dynatrace

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.

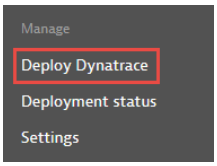
You have seen the easyTravel native application. We will now work with another flavor of easyTravel for mobile, an hybrid easyTravel app build on Ionic/Cordova.

Instructions

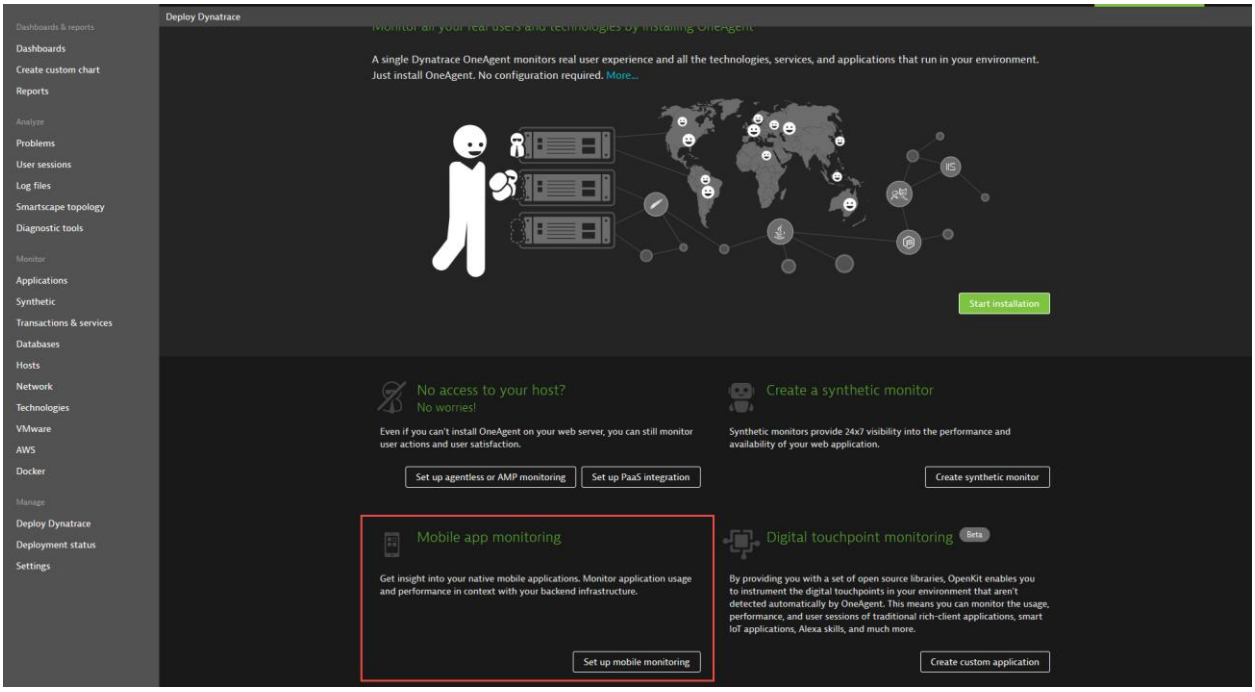
1. From your command line interface, navigate to the **Dynatrace-Ionic-Monitoring** directory
2. Execute the following command : **npm install**
   - This downloads and installs the easyTravel Ionic application project dependencies from the NPM repository
3. Execute the following command : **ionic cordova 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-XXXXX?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
```

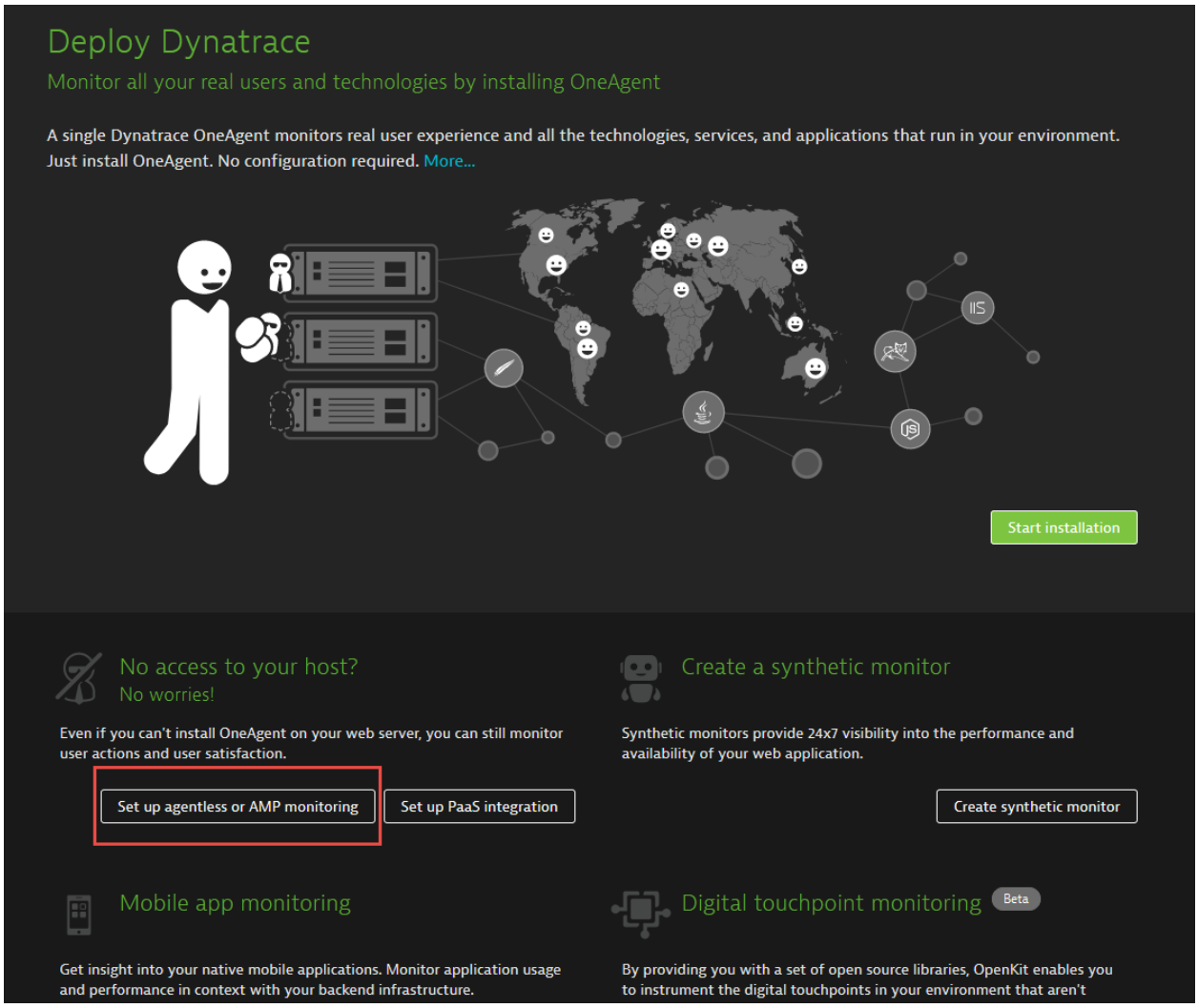
You can now install and run your app using your IDE.

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

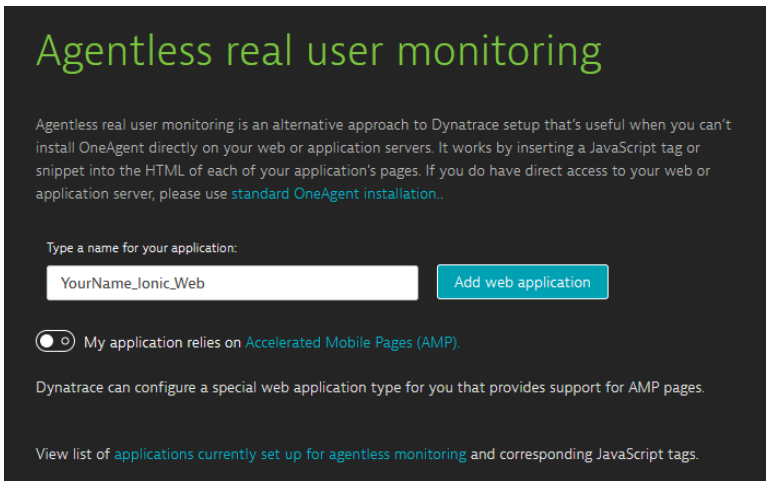
For example, your XML should now look something like this:

```
<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-XXXXX?Api-
Token=YOUR_TOKEN</url>
    <apitoken>YOUR_TOKEN</apitoken>
</JSAGENT>
```

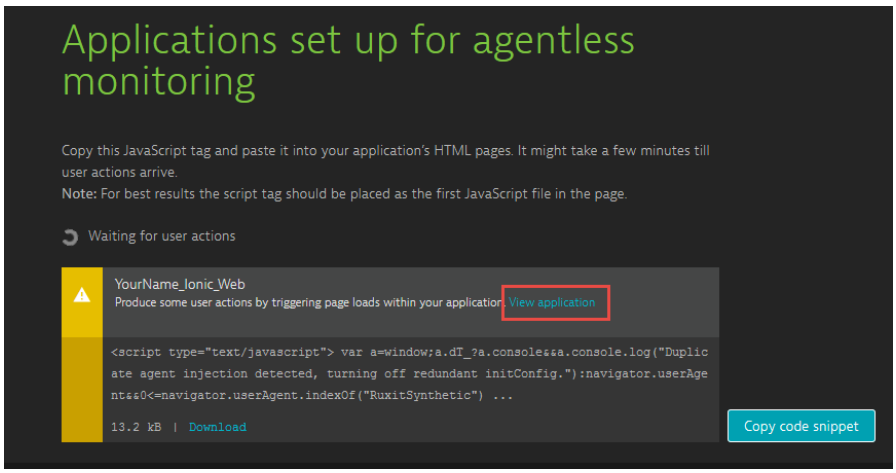
17. Save the file in the **easytravel-ionic** directory as the following name: **dynatrace.config**
18. Now we will define an agentless web application. Remember that an agentless web application is one for which the Javascript tag in the HTML pages is not injected on-the-fly by a server-side agent but 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**.



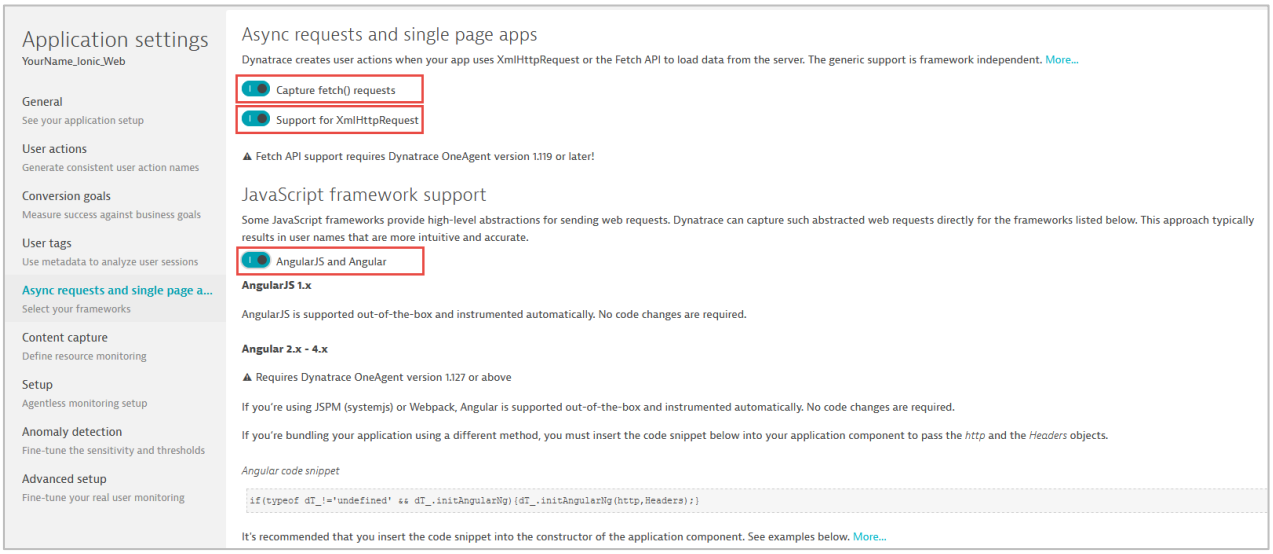
19. Enter an application name. For consistency, please use the following syntax: YourName_Ionic_Web (replace the YourName string by your First_LastName ☺☺). Click on **Add web application**.



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

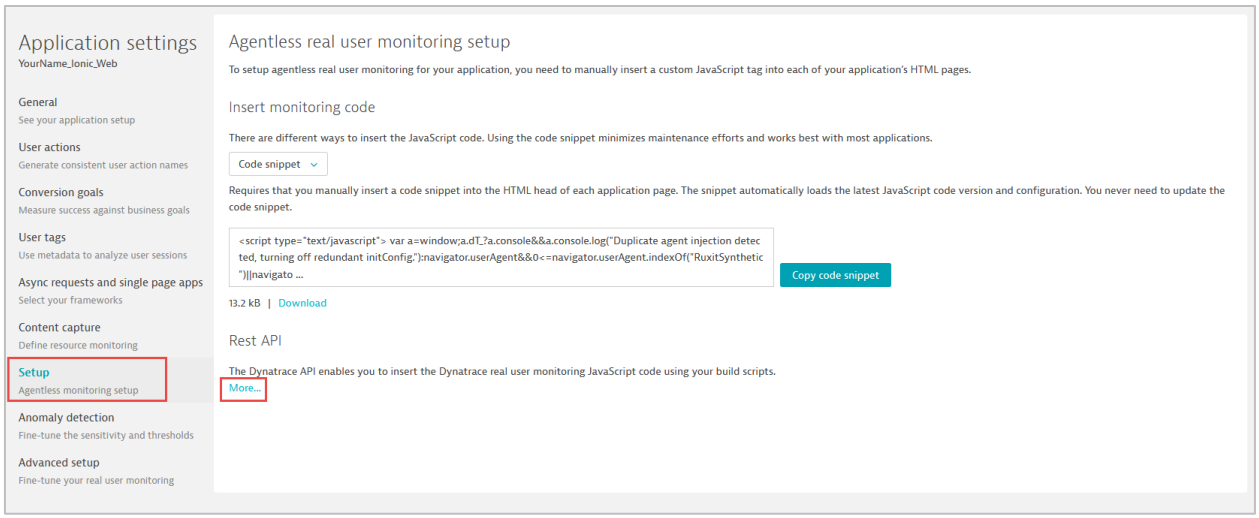


21. Edit the application settings by clicking on the **…** button (top right) and clicking **Edit**.
22. In the **Application Settings** menu, select **Async requests and single page apps**. Enable the following:
    - Capture fetch() requests
    - Support for XmlHttpRequest
    - Angular JS and Angular

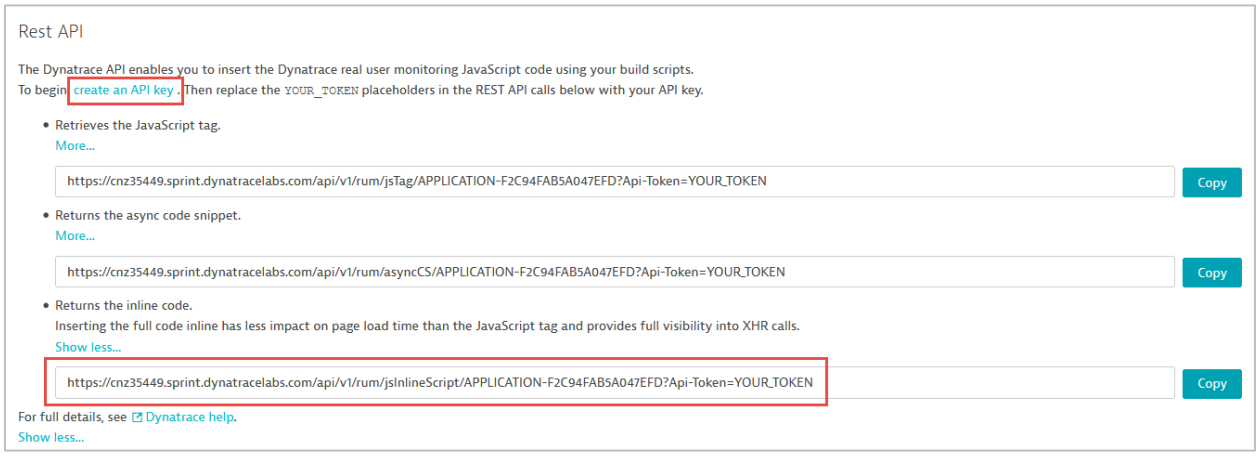


Don't forget to click on the **Save** button (bottom right – you might have to scroll down).

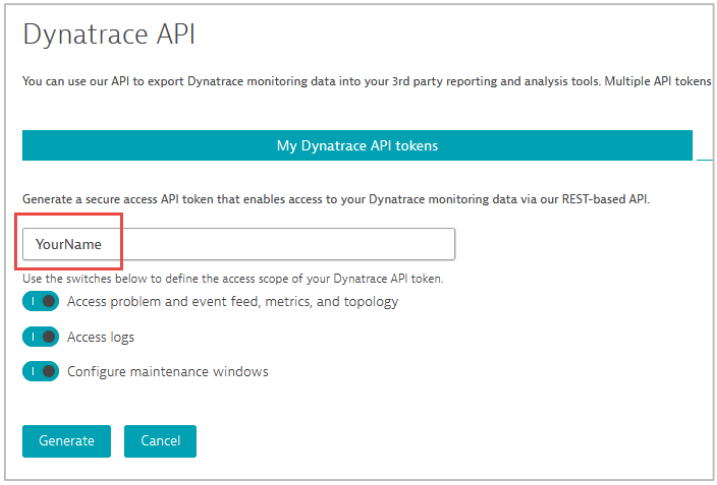
23. In the **Application Settings** menu, select **Setup**. In the **Rest API** section, click on **More…**



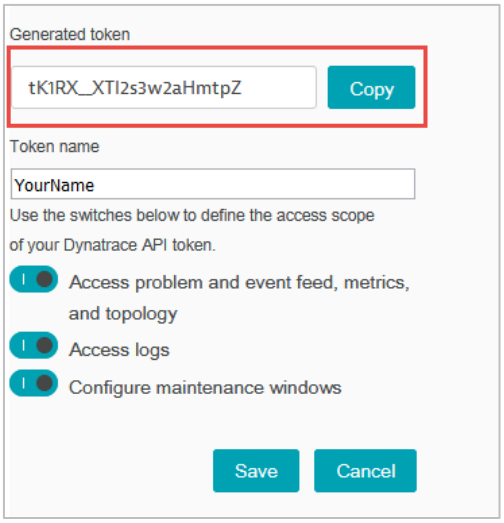
24. Copy the inline code and paste it in your **dynatrace.config** XML file, replacing the URL in the JSAGENT section. Then click on the **create an API key** link.



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



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



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