

ReactNative利用CodePush实现热更新

ReactNative的优势之一就是它可以进行热更新，需要热更新的应用场景有很多种，例如：

- 各大应用市场审核时间过长
- 一些重业务逻辑，经常需要更新的页面
- 一个较为庞大的App，可能每一块都由一个部门负责，不能每个部门一有变动就推一个新的版本
- 如果app走自己平台的cdn的时候，需要考虑流量的成本

下面我们说一下，在ReactNative项目中如何接入CodePush，如果英语还ok的小伙伴，可以直接照着英文文档撸一遍[文档地址](#)。

CodePush

- 直接对用户部署代码更新
- 管理 Alpha, Beta 和生产环境应用
- 支持 React Native 和 Cordova
- 支持JavaScript 文件与图片资源的更新

安装CodePush CLI

在终端输入命令：`npm install -g code-push-cli`

创建CodePush账号

在终端输入命令：`code-push register`

命令输入完之后，会弹出网页，我们可以在上面注册账号登录，也可以直接用GitHub登录，登录之后，网页会返回给我们一个key，我们把这个key再粘贴回命令行就可以了。

登录

登录：`code-push login`

其它相关命令：`code-push logout`

列出登录的token：`code-push access-key ls`

删除某个key：`code-push access-key rm <accessKey>`

向CodePush注册app

Android与ios需要注册两次：

```
code-push app add MyApp-Android
code-push app add MyApp-ios
```

注册之后会返回key，这个key我们需要保存一下

在项目中集成CodePush SDK

在项目的根目录运行：`npm install --save react-native-code-push`

在Android目录运行：`npm i -g rnpm` //一般来说我们的React里面都会有，在V0.27以后

在Android中添加配置文件：`rnpm link react-native-code-push`

运行这条命令的过程中，它会让我们输入`key`，就是刚刚保存的那个，当然不输入也可以

配置Android工程

需要在 `android/app/build.gradle` 检查是否有如下内容，如果没有则添加：

```
apply from: "../../node_modules/react-native/react.gradle"
apply from: "../../node_modules/react-native-code-push/android/codepush.gradle"

...
dependencies {
    ...
    compile project(':react-native-code-push')
}
```

需要在 `android/settings.gradle` 检查是否有如下内容，如果没有则添加：

```
include ':app', ':react-native-code-push'
project(':react-native-code-push').projectDir = new File(rootProject.projectDir, '../node_modules/rea
```

获取部署的密钥

运行：`code-push deployment ls Daily -k`

进行配置

在 `MainApplication.java` 中添加如下代码：

```
...
// 1. Import the plugin class.
import com.microsoft.codepush.react.CodePush;

public class MainApplication extends Application implements ReactApplication {

    private final ReactNativeHost mReactNativeHost = new ReactNativeHost(this) {
        ...
        // 2. Override the getJSBundleFile method in order to let
        // the CodePush runtime determine where to get the JS
        // bundle location from on each app start
        @Override
        protected String getJSBundleFile() {
            return CodePush.getJSBundleFile();
        }

        @Override
        protected List<ReactPackage> getPackages() {
            // 3. Instantiate an instance of the CodePush runtime and add it to the list of
            // existing packages, specifying the right deployment key. If you don't already
            // have it, you can run "code-push deployment ls <appName> -k" to retrieve your key.
            return Arrays.<ReactPackage>asList(
                new MainReactPackage(),
                new CodePush("deployment-key-here", MainApplication.this, BuildConfig.DEBUG)
            );
        }
    };
}
```

```
...
// 1. Declare your ReactNativeHost to extend ReactInstanceHolder. ReactInstanceHolder is a subset of
import com.microsoft.codepush.react.ReactInstanceHolder;

public class MyReactNativeHost extends ReactNativeHost implements ReactInstanceHolder {
    // ... usual overrides
}

// 2. Provide your ReactNativeHost to CodePush.

public class MainApplication extends Application implements ReactApplication {

    private final MyReactNativeHost mReactNativeHost = new MyReactNativeHost(this);

    @Override
    public void onCreate() {
        CodePush.setReactInstanceHolder(mReactNativeHost);
        super.onCreate();
    }
}
```

这样就完事啦～

如果您遇到什么问题，欢迎给我提issue～