无感验证SDK-Android ReactNative接入

Android下原生程序编写

Android Studio添加aar依赖

- SDK中libs和assets目录下的文件，需配置到项目对应的位置
- 在该模块的build.gradle中如下配置:

```
defaultConfig {
  xxxx

}

repositories{
    flatDir{
        dirs 'libs'
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation (name:'dx-risk-x.x.x', ext:'aar')
    implementation (name:'dx-captcha-x.x.x', ext:'aar')
}
```

- 在build.gradle的android块内加入如下配置:

```
packagingOptions{
        doNotStrip "*/armeabi/*.so"
        doNotStrip "*/armeabi-v7a/*.so"
        doNotStrip "*/x86/*.so"
        doNotStrip "*/arm64-v8a/*.so"
    }
```

原生模块调用dx-captcha.aar

- 声明一个继承`SimpleViewManager.java`的类`ReactWebViewManager.java`

```
public class ReactWebViewManager extends SimpleViewManager<WebView> {

    private static final String TAG = "DXCaptchaView";
    ReactApplicationContext mRNContext;

    public ReactWebViewManager(ReactApplicationContext rnContext) {
        mRNContext = rnContext;
    }

    @Override
    public String getName() {
        return "DXCaptchaView";
    }

    @Override
    protected DXCaptchaView createViewInstance(ThemedReactContext rea
ctContext) {
        DXCaptchaView dxCaptchaView = new DXCaptchaView(reactContext,
 null);
        dxCaptchaView.setWebViewClient(new WebViewClient(){
            @Override
            public boolean shouldOverrideUrlLoading(WebView view, Str
ing url) {
                view.loadUrl(url);
                return true;
            }
        });
        return dxCaptchaView;
    }

    // 向RN发送事件
    public void sendEventToRN(Object data) {
        mRNContext.getJSModule(DeviceEventManagerModule.RCTDeviceEven
tEmitter.class)
                .emit("DXEvent", data);
    }

    @ReactProp(name = "appid")
    public void init(DXCaptchaView view, @Nullable String appid) {
        Log.e(TAG, "init appid:"+appid);
        ////可以将appid从这里传入
        view.init(appid);

        view.startToLoad(new DXCaptchaListener() {
            @Override
            public void handleEvent(WebView webView, DXCaptchaEvent d
xCaptchaEvent, Map<String, String> map) {
```

```
                switch(dxCaptchaEvent) {
                    case DXCAPTCHA_SUCCESS: // 验证成功的回调
                        String token = map.get("token"); // 成功时会传
递token参数
                        Log.i(TAG, "Verify Success. token: " + token)
;
                        sendEventToRN(token);
                        break;
                    case DXCAPTCHA_FAIL:      // 验证失败回调
                        Log.i(TAG, "Verify Failed.");
                        break;
                    default:
                        Log.i(TAG, "Unknown");
                        break;
                }
            }
        });
    }
}
```

1.注意重写的getName()方法一定要return "DXCaptchaView"，以让后面的js代码识别

2.方法加上`@ReactProp`标注

3.返回类型为`void`，在后面的js中以回调方法的形式获取原生程序中Java方法的返回值

注册ReactWebViewManager

- 在ReactWebViewManager.java同级目录下添加DxPackage类实现ReactPackage接口

```
public class DxPackage implements ReactPackage{

  public List<Class<? extends JavaScriptModule>> createJSModules() {
      return Collections.emptyList();
  }

  @Override
  public List<NativeModule> createNativeModules(ReactApplicationConte
xt reactContext) {
      return Collections.emptyList();
  }

  @Override
  public List<ViewManager> createViewManagers(ReactApplicationContext
 reactContext) {
      return Arrays.<ViewManager>asList(
```

```
            //注册ReactWebViewManager
                new ReactWebViewManager());
    }
  }
```

在MainApplication.java中添加DxPackage实例

- 在重写的getPackages()方法中添加DxPackage实例

```
public class MainApplication extends Application implements ReactApplic
ation {

  private final ReactNativeHost mReactNativeHost = new ReactNativeHost(
this) {
    @Override
    public boolean getUseDeveloperSupport() {
      return BuildConfig.DEBUG;
    }

    @Override
    protected List<ReactPackage> getPackages() {
      return Arrays.<ReactPackage>asList(
          new MainReactPackage(),
          //添加DxPackage实例
          new DxPackage()
      );
    }

    @Override
    protected String getJSMainModuleName() {
      return "index";
    }
  };

  @Override
  public ReactNativeHost getReactNativeHost() {
    return mReactNativeHost;
  }

  @Override
  public void onCreate() {
    super.onCreate();
    SoLoader.init(this, /* native exopackage */ false);
  }
}
```

ReactNative中js调用Android模块方法

- 在项目根目录新建一个DXCaptchaView.js文件

```
import React from 'react';
import {requireNativeComponent,View} from 'react-native';
import PropTypes from 'prop-types';

var iface = {
    //这里的name的值可自行定义
    name: 'DXCaptchaView',
    propTypes: {
        appid: PropTypes.string, //appid:对应Android原生程序中的DXCaptchaV
iew的init方法
        ...View.propTypes
    },
}
//下面的DXCaptchaView对应ReactWebViewManager.java中getName()方法的返回值
module.exports = requireNativeComponent('DXCaptchaView', iface,{
    nativeOnly:{
        "testID": true,
        'renderToHardwareTextureAndroid': true,
        'accessibilityComponentType': true,
        'accessibilityLabel': true,
        'importantForAccessibility': true,
        'accessibilityLiveRegion': true,
        'onLayout':true,
    }
});
```

- 在需要调用无感验证的js文件头导入模块库(这里以App.js为例)

```
import React, {Component,PropTypes } from 'react';
import {Platform, StyleSheet, Text, View} from 'react-native';
import { NativeModules } from 'react-native';
```

- 利用传递参数`appid`与`msg`的方式调用`ReactWebViewManager.java`中被打上标注`ReactProp`的方法

```
export default class App extends Component<Props> {

    // 监听无感验证的token返回
    componentDidMount() {
            this.subscription = DeviceEventEmitter.addListener('DXEve
```

```
nt', function (msg) {
                alert(msg)
            });
    }

    componentWillUnmount() {
        this.subscription.remove();
    }

  render() {
    var DXCaptchaView = require('./DXCaptchaView');
    return (
      //appid：无感验证appid，也可以直接将appid写死在原生app的Java代码中，如有需要请在官网控制台获取或者联系顶象工作人员，
      <DXCaptchaView  appid='appid' msg='msg'style={{width:300,height:200}}></DXCaptchaView>
    );
  }
}
```

以

```
<DXCaptchaView appid='appid' style={{width:300,height:200}}>
</DXCaptchaView>
```

调用验证码页面

编译运行

- RN 项目根目录下，执行命令 `react-native run-android`

最后呈现效果

验证成功

请向右拖动滑块完成拼图

顶象无感验证成功

验证未通过