齐 齐 哈 尔 大 学

程序源代码

题 目 基于安卓的齐大校园APP的设计与开发

学 院 计算机与控制工程学院

专业班级 软件171

学生姓名 贾竞超

指导教师 吴迪

2021 年 6月 18日

MainActivity,java

public class MainActivity extends BaseActivity {

private static final String TAG = MainActivity.class.getSimpleName();

private TextRoundProgressDialog loadingDialog;

private long exitTime;//获取第一次点击返回键的系统时间

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

PageManager.addFragment(new MainFragment(), getSupportFragmentManager(), R.id.main\_container);

EventBusManager.register(this);

}

@Override

public void initData() {

if (LoginController.isLogin()) {

LogHelper.i(TAG, "initData -> LoginSuccessEvent");

EventBusManager.postSticky(new EventBusEvents.LoginSuccessEvent());

}

}

@Override

public int getRootLayout() {

return R.layout.activity\_main;

}

@Override

protected void onDestroy() {

super.onDestroy();

EventBusManager.unregister(this);

}

@Subscribe

public void onEvent(EventBusEvents.LoadingEvent event) {

if (event.isLoading) {

if (loadingDialog == null) {

loadingDialog = new TextRoundProgressDialog();

}

if (!loadingDialog.isAdded()) {

loadingDialog.show(getSupportFragmentManager(), TextRoundProgressDialog.class.getSimpleName());

}

} else {

if (loadingDialog != null) {

loadingDialog.dismissAllowingStateLoss();

}

}

}

@Override

public void onBackPressed() {

if ((System.currentTimeMillis() - exitTime) > 2000) { //系统时间减去exitTime

ToastManager.show(MainActivity.this, "再按一次退出应用");

exitTime = System.currentTimeMillis();

} else {

finish();

System.exit(0);

}

}

}

MainFragment.java

public class MainFragment extends BaseFragment {

private static final String TAG = MainFragment.class.getSimpleName();

private static final String[] titles = {"首页", "发现", "我的"};

private int[] icons = {R.drawable.selector\_tab\_home, R.drawable.selector\_tab\_discover, R.drawable.selector\_tab\_me};

private ViewPager viewPager;

private MainActivity mainActivity;

@Override

public int getRootLayout() {

return R.layout.fragment\_main;

}

@Override

public void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

mainActivity = (MainActivity) getActivity();

mainActivity.hideStatusBarMode(false, getResources().getColor(R.color.theme\_blue));

}

@Override

public void onResume() {

super.onResume();

fetchData(false);

}

@Override

public void onDestroy() {

super.onDestroy();

}

@Override

public void initData() {

super.initData();

}

@Override

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

super.initView(view, savedInstanceState);

viewPager = view.findViewById(R.id.vp\_main);

MainFragmentAdapter adapter = new MainFragmentAdapter(getChildFragmentManager(), titles);

viewPager.setAdapter(adapter);

viewPager.setOffscreenPageLimit(3);

TabLayout tabLayout = view.findViewById(R.id.tl\_main);

tabLayout.addTab(tabLayout.newTab());

tabLayout.addTab(tabLayout.newTab());

tabLayout.setupWithViewPager(viewPager);

//noinspection ConstantConditions

tabLayout.getTabAt(0).setCustomView(getTabView(0));

//noinspection ConstantConditions

tabLayout.getTabAt(1).setCustomView(getTabView(1));

//noinspection ConstantConditions

tabLayout.getTabAt(2).setCustomView(getTabView(2));

tabLayout.addOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {

@Override

public void onTabSelected(TabLayout.Tab tab) {

if (tab.getPosition() == 0) {

mainActivity.hideStatusBarMode(false, getResources().getColor(R.color.theme\_blue));

} else if (tab.getPosition() == 1) {

mainActivity.hideStatusBarMode(false, getResources().getColor(R.color.white));

} else if (tab.getPosition() == 2) {

mainActivity.hideStatusBarMode(false, getResources().getColor(R.color.white));

}

}

@Override

public void onTabUnselected(TabLayout.Tab tab) {

}

@Override

public void onTabReselected(TabLayout.Tab tab) {

}

});

}

private void fetchData(boolean refreshCoachInfo) {

}

public View getTabView(int position) {

View view = LayoutInflater.from(getActivity()).inflate(R.layout.item\_tab, null);

TextView title = view.findViewById(R.id.tv\_tab);

if (title.getVisibility() == View.VISIBLE) {

title.setText(titles[position]);

}

ImageView icon = view.findViewById(R.id.iv\_tab);

icon.setImageResource(icons[position]);

return view;

}

}

MainFragmentAdapter.java

public class MainFragmentAdapter extends FragmentStatePagerAdapter {

private List<Fragment> mFragmentList;

private String[] mTitleArray;

public MainFragmentAdapter(FragmentManager fm, String[] titleArray) {

super(fm, BEHAVIOR\_RESUME\_ONLY\_CURRENT\_FRAGMENT);

mFragmentList = new ArrayList<>();

mFragmentList.add(new HomeFragment());

mFragmentList.add(new DiscoverFragment());

mFragmentList.add(new MineFragment());

mTitleArray = titleArray;

}

@Override

public Fragment getItem(int position) {

return mFragmentList.get(position);

}

@Override

public int getCount() {

return mFragmentList.size();

}

@Nullable

@Override

public CharSequence getPageTitle(int position) {

return mTitleArray[position];

}

}

SplashActivity.java

public class SplashActivity extends BaseActivity {

private static final String[] PERMISSIONS = {

Manifest.permission.WRITE\_EXTERNAL\_STORAGE,

Manifest.permission.READ\_EXTERNAL\_STORAGE,

Manifest.permission.READ\_PHONE\_STATE };

private PermissionHelper.PermissionsListener permissionsListener;

@Override

public int getRootLayout() {

return R.layout.activity\_splash;

}

@Override

protected void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

hideStatusBarMode(true, getResources().getColor(R.color.theme\_blue));

if (DeviceManager.getSDKint() < Build.VERSION\_CODES.LOLLIPOP) {

return;

}

if (PermissionHelper.isGranted(this, PERMISSIONS)) {

doPermissionCheckSuccess();

} else {

ActivityCompat.requestPermissions(this, PERMISSIONS, 0);

permissionsListener = new PermissionHelper.PermissionsListener() {

@Override

public void onGranted() {

doPermissionCheckSuccess();

}

@Override

public void onDenied() {

if (PermissionHelper.isGranted(SplashActivity.this, PERMISSIONS)) {

doPermissionCheckSuccess();

} else {

String msg = "检测到您未授权相关权限，部分功能将无法使用";

PermissionHelper.showPermissionsDeniedAlert(SplashActivity.this, msg);

}

}

};

}

}

private void doPermissionCheckSuccess() {

Timer timer = new Timer();

timer.schedule(new TimerTask() {

@Override

public void run() {

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

startActivity(intent);

finish();

}

}, 1000);

}

@Override

public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {

super.onRequestPermissionsResult(requestCode, permissions, grantResults);

if (permissionsListener == null) {

return;

}

if (PermissionHelper.isGranted(grantResults)) {

permissionsListener.onGranted();

} else {

permissionsListener.onDenied();

}

}

@Override

public void onBackPressed() {

// Empty

}

}

RequestManager.java

public class RequestManager {

private static volatile RequestManager mInstance;

private OkHttpClient.Builder mBuilder;

private List<Interceptor> mInterceptors;

public static RequestManager getInstance() {

if (mInstance == null) {

synchronized (RequestManager.class) {

if (mInstance == null) {

mInstance = new RequestManager();

}

}

}

return mInstance;

}

public RequestManager() {

mInterceptors = new ArrayList<>();

mBuilder = new OkHttpClient.Builder()

.connectTimeout(5, TimeUnit.SECONDS)

.readTimeout(5, TimeUnit.SECONDS)

.writeTimeout(5, TimeUnit.SECONDS);

}

public <T> T createService(@NonNull String baseUrl, @NonNull Class<T> service) {

if (!ObjectHelper.isIllegal(mInterceptors)) {

for (Interceptor interceptor : mInterceptors) {

mBuilder.addInterceptor(interceptor);

}

}

Retrofit retrofit = new Retrofit.Builder()

.client(mBuilder.build())

.baseUrl(baseUrl)

.addConverterFactory(GsonConverterFactory.create())

.build();

return retrofit.create(service);

}

public RequestManager addInterceptors(@NonNull Interceptor interceptor) {

mInterceptors.add(interceptor);

return this;

}

}

BizRequest.java

public class BizRequest extends BaseRequest {

private static final String TAG = BizRequest.class.getSimpleName();

private static volatile BizRequest mInstance;

private BizHttpService mBizHttpService;

private BasicParamsInterceptor mParamsInterceptor;

public static BizRequest getInstance() {

if (mInstance == null) {

synchronized (BizRequest.class) {

if (mInstance == null) {

mInstance = new BizRequest();

}

}

}

return mInstance;

}

public BizRequest() {

mParamsInterceptor = new BasicParamsInterceptor.Builder()

.signPrivateParams(BizCommonFieldMap.buildCommonFieldMap())

// .addQueryParam(BizHttpConstants.TOKEN, LoginController.getToken())

.build();

String baseUrl = BizHttpConstants.BASE\_URL;

mBizHttpService = RequestManager.getInstance()

.addInterceptors(mParamsInterceptor)

.createService(baseUrl, BizHttpService.class);

}

public void refresh() {

// mParamsInterceptor.updateQueryParam(BizHttpConstants.TOKEN, LoginController.getToken());

}

@Override

public void bizIntercept(Throwable t) {

super.bizIntercept(t);

if (AppManager.isDebug()) {

ToastManager.showLengthLong(BizApplication.getInstance(), t.getMessage());

} else {

ToastManager.show(BizApplication.getInstance(), BizApplication.getInstance().getResources().getString(R.string.network\_fail\_toast\_text));

}

}

public void login(Map<String, String> params, RequestListener<BaseModel<UserModel>> listener) {

mBizHttpService.login(params).enqueue(callback(listener));

}

public void sendCode(Map<String, String> params, RequestListener<BaseModel> listener) {

mBizHttpService.sendCode(params).enqueue(callback(listener));

}

public void verifyCode(Map<String, String> params, RequestListener<BaseModel> listener) {

mBizHttpService.verifyCode(params).enqueue(callback(listener));

}

public void getNewsList(String type, String key, RequestListener<ListNewsVO> listener) {

mBizHttpService.getNewsList(type, key).enqueue(callback(listener));

}

}

NewsFragment.java

public class NewsFragment extends BaseFragment {

private static final String TAG = NewsFragment.class.getSimpleName();

private RecyclerView rvNews;

private NewsItemListAdapter newsItemListAdapter;

private ListNewsVO listNewsVO;

private ListNewsResVO resVO;

private Banner banner;

// 轮播图列表

private List<Integer> imgList = Arrays.asList(R.drawable.banner\_img1, R.drawable.banner\_img2, R.drawable.banner\_img3);

private List<String> titleList = Arrays.asList("2021牛年大吉", "众志成城，战胜疫情", "齐齐哈尔大学冬日摄影");

private List<String> bmobImageUrlList = new ArrayList<>();

private List<String> bmobImageTitleList = new ArrayList<>();

private RefreshLayout news\_refresh;

@Override

public int getRootLayout() {

return R.layout.fragment\_news;

}

@Override

public void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

EventBusManager.register(this);

}

@Override

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

banner = view.findViewById(R.id.banner);

// 设置图片加载器

banner.setImageLoader(new BannerLoader());

// 显示图片

banner.setBannerStyle(BannerConfig.CIRCLE\_INDICATOR\_TITLE\_INSIDE);//设置页码与标题

initBanner();

rvNews = view.findViewById(R.id.rv\_news);

rvNews.setLayoutManager(new LinearLayoutManager(view.getContext()));

resVO = new ListNewsResVO();

resVO.type = "top";

// 今日头条api的key

resVO.key = getResources().getString(R.string.my\_toutiao\_key);

getNewsList(resVO);

news\_refresh = view.findViewById(R.id.news\_refresh);

// 是否在刷新的时候禁止列表的操作

news\_refresh.setDisableContentWhenRefresh(false);

}

public void initBanner() {

BmobQuery<BannerModel> query = new BmobQuery<>();

query.addWhereEqualTo("isShow", Boolean.TRUE);

query.order("ImageIndex");

query.findObjects(new FindListener<BannerModel>() {

@Override

public void done(List<BannerModel> list, BmobException e) {

if (e == null) {

for (BannerModel model : list) {

if (!ObjectHelper.isIllegal(model)) {

bmobImageUrlList.add(model.getImageUrl());

bmobImageTitleList.add(model.getImageTitle());

}

}

banner.setImages(bmobImageUrlList);

banner.setBannerTitles(bmobImageTitleList);

banner.setDelayTime(3000);

banner.setIndicatorGravity(BannerConfig.CENTER);

banner.start();

ToastManager.show(getContext(), "图片加载成功");

} else {

LogHelper.i(TAG, e.getMessage());

banner.setImages(imgList);

banner.setBannerTitles(titleList);

banner.setDelayTime(3000);

banner.setIndicatorGravity(BannerConfig.CENTER);

banner.start();

ToastManager.show(getContext(), "图片加载失败");

}

}

});

}

@Override

public void onDestroy() {

super.onDestroy();

EventBusManager.unregister(this);

}

protected void initListener() {

if (newsItemListAdapter != null) {

newsItemListAdapter.setListener((view, position) -> {

String detailsUrl = listNewsVO.getResult().getData().get(position).getUrl();

NewsDetailsWebViewActivity.start(getContext(), detailsUrl);

});

}

// 下拉刷新

news\_refresh.setOnRefreshListener(new OnRefreshListener() {

@Override

public void onRefresh(RefreshLayout refreshlayout) {

listNewsVO = null;

getNewsList(resVO);

}

});

}

/\*\*

\* 访问头条接口获取新闻列表

\* @param resVO

\*/

private void getNewsList(ListNewsResVO resVO) {

BizRequest.getInstance().getNewsList(resVO.type, resVO.key, new RequestListener<ListNewsVO>() {

@Override

public void onResponse(ListNewsVO listNewsVO) {

if (ObjectHelper.isIllegal(listNewsVO)) {

ToastManager.show(getContext(), "获取失败，请检查网络或服务器出错");

news\_refresh.finishRefresh(false);

return;

}

news\_refresh.finishRefresh(true);

LogHelper.i(TAG, "getNewsList Success");

EventBusEvents.GetNewsListSuccessEvent event = new EventBusEvents.GetNewsListSuccessEvent();

event.listNewsVO = listNewsVO;

EventBusManager.post(event);

}

@Override

public void onFailure(Throwable t) {

LogHelper.i(TAG, "getNewsList Failed");

t.printStackTrace();

}

});

}

@Subscribe

public void onEvent(EventBusEvents.GetNewsListSuccessEvent event) {

listNewsVO = event.listNewsVO;

newsItemListAdapter = new NewsItemListAdapter(getContext(), listNewsVO);

initListener();

rvNews.setAdapter(newsItemListAdapter);

}

}

}

NewsDetailsWebViewActivity.java

public class NewsDetailsWebViewActivity extends BaseWebViewActivity {

private static final String KEY\_URL = "url";

@Override

public int getRootLayout() {

return R.layout.activity\_webview;

}

@Override

public String getUrl() {

return getIntent().getStringExtra(KEY\_URL);

}

public static void start(Context context, String url) {

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

intent.putExtra(KEY\_URL, url);

context.startActivity(intent);

}

@Override

public void initWebSetting() {

mWebView.setWebViewClient(new WebViewClient() {

@Override

public void onPageStarted(WebView view, String url, Bitmap favicon) {

super.onPageStarted(view, url, favicon);

showLoading();

}

@Override

public void onPageFinished(WebView view, String url) {

super.onPageFinished(view, url);

hideLoading();

}

@Override

public void onReceivedError(WebView view, int errorCode, String description, String failingUrl) {

super.onReceivedError(view, errorCode, description, failingUrl);

// 断网或者网络连接超时

if (errorCode == ERROR\_HOST\_LOOKUP || errorCode == ERROR\_CONNECT || errorCode == ERROR\_TIMEOUT) {

view.loadUrl("about:blank"); // 避免出现默认的错误界面

showNetFail();

}

}

});

}

@Override

public void initOtherView() {

super.initOtherView();

initTitleBar();

titleBarView.setCenterText("新闻快讯");

}

}

BaseActivity.java

public abstract class BaseActivity extends AppCompatActivity {

public TitleBarView titleBarView;

public static final String KEY\_DATA = "data";

@Override

protected void onCreate(@Nullable Bundle savedInstanceState) {

setScreenOrientation();

super.onCreate(savedInstanceState);

requestWindowFeature(Window.FEATURE\_NO\_TITLE);

setContentView(getRootLayout());

hideStatusBarMode(true, Color.BLACK);

initView(savedInstanceState);

initData();

getWindow().addFlags(WindowManager.LayoutParams.FLAG\_KEEP\_SCREEN\_ON);

}

public void setScreenOrientation() {

}

@Override

protected void onResume() {

super.onResume();

}

@Override

protected void onPause() {

super.onPause();

}

public void initView(@Nullable Bundle savedInstanceState) {

}

public void initData() {

}

public abstract int getRootLayout();

@SuppressLint("MissingSuperCall")

@Override

protected void onSaveInstanceState(Bundle outState) {

// 用于防止崩溃时，Fragment的重叠问题

// super.onSaveInstanceState(outState);

}

public void initTitleBar() {

titleBarView = findViewById(R.id.title\_bar);

}

public static void start(Context context, Class<?> cls) {

Intent intent = new Intent(context, cls);

context.startActivity(intent);

}

public void showContent() {

}

public int getScreenOrientation() {

return getResources().getConfiguration().orientation;

}

public boolean isLandscape() {

return getScreenOrientation() == ActivityInfo.SCREEN\_ORIENTATION\_LANDSCAPE;

}

public void release() {

}

public void pause() {

}

public void resume() {

}

public ViewGroup getRootView() {

return getWindow().getDecorView().findViewById(android.R.id.content);

}

/\*\*

\* 修改状态栏模式，支持6.0以上版本

\*

\* @param isTextWhite 文字、图标是否为黑色 （true为默认的白色）

\* @param color 状态栏颜色

\*/

public void hideStatusBarMode(boolean isTextWhite, int color) {

//Android6.0（API 23）以上使用系统方法，6.0以下使用默认方法

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.M) {

//默认显示导航栏

getWindow().getDecorView().setSystemUiVisibility(View.SYSTEM\_UI\_FLAG\_VISIBLE);

//取消状态栏透明和变为全屏模式模式。

getWindow().clearFlags(WindowManager.LayoutParams.FLAG\_TRANSLUCENT\_STATUS);

//表明会Window负责系统bar的background 绘制，绘制透明背景的系统bar,设置setStatusBarColor需要配合这个参数

getWindow().addFlags(WindowManager.LayoutParams.FLAG\_DRAWS\_SYSTEM\_BAR\_BACKGROUNDS);

getWindow().setStatusBarColor(color);

if (!isTextWhite) {

//设置字体的颜色为亮色模式

getWindow().getDecorView().setSystemUiVisibility(View.SYSTEM\_UI\_FLAG\_LIGHT\_STATUS\_BAR);

}

} else {

getWindow().setFlags(WindowManager.LayoutParams.FLAG\_FULLSCREEN,

WindowManager.LayoutParams.FLAG\_FULLSCREEN);

}

}

public void hideStatusBarMode() {

getWindow().setFlags(WindowManager.LayoutParams.FLAG\_FULLSCREEN,

WindowManager.LayoutParams.FLAG\_FULLSCREEN);

}

@Override

protected void onDestroy() {

super.onDestroy();

getWindow().clearFlags(WindowManager.LayoutParams.FLAG\_KEEP\_SCREEN\_ON);

}

}