SALESmanago Mobile - React Native libraries

1. Configuration

For both, Android and iOS, you should include SALESmanago Mobile libraries and follow the steps as described in https://support.salesmanago.com/developers/

Android

Import `appmanagolibrary-react-adapter-release.aar` to the project using Gradle (in the same way as `appmanagolibrary-firebase-release.aar`). Android requires additional module registration (https://reactnative.dev/docs/native-modules-android#register-the-module-android-specific). To do this, add an instance of the `AmPackage` class to the `getPackages()` method in the `MainApplication` class:

```
import com.appmanago.lib.AmPackage;
//...

@Override
protected List<ReactPackage> getPackages() {
 @SuppressWarnings("UnnecessaryLocalVariable")
 List<ReactPackage> packages = new PackageList(this).getPackages();
 packages.add(new AmPackage());
 return packages;
}
```

iOS

Add `AppmanagoLibraryReactAdapter.framework` to the project along with `AmMonitor.framework`. React native module should be installed automatically, without any additional configuration.

2. Using the libraries in JavaScript

You can import the library adapter in JavaScript using:

```
import { NativeModules } from 'react-native';
const { AmMonitorAdapter } = NativeModules;
```

Details for each method have been listed below. You can learn more about those methods here: https://support.salesmanago.com/developers/

`create(), destroy(), eventStarted(), eventEnded()`:

Note: While using the Android library it's important that calls to `eventStarted()` and `eventEnded()` methods were in between `create()` i `destroy()` calls for the same module. You can achieve this by calling `create()` when the component is being created. Call `destroy()` before the unmounting of the component (before the end of the component's lifecycle).

```
AmMonitorAdapter.create('moduleSimpleId');
AmMonitorAdapter.eventStarted('moduleSimpleId');
AmMonitorAdapter.eventEnded('moduleSimpleId');
AmMonitorAdapter.create('moduleSimpleId');
```

`clicked()`:
Calling this method should additionally contain the ID of the module to which the function belongs. It should also be placed in between `create()` i `destroy()` for the same module..

```
AmMonitorAdapter.clicked('moduleSimpleId', 'functionSimpleId');
```

`syncEmail(), syncMsisdn()`:

```
AmMonitorAdapter.syncEmail('mail@example.com');
AmMonitorAdapter.syncMsisdn('+15554443332');
```

`sendLocation()`:

```
Geolocation.getCurrentPosition(
 async(position) => {
   let coords = position.coords;
   AmMonitorAdapter.sendLocation(coords.latitude.toString(),
     coords.longitude.toString());
   //e. g.
   //AmMonitorAdapter.sendLocation('37.33233141', '-122.0312186');
 };
```

eventCustom():

```
AmMonitorAdapter.eventCustom('u_event_id', {
 "textParamterName": {
   "type": "text",
   "value": "Sample text"
 },
 "numberParamterName1": {
   "type": "integer",
   "value": 123
 },
 "numberParamterName2": {
   "type": "double",
   "value": 123.678
 },
 "dateParameterName": {
   "type": "date",
   "value": new Date().toISOString()
 },
 "booleanParameterName": {
   "type": "boolean",
   "value": true
}
});
```

sendUserProperties():

```
AmMonitorAdapter.sendUserProperties({
 "textPropertyName": {
   "type": "text",
   "value": "Sample text"
 },
 "numberPropertyName1": {
   "type": "integer",
   "value": 666
 },
 "numberPropertyName2": {
   "type": "double",
   "value": 123.678
 },
 "datePropertyName": {
   "type": "date",
   "value": new Date().toISOString()
 },
```

```
 "booleanPropertyName": {
   "type": "boolean",
   "value": false
 }
});
```