

Perform 2018

Mobile Performance Management for Native Mobile Applications

Hands-on Training Instructions

Lesson 6 – Hybrid Ionic Mobile Application Monitoring





Preparation - Download the mobile applications projects

1. Launch your command line interface
 - a. Windows : Windows button -> type "cmd" at the prompt
 - b. MacOS : Search for "Terminal" in the Spotlight Search
2. Create a directory structure for your projects. **It is important that the directory path does not contain any space.**
For example, create directly from the C drive or from your MacOS drive
 - a. Windows : md c:\projects
 - b. MacOS : mkdir projects
3. Switch to your projects directory :
 - a. Windows : cd c:\projects
 - b. MacOS : cd projects
4. Clone the mobile-hotday-2018 repository from Github
 - a. git clone <https://github.com/Dynatrace/mobile-hotday-2018.git>

Lesson 4 - Hybrid Ionic Mobile Application Monitoring

So what we have done so far applies to native mobile applications for Android and iOS. These applications are built for a specific platform using the development language of the platform (Java for Android, Swift/Objective-C for iOS). So if you want to target both Android and iOS users, you need to develop the application twice because the languages and frameworks are different.

In order to make it easier to develop applications once for all platforms, frameworks using web technologies (HTML5, Angular, React, etc) embedded in mobile applications have emerged. Such frameworks are Cordova/PhoneGap and Ionic. Because they use different frameworks and a mix of native and web technologies, these apps are referred as hybrid mobile applications.

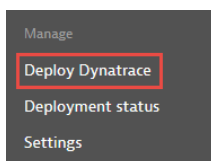
You have seen the easyTravel native application. We will now work with another flavor of easyTravel for mobile, an hybrid easyTravel app build on Ionic/Cordova.

Instructions

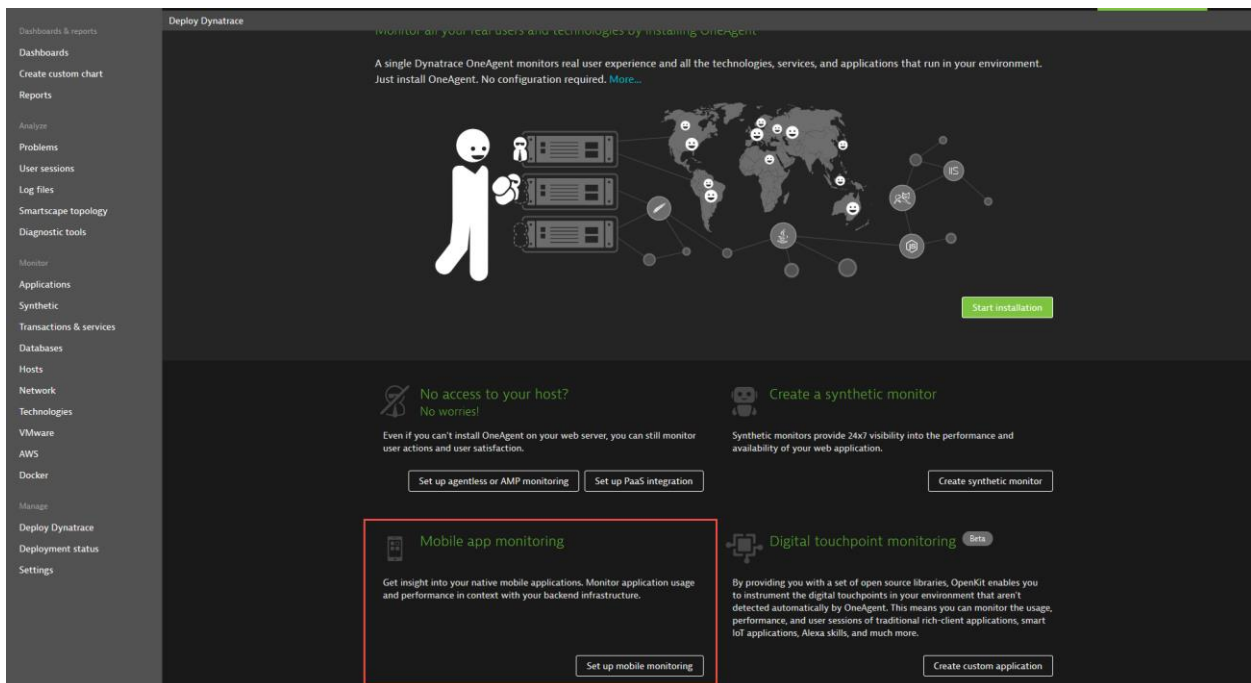
1. From your command line interface, navigate to the **Dynatrace-Ionic-Monitoring** directory
2. Execute the following command : **npm install**
 - This downloads and installs the easyTravel Ionic application project dependencies from the NPM repository
3. Execute the following command : **ionic cordova platform add android**
 - This command tells Ionic to add the necessary files to enable and configure the project to run the application on the Android platform
 - **(Optional)** For iOS, execute this command : **ionic cordova platform add ios**
4. Build the application by executing the following command : **ionic cordova build android**
 - **(Optional)** For iOS, execute this command : **ionic cordova build ios**
 - If the build fails, you might not have **ios-deploy** installed on your Mac.
 - Run this command to install it : **sudo npm install -g ios-deploy --unsafe-perm=true**
5. If your build is successful, you should have a line displayed with the following : **BUILD SUCCESSFUL in XXs**. If your build is not successful, let us know!!!
6. Test your application with the following command : **ionic cordova run android**.



- a. **(Optional)** Use this command for iOS : **ionic cordova run ios**
7. Navigate in the **easyTravel** application. This application is similar to the **easyTravel** native Android application of the previous exercise but it is instead an Angular web application running in a webview.
8. So you now have a nice Ionic easyTravel application. But how to monitor it with Dynatrace?
9. Close the application and uninstall it from the device.
10. Return to your command line interface and install the Dynatrace Cordova Plugin in your project by executing the following command : **cordova plugin add dynatrace-cordova-plugin --save**
11. For this application, because it is hybrid, with a native part and a web part, we need to define 2 applications in Dynatrace
12. First, we will define the native mobile application, the same way we did in the Lesson 2. In the Dynatrace portal, on the left menu at the bottom, click on **Deploy Dynatrace**



13. On the Deploy Dynatrace screen, go to Mobile app monitoring and click on **Set up mobile monitoring**



14. Enter an application name. For consistency, please use the following syntax: YourName_Ionic_Native (replace the YourName string by your First_LastName ☺). Click on **Create mobile app**.



15. Open your text editor and paste the following XML :

```
<GENERAL>
  <AUTO_UPDATE>true</AUTO_UPDATE>
</GENERAL>

<NATIVEAGENT>
  <DTXLogLevel>debug</DTXLogLevel>
  <DTXApplicationID>Application ID</DTXApplicationID>
  <DTXAgentEnvironment>Agent Environment</DTXAgentEnvironment>
  <DTXClusterUrl>Cluster Url</DTXClusterUrl>

  <platform name="android">
    <DTXAllowFileCookies>true</DTXAllowFileCookies>
  </platform>
  <platform name="ios">
  </platform>
</NATIVEAGENT>

<JSAGENT>
  <!-- IMPORTANT TO USE jsInlineScript URL -->
  <url>https://XXX.com/api/v1/rum/jsInlineScript/APPLICATION-XXXXXX?Api-
Token=YOUR_TOKEN</url>
  <apitoken>YOUR_TOKEN</apitoken>
</JSAGENT>
```

16. In the XML you just pasted, replace the following strings with the values from your Dynatrace screen (click on the **Command line** tab):

- Application ID
- Agent Environment
- Cluster Url





Select method for instrumentation:

Gradle

Command line

Add Dynatrace OneAgent for Android:

Download the [OneAgent](#) for Android.

Start the Auto-Instrumentor for your Android app:

Create a **Dynatrace.properties** file with the contents below:

```
DTXApplicationID=93114022-293f-4a07-9240-12e358bb8b33
DTXAgentEnvironment=bf32559dsi
DTXClusterURL=https://bf-sprint.dynatracelabs.com
```

Call the following command to instrument your app on Linux systems:

```
instrument.sh apk=<AppName>.apk prop=Dynatrace.properties
```

Call the following command to instrument your app on Windows systems:

```
instrument.cmd apk=<AppName>.apk prop=Dynatrace.properties
```

You can now install and run your app using your IDE.

Instrumented APK: <PathToAPK>/<AppName>/dist/<AppName>-final.apk



For example, your XML should now look something like this:

```
<GENERAL>
  <AUTO_UPDATE>true</AUTO_UPDATE>
</GENERAL>

<NATIVEAGENT>
  <DTXLogLevel>debug</DTXLogLevel>
  <DTXApplicationID>93114022-293f-4a07-9240-12e358bb8b33</DTXApplicationID>
  <DTXAgentEnvironment> bf32559dsi</DTXAgentEnvironment>
  <DTXClusterUrl> https://bf-sprint.dynatracelabs.com</DTXClusterUrl>

  <platform name="android">
    <DTXAllowFileCookies>true</DTXAllowFileCookies>
  </platform>
  <platform name="ios">
  </platform>
</NATIVEAGENT>

<JSAGENT>
  <!-- IMPORTANT TO USE jsInlineScript URL -->
  <url>https://XXX.com/api/v1/rum/jsInlineScript/APPLICATION-XXXXXX?Api-
Token=YOUR_TOKEN</url>
  <apitoken>YOUR_TOKEN</apitoken>
</JSAGENT>
```

17. Save the file in the **easytravel-ionic** directory as the following name: **dynatrace.config**
18. Now we will define an agentless web application. Remember that an agentless web application is one for which the Javascript tag in the HTML pages is not injected on-the-fly by a server-side agent but rather injected before or during the build. In the context of an Ionic application, this is a single page application with the HTML page embedded in the application (i.e. the page is not served by a web server). This is why we need to have the Javascript agent injected during the build – hence the agentless application configuration.
Return to the **Deploy Dynatrace screen**. This time you will click on **Set up agentless or AMP monitoring**.





Deploy Dynatrace

Monitor all your real users and technologies by installing OneAgent

A single Dynatrace OneAgent monitors real user experience and all the technologies, services, and applications that run in your environment. Just install OneAgent. No configuration required. [More...](#)



[Start installation](#)



No access to your host? No worries!

Even if you can't install OneAgent on your web server, you can still monitor user actions and user satisfaction.

[Set up agentless or AMP monitoring](#)

[Set up PaaS integration](#)



Create a synthetic monitor

Synthetic monitors provide 24x7 visibility into the performance and availability of your web application.

[Create synthetic monitor](#)



Mobile app monitoring

Get insight into your native mobile applications. Monitor application usage and performance in context with your backend infrastructure.



Digital touchpoint monitoring Beta

By providing you with a set of open source libraries, OpenKit enables you to instrument the digital touchpoints in your environment that aren't

19. Enter an application name. For consistency, please use the following syntax: YourName_Ionic_Web (replace the YourName string by your First_LastName 😊😊). Click on **Add web application**.

Agentless real user monitoring

Agentless real user monitoring is an alternative approach to Dynatrace setup that's useful when you can't install OneAgent directly on your web or application servers. It works by inserting a JavaScript tag or snippet into the HTML of each of your application's pages. If you do have direct access to your web or application server, please use [standard OneAgent installation](#).

Type a name for your application:

YourName_Ionic_Web

[Add web application](#)

☒ My application relies on [Accelerated Mobile Pages \(AMP\)](#)

Dynatrace can configure a special web application type for you that provides support for AMP pages.

[View list of applications currently set up for agentless monitoring and corresponding JavaScript tags.](#)



20. Skip the next screen and go to the application dashboard by clicking on the **View application** link.

Applications set up for agentless monitoring

Copy this JavaScript tag and paste it into your application's HTML pages. It might take a few minutes till user actions arrive.
Note: For best results the script tag should be placed as the first JavaScript file in the page.

Waiting for user actions

YourName_Ionic_Web
Produce some user actions by triggering page loads within your application. [View application](#)

```
<script type="text/javascript"> var a=window;a.dT_7a.console$$$a.console.log("Duplicate agent injection detected, turning off redundant initConfig.");navigator.userAgent$$$0<=navigator.userAgent.indexOf("RuxitSynthetic") ...
```

13.2 kB | [Download](#) [Copy code snippet](#)

21. Edit the application settings by clicking on the ... button (top right) and clicking **Edit**.

22. In the **Application Settings** menu, select **Async requests and single page apps**. Enable the following:

- Capture fetch() requests
- Support for XMLHttpRequest
- Angular JS and Angular

Application settings
YourName_Ionic_Web

General
See your application setup

User actions
Generate consistent user action names

Conversion goals
Measure success against business goals

User tags
Use metadata to analyze user sessions

Async requests and single page a...
Select your frameworks

Content capture
Define resource monitoring

Setup
Agentless monitoring setup

Anomaly detection
Fine-tune the sensitivity and thresholds

Advanced setup
Fine-tune your real user monitoring

Async requests and single page apps
DynaTrace creates user actions when your app uses XMLHttpRequest or the Fetch API to load data from the server. The generic support is framework independent. [More...](#)

☒ Capture fetch() requests

☒ Support for XMLHttpRequest

▲ Fetch API support requires DynaTrace OneAgent version 1.119 or later!

JavaScript framework support
Some JavaScript frameworks provide high-level abstractions for sending web requests. DynaTrace can capture such abstracted web requests directly for the frameworks listed below. This approach typically results in user names that are more intuitive and accurate.

☒ AngularJS and Angular

AngularJS 1.x
AngularJS is supported out-of-the-box and instrumented automatically. No code changes are required.

Angular 2.x - 4.x
▲ Requires DynaTrace OneAgent version 1.127 or above
If you're using JSPM (systemjs) or Webpack, Angular is supported out-of-the-box and instrumented automatically. No code changes are required.
If you're bundling your application using a different method, you must insert the code snippet below into your application component to pass the *http* and the *Headers* objects.

Angular code snippet

```
if(typeof dT_!=="undefined" && dT_.initAngularNg)(dT_.initAngularNg(http,Headers);)
```

It's recommended that you insert the code snippet into the constructor of the application component. See examples below. [More...](#)

Don't forget to click on the **Save** button (bottom right – you might have to scroll down).

23. In the **Application Settings** menu, select **Setup**. In the **Rest API** section, click on **More...**



Application settings

YourName_Ionic_Web

General

See your application setup

User actions

Generate consistent user action names

Conversion goals

Measure success against business goals

User tags

Use metadata to analyze user sessions

Async requests and single page apps

Select your frameworks

Content capture

Define resource monitoring

Setup

Agentless monitoring setup

Anomaly detection

Fine-tune the sensitivity and thresholds

Advanced setup

Fine-tune your real user monitoring

Agentless real user monitoring setup

To setup agentless real user monitoring for your application, you need to manually insert a custom JavaScript tag into each of your application's HTML pages.

Insert monitoring code

There are different ways to insert the JavaScript code. Using the code snippet minimizes maintenance efforts and works best with most applications.

Code snippet

Requires that you manually insert a code snippet into the HTML head of each application page. The snippet automatically loads the latest JavaScript code version and configuration. You never need to update the code snippet.

```
<script type="text/javascript"> var a=window;a.T?a.console&&a.console.log("Duplicate agent injection detected, turning off redundant initConfig");navigator.userAgent&&0<=navigator.userAgent.indexOf("RuxitSynthetic")||navigato ...
```

13.2 kB | Download

Copy code snippet

Rest API

The Dynatrace API enables you to insert the Dynatrace real user monitoring JavaScript code using your build scripts.

More...

24. Copy the inline code and paste it in your **dynatrace.config** XML file, replacing the URL in the JSAGENT section. Then click on the **create an API key** link.

Rest API

The Dynatrace API enables you to insert the Dynatrace real user monitoring JavaScript code using your build scripts. To begin, **create an API key**. Then replace the YOUR_TOKEN placeholders in the REST API calls below with your API key.

- Retrieves the JavaScript tag.
[More...](#)
 [Copy](#)
- Returns the async code snippet.
[More...](#)
 [Copy](#)
- Returns the inline code.
Inserting the full code inline has less impact on page load time than the JavaScript tag and provides full visibility into XHR calls.
[Show less...](#)
 [Copy](#)

For full details, see [Dynatrace help](#).
[Show less...](#)

25. Enter an API key name. For consistency, please use your name as the API key name. Click **Generate**.

Dynatrace API

You can use our API to export Dynatrace monitoring data into your 3rd party reporting and analysis tools. Multiple API tokens

My Dynatrace API tokens

Generate a secure access API token that enables access to your Dynatrace monitoring data via our REST-based API.

Use the switches below to define the access scope of your Dynatrace API token.

☒ Access problem and event feed, metrics, and topology

☒ Access logs

☒ Configure maintenance windows

[Generate](#) [Cancel](#)



26. You should now have token under your name added to the list of available token. Expand your token view and copy the generated token value

Generated token

tK1RX__XTI2s3w2aHmtpZ

Copy

Token name

YourName

Use the switches below to define the access scope of your Dynatrace API token.

☒ Access problem and event feed, metrics, and topology

☒ Access logs

☐ Configure maintenance windows

Save Cancel

27. Paste the token value in the **dynatrace.config** file, replacing **YOUR_TOKEN** in both the **url** and the **apitoken** tags of the **JSAGENT** section. The XML of your file should now look similar to this :

```
<GENERAL>
  <AUTO_UPDATE>true</AUTO_UPDATE>
</GENERAL>

<NATIVEAGENT>
  <DTXLogLevel>debug</DTXLogLevel>
  <DTXApplicationID>93114022-293f-4a07-9240-12e358bb8b33</DTXApplicationID>
  <DTXAgentEnvironment> bf32559dsi</DTXAgentEnvironment>
  <DTXClusterUrl> https://bf-sprint.dynatracelabs.com</DTXClusterUrl>

  <platform name="android">
    <DTXAllowFileCookies>true</DTXAllowFileCookies>
  </platform>
  <platform name="ios">
  </platform>
</NATIVEAGENT>

<JSAGENT>
  <!-- IMPORTANT TO USE jsInlineScript URL -->

  <url>https://cnz35449.sprint.dynatracelabs.com/api/v1/rum/jsInlineScript/APPLICATIO
N-F2C94FAB5A047EFD?Api-Token=tK1RX__XTI2s3w2aHmtpZ</url>
  <apitoken>tK1RX__XTI2s3w2aHmtpZ</apitoken>
</JSAGENT>
```

28. Save your **dynatrace.config** file.
29. Return to your command line interface and rebuild and run the **easyTravel Ionic** application by executing this command : **ionic cordova run android**
- (Optional) For iOS, use this command **ionic cordova run ios**



30. Navigate into the application and take a look at the following dashboards in Dynatrace:

- Application -> Your_Name_Ionic_Native
- Application -> Your_Name_Ionic_Web
- User Sessions

