**package** com.htl.baubi.ui.screen.community.adapter;  
  
**import** android.app.Dialog;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.res.TypedArray;  
**import** android.graphics.Color;  
**import** android.graphics.drawable.ColorDrawable;  
**import** android.os.Bundle;  
**import** android.support.design.widget.BottomSheetDialog;  
**import** android.support.v4.app.Fragment;  
**import** android.support.v4.app.FragmentActivity;  
**import** android.support.v4.app.FragmentManager;  
**import** android.support.v4.app.FragmentTransaction;  
**import** android.support.v4.view.ViewPager;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.widget.LinearLayout;  
**import** android.widget.TextView;  
  
**import** com.chad.library.adapter.base.BaseQuickAdapter;  
**import** com.chad.library.adapter.base.BaseViewHolder;  
**import** com.htl.baubi.R;  
**import** com.htl.baubi.common.Constants;  
**import** com.htl.baubi.manager.ReportOnClickEvent;  
**import** com.htl.baubi.manager.SaveOnClickEvent;  
**import** com.htl.baubi.model.Data;  
**import** com.htl.baubi.model.diary.Image;  
**import** com.htl.baubi.model.post.DataListPost;  
**import** com.htl.baubi.ui.base.OnItemRvClickListener;  
**import** com.htl.baubi.ui.customview.CircleIndicator;  
**import** com.htl.baubi.ui.customview.ImageReviewFragment;  
**import** com.htl.baubi.ui.customview.ImageReviewPagerAdapter;  
**import** com.htl.baubi.ui.screen.infouser.InfoUserActivity;  
**import** com.htl.baubi.ui.screen.photodiary.fragment.SlideshowDialogFragment;  
**import** com.htl.baubi.ui.screen.post.detailpost.PostDetailActivity;  
**import** com.htl.baubi.utils.FormatUtils;  
**import** com.htl.baubi.utils.ImageUtils;  
**import** com.htl.baubi.utils.sliderimage.PicassoImageLoadingService;  
**import** com.squareup.picasso.Picasso;  
  
**import** org.greenrobot.eventbus.EventBus;  
  
**import** java.io.Serializable;  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
**import** de.hdodenhof.circleimageview.CircleImageView;  
**import** ss.com.bannerslider.Slider;  
**import** ss.com.bannerslider.adapters.SliderAdapter;  
**import** ss.com.bannerslider.event.OnSlideClickListener;  
**import** ss.com.bannerslider.indicators.IndicatorShape;  
**import** ss.com.bannerslider.viewholder.ImageSlideViewHolder;  
  
**public class** QuickCommunityNewsAdapter **extends** BaseQuickAdapter<DataListPost, BaseViewHolder> {  
  
 **private static final int *TYPE\_ITEM*** = 1;  
 **private static final int *TYPE\_HEADER*** = 0;  
 **protected** Context **mContext**;  
 OnItemRvClickListener **onItemRvClickListener**;  
 Fragment **fragment**;  
  
  
 **public** QuickCommunityNewsAdapter(Context context, Fragment fragment, OnItemRvClickListener onItemRvClickListener, **int** layoutResId, List data) {  
 **super**(layoutResId, data);  
 **this**.**mContext** = context;  
 **this**.**fragment** = fragment;  
 **this**.**onItemRvClickListener** = onItemRvClickListener;  
 }  
  
 @Override  
 **protected void** convert(BaseViewHolder holder, DataListPost item) {  
 **if** (item.getType() != **null**) {  
 **switch** (item.getType()) {  
 **case "text"**:  
 holder.setVisible(R.id.***viewPager***, **false**);  
 **break**;  
 **case "background"**:  
 holder.setVisible(R.id.***viewPager***, **false**);  
 holder.setVisible(R.id.***rlBackgroundPost***, **true**);  
 TypedArray typeArray = **mContext**.getResources().obtainTypedArray(  
 R.array.***background\_resource***);  
 **int** count = typeArray.length();  
 **int**[] arrBg = **new int**[count];  
 **for** (**int** i = 0; i < arrBg.**length**; i++) {  
 arrBg[i] = typeArray.getResourceId(i, 0);  
 }  
 **if** (item.getBackground() != **null**) {  
 holder.setImageResource(R.id.***imgBackgroundPost***, arrBg[Integer.*parseInt*(item.getBackground())]);  
 }  
 holder.setText(R.id.***txtPostContentWithBackground***, item.getContent());  
 holder.setVisible(R.id.***txtPostContent***, **false**);  
 **break**;  
 **case "image"**:  
 holder.setVisible(R.id.***viewPager***, **true**);  
*// ImageUtils.loadImageInsideFragmentByGlide(fragment, item.getImages().get(0), imgTest);  
// ImageUtils.loadImageByGlide(mContext, item.getImages().get(0), imgTest);  
// bgaBanner.setAdapter(new BGABanner.Adapter<ImageView, String>() {  
// @Override  
// public void fillBannerItem(BGABanner banner, ImageView itemView, String model, int position) {  
// ImageUtils.loadImageByGlide(mContext, model, itemView);  
// }  
// });  
// bgaBanner.setData(Arrays.asList(item.getImages()), Arrays.asList("提示文字1", "提示文字2", "提示文字3"));  
  
// holder.setImageUrl(R.id.imgTest, item.getImages().get(0));  
// List<Fragment> fragmentList = new ArrayList<>();  
// for (int i = 0; i < item.getImages().size(); i++) {  
// ImageReviewFragment imageReviewFragment = new ImageReviewFragment();  
// Bundle bundle = new Bundle();  
// bundle.putString("imageReviewUrl", item.getImages().get(i));  
// imageReviewFragment.setArguments(bundle);  
// fragmentList.add(imageReviewFragment);  
// }* ImageReviewPagerAdapter imageReviewPagerAdapter = **new** ImageReviewPagerAdapter(**fragment**.getChildFragmentManager(), item.getImages());  
 ViewPager viewPager = holder.getView(R.id.***viewPager***);  
 CircleIndicator indicator = holder.getView(R.id.***indicator***);  
 viewPager.setOffscreenPageLimit(2);  
 viewPager.setAdapter(imageReviewPagerAdapter);  
 indicator.setViewPager(viewPager);  
 imageReviewPagerAdapter.notifyDataSetChanged();  
 **break**;  
 **default**:  
 **break**;  
 }  
 }  
  
 holder.setText(R.id.***txtPostContent***, item.getContent());  
 holder.setText(R.id.***txtCommentDate***, item.getContent());  
 holder.setText(R.id.***txtLikeCount***, String.*valueOf*(item.getLikeNumber()));  
 holder.setText(R.id.***txtPostTime***, FormatUtils.*getDescriptionTimeFromDateString*(item.getCreatedAt()));  
  
 **if** (!item.getAnonymous()) {  
 holder.setText(R.id.***txtUserName***, item.getUser().getName());  
 holder.setText(R.id.***txtUserPost***, item.getUser().getName() + **"đã gửi vào nhóm người cao tuổi"**);  
 holder.getView(R.id.***imgUserAvatar***).setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 ShowPopup(v, item);  
 }  
 });  
 ImageUtils.*loadImageByGlide*(**mContext**, item.getUser().getAvatar(), holder.getView(R.id.***imgUserAvatar***));  
  
 holder.getView(R.id.***imgAvatarTopComment***).setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 ShowPopup(v, item);  
 }  
 });  
  
 } **else** {  
 holder.setText(R.id.***txtUserName***, **"Người dùng ẩn danh"**);  
 holder.setImageDrawable(R.id.***imgUserAvatar***, **mContext**.getResources().getDrawable(R.drawable.***question\_avatar***));  
 }  
  
 holder.getView(R.id.***imgAvatarTopComment***).setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 ShowPopup(view, item);  
 }  
 });  
  
 holder.getView(R.id.***btnViewMore***).setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent intent = **new** Intent(**mContext**, PostDetailActivity.**class**);  
 intent.putExtra(**"id"**, item.getId());  
 **mContext**.startActivity(intent);  
 }  
 });  
  
 holder.getView(R.id.***rlTopComment***).setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Intent intent = **new** Intent(**mContext**, PostDetailActivity.**class**);  
 intent.putExtra(**"id"**, item.getId());  
 **mContext**.startActivity(intent);  
 }  
 });  
  
 holder.setText(R.id.***txtCommentCount***, String.*valueOf*(item.getCommentNumber()));  
 **if** (item.getDefaultComment().getId() == **null**) {  
 holder.setVisible(R.id.***rlTopComment***, **false**);  
 } **else** {  
 holder.setVisible(R.id.***rlTopComment***, **true**);  
 holder.setText(R.id.***txtTopComment***, item.getDefaultComment().getContent());  
 holder.setText(R.id.***txtUserTopComment***, item.getDefaultComment().getUser().getName());  
 holder.getView(R.id.***txtUserTopComment***).setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 ShowPopup(view, item);  
 *// openInfoUser(dataListPosts.get(position - 1));* }  
 });  
 ImageUtils.*loadImageByGlide*(**mContext**, item.getDefaultComment().getUser().getAvatar(), holder.getView(R.id.***imgAvatarTopComment***));  
  
 }  
 ImageUtils.*loadImageByGlide*(**mContext**, Data.*imageMom*, holder.getView(R.id.***imgUserAvatarComment***));  
  
  
 holder.getView(R.id.***btnLike***).setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **onItemRvClickListener**.onItemRvClick(v, item, holder.getAdapterPosition());  
 }  
 });  
 **if** (item.getLiked()) {  
 holder.setImageDrawable(R.id.***btnLike***, **mContext**.getResources().getDrawable(R.drawable.***ic\_action\_liked***));  
  
 } **else** {  
 holder.setImageDrawable(R.id.***btnLike***, **mContext**.getResources().getDrawable(R.drawable.***ic\_action\_like1***));  
 }  
 holder.getView(R.id.***btnComment***).setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **onItemRvClickListener**.onItemRvClick(v, item, holder.getAdapterPosition());  
 }  
 });  
 holder.getView(R.id.***btnSetting***).setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 showDialogMenu(item, holder.getAdapterPosition());  
 }  
 });  
  
 holder.getView(R.id.***edtComment***).setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **onItemRvClickListener**.onItemRvClick(v, item, holder.getAdapterPosition());  
 }  
 });  
  
 holder.getView(R.id.***btnAddImageVideo***).setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **onItemRvClickListener**.onItemRvClick(v, item, holder.getAdapterPosition());  
 }  
 });  
  
 }  
  
 **private void** initSlider(DataListPost item, List<Image> imageHomeDiaries, Slider slider) {  
  
 Slider.*init*(**new** PicassoImageLoadingService(**mContext**));  
 slider.setAdapter(**new** SliderAdapter() {  
 @Override  
 **public int** getItemCount() {  
 **return** item.getImages().size();  
 }  
  
 @Override  
 **public void** onBindImageSlide(**int** position, ImageSlideViewHolder imageSlideViewHolder) {  
 **switch** (position) {  
 **case** 0:  
 imageSlideViewHolder.bindImageSlide(Constants.***API\_ENDPOINT2*** + item.getImages().get(position).toString());  
 **break**;  
 **case** 1:  
 imageSlideViewHolder.bindImageSlide(Constants.***API\_ENDPOINT2*** + item.getImages().get(position).toString());  
 **break**;  
 **case** 2:  
 imageSlideViewHolder.bindImageSlide(Constants.***API\_ENDPOINT2*** + item.getImages().get(position).toString());  
 **break**;  
 **case** 3:  
 imageSlideViewHolder.bindImageSlide(Constants.***API\_ENDPOINT2*** + item.getImages().get(position).toString());  
 **break**;  
 **case** 4:  
 imageSlideViewHolder.bindImageSlide(Constants.***API\_ENDPOINT2*** + item.getImages().get(position).toString());  
 **break**;  
 }  
 }  
 });  
 slider.setSelectedSlide(0);  
 slider.setIndicatorStyle(IndicatorShape.***ROUND\_SQUARE***);  
 slider.setOnSlideClickListener(**new** OnSlideClickListener() {  
 @Override  
 **public void** onSlideClick(**int** position) {  
 openImage(position, imageHomeDiaries);  
 }  
 });  
 slider.setVisibility(View.***VISIBLE***);  
 }  
  
 @Override  
 **public long** getItemId(**int** position) {  
 **return super**.getItemId(position);  
 }  
  
 **private void** openImage(**int** index, List<Image> imageHomeDiaries) {  
 Bundle bundle = **new** Bundle();  
 bundle.putSerializable(**"images"**, (Serializable) imageHomeDiaries);  
 bundle.putInt(**"position"**, index);  
  
 FragmentTransaction ft = ((FragmentActivity) **mContext**).getSupportFragmentManager().beginTransaction();  
 SlideshowDialogFragment newFragment = SlideshowDialogFragment.*newInstance*(**false**);  
 newFragment.setArguments(bundle);  
 newFragment.show(ft, **"slideshow"**);  
 }  
  
 **private void** openInfoUser(DataListPost myQuestion) {  
 Intent intent = **new** Intent(**mContext**, InfoUserActivity.**class**);  
 intent.putExtra(**"id"**, myQuestion.getUser().getId());  
 intent.putExtra(**"name"**, myQuestion.getUser().getName());  
 intent.putExtra(**"image"**, myQuestion.getUser().getAvatar());  
 **mContext**.startActivity(intent);  
 }  
  
 **private** Dialog **myDialog**;  
 **private** BottomSheetDialog **bottomSheetDialog**;  
  
 **private void** ShowPopup(View v, DataListPost myQuestion) {  
 **myDialog** = **new** Dialog(LayoutInflater.*from*(**mContext**).getContext());  
 **myDialog**.setContentView(R.layout.***dialog\_user\_info***);  
  
 TextView txt\_Name = (TextView) **myDialog**.findViewById(R.id.***txt\_Name***);  
 txt\_Name.setText(myQuestion.getUser().getName());  
 CircleImageView logo = **myDialog**.findViewById(R.id.***logo***);  
 **if** (myQuestion.getUser().getAvatar().contains(**"http"**)) {  
 Picasso.*get*().load(myQuestion.getUser().getAvatar()).fit().centerCrop().into(logo);  
 } **else** {  
 Picasso.*get*().load(Constants.***API\_ENDPOINT2*** + myQuestion.getUser().getAvatar()).fit().centerCrop().into(logo);  
 }  
  
 **myDialog**.findViewById(R.id.***btnfollow***).setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 openInfoUser(myQuestion);  
 **myDialog**.dismiss();  
 }  
 });  
 TextView txtclose = (TextView) **myDialog**.findViewById(R.id.***txtclose***);  
 txtclose.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **myDialog**.dismiss();  
 }  
 });  
 **myDialog**.getWindow().setBackgroundDrawable(**new** ColorDrawable(Color.***TRANSPARENT***));  
 **myDialog**.show();  
 }  
  
 **private void** showDialogMenu(DataListPost item, **int** position) {  
 **bottomSheetDialog** = **new** BottomSheetDialog(LayoutInflater.*from*(**mContext**).getContext());  
 LayoutInflater li = (LayoutInflater) **mContext**.getSystemService(Context.***LAYOUT\_INFLATER\_SERVICE***);  
 View dialog = li.inflate(R.layout.***dialog\_show\_report***, **null**);  
 **bottomSheetDialog**.setContentView(dialog);  
  
 LinearLayout ll\_Save = (LinearLayout) dialog.findViewById(R.id.***ll\_Save***);  
 LinearLayout ll\_Report = (LinearLayout) dialog.findViewById(R.id.***ll\_Report***);  
  
 ll\_Save.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 EventBus.*getDefault*().post(**new** SaveOnClickEvent(item, position));  
 **bottomSheetDialog**.dismiss();  
 }  
 });  
 ll\_Report.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 EventBus.*getDefault*().post(**new** ReportOnClickEvent(item, position));  
 **bottomSheetDialog**.dismiss();  
 }  
 });  
 **bottomSheetDialog**.show();  
 }  
}