推荐列表布局

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

<RelativeLayout 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"

android:padding="10dp">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@drawable/btn\_bg\_white\_1\_r10"

android:gravity="center\_vertical"

android:orientation="horizontal"

android:padding="20dp">

<com.itheima.roundedimageview.RoundedImageView

android:id="@+id/mRoundedImageView"

android:layout\_width="100dp"

android:layout\_height="120dp"

android:scaleType="centerCrop"

android:src="@mipmap/default\_img\_1"

app:riv\_corner\_radius="10dp"

app:riv\_mutate\_background="true"

app:riv\_oval="false"

app:riv\_tile\_mode="clamp" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="12dp"

android:orientation="vertical">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:gravity="center\_vertical">

<TextView

android:id="@+id/tvRecommend"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:background="@drawable/bg\_red\_1\_r10"

android:gravity="center\_vertical"

android:paddingLeft="6dp"

android:paddingTop="2dp"

android:paddingRight="6dp"

android:paddingBottom="2dp"

android:layout\_marginRight="8dp"

android:text="推荐"

android:textColor="#FFFFFF"

android:textSize='12sp' />

<TextView

android:id="@+id/tvShiJianMingCheng"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:gravity="center\_vertical"

android:hint="暂无标签"

android:maxLines="2"

android:textColor="#000000"

android:textSize='16sp' />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="6dp"

android:layout\_marginBottom="12dp"

android:orientation="horizontal">

<TextView

android:id="@+id/tvDes"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:ellipsize="end"

android:maxLines="3"

android:hint="暂无描述内容"

android:textColor="#958F8F"

android:textSize="13sp" />

</LinearLayout>

</LinearLayout>

</LinearLayout>

</RelativeLayout>

推荐列表java代码和网络请求的实现代码

/\*\*

\* @des: 历史事件列表

\* @Author:

\* @time: 2022年08月20日

\*/

public class HistoryEventListActivity extends BaseActivity {

@BindView(R.id.llSearch)

LinearLayout llSearch;

@BindView(R.id.mPullToRefreshLayout)

PullToRefreshLayout mPullToRefreshLayout;

@BindView(R.id.mRecyclerView)

RecyclerView mRecyclerView;

private HistoryEventListActivityAdapter mAdapter;

private HistoryEventListActivityRequest mRequest;

private boolean mCanLoadingMore = true;

private int mPageNo = 1;

private String mId;

private String mSearchStr = "";

public static void open(Context context, String id) {

Intent intent = new Intent(context, HistoryEventListActivity.class);

intent.putExtra("id", id);

context.startActivity(intent);

}

@Override

protected void init(Bundle savedInstanceState) {

mId = getIntent().getStringExtra("id");

mRequest = new HistoryEventListActivityRequest(this);

}

@Override

protected int attachLayout() {

return R.layout.activity\_history\_event\_list;

}

@Override

protected void initView() {

mRecyclerView.setLayoutManager(new LinearLayoutManager(mActivity));

mRecyclerView.setAdapter(mAdapter = new HistoryEventListActivityAdapter(mActivity));

mAdapter.setOnItemClickListener(new OnItemClickListener() {

@Override

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

HistoryEventListActivityBean.DataBean bean = (HistoryEventListActivityBean.DataBean) adapter.getItem(position);

String id = bean.getId();

if (TextUtils.isEmpty(id)) {

ToastUtils.showShort("id错误");

return;

}

HistoryInfoActivity.open(mActivity, id);

}

});

mPullToRefreshLayout.canRefresh(true);

mPullToRefreshLayout.canLoadMore(true);

mPullToRefreshLayout.setOnRefreshListener(new OnRefreshListener() {

@Override

public void onRefresh() {

mCanLoadingMore = true;

mPageNo = 1;

mRequest.getList2(mId, mSearchStr, mPageNo);

}

@Override

public void onLoadMore() {

if (mCanLoadingMore) {

mPageNo++;

mRequest.getList2(mId, mSearchStr, mPageNo);

} else {

mPullToRefreshLayout.finishLoadMore();

ToastUtils.showShort("没有更多数据了");

}

}

});

mPullToRefreshLayout.autoRefresh();

}

@OnClick({R.id.ivBack, R.id.llSearch})

public void onViewClicked(View view) {

MySoftKeyboardUtils.closeSoftKeyboard(mActivity);

if (MyClickUtils.isDoubleClick()) {

return;

}

switch (view.getId()) {

case R.id.ivBack:

finish();

break;

case R.id.llSearch:

SearchActivity.open(mActivity);

break;

}

}

@Override

protected void eventMsg(DataEvent event) {

switch (event.type) {

case HISTORY\_EVENT\_LIST\_SUCC:

mPullToRefreshLayout.finishRefresh();

mPullToRefreshLayout.finishLoadMore();

// List<HistoryTimeListActivityBean> list = new ArrayList<>();

// list.add(new HistoryTimeListActivityBean("元朝", "时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服"));

// list.add(new HistoryTimeListActivityBean("元朝", "时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服"));

// list.add(new HistoryTimeListActivityBean("元朝", "时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服"));

// list.add(new HistoryTimeListActivityBean("元朝", "时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服"));

// list.add(new HistoryTimeListActivityBean("元朝", "时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服"));

// list.add(new HistoryTimeListActivityBean("元朝", "时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服"));

// list.add(new HistoryTimeListActivityBean("元朝", "时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服"));

// list.add(new HistoryTimeListActivityBean("元朝", "时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服"));

// list.add(new HistoryTimeListActivityBean("元朝", "时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服"));

// list.add(new HistoryTimeListActivityBean("元朝", "时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服"));

// list.add(new HistoryTimeListActivityBean("元朝", "时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服"));

// list.add(new HistoryTimeListActivityBean("元朝", "时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服"));

// list.add(new HistoryTimeListActivityBean("元朝", "时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服"));

// list.add(new HistoryTimeListActivityBean("元朝", "时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服"));

// list.add(new HistoryTimeListActivityBean("元朝", "时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服时代俊峰乐山大佛连接速度发历史地看杰弗里斯客服"));

// mAdapter.setNewInstance(list);

HistoryEventListActivityBean bean = (HistoryEventListActivityBean) event.data;

List<HistoryEventListActivityBean.DataBean> data = bean.getData();

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

mPullToRefreshLayout.showView(ViewStatus.CONTENT\_STATUS);

if (mPageNo == 1) {

mAdapter.setNewInstance(data);

} else {

mAdapter.addData(data);

}

if (data.size() >= Constants.PAGE\_SIZE\_VALUE) {

mCanLoadingMore = true;

} else {

mCanLoadingMore = false;

}

} else {

mPullToRefreshLayout.showView(ViewStatus.EMPTY\_STATUS);

ToastUtils.showShort("没有数据了");

}

break;

case HISTORY\_EVENT\_LIST\_ERR:

mPullToRefreshLayout.finishRefresh();

mPullToRefreshLayout.finishLoadMore();

mPullToRefreshLayout.showView(ViewStatus.EMPTY\_STATUS);

ToastUtils.showShort(event.data.toString());

break;

}

}

}

推荐列表recyclerView的数据适配器Adapter代码

/\*\*

\* @des: 历史事件列表 - 数据适配器

\* @Author:

\* @time: 2022年08月20日

\*/

public class HistoryEventListActivityAdapter extends BaseQuickAdapter<HistoryEventListActivityBean.DataBean, BaseViewHolder> {

private Context mContext;

private final String TAG = "HistoryEventListActivityAdapter";

public HistoryEventListActivityAdapter(Context context) {

super(R.layout.activity\_history\_event\_list\_item, null);

this.mContext = context;

}

@Override

protected void convert(@NonNull BaseViewHolder holder, HistoryEventListActivityBean.DataBean item) {

try {

RoundedImageView mRoundedImageView = (RoundedImageView) holder.getView(R.id.mRoundedImageView);

TextView tvShiJianMingCheng = (TextView) holder.getView(R.id.tvShiJianMingCheng);

TextView tvDes = (TextView) holder.getView(R.id.tvDes);

TextView tvRecommend = (TextView) holder.getView(R.id.tvRecommend);

tvRecommend.setVisibility(View.GONE);

String title = item.getTitle();

String contents = item.getContents();

String description = item.getDescription();

String url = item.getImage();

String isRecommend = item.getIsRecommend();

tvShiJianMingCheng.setText(title);

tvDes.setText(description);

if (!TextUtils.isEmpty(url)) {

url = UrlManager.RequestUrl.baseUrl + url;

}

Log.d(TAG, "------>> url = " + url);

RequestOptions options = new RequestOptions()

.error(R.mipmap.default\_img\_1)

.placeholder(R.mipmap.default\_img\_1);

Glide.with(mContext)

.load(url)

.thumbnail(0.5f)

.apply(options)

.into(mRoundedImageView);

if (!TextUtils.isEmpty(isRecommend)) {

if (isRecommend.equals("Y")) {

tvRecommend.setVisibility(View.VISIBLE);

}

}

} catch (Exception e) {

ToastUtils.showShort("数据异常");

}

}

}

文章详情

/\*\*

\* @des: 文章详情

\* @Author:

\* @time: 2022年08月20日

\*/

public class HistoryInfoActivity extends BaseActivity {

// butterknife注解框架（ 替代findViewById(R.id.xxx) ）

@BindView(R.id.ivBack)

ImageView ivBack;

@BindView(R.id.tvTitle)

TextView tvTitle;

@BindView(R.id.mPullToRefreshLayout)

PullToRefreshLayout mPullToRefreshLayout;

@BindView(R.id.tvDes)

TextView tvDes;

private String mId;

// 网络请求

private HistoryInfoActivityRequest mRequest;

/\*\*

\* 当前界面启动函数

\* @param context 上下文

\* @param id 文章id

\*/

public static void open(Context context, String id) {

Intent intent = new Intent(context, HistoryInfoActivity.class);

// 打开新页面的时候传递参数

intent.putExtra("id", id);

context.startActivity(intent);

}

@Override

protected void init(Bundle savedInstanceState) {

// 接收参数

mId = getIntent().getStringExtra("id");

// 实例化网络请求

mRequest = new HistoryInfoActivityRequest(this);

}

@Override

protected int attachLayout() {

// 加载布局

return R.layout.activity\_history\_info;

}

@Override

protected void initView() {// 布局已经加载完成

tvTitle.setText("详情");// 设置当前页面的标题

mRequest.addZuJi(mId);// 网络请求 - 添加足迹，用于数据分析和算法推荐

mPullToRefreshLayout.canRefresh(true);// 能否下拉刷新

mPullToRefreshLayout.canLoadMore(false);// 能否上拉加载更多数据

mPullToRefreshLayout.setOnRefreshListener(new OnRefreshListener() {// 下拉上拉监听器

@Override

public void onRefresh() {// 下拉的回调函数

mRequest.getInfo(mId);// 网络请求 - 加载文章详情数据

}

@Override

public void onLoadMore() {// 上拉的回调函数

}

});

mPullToRefreshLayout.autoRefresh();// 自动刷新函数

}

// butterknife注解框架（ 替代setOnClicklistener（）函数 ）

@OnClick({R.id.ivBack})

public void onViewClicked(View view) {

MySoftKeyboardUtils.closeSoftKeyboard(mActivity);// 关闭软键盘

if (MyClickUtils.isDoubleClick()) {// 防止双击

return;

}

switch (view.getId()) {

case R.id.ivBack:// 返回按钮的点击事件

finish();// 结束当前页面

break;

}

}

@Override

protected void eventMsg(DataEvent event) {// EventBus的数据接收函数

switch (event.type) {// 根据type类型接收对应的事件信息

case INFO\_SUCC:// 加载文章详情成功

mPullToRefreshLayout.finishRefresh();// 结束刷新动画

getInfoSucc(event);// 处理数据

break;

case INFO\_ERR:// 加载文章详情失败

mPullToRefreshLayout.finishRefresh();// 结束刷新动画

mPullToRefreshLayout.showView(ViewStatus.EMPTY\_STATUS);// 显示空布局

ToastUtils.showShort(event.data.toString());// 显示提示信息

break;

case ADD\_ZU\_JI\_SUCC:

break;

case ADD\_ZU\_JI\_ERR:

break;

}

}

private void getInfoSucc(DataEvent event) {

HistoryInfoActivityBean bean = (HistoryInfoActivityBean) event.data;// 将数据转成对应的javaBean

HistoryInfoActivityBean.DataBean data = bean.getData();

if (data == null) {// 判断数据是否为空

ToastUtils.showShort("暂无数据");

mPullToRefreshLayout.showView(ViewStatus.EMPTY\_STATUS);// 显示空布局

return;

}

String contents = data.getContents();// 获取富文本

if (!TextUtils.isEmpty(contents)) {// 判断富文本是否为空

// 富文本内容不为空

mPullToRefreshLayout.showView(ViewStatus.CONTENT\_STATUS);// 显示内容布局

MovementMethod instance = LinkMovementMethod.getInstance();

RichText.initCacheDir(mActivity);// 初始化富文本显示框架

RichText.fromHtml(contents).into(tvDes);// 显示富文本

} else {

// 富文本内容为空

ToastUtils.showShort("暂无数据");// 提示

mPullToRefreshLayout.showView(ViewStatus.EMPTY\_STATUS);// 显示空布局

}

}

}

文章详情布局

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="#5064EB"

android:clipChildren="true"

android:fitsSystemWindows="true"

android:orientation="vertical">

<include layout="@layout/action\_bar\_blue\_2" />

<com.geaosu.refreshx.PullToRefreshLayout

android:id="@+id/mPullToRefreshLayout"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="#FFFFFF">

<ScrollView

android:layout\_width="match\_parent"

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

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical">

<TextView

android:id="@+id/tvDes"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:textSize="16sp" />

</LinearLayout>

</ScrollView>

</com.geaosu.refreshx.PullToRefreshLayout>

</LinearLayout>