react-native-zello-sdk

Zello SDK for React Native.

Our React Native SDK offers itself as a thin wrapper around our native iOS and Android SDK's. It is built with TypeScript, and communicates with the native SDK's via React Native's native module support.

This repository contains both the React Native Bridge, as well as an example app to demonstrate usage.

To see an example of the React Native SDK, check out the example folder.

The following instructions are for the SDK, not for the example app.

Prerequisites

A React Native application and environment.

Installation

Install the package via NPM: `sh npm install @zelloptt/react-native-zello-sdk`

iOS

Please reference the iOS Installation Guide.

Android

Before getting started, please reference the Android Installation Guide.

Dependencies

Add the repositories to your app's `build.gradle` :

```
repositories {
  google()
  mavenCentral()
  maven {
    url = uri("https://zello-sdk.s3.amazonaws.com/android/latest")
  }
}
```

Linking

Since the Zello Android SDK requires Hilt, and we must inject the `Zello` instance into the `ReactPackage`, the project does not work with autolinking.

As such, you will need to create a `ReactPackage` wrapper like this:

```
/// ZelloAndroidSdkPackage.kt

import com.facebook.react.ReactPackage
import com.facebook.react.bridge.NativeModule
import com.facebook.react.bridge.ReactApplicationContext
import com.facebook.react.uimanager.ViewManager
import com.zello.sdk.Zello
import com.zellosdk.ZelloAndroidSdkModule
import javax.inject.Inject

class ZelloAndroidSdkPackage @Inject constructor(private val zello: Zello) : ReactPacka

  override fun createNativeModules(reactContext: ReactApplicationContext): List<NativeM
    return listOf(ZelloAndroidSdkModule(reactContext, zello))
  }

  override fun createViewManagers(reactContext: ReactApplicationContext): List<ViewMana
    return emptyList()
  }
}
```

Then, add it to the packages list:

```
/// MainApplication.kt

@HiltAndroidApp
class MainApplication : Application(), ReactApplication {

    @Inject lateinit var zello: Zello

    override val reactNativeHost: ReactNativeHost =
      object : DefaultReactNativeHost(this) {
        override fun getPackages(): List<ReactPackage> =
            PackageList(this).packages.apply {
              // Packages that cannot be autolinked yet can be added manually here, for
              add(ZelloAndroidSdkPackage(zello))
            }

        override fun getJSMainModuleName(): String = "index"

        override fun getUseDeveloperSupport(): Boolean = BuildConfig.DEBUG

        override val isNewArchEnabled: Boolean = BuildConfig.IS_NEW_ARCHITECTURE_ENABLE
        override val isHermesEnabled: Boolean = BuildConfig.IS_HERMES_ENABLED
      }

      override fun onCreate() {
        super.onCreate()
        zello.start()
      }
}
```

Contributing

See the contributing guide to learn how to contribute to the repository and the development workflow.

License

MIT

Made with create-react-native-library