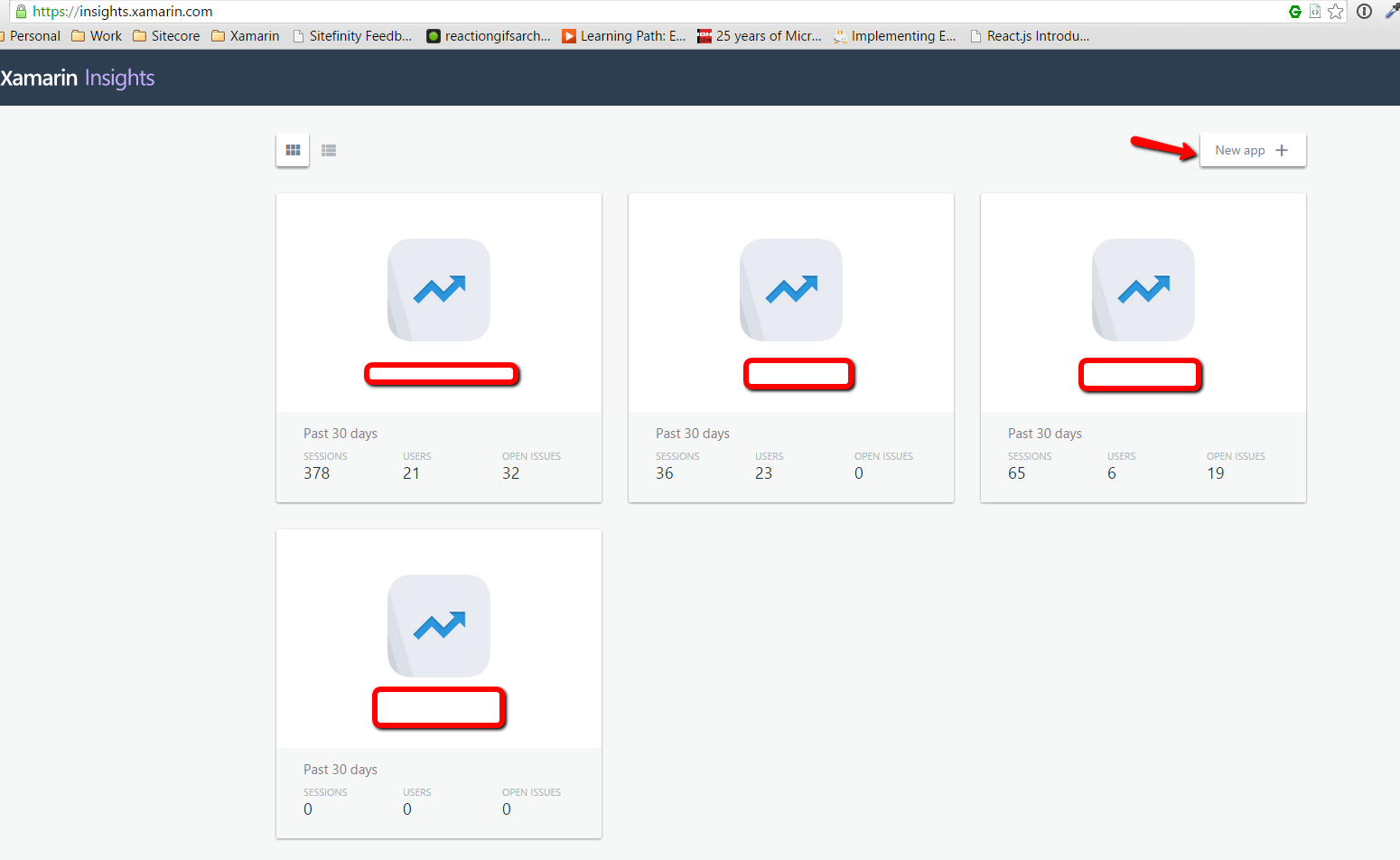
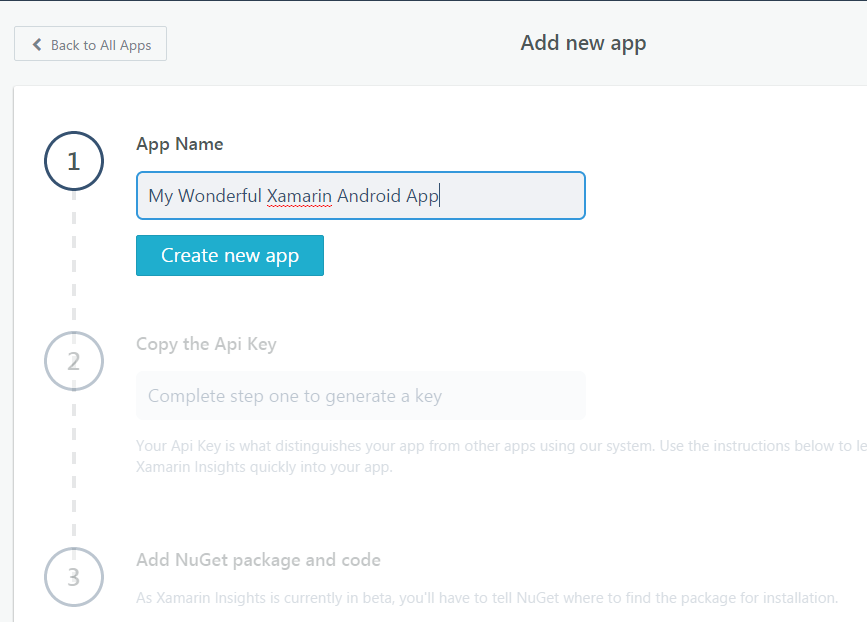
Day 30: Monitoring your Xamarin.Android applications using Xamarin Insights

Monitoring how well your app is doing is an absolute necessary thing now a days. Even the small company / indie developers are actually competing for app usage time with big companies like Facebook, Twitter, etc. So if your app crashes and you wait for your users to tell your about it, chances are your users will never ever come back to your app. Xamarin provides a wonderful tool called [Xamarin Insights](http://xamarin.com/insights) for monitoring your mobile applications (be it be Android or iOS). In today’s post we will look at how Xamarin Insights helps us monitor our Xamarin.Android app.

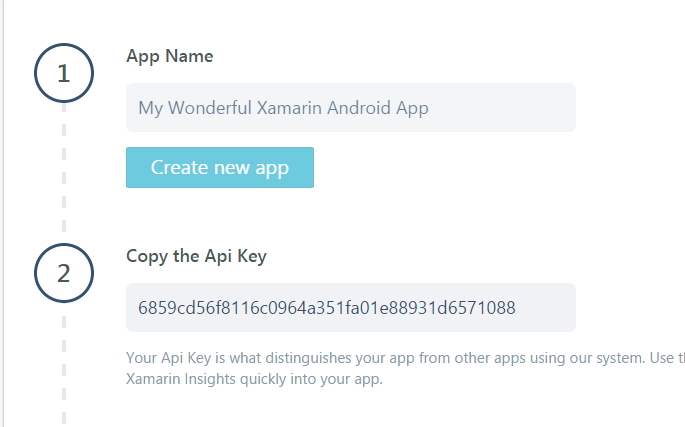
To get started with Xamarin Insights, you need to first create a new app in the Xamarin Insights [portal](https://insights.xamarin.com/)



In the “Add new App” screen, we have to define the “App Name” and click “Create new app” button.



Once we create the new app, we will be presented with the “Api Key”. We will need this Api Key to communicate with the Xamarin Insights Portal –



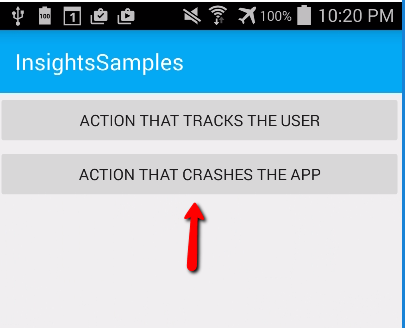
And finally, back on our Xamarin.Android application, we need to add the Xamarin Insights Nuget Package

|  |
| --- |
| PM> Install-Package Xamarin.Insights |

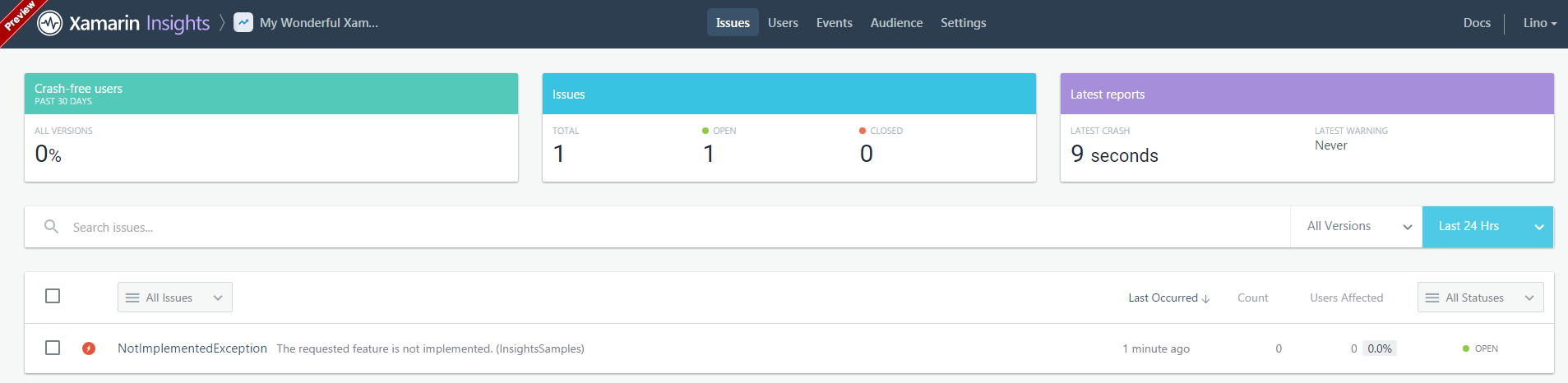
After the package is done installing, we need to connect our Xamarin.Android App with the Insights portal using the Api Key from above. To do this, we will use the Insights.Intialize api from our Xamarin.Android OnCreate method of the main launcher activity –

|  |
| --- |
| protected override void OnCreate(Bundle bundle)  {  base.OnCreate(bundle);  Insights.Initialize("6859cd56f8116c0964a351fa01e88931d6571088", this);  } |

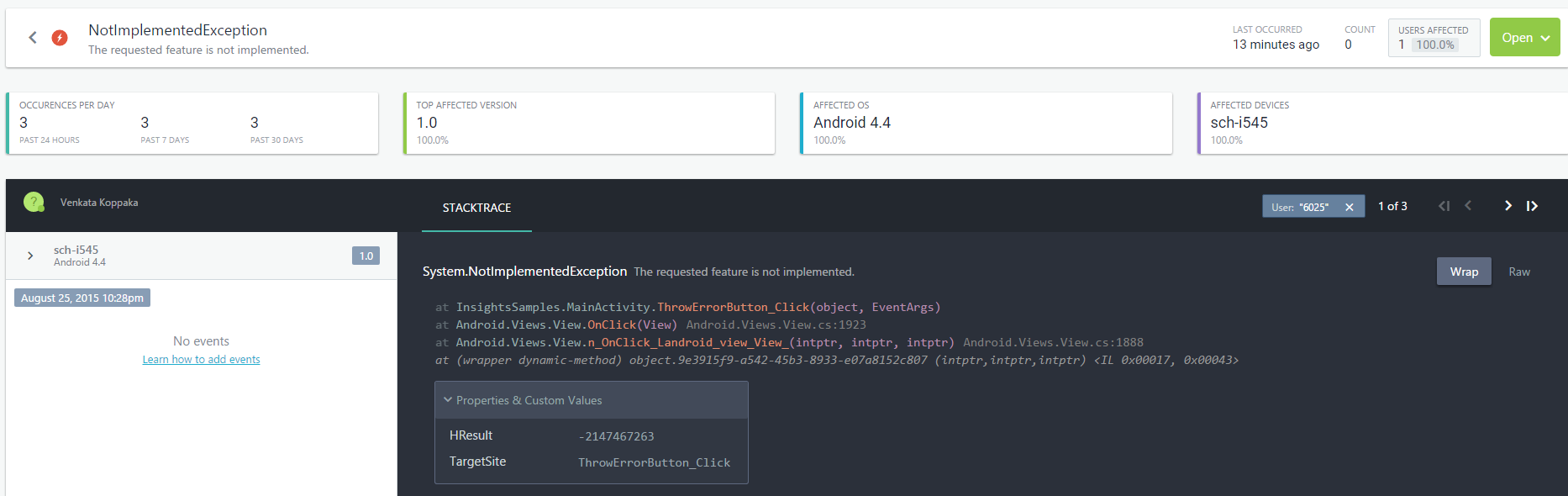
And that’s it. Your Android application is now monitored by Xamarin Insights. To see how it all works, let’s launch our application and do some action that makes the app crash.



I intentionally added a button that causes a crash by throwing a NotImplementedException. And as soon as I click the button, the Xamarin Insights Portal shows the crash



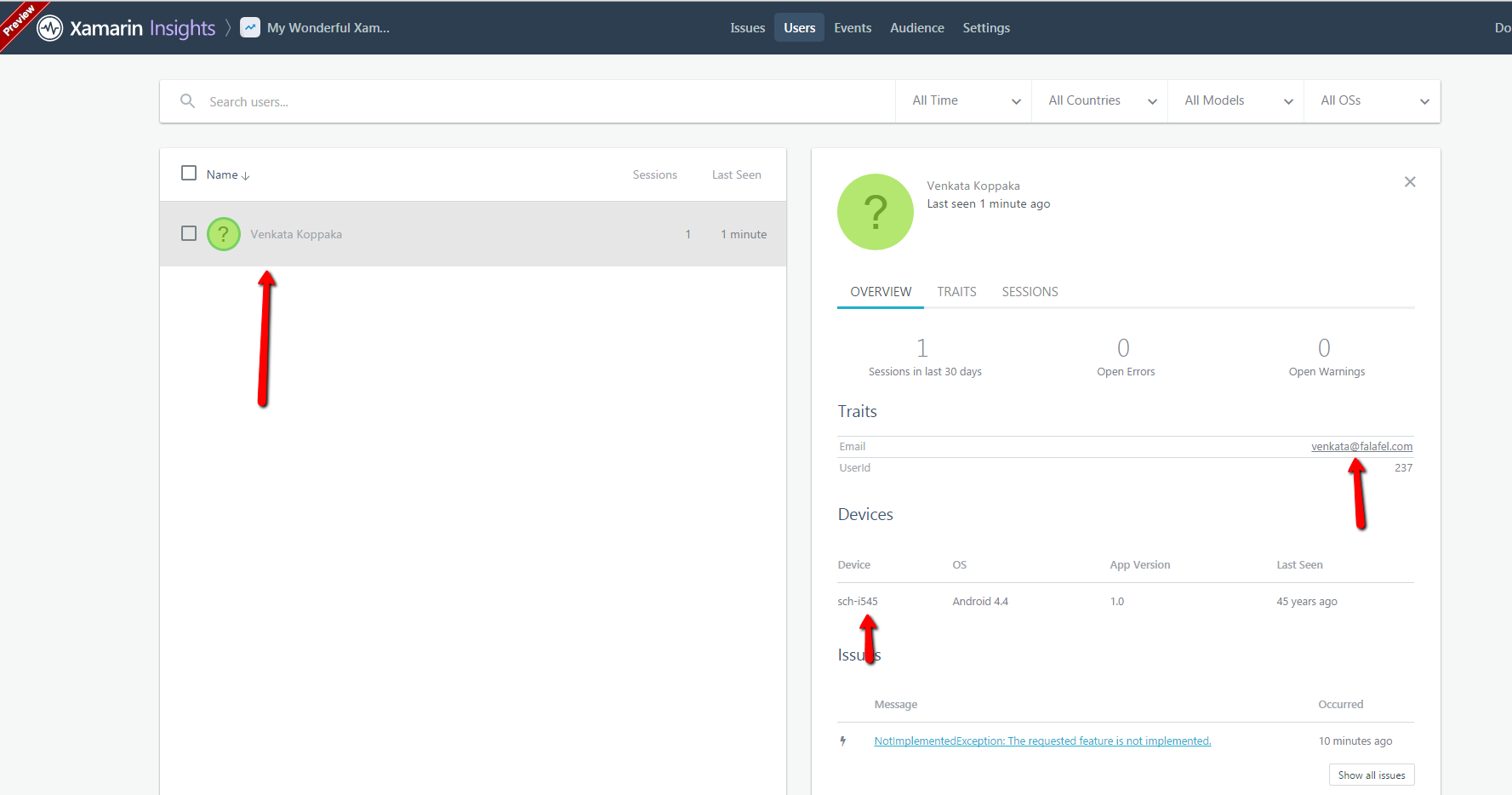
And if we drill down into the details of the crash, we get full .NET Exception stack trace, Device information, when it happened, how many users are experiencing this crash etc –



Insights can also identify the user (if tracked) that experienced the crash. If no user information is specified, Insights portal just identifies the user as “Guest”. To track the user more intelligently, we can use the following snippet –

|  |
| --- |
| private void TrackUserButton\_Click(object sender, EventArgs e)  {  var userAttributes = new Dictionary<string, string>  {  {"Email", "venkata@falafel.com"},  {"UserId", "237"}  };  Insights.Identify("Venkata Koppaka", userAttributes);  } |

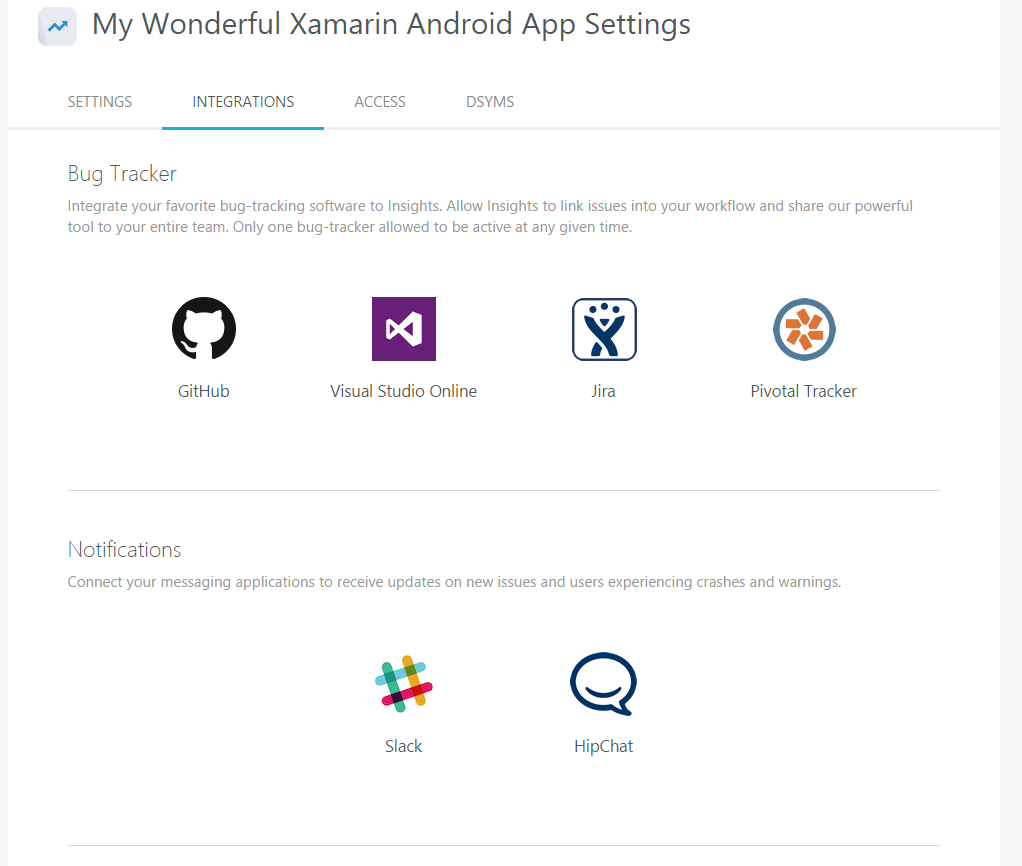
The Insights.Identify call tracks the user and each tracked user can have additional “Traits” that you can store in the insights portal. In the snippet above, I am specifying my email and UserId as additional Traits. The result of executing this code would look like the following in the Insights portal –



Xamarin Insights is extremely smart about when it communicates to the Insights portal about the crashes and other information, if there is slow network, low battery scenarios, Insights handles them well. But, there might be cases where you want the data transmission to always happen right away. Xamarin Insights supports this functionality as well when you set the ForceDataTransmission property to true.

|  |
| --- |
| Insights.ForceDataTransmission = true; |

On top of fantastic user and crash analytics and monitoring, Xamarin Insights provides options to automatically send emails whenever a crash occurs, and it also provides integrations with Slack and Hipchat for notify right away in an event of crash. Insights also can automatically create bugs for crashes on Github, Visual Studio Online, Jira etc.



Xamarin Insights is currently in beta and I can only assume that the product will improve immensely as it marches towards V1.

That’s it for today, see you all tomorrow.