



Mobile User Experience Monitoring Hybrid Apps – Capacitor

Table of Contents

Steps for Bundling eG Mobile Agent with Application.....	3
How to Install eG Capacitor Agent for Angular Applications - Android.....	3
Build and Run the application.....	9
How to Install eG Capacitor Agent for React Applications - Android.....	9
Build and Run the application.....	11
How to Install eG Capacitor Agent for Angular Applications - iOS	12
Build and Run the application.....	17
How to Install eG Capacitor Agent for React Applications - iOS	17
Build and Run the application.....	18

Steps for Bundling eG Mobile Agent with Application

The following steps are required to bundle the 'eG Mobile Agent' into Hybrid apps(Capacitor) to activate user experience monitoring. Capacitor apps monitored using following components.

1. eGCapacitor Plugin
2. eGJS Agent.

The above components need to be installed on the apps to activate mobile user experience.

How to Install eG Capacitor Agent for Angular Applications - Android

- a. Add 'eG-Hybrid' Apps Component into eG Manager Console.
- b. Run the below commands from command prompt from application's root folder to install eG Agent capacitor plugin into the application.

```
➤ npm i @eginnovations/eginnovations-capacitor-plugin
➤ npx cap sync
➤ ionic build
```

- c. Open the application in android studio or any IDE and open the MainActivity.java Add the below import statements

```
import com.eginnovations.capacitor.plugin.eGCapacitorPlugin;
import com.eg.agent.android.eGAndroidAgent;
import com.eg.agent.android.AppPlatform;
```

In the same file, inside onCreate method, add the below bolded lines and replace the app token and URL values that are obtained from add component page.

```
eGAndroidAgent.withCollectorHost("<APP_TOKEN>", "URL").start(getApplicationContext());
eGAndroidAgent.setAppliactionPlatform(AppPlatform.Capacitor);

// Initializes the Bridge
this.init(savedInstanceState, new ArrayList<Class<? extends Plugin>>() {{
    // Additional plugins you've installed go here
    // Ex: add(TotallyAwesomePlugin.class);
    add(eGCapacitorPlugin.class);
}});
```

- d. Open build.gradle under android folder)and add the below lines at the end of file.

```
buildscript {
    repositories {
        flatDir {
            dirs '../node_modules/@eginnovations/eginnovations-capacitor-plugin/eg-
agent'
        }
    }
}
```

```
dependencies {
    classpath fileTree(include: ['*.jar'], dir:
'../node_modules/@eginnovations/eginnovations-capacitor-plugin/eg-agent')
}
```

- e. Open build.gradle under app folder which is under android folder and add the below lines at the end of file.

```
repositories {
    flatDir {
        dirs '../node_modules/@eginnovations/eginnovations-capacitor-plugin/eg-agent'
    }
}
apply plugin: 'eg'
dependencies {
    compile fileTree(dir: '../node_modules/@eginnovations/eginnovations-capacitor-
plugin/eg-agent' , include: ['*.jar'])
}
```

- f. Open gradle.properties under android folder and add the below lines at the end of file.

```
android.jetifier.blacklist=rewriter-agent-1.0.0.jar,roboelectric-1.2-8.1-jar-with-
dependencies.jar
```

- g. Open index.html under applications src folder and add the below lines. Replace the app token and URL with the values from component page.

To instrument your Hybrid Ionic Capacitor apps using eG Hybrid Capacitor agent

Download the PDF and follow the steps to inject eG Hybrid Android agent into your mobile apps: [Download](#)

Download eG Capacitor Plugin: [Hybrid_Capacitor_Agent.zip](#) [Download](#)

Include this line into applications index.html to inject eG JS agent into your mobile apps

```
<!-- RUM Header -->
<script charset='UTF-8' type='text/javascript'>
window['egrum-start_time'] = new Date().getTime();
window['Site_Name'] = '1979b2f7-6ea5-4373-b243-f2292742e602-1601356122936';
window['beacon-url'] = 'http://192.168.11.136:7077';
</script>
<script src='egmobilerum.js' async> </script>
```

- h. In AndroidManifest.xml add **android:usesCleartextTraffic="true"** in application tag. Sample below

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="io.ionic.starter">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
```

```
android:theme="@style/AppTheme"
android:usesCleartextTraffic="true">
...
```

- i. In angular.json under the application root folder, include below lines under assets

```
"assets": [
  {
    "glob": "**/*.svg",
    "input": "node_modules/ionicons/dist/ionicons/svg",
    "output": "./svg"
  },
  {
    "glob": "**/egmobilerum.js",
    "input": "node_modules/@eginnovations/eginnovations-capacitor-
plugin/src",
    "output": "./"
  }
],
```

- j. In the app.component.ts file present under src/app folder add the below lines under import. If any of the import statements are already present, please ignore them

```
import { OnInit } from '@angular/core';

import { NavigationStart, NavigationEnd, NavigationError, NavigationCancel,
RoutesRecognized, Router, ActivatedRoute } from '@angular/router';

import { Plugins } from '@capacitor/core';

import '@eginnovations/eginnovations-capacitor-plugin';

const { eGCapacitorPlugin } = Plugins;

declare var EGRUM: any;
```

- k. In the same file under Component class the following bolded lines must be added.

```
export class AppComponent implements OnInit {
  deviceInformation = "";
  resourceInformation = "";
  applicationInformation = "";
  activityName = "";
  country = "";
  device = "";
  os = "";
  applicationName = "";
  osVersion = "";
  isEnabled = true;
```

```

constructor(
  private platform: Platform,
  private router: Router,
  private splashScreen: SplashScreen,
  private activatedRoute: ActivatedRoute,
  private statusBar: StatusBar
) {
  this.initializeApp();
}
initializeApp() {
  this.platform.ready().then(() => {
    this.statusBar.styleDefault();
    this.splashScreen.hide();
  });
}
// eG RUM Monitoring Starts
ngOnInit(): void {
  this.router.events.subscribe(event => {
    let componentName = (this.activatedRoute.component !== null) ?
this.activatedRoute.component['name'] : 'default';
    if (event instanceof RoutesRecognized) {
      let actSnap = event.state.root;
      const childLength = actSnap.children.length;
      while (actSnap && childLength > 0) {
        actSnap = actSnap.children[0];
        if (actSnap && actSnap.component && actSnap.component['name']) {
          componentName = actSnap.component['name'] || componentName;
        }
      }
    }
    if (event instanceof NavigationStart) {
      if (typeof EGRUM !== 'undefined') {
        EGRUM.vPage('markVirtualPageStart');
        EGRUM.vPage('page', componentName);
        EGRUM.vPage('meta', 'component', componentName);
      }
    } else if (event instanceof NavigationEnd) {
      if (typeof EGRUM !== 'undefined') {
        EGRUM.vPage('markVirtualPageEnd', { status: 'completed', url:
window.location.href });
      }
    } else if (event instanceof NavigationError) {
      if (typeof EGRUM !== 'undefined') {
        EGRUM.vPage('markVirtualPageEnd', {
          status: 'error',

```

```

        url: window.location.href,
        explanation: event['error'] && event['error'].toString()
    });
}
} else if (event instanceof NavigationCancel) {
    if (typeof EGRUM !== 'undefined') {
        EGRUM.vPage('markVirtualPageEnd', {
            status: 'aborted',
            url: window.location.href,
            explanation: event['reason'] || ''
        });
    }
}
});
this.load();
}
async load() {
    try {
        let result = await eGCapacitorPlugin.isEnabled();
        this.isEnabled = result['isEnabled'];
    } catch(err) {
        console.log('AppComponent::din:error::'+err);
    }
    try {
        let result = await eGCapacitorPlugin.getOSVersion();
        const JSobj = JSON.parse(JSON.stringify(result));
        this.osVersion = JSobj.osVersion;
    } catch(err) {
        console.log('AppComponent::osVersion:error::'+err);
    }
    try {
        let result = await eGCapacitorPlugin.getDevInfo();
        this.deviceInformation = result['deviceInformation'];
    } catch(err) {
        console.log('AppComponent::din:error::'+err);
    }
    try {
        this.applicationInformation = (await
Plugins.eGCapacitorPlugin.getAppInfo())['applicationInformation'];
    } catch(err) {
        console.log('AppComponent::ain:error::'+err);
    }
    try {
        this.resourceInformation = (await
Plugins.eGCapacitorPlugin.getResInfo())['resourceInformation'];
    }
}

```

```

    } catch(err) {
        console.log('AppComponent::rin:error:::' + err);
    }
    try {
        this.activityName = (await
Plugins.eGCapacitorPlugin.getCurrentActivityName())[ 'activityName'];
    } catch(err) {
        console.log('AppComponent::at:error:::' + err);
    }
    try {
        this.country = (await Plugins.eGCapacitorPlugin.getUserCountry())[ 'country'];
    } catch(err) {
        console.log('AppComponent::cty:error:::' + err);
    }
    try {
        this.device = (await Plugins.eGCapacitorPlugin.getDeviceName())[ 'deviceName'];
    } catch(err) {
        console.log('AppComponent::odvn:error:::' + err);
    }
    try {
        this.os = (await Plugins.eGCapacitorPlugin.getBaseOs())[ 'osName'];
    } catch(err) {
        console.log('AppComponent::os:error:::' + err);
    }
    try {
        this.applicationName = (await
Plugins.eGCapacitorPlugin.getAppname())[ 'appName'];
    } catch(err) {
        console.log('AppComponent::appn:error:::' + err);
    }
    if (typeof EGRUM !== 'undefined') {
        EGRUM.setDeviceDetails(this.deviceInformation);
        EGRUM.setAppDetails(this.applicationInformation);
        EGRUM.setResDetails(this.resourceInformation);
        EGRUM.setActivity(this.activityName);
        EGRUM.setUserCountry(this.country);
        EGRUM.setDeviceName(this.device);
        EGRUM.setBaseOS(this.os);
        EGRUM.setAppName(this.applicationName);
        EGRUM.setOSVersion(this.osVersion);
        EGRUM.isEnabled(this.isEnabled);
    }
}
} // eG RUM SPA Monitoring Ends

```


1. For capturing error events in virtual pages, edit the app.module.ts file in the <APP_HOME>\src\app directory, and insert the following code block after all the import statements in that file:

```
import { NgModule, Injectable, ErrorHandler } from '@angular/core';
declare var EGRUM: any;
@Injectable()
export class EGRUMErrorHandler implements ErrorHandler {
  handleError(err: any): void {
    EGRUM.onerror({
      message: err.message,
      filename: err.filename,
      lineNumber: err.lineNumber,
      columnNumber: err.columnNumber,
      stack: err.stack,
      meta: { type: "Internal Error." }
    });
  }
}
```

Build and Run the application

Clean the project and Build the application. After successful build, run the application in an emulator or mobile device and login to eG manager application to start seeing data.

Use the below commands to build and copy the source files to android

- ionic build
- npx cap sync android

How to Install eG Capacitor Agent for React Applications - Android

- a. Add 'eG-Hybrid' Apps Component into eG Manager Console.
- b. Open capacitor.config.json and change bundledWebRuntime to true and webDir to "public" (The directory name where index.html is present)

```
{
  "bundledWebRuntime": true,
  "webDir": "public"
}
```

- c. Run the below commands from command prompt from application's root folder to install eG Agent capacitor plugin into the application.

- npm i [@eginnovations/eginnovations-capacitor-plugin](https://www.npmjs.com/package/@eginnovations/eginnovations-capacitor-plugin)

- ionic build
- npx cap copy web
- npx cap sync

- d. Copy egmobilerum_react.js from node_modules\@eginnovations\eginnovations-capacitor-plugin\src to public folder (where index.html is located).
- e. Open the application in android studio or any IDE and open the MainActivity.java Add the below import statements

```
import com.eginnovations.capacitor.plugin.eGCapacitorPlugin;
import com.eg.agent.android.eGAndroidAgent;
import com.eg.agent.android.AppPlatform;
```

In the same file, inside onCreate method, add the below bolded lines and replace the app token and URL values that are obtained from add component page.

```
eGAndroidAgent.withCollectorHost("<APP_TOKEN>","URL").start(getApplicationContext());
eGAndroidAgent.setAppliactionPlatform(AppPlatform.Capacitor);

// Initializes the Bridge
this.init(savedInstanceState, new ArrayList<Class<? extends Plugin>>() {{
    // Additional plugins you've installed go here
    // Ex: add(TotallyAwesomePlugin.class);
    add(eGCapacitorPlugin.class);
}});
```

- f. Open build.gradle under android folder)and add the below lines at the end of file.

```
buildscript {
    repositories {
        flatDir {
            dirs '../node_modules/@eginnovations/eginnovations-capacitor-plugin/eg-agent'
        }
    }
    dependencies {
        classpath fileTree(include: ['*.jar'], dir:
        '../node_modules/@eginnovations/eginnovations-capacitor-plugin/eg-agent')
    }
}
```

- g. Open build.gradle under app folder which is under android folder and add the below lines at the end of file.

```
repositories {
    flatDir {
        dirs '../..node_modules/@eginnovations/eginnovations-capacitor-plugin/eg-agent'
    }
}
apply plugin: 'eg'
dependencies {
```

```
    compile fileTree(dir: '../node_modules/@eginnovations/eginnovations-capacitor-  
    plugin/eg-agent' , include: ['*.jar'])  
}
```

- h. Open gradle.properties under android folder and add the below lines at the end of file.

```
android.jetifier.blacklist=rewriter-agent-1.0.0.jar,robolectric-1.2-8.1-jar-with-  
dependencies.jar
```

- i. Open index.html under applications src folder and add the below lines. Replace the app token and URL with the values from component page.

```
<!-- RUM Header -->  
    <script src="capacitor.js"></script>  
    <script charset='UTF-8' type='text/javascript'>  
        window['egrum-start_time'] = new Date().getTime();  
        window['Site_Name'] = <APP_TOKEN>;  
        window['beacon-url'] = <COLLECTOR_URL>;  
        if(!window['egrum-config']) window['egrum-config'] = {}; (function (config) {config.capture =  
        { jsError:true, resourceDetails:true, ajax:true, ajaxCorrelation:false, fetch:true, spa:true,  
        overwriteBtmUName:true, excludeURLPattern:'none', ajaxExcludeURLPattern:'none',  
        includeURLPattern:'*'}; config.username ={ username_enabled:true, selectortype:'xpath',  
        selectoridorphpath:'/html/body/div/h1/p', selectorindex:1;}})(window['egrum-config']);  
    </script>  
    <script id="egrum" src='egmobilerum_react.js' async > </script> <!-- RUM Header -->
```

- j. In AndroidManifest.xml add **android:usesCleartextTraffic="true"** in application tag.
Sample below

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="io.ionic.starter">  
  
    <application  
        android:allowBackup="true"  
        android:icon="@mipmap/ic_launcher"  
        android:label="@string/app_name"  
        android:roundIcon="@mipmap/ic_launcher_round"  
        android:supportsRtl="true"  
        android:theme="@style/AppTheme"  
        android:usesCleartextTraffic="true">  
  
    ...
```

Build and Run the application

Clean the project and Build the application. After successful build, run the application in an emulator or mobile device and login to eG manager application to start seeing data.

Use the below commands to build and copy the source files to android

- ionic build
- npx cap sync android

How to Install eG Capacitor Agent for Angular Applications – iOS

- Add 'eG-Hybrid' Apps Component into eG Manager Console.
- Run the below commands from command prompt from application's root folder to install eG Agent capacitor plugin into the application.

- npm i [@eginnovations/eginnovations-capacitor-plugin](#)
- npx cap sync
- ionic build

- Open the application in Xcode and open the AppDelegate.Swift Add the below import statements

```
import eGMobileAPM
```

In the same file, inside didFinishLaunchingWithOptions method, add the below bolded lines and replace the app token and URL values that are obtained from add component page.

```
egAPM.sharedInstance().start(appToken:"APP_TOKEN", collectorHost:"URL")
```

- Open index.html under applications src folder and add the below lines. Replace the app token and URL with the values from component page.

To instrument your Hybrid Ionic Capacitor apps using eG Hybrid Capacitor agent

Download the PDF and follow the steps to inject eG Hybrid Android agent into your mobile apps:

[egMobileAgent_Installation_Hybrid-Capacitor.pdf](#)

Download

Download eG Capacitor Plugin:

[Hybrid_Capacitor_Agent.zip](#)

Download

Include this line into applications index.html to inject eG JS agent into your mobile apps

```
<!-- RUM Header -->
<script charset='UTF-8' type='text/javascript'>
window['egrum-start_time'] = new Date().getTime();
window['Site_Name'] = '1979b2f7-6ea5-4373-b243-f2292742e602-1601356122936';
window['beacon-url'] = 'http://192.168.11.136:7077';
</script>
<script src='egmobilerum.js' async> </script>
```

- e. In angular.json under the application root folder, include below lines under assets

```
"assets": [  
  {  
    "glob": "**/*.svg",  
    "input": "node_modules/ionicons/dist/ionicons/svg",  
    "output": "./svg"  
  },  
  {  
    "glob": "**/egmobilerum.js",  
    "input": "node_modules/eginnovations-capacitor-plugin/src",  
    "output": "./"  
  }  
],
```

- f. In the app.component.ts file present under a src/app folder add the below lines under import. If any of the import statements are already present, please ignore them

```
import { OnInit } from '@angular/core';  
  
import { NavigationStart, NavigationEnd, NavigationError, NavigationCancel,  
RoutesRecognized, Router, ActivatedRoute } from '@angular/router';  
  
import { Plugins } from '@capacitor/core';  
  
import 'eginnovations-capacitor-plugin';  
  
const { eGCapacitorPlugin } = Plugins;  
  
declare var EGRUM: any;
```

- g. In the same file under Component class the following bolded lines must be added.

```
export class AppComponent implements OnInit {  
  deviceInformation = "";  
  resourceInformation = "";  
  applicationInformation = "";  
  activityName = "";  
  country = "";  
  device = "";  
  os = "";  
  applicationName = "";  
  osVersion = "";  
  isEnabled = true;  
  constructor(  

```

```

private platform: Platform,
private router: Router,
private splashScreen: SplashScreen,
private activatedRoute: ActivatedRoute,
private statusBar: StatusBar
) {
  this.initializeApp();
}
initializeApp() {
  this.platform.ready().then(() => {
    this.statusBar.styleDefault();
    this.splashScreen.hide();
  });
}
// eG RUM Monitoring Starts
ngOnInit(): void {
  this.router.events.subscribe(event => {
    let componentName = (this.activatedRoute.component !== null) ?
this.activatedRoute.component['name'] : 'default';
    if (event instanceof RoutesRecognized) {
      let actSnap = event.state.root;
      const childLength = actSnap.children.length;
      while (actSnap && childLength > 0) {
        actSnap = actSnap.children[0];
        if (actSnap && actSnap.component && actSnap.component['name']) {
          componentName = actSnap.component['name'] || componentName;
        }
      }
    }
    if (event instanceof NavigationStart) {
      if (typeof EGRUM !== 'undefined') {
        EGRUM.vPage('markVirtualPageStart');
        EGRUM.vPage('page', componentName);
        EGRUM.vPage('meta', 'component', componentName);
      }
    } else if (event instanceof NavigationEnd) {
      if (typeof EGRUM !== 'undefined') {
        EGRUM.vPage('markVirtualPageEnd', { status: 'completed', url:
window.location.href });
      }
    } else if (event instanceof NavigationError) {
      if (typeof EGRUM !== 'undefined') {
        EGRUM.vPage('markVirtualPageEnd', {
          status: 'error',
          url: window.location.href,

```

```

        explanation: event['error'] && event['error'].toString()
    });
}
} else if (event instanceof NavigationCancel) {
    if (typeof EGRUM !== 'undefined') {
        EGRUM.vPage('markVirtualPageEnd', {
            status: 'aborted',
            url: window.location.href,
            explanation: event['reason'] || ''
        });
    }
}
});
this.load();
}
async load() {
    try {
        let result = await eGCapacitorPlugin.isEnabled();
        this.isEnabled = result['isEnabled'];
    } catch(err) {
        console.log('AppComponent::din:error::'+err);
    }
    try {
        let result = await eGCapacitorPlugin.getOSVersion();
        const JSobj = JSON.parse(JSON.stringify(result));
        this.osVersion = JSobj.osVersion;
    } catch(err) {
        console.log('AppComponent::osVersion:error::'+err);
    }
    try {
        let result = await eGCapacitorPlugin.getDevInfo();
        this.deviceInformation = result['deviceInformation'];
    } catch(err) {
        console.log('AppComponent::din:error::'+err);
    }
    try {
        this.applicationInformation = (await
Plugins.eGCapacitorPlugin.getAppinfo())['applicationInformation'];
    } catch(err) {
        console.log('AppComponent::ain:error::'+err);
    }
    try {
        this.resourceInformation = (await
Plugins.eGCapacitorPlugin.getResInfo())['resourceInformation'];
    } catch(err) {

```

```

    console.log('AppComponent::rin:error:::' + err);
  }
  try {
    this.activityName = (await
Plugins.eGCapacitorPlugin.getCurrentActivityName())[ 'activityName'];
  } catch(err) {
    console.log('AppComponent::at:error:::' + err);
  }
  try {
    this.country = (await Plugins.eGCapacitorPlugin.getUserCountry())[ 'country'];
  } catch(err) {
    console.log('AppComponent::cty:error:::' + err);
  }
  try {
    this.device = (await Plugins.eGCapacitorPlugin.getDeviceName())[ 'deviceName'];
  } catch(err) {
    console.log('AppComponent::odvn:error:::' + err);
  }
  try {
    this.os = (await Plugins.eGCapacitorPlugin.getBaseOs())[ 'osName'];
  } catch(err) {
    console.log('AppComponent::os:error:::' + err);
  }
  try {
    this.applicationName = (await
Plugins.eGCapacitorPlugin.getAppname())[ 'appName'];
  } catch(err) {
    console.log('AppComponent::appn:error:::' + err);
  }

  if (typeof EGRUM !== 'undefined') {
    EGRUM.setDeviceDetails(this.deviceInformation);
    EGRUM.setAppDetails(this.applicationInformation);
    EGRUM.setResDetails(this.resourceInformation);
    EGRUM.setActivity(this.activityName);
    EGRUM.setUserCountry(this.country);
    EGRUM.setDeviceName(this.device);
    EGRUM.setBaseOS(this.os);
    EGRUM.setAppName(this.applicationName);
    EGRUM.setOSVersion(this.osVersion);
    EGRUM.isEnabled(this.isEnabled);
  }
}
} // eG RUM SPA Monitoring Ends

```


- h. For capturing error events in virtual pages, edit the app.module.ts file in the <APP_HOME>\src\app directory, and insert the following code block after all the import statements in that file:

```
import { NgModule, Injectable, ErrorHandler } from '@angular/core';
declare var EGRUM: any;
@Injectable()
export class EGRUMErrorHandler implements ErrorHandler {
  handleError(err: any): void {
    EGRUM.onerror({
      message: err.message,
      filename: err.filename,
      lineNumber: err.lineNumber,
      columnNumber: err.columnNumber,
      stack: err.stack,
      meta: { type: "Internal Error." } }
    });
  }
}
```

Build and Run the application

Clean the project and Build the application. After successful build, run the application in an simulator or mobile device and login to eG manager application to start seeing data.

Use the below commands to build and copy the source files to iOS

- ionic build
- npx cap sync ios

How to Install eG Capacitor Agent for React Applications - iOS

- a. Add 'eG-Hybrid' Apps Component into eG Manager Console.
- b. Open capacitor.config.json and change bundledWebRuntime to true and webDir to "public" (The directory name where index.html is present)

```
{
  "bundledWebRuntime": true,
  "webDir": "public"
}
```

- c. Run the below commands from command prompt from application's root folder to install eG Agent capacitor plugin into the application.

```
➤ npm i @eginnovations/eginnovations-capacitor-plugin
➤ ionic build
➤ npx cap copy web
➤ npx cap sync
```

- d. Copy egmobilerum_react.js from node_modules\@eginnovations\eginnovations-capacitor-plugin\src to public folder (where index.html is located).
- e. Open the application in Xcode and open the AppDelegate.swift Add the below import statements

```
import eGMobileAPM
```

In the same file, inside didFinishLaunchingWithOptions method, add the below bolded lines and replace the app token and URL values that are obtained from add component page.

```
egAPM.sharedInstance().start(appToken:"APP_TOKEN", collectorHost:"URL")
```

- f. Open index.html under applications src folder and add the below lines. Replace the app token and URL with the values from component page.

```
<!-- RUM Header -->
  <script src="capacitor.js"></script>
  <script charset='UTF-8' type='text/javascript'>
window['egrum-start_time'] = new Date().getTime();
window['Site_Name'] = <APP_TOKEN>;
window['beacon-url'] = <COLLECTOR_URL>;
if(!window['egrum-config']) window['egrum-config'] = {}; (function (config) {config.capture =
{ jsError:true, resourceDetails:true, ajax:true, ajaxCorrelation:false, fetch:true, spa:true,
overwriteBtmUName:true, excludeURLPattern:'none', ajaxExcludeURLPattern:'none',
includeURLPattern:'*'}; config.username = { username_enabled:true, selectortype:'xpath',
selectoridorphpath:'/html/body/div/h1/p', selectorindex:1;}})(window['egrum-config']);
</script>
  <script id="egrum" src='egmobilerum_react.js' async > </script> <!-- RUM Header -->
```

Build and Run the application

Clean the project and Build the application. After successful build, run the application in an simulator or mobile device and login to eG manager application to start seeing data.

Use the below commands to build and copy the source files to ios.

- ionic build
- npx cap sync ios