导向组件以及欢迎界面

**导向组件**

//导向  
compile 'com.stephentuso:welcome:0.7.2'

**欢迎界面**

联网时,读取服务器图片。不联网时，读取默认图片。

##### 导向组件

1. 引入

compile 'com.stephentuso:welcome:0.7.2'

1. 继承组件类 WelcomeActivity

public class MyWelcomeActivity extends WelcomeActivity {  
  
 @Override  
 protected WelcomeScreenConfiguration configuration() {  
 return new WelcomeScreenBuilder(this)  
 .theme(R.style.*CustomWelcomeScreenTheme*)  
 .defaultTitleTypefacePath("Montserrat-Bold.ttf")  
 .defaultHeaderTypefacePath("Montserrat-Bold.ttf")  
 .titlePage(R.drawable.*photo*, "Welcome", R.color.*orange\_background*)  
 .basicPage(R.drawable.*photo*, "Simple to use", "Add a welcome screen to your app with only a few lines of code.", R.color.*red\_background*)  
 .parallaxPage(R.layout.*parallax\_example*, "Easy parallax", "Supply a layout and parallax effects will automatically be applied", R.color.*purple\_background*, 0.2f, 2f)  
 .basicPage(R.drawable.*photo*, "Customizable", "All elements of the welcome screen can be customized easily.", R.color.*blue\_background*)  
 .swipeToDismiss(true)  
 .exitAnimation(android.R.anim.*fade\_out*)  
 .build();  
 }  
 public static String welcomeKey() {  
 return "开始体验,旧时光";  
 }  
}

<activity android:name=".MyWelcomeActivity"

android:theme="@style/WelcomeScreenTheme"/>

1. 在导向界面结束之后应该显示的界面类之中配置

@Override  
protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_blank*);  
 welcomeScreen = new WelcomeScreenHelper(this, MyWelcomeActivity.class);  
 //是否显示了welcomeScreen  
 Boolean isWelcome = welcomeScreen.show(savedInstanceState);

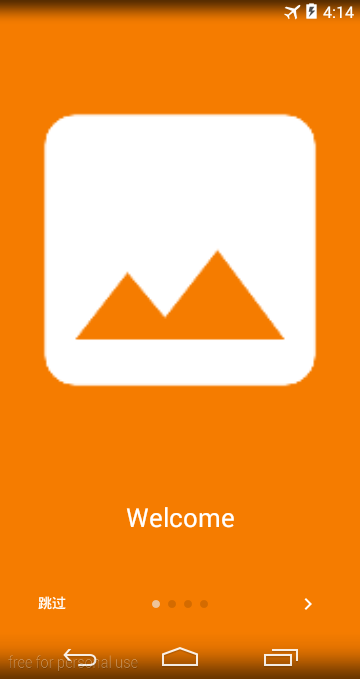
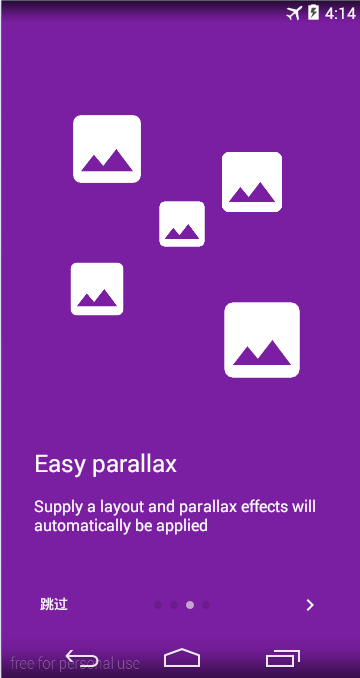
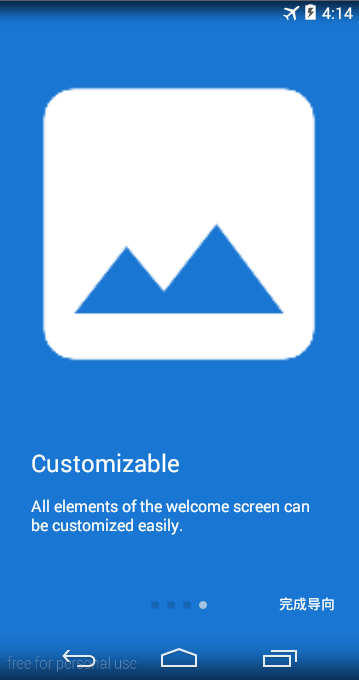
……….todo

*/\*\*  
 \* 在这里接收导向结束的“事件”  
 \** ***@param*** *requestCode  
 \** ***@param*** *resultCode  
 \** ***@param*** *data  
 \*/*

@Override  
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
  
 if (requestCode == WelcomeScreenHelper.*DEFAULT\_WELCOME\_SCREEN\_REQUEST*) {  
 String welcomeKey = data.getStringExtra(WelcomeActivity.*WELCOME\_SCREEN\_KEY*);  
  
 if (resultCode == *RESULT\_OK*) {  
 Toast.*makeText*(getApplicationContext(), welcomeKey , Toast.*LENGTH\_SHORT*).show();  
 } else {  
 Toast.*makeText*(getApplicationContext(), welcomeKey , Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
}  
@Override  
protected void onSaveInstanceState(Bundle outState) {  
 super.onSaveInstanceState(outState);  
 welcomeScreen.onSaveInstanceState(outState);  
}

1. 引入必要的UI资源
2. demo效果

动图不会显示，静态图凑活看



1. 需要注意的地方
2. 步骤2里面提到的方法中 return值决定的下次是否显示。如果想重复显示，请保证返回值的唯一性。

public static String welcomeKey() {  
 return "开始体验,旧时光";  
}

1. 在混淆的时候

-keepclassmembers class \* extends com.stephentuso.welcome.ui.WelcomeActivity {

public static java.lang.String welcomeKey();

}

1. 改变导向的一些样式或者字段信息

在style 里面覆盖WelcomeScreenTheme的一些属性

<style name="CustomWelcomeScreenTheme" parent="WelcomeScreenTheme">  
 <item name="welcomeIndicatorStyle">@style/WelcomeScreenPageIndicator.Custom</item>  
 <item name="welcomeTitleTextStyle">@style/WelcomeScreenText.Title.Centered</item>  
 <item name="welcomeButtonDoneText">@string/idone</item>  
 <item name="welcomeButtonSkipText">@string/iskip</item>  
</style>

1. 设置导向界面内容

图片+标题的形式：

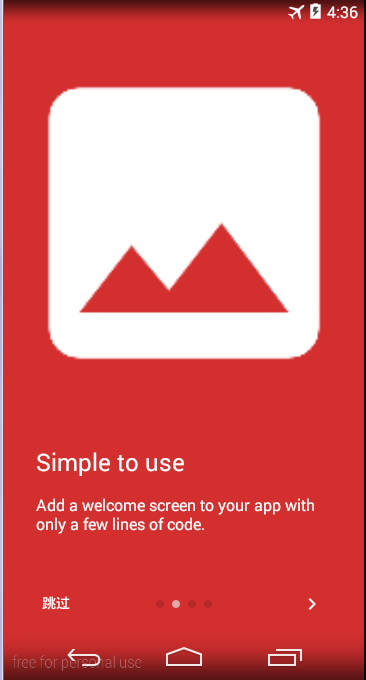
titlePage(int resId, String title)

titlePage(int resId, String title, int colorResId)

titlePage(int resId, String title, int colorResId, boolean showParallaxAnim)

titlePage(int resId, String title, int colorResId, boolean showParallaxAnim, String titleTypefacePath) 

单个图片+标题+解释说明



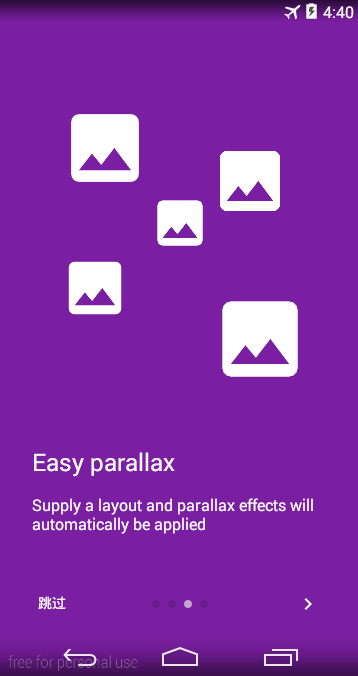
basicPage(int resId, String title, String description)

basicPage(int resId, String title, String description, int colorResId)

basicPage(int drawableId, String title, String description, int colorResId, boolean showParallaxAnim)

basicPage(int drawableId, String title, String description, int colorResId, boolean showParallaxAnim,

String headerTypefacePath, String descriptionTypefacePath)



parallaxPage(int resId, String title, String description)

parallaxPage(int resId, String title, String description, int colorResId)

parallaxPage(int resId, String title, String description, int colorResId, float startParallaxFactor,

float endParallaxFactor)

parallaxPage(int resId, String title, String description, int colorResId, float startParallaxFactor,

float endParallaxFactor, String headerTypefacePath, String descriptionTypefacePath)

无说明

fullScreenParallaxPage(int resId)

fullScreenParallaxPage(int resId, int colorResId)

fullScreenParallaxPage(int resId, int colorResId, float startParallaxFactor, float endParallaxFactor)

添加自己的Fragment作为页面

@Override

protected WelcomeScreenConfiguration configuration() {

return new WelcomeScreenBuilder(this)

...

.page(new WelcomeFragmentHolder() {

@Override

protected Fragment fragment() {

return new YourFragmentHere();

}

}, R.color.background-color)

...

}

如果需要添加动画，则需要自己的Fragment实现[WelcomeScreenPage.OnChangeListener](http://stephentuso.github.io/welcome-android/javadoc/com/stephentuso/welcome/ui/WelcomeScreenPage.OnChangeListener.html)方法

比如：

@Override

public void onScrolled(int pageIndex, float offset, int offsetPixels) {

if (Build.VERSION.SDK\_INT >= 11 && imageView != null) {

imageView.setAlpha(1-Math.abs(offset));

}

}

又或者：

@Override

public void onScrolled(int pageIndex, float offset, int offsetPixels) {

if (parallaxLayout != null)

WelcomeUtils.applyParallaxEffect(parallaxLayout, false, offsetPixels, 0.3f, 0.2f);

}

##### 欢迎功能

**方案**

服务器提供路径，客户端根据路径读取图片并展示。

**体验点**

图片读取速度 尽量保证服务器端图片的size够小

无网络环境 显示默认

读取失败时 显示默认

方法体：

*/\*\*  
 \* 装载图片  
 \* 无网络环境 显示默认  
 \* 读取失败时 显示默认  
 \*/*private void init() {  
 if(!NetUtils.*isNetworkConnected*(BaseApplication.*getInstance*())){  
 welcomeImage.setImageResource(R.drawable.*wall\_picture*);  
 animateImage();  
 return;  
 }  
 final RequestListener listener = new RequestListener<String, GlideDrawable>() {  
 @Override  
 public boolean onException(Exception e, String model, Target<GlideDrawable> target, boolean isFirstResource) {  
 //image.setImageResource(R.drawable.wall\_picture);  
 animateImage();  
 return false;  
 }  
 @Override  
 public boolean onResourceReady(GlideDrawable resource, String model, Target<GlideDrawable> target, boolean isFromMemoryCache, boolean isFirstResource) {  
 animateImage();  
 return false;  
 }  
 };  
 RetrofitSingleton.*apiService* .getWelcomePic()  
 .subscribeOn(Schedulers.*io*())  
 .observeOn(AndroidSchedulers.*mainThread*())  
 .map(new Func1<BaseResponse<WelcomePic>, String[]>() {  
 @Override  
 public String[] call(BaseResponse<WelcomePic> welcomePicBaseResponse) {  
 return new String[]{welcomePicBaseResponse.model.info==null?"":welcomePicBaseResponse.model.info,FrHereApi.*API\_URL* + welcomePicBaseResponse.model.picurl};  
 }  
 })  
 .subscribe(new Observer<String[]>() {  
 @Override  
 public void onCompleted() {  
 //Log.e("666","onCompleted");  
 }  
 @Override  
 public void onError(Throwable e) {  
 PLog.*e*("welcome", "onError");  
 welcomeImage.setImageResource(R.drawable.*wall\_picture*);  
 animateImage();  
 }  
 @Override  
 public void onNext(String[] url) {  
 info.setText(url[0]);  
 Glide.*with*(BaseApplication.*getInstance*())  
 .load(url[1])  
 .fitCenter()  
 .placeholder(R.drawable.*wall\_picture*)  
 .diskCacheStrategy(DiskCacheStrategy.*ALL*)  
 .error(R.drawable.*wall\_picture*)  
 .listener(listener)  
 .into(welcomeImage);  
 }  
 });  
}

动画方法

private void animateImage() {  
 AnimatorSet set = new AnimatorSet();  
 set.playTogether(ObjectAnimator.*ofFloat*(welcomeImage, "scaleX", 1.5f, 1f)  
 , ObjectAnimator.*ofFloat*(welcomeImage, "scaleY", 1.5f, 1f)  
 , ObjectAnimator.*ofFloat*(welcomeImage, "alpha", 0.5f, 1f, 1));  
  
 //image.startAnimation(scaleAnimation);  
 set.setDuration(2000).start();  
 //缩放动画监听  
 set.addListener(new Animator.AnimatorListener() {  
 @Override  
 public void onAnimationStart(Animator animation) {  
 }  
 @Override  
 public void onAnimationEnd(Animator animation) {  
 startWithPop( LoginFragment.*newInstance*());;  
 }  
 @Override  
 public void onAnimationCancel(Animator animation) {  
 }  
 @Override  
 public void onAnimationRepeat(Animator animation) {  
 }  
 });  
}

效果：

无网络

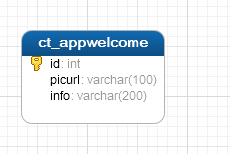


联网后

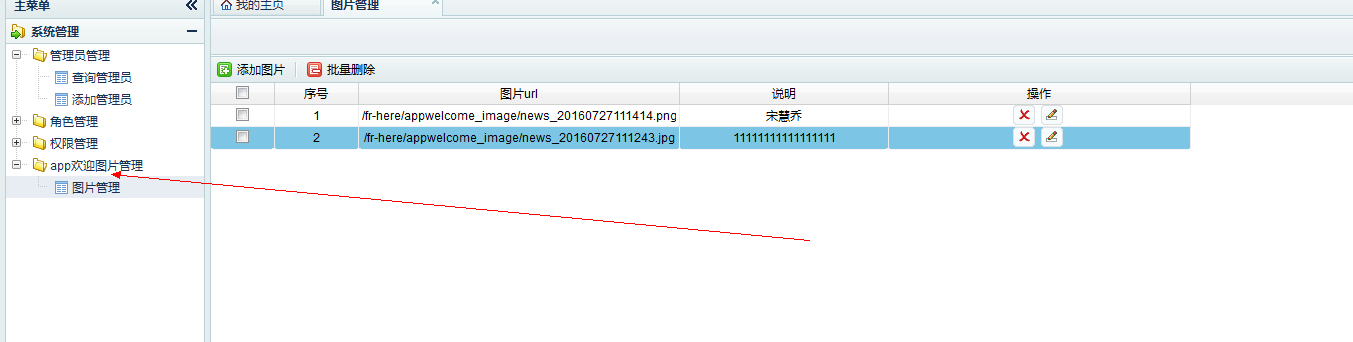


##### *欢迎界面服务端实现参考*

创建表



提供后台录入



控制类

com.fr.here.controller.AppWelcomeController

提供给app访问的接口

<http://localhost/fr-here/rest/app/welcome/newpic>

返回数据

{"result":{"requltType":null,"errorMessage":null},"list":null,"model":{"id":2,"picurl":"/fr-here/appwelcome\_image/news\_20160727111414.png","info":"宋慧乔"}}