mType.setTypeface(Typeface.DEFAULT\_BOLD);  
 }else{  
 //灰底灰字 #b9dedcdc  
 mView.setBackgroundColor(Color.parseColor("#b9dedcdc"));  
 mType.setTextColor(Color.GRAY);  
 mType.setTypeface(Typeface.DEFAULT);  
 }  
 mType.setText(goodsTypeInfo.getName());

}

2.左侧选中后，查找右侧应该跳转的位置

int typeId = mGoodsTypeInfoList.get(selectIndex).getId();  
int position = mGoodsFragment.mGoodsFragmentPresenter.getGoodsPositionByTypeId(typeId);

public int getGoodsPositionByTypeId(int typeId){  
 int position = -1;  
 for(int j=0;j < mAllTypeGoodsList.size();j++){  
 GoodsInfo goodsInfo = mAllTypeGoodsList.get(j);  
 if(goodsInfo.getTypeId() == typeId){  
 position = j;  
 break;  
 }  
 }  
 return position;  
}

3.让右侧列表跳转到此position

mGoodsFragment.mSlhlv.setSelection(position);

**双列表视图联动效果(右侧滚动，左侧跳选）**

1.右侧滚动效果

mGoodsFragment.mSlhlv.setOnScrollListener(new AbsListView.OnScrollListener() {  
 @Override  
 public void onScrollStateChanged(AbsListView view, int scrollState) {  
  
 }  
  
 @Override  
 public void onScroll(AbsListView view, int firstVisibleItem, int visibleItemCount, int totalItemCount) {  
 int oldIndex = mGoodsFragment.mGoodsTypeRvAdapter.selectIndex;  
 int oldTypeId = mGoodsTypeInfoList.get(oldIndex).getId();

}

2.右侧滚动后，根据firstvisibleItem查找左侧

int newTypeId = mAllTypeGoodsList.get(firstVisibleItem).getTypeId();

public int getGoodsPositionByTypeId(int typeId){  
 int position = -1;  
 for(int j=0;j < mAllTypeGoodsList.size();j++){  
 GoodsInfo goodsInfo = mAllTypeGoodsList.get(j);  
 if(goodsInfo.getTypeId() == typeId){  
 position = j;  
 break;  
 }  
 }  
 return position;  
}

3.typeId变化时，切换selectIndex

if(newTypeId != oldTypeId){  
 //3.切换到新的index,根据用心营养套餐的typeid算出来index  
 int newIndex = getTypePositionByTypeId(newTypeId);  
 mGoodsFragment.mGoodsTypeRvAdapter.selectIndex = newIndex;  
 mGoodsFragment.mGoodsTypeRvAdapter.notifyDataSetChanged();  
}

**添加商品**

1.按钮点击事件

@OnClick({R.id.ib\_add,R.id.ib\_minus})  
public void onAddOrMinusClick(View view){  
 boolean isAdd = false;  
 switch (view.getId()){  
 case R.id.ib\_add:  
 isAdd = true;  
 doAddOperation();  
 break;  
 case R.id.ib\_minus:  
 doMinusOperation();  
 break;  
 }

2.显示动画

private AnimationSet getShowAnimation() {  
 AnimationSet animationSet = new AnimationSet(false);  
 animationSet.setDuration(SHOW\_DURATION);  
  
 AlphaAnimation alphaAnimation = new AlphaAnimation(0,1);  
 animationSet.setDuration(SHOW\_DURATION);  
 animationSet.addAnimation(alphaAnimation);  
  
 RotateAnimation rotateAnimation = new RotateAnimation(0,720, Animation.RELATIVE\_TO\_SELF,0.5f,  
 Animation.RELATIVE\_TO\_SELF,0.5f);  
 rotateAnimation.setDuration(SHOW\_DURATION);  
 animationSet.addAnimation(rotateAnimation);  
  
 TranslateAnimation translateAnimation = new TranslateAnimation(  
 Animation.RELATIVE\_TO\_SELF,2,  
 Animation.RELATIVE\_TO\_SELF,0,  
 Animation.RELATIVE\_TO\_SELF,0,  
 Animation.RELATIVE\_TO\_SELF,0  
 );  
 translateAnimation.setDuration(SHOW\_DURATION);  
 animationSet.addAnimation(translateAnimation);  
 return animationSet;  
}

3.刷新数量

count ++;  
mGoodsInfo.setCount(count);  
notifyDataSetChanged();

**减少商品**

1.按钮点击事件

@OnClick({R.id.ib\_add,R.id.ib\_minus})  
public void onAddOrMinusClick(View view){  
 boolean isAdd = false;  
 switch (view.getId()){  
 case R.id.ib\_add:  
 isAdd = true;  
 doAddOperation();  
 break;  
 case R.id.ib\_minus:  
 doMinusOperation();  
 break;  
 }

2.显示动画

private AnimationSet getHideAnimation() {  
 AnimationSet animationSet = new AnimationSet(false);  
 animationSet.setDuration(SHOW\_DURATION);  
  
 AlphaAnimation alphaAnimation = new AlphaAnimation(1,0);  
 alphaAnimation.setDuration(SHOW\_DURATION);  
 animationSet.addAnimation(alphaAnimation);  
  
 RotateAnimation rotateAnimation = new RotateAnimation(720,0, Animation.RELATIVE\_TO\_SELF,0.5f,  
 Animation.RELATIVE\_TO\_SELF,0.5f);  
 rotateAnimation.setDuration(SHOW\_DURATION);  
 animationSet.addAnimation(rotateAnimation);  
  
 TranslateAnimation translateAnimation = new TranslateAnimation(  
 Animation.RELATIVE\_TO\_SELF,0,  
 Animation.RELATIVE\_TO\_SELF,2,  
 Animation.RELATIVE\_TO\_SELF,0,  
 Animation.RELATIVE\_TO\_SELF,0  
 );  
 translateAnimation.setDuration(SHOW\_DURATION);  
 animationSet.addAnimation(translateAnimation);  
 return animationSet;  
}

3.刷新数量

count --;  
mGoodsInfo.setCount(count);  
notifyDataSetChanged();

**红点功能**

1. 红点4步骤

private void processRedDot(boolean isAdd) {  
 //点击增加或者减少馒头  
 int typeId = mGoodsInfo.getTypeId(); //粗粮主食  
 //2.根据粗粮主食id找到它的排序position  
 int position = mGoodsFragment.mGoodsFragmentPresenter.getTypePositionByTypeId(typeId);  
 //3.从左侧列表中找到第postion个GoodstypeInfo  
 GoodsTypeInfo goodsTypeInfo = mGoodsFragment.mGoodsFragmentPresenter.mGoodsTypeInfoList.get(position);  
 int count = goodsTypeInfo.getCount();  
 if (isAdd) {  
 count++;  
 } else {  
 count--;  
 }  
 goodsTypeInfo.setCount(count);  
 //4.刷新左侧列表  
 mGoodsFragment.mGoodsTypeRvAdapter.notifyDataSetChanged();  
}

2. GoodsTypeAdapter增加count

if(goodsTypeInfo.getCount()>0) {  
 mTvCount.setVisibility(View.VISIBLE);  
}else{  
 mTvCount.setVisibility(View.GONE);  
}  
mTvCount.setText(goodsTypeInfo.getCount() + "");

**抛物线动画（view绝对位置）**

1. 克隆一个加号按钮，位置完全相同

ImageButton ib = new ImageButton(mContext);  
**int[] srcLocation = new int[2];  
mIbAdd.getLocationInWindow(srcLocation); //输入参数也是输出参数  
Log.e("donghua", "x:" + srcLocation[0] + ",y:" + srcLocation[1]);**  
ib.setX(srcLocation[0]);  
ib.setY(srcLocation[1]);  
ib.setBackgroundResource(R.drawable.button\_add);

2. 添加到activity上

((BusinessActivity) mGoodsFragment.getActivity()).addImageButton(ib, mIbAdd.getWidth(),mIbAdd.getHeight());

3. 抛物线终点确定

int[] destLocation = ((BusinessActivity) mGoodsFragment.getActivity()).getCartLocation();  
AnimationSet parabolaAnimation = getParabolaAnimation(ib, srcLocation, destLocation);  
ib.startAnimation(parabolaAnimation);

4. 绘制抛物线

private AnimationSet getParabolaAnimation(final ImageButton ib, int[] srcLocation, int[] destLocation) {  
 //先做斜线  
 AnimationSet animationSet = new AnimationSet(false);  
 animationSet.setDuration(SHOW\_DURATION);  
 TranslateAnimation translateX = new TranslateAnimation(  
 Animation.ABSOLUTE, 0,  
 Animation.ABSOLUTE, destLocation[0] - srcLocation[0],  
 Animation.ABSOLUTE, 0,  
 Animation.ABSOLUTE, 0  
 );  
 translateX.setDuration(SHOW\_DURATION);  
 animationSet.addAnimation(translateX);  
  
 TranslateAnimation translateY = new TranslateAnimation(  
 Animation.ABSOLUTE, 0,  
 Animation.ABSOLUTE, 0,  
 Animation.ABSOLUTE, 0,  
 Animation.ABSOLUTE, destLocation[1] - srcLocation[1]  
 );  
 translateY.setInterpolator(new AccelerateInterpolator());  
 translateY.setDuration(SHOW\_DURATION);  
 animationSet.addAnimation(translateY);  
 return animationSet;  
}

4. 结束后回收克隆的按钮

@Override  
public void onAnimationEnd(Animation animation) {  
 //3.投入购物栏以后回收  
 ViewParent viewParent = ib.getParent();  
 if (viewParent instanceof ViewGroup) {  
 ViewGroup viewGroup = (ViewGroup) viewParent;  
 viewGroup.removeView(ib);  
 }  
}

## 每日作业

1.数据外键（商品类别与商品双向绑定）

2.双列表视图的联动

3.点餐抛物线动画实现