<activity  
 android:name=".activity.SSSecurityActivity"  
 android:theme="@style/ss\_gyFloatTheme"  
 android:windowSoftInputMode="stateHidden" >  
 <intent-filter>  
 <!--协议部分，随便设置-->  
 <data  
 android:host="host"  
 android:path="/security"  
 android:port="8888"  
 android:scheme="intergrowthsettings" />  
 <!--下面这几行也必须得设置-->  
 <category android:name="android.intent.category.DEFAULT" />  
 <action android:name="android.intent.action.VIEW" />  
 </intent-filter>  
</activity>

package com.gy.code.router;

*/\*\*  
 \* URI协议接口  
 \*/*public interface IRouterUri {

@RouterUri(routerUri = "intergrowthsettings://host:8888/security")  
String jumpToIntergrowthsettingsSecurity(@RouterParam("urlKey") String urlKey,@RouterParam("forget") String forget);

}

Intent intent = getIntent();  
String action = intent.getAction();  
LogUtil.*e*("action==" + action);  
if (Intent.*ACTION\_VIEW*.equals(action)) {  
 Uri uri = intent.getData();  
 LogUtil.*e*("uri==" + uri);  
 if (uri != null) {  
 String urlKey = uri.getQueryParameter("urlKey");  
 LogUtil.*e*("urlKey==" + urlKey);  
 String forget = uri.getQueryParameter("forget");  
 LogUtil.*e*("forget==" + forget);  
 }  
}

打印：

IReqApi---reqUrl->intergrowthsettings://host:8888/security

reqParam---reqParam->urlKey=bFNnSDNaemNYOTNXTTMzMw==

eqParam---reqParam->forget=forget

action==android.intent.action.VIEW

uri==intergrowthsettings://host:8888/security?urlKey=bFNnSDNaemNYOTNXTTMzMw==&forget=forget

urlKey==bFNnSDNaemNYOTNXTTMzMw==

forget==forget

*/\*\*  
 \* URI注解  
 \*/*@Documented  
@Target(*METHOD*)  
@Retention(*RUNTIME*)  
public @interface RouterUri {  
 String routerUri() default "";  
}

*/\*\*  
 \* Router参数  
 \*/*@Documented  
@Target(*PARAMETER*)  
@Retention(*RUNTIME*)  
public @interface RouterParam {  
 String value() default "";  
}

package com.gy.code.router;  
  
import android.content.Context;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.content.pm.ResolveInfo;  
import android.net.Uri;  
import android.util.Log;  
  
import com.gy.code.BaseApplication;  
  
import java.lang.annotation.Annotation;  
import java.lang.reflect.InvocationHandler;  
import java.lang.reflect.Method;  
import java.lang.reflect.Proxy;  
import java.util.List;  
  
*/\*\*  
 \* 路由操作  
 \*/*public class Router {  
  
 private final static String *TAG* = Router.class.getSimpleName();  
 private static Router *mInstance*;  
 private IRouterUri mRouterUri;  
 private Context mContext;  
  
 */\*\*  
 \* 获取单例引用  
 \*/* public static Router getInstance(Context context) {  
 Router inst = *mInstance*;  
 if (inst == null) {  
 synchronized (Router.class) {  
 inst = *mInstance*;  
 if (inst == null) {  
 inst = new Router(context.getApplicationContext());  
 *mInstance* = inst;  
 }  
 }  
 }  
 return inst;  
 }  
  
 */\*\*  
 \* 构造函数  
 \*  
 \** ***@param*** *mContext application  
 \*/* private Router(Context mContext) {  
 this.mContext = mContext;  
 mRouterUri = create(IRouterUri.class);  
 }  
  
 */\*\*  
 \* 返回Api  
 \*/* public IRouterUri routerUri() {  
 return mRouterUri;  
 }  
  
 public IRouterUri create(Class<?> aClass) {  
 return (IRouterUri) Proxy.*newProxyInstance*(aClass.getClassLoader(), new Class<?>[]{aClass},  
 new InvocationHandler() {  
  
 @Override  
 public Object invoke(Object proxy, Method method, Object... args) throws Throwable {  
 StringBuilder stringBuilder = new StringBuilder();  
 RouterUri reqUrl = method.getAnnotation(RouterUri.class);  
 Log.*e*(*TAG*, "IReqApi---reqUrl->" + reqUrl.routerUri());  
 stringBuilder.append(reqUrl.routerUri());  
 //Type[] parameterTypes = method.getGenericParameterTypes();//获取注解参数类型  
 Annotation[][] parameterAnnotationsArray = method.getParameterAnnotations();//拿到参数注解  
 //Annotation[] annotation = method.getDeclaredAnnotations();  
 int pos = 0;  
 for (int i = 0; i < parameterAnnotationsArray.length; i++) {  
 Annotation[] annotations = parameterAnnotationsArray[i];  
 if (annotations != null && annotations.length != 0) {  
 if (pos == 0) {  
 stringBuilder.append("?");  
 } else {  
 stringBuilder.append("&");  
 }  
 pos++;  
 RouterParam reqParam = (RouterParam) annotations[0];  
 stringBuilder.append(reqParam.value());  
 stringBuilder.append("=");  
 stringBuilder.append(args[i]);  
 Log.*e*(*TAG*, "reqParam---reqParam->" + reqParam.value() + "=" + args[i]);  
 if ("urlKey".equals(reqParam.value())) {  
 if (!args[i].equals(BaseApplication.*urlKey*)) {  
 return null;  
 }  
 }  
 }  
 }  
 //下面就可以执行相应的跳转操作  
 openRouterUri(stringBuilder.toString());  
 return null;  
 }  
 });  
 }  
  
 */\*\*  
 \* 通过uri跳转指定页面  
 \*/* private void openRouterUri(String url) {  
 PackageManager packageManager = mContext.getPackageManager();  
 Intent intent = new Intent(Intent.*ACTION\_VIEW*, Uri.*parse*(url));  
 intent.setFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 List<ResolveInfo> activities = packageManager.queryIntentActivities(intent, 0);  
 boolean isValid = !activities.isEmpty();  
 if (isValid) {  
 mContext.startActivity(intent);  
 }  
 }  
  
}