

# How To Install Firebase Crashlytics

Author: [Ryuichi.X.Ishitsuka@sony.com](mailto:Ryuichi.X.Ishitsuka@sony.com)

## Firebase Consoleにプロジェクトを作成する

Firebase Consoleにログインするためには、Googleアカウントが必要です。本項は、Googleアカウントを取得しており、Firebase Consoleにログイン済みであるものといたします。

1. ブラウザで、[Firebase Console](#)を開き、Projectの追加をクリックします。



2. 必要事項を入力して、プロジェクトを作成をクリックします。

本項では、iem-clashlytics-researchという名前のプロジェクトを作成します。



プロジェクトの作成が完了すると以下のように、当該プロジェクトへのリンクが、メニューに表示されます。



なお、これ以降は、本項で作成したiem-clashlytics-researchプロジェクト上に、iOS/Androidサンプルアプリのクラッシュレポートを収集していく流れで、説明させていただきます。

## AndroidアプリにCrashlytics導入する

本項では、AndroidアプリにCrashlyticsを導入する手順をサンプルアプリを通して説明します。

## アプリにCrashlyticsのライブラリを追加する

### ■ build.gradle

```
1 // Top-level build file where you can add configuration options
  common to all sub-projects/modules.
2
3 buildscript {
4
5     repositories {
6
7         // Add repository
7 +         maven {
8 +             url 'https://maven.fabric.io/public'
9 +         }
10
11         google()
12         jcenter()
13     }
14     dependencies {
15
16         classpath 'com.android.tools.build:gradle:3.2.1'
17
18         // Check for v3.1.2 or higher
19 +         classpath 'com.google.gms:google-services:4.1.0'
20
21         // Add dependency
22 +         classpath 'io.fabric.tools:gradle:1.25.4'
23
24     }
25 }
26
27
28 allprojects {
29     repositories {
30
31         // Add repository
32 +         maven {
33 +             url 'https://maven.fabric.io'
34 +         }
35
36         google()
37         jcenter()
38     }
39 }
40
41 task clean(type: Delete) {
42     delete rootProject.buildDir
43 }
44
```

### ■ app/build.gradle

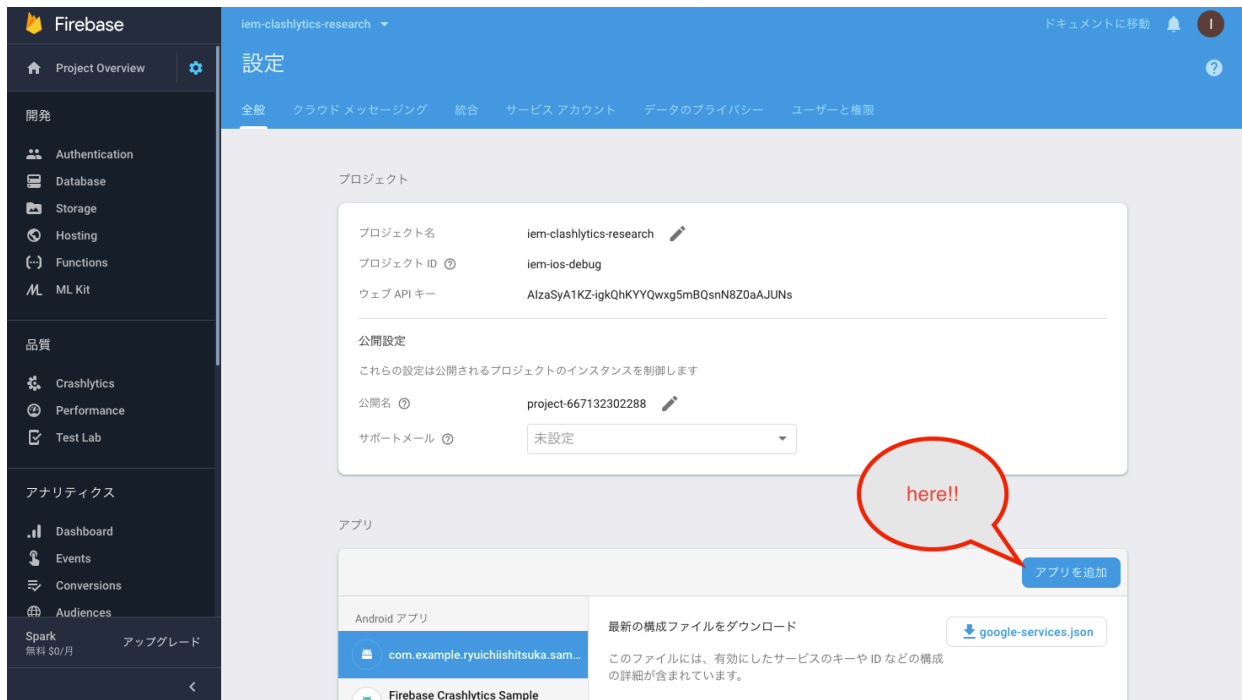
```

1  apply plugin: 'com.android.application'
2  + apply plugin: 'io.fabric'
3  + apply plugin: 'com.google.gms.google-services'
4
5  android {
6      compileSdkVersion 28
7      defaultConfig {
8          applicationId
9          "com.sips.firebase.crashlytics.sample.firebasecrashlyticssample"
10         minSdkVersion 22
11         targetSdkVersion 28
12         versionCode 1
13         versionName "1.0"
14         testInstrumentationRunner
15         "android.support.test.runner.AndroidJUnitRunner"
16     }
17     buildTypes {
18         release {
19             minifyEnabled false
20             proguardFiles getDefaultProguardFile('proguard-
21             android.txt'), 'proguard-rules.pro'
22         }
23     }
24 }
25
26 dependencies {
27     implementation fileTree(dir: 'libs', include: ['*.jar'])
28     implementation 'com.android.support:appcompat-v7:28.0.0'
29     implementation 'com.android.support.constraint:constraint-
30     layout:1.1.3'
31     testImplementation 'junit:junit:4.12'
32     androidTestImplementation 'com.android.support.test:runner:1.0.2'
33     androidTestImplementation
34     'com.android.support.test.espresso:espresso-core:3.0.2'
35
36 +    // Check for v11.4.2 or higher
37 +    implementation 'com.google.firebase:firebase-core:16.0.4'
38
39 +    // Add dependency
40 +    implementation 'com.crashlytics.sdk.android:crashlytics:2.9.5'
41 }

```

*Firestore Console*上に作成したプロジェクトにアプリを登録する

当作業は、Firebase Console > 対象プロジェクト(iem-clashlytics-research) > プロジェクトの設定 > アプリを追加を開いて行います。



当画面には、手順も併記されていますので、記載どおりに作業をします。また、この作業は、基本的に、Android Package名が変わることがなければ、一度のみ行なえばよいです。

Step 1. 必要事項を入力してアプリを登録します。

### × Android アプリに Firebase を追加

- 1 アプリの登録
  - Android パッケージ名 ⓘ
  - アプリのニックネーム (省略可) ⓘ
  - デバッグ用の署名証明書 SHA-1 (省略可) ⓘ
  - Auth で Dynamic Links、Invites、Google ログイン、電話番号をサポートするために必須です。[設定] で SHA-1 を編集してください。
  -
- 2 設定ファイルのダウンロード
- 3 Firebase SDK の追加
- 4 アプリを実行してインストールを確認

Step 2. google-services-json ファイルを取得して、Android プロジェクトに配置します。

## × Android アプリに Firebase を追加



### アプリの登録

Android パッケージ名: com.sips.firebase.crashlytics.sample.firebasecrashlyticssample、アプリのニックネーム: Firebase Crashlytics Sample



### 設定ファイルのダウンロード

Android Studio については下記参照 | [Unity C++](#)

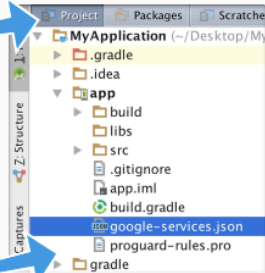
📄 google-services.json をダウンロード

Android Studio の [Project] 表示に切り替え、プロジェクトのルート ディレクトリを表示します。

ダウンロードした google-services.json ファイルを Android アプリ モジュールのルート ディレクトリに移動します。



google-services.json



前へ

次へ



### Firebase SDK の追加



### アプリを実行してインストールを確認

Step 3. Firebase SDKは追加済みなので先へ進みます。

3

## Firebase SDK の追加

Gradle の手順 | [Unity C++](#)

[Gradle](#) 用の Google サービス プラグインは、ダウンロードした `google-services.json` ファイルを読み込みます。このプラグインを使用するよう、`build.gradle` ファイルを修正してください。

プロジェクトレベルの `build.gradle` (<project>/`build.gradle`) :

```
buildscript {  
    dependencies {  
        // Add this line  
        classpath 'com.google.gms:google-services:4.0.1'  
    }  
}
```



アプリレベルの `build.gradle` (<project>/<app-module>/`build.gradle`) :

```
dependencies {  
    // Add this line  
    implementation 'com.google.firebase:firebase-core:16.0.1'  
}  
...  
// Add to the bottom of the file  
apply plugin: 'com.google.gms.google-services'
```



デフォルトでアナリティクスを含める ⓘ

最後に、IDE に表示されるバーの **[Sync now]** をクリックします。

Gradle files have changed since

**Sync now**

前へ

次へ

4

アプリを実行してインストールを確認

Step 4. 初期状態では、Waiting していますので、Android Studio 上で、アプリを Run します。

## × Android アプリに Firebase を追加



### アプリの登録

Android パッケージ名: com.sips.firebase.crashlytics.sample.firebasecrashlyticssample、アプリのニックネーム: Firebase Crashlytics Sample



### 設定ファイルのダウンロード



### Firebase SDK の追加



### アプリを実行してインストールを確認

🔗 アプリがサーバーと通信したかどうかを確認中です。場合によってはアプリのアンインストールと再インストールが必要です。

前へ

コンソールに進む

[このステップをスキップ](#)

アプリが起動されたら、以下のようなメッセージが表示されます。切り替わらない場合は、複数回、アプリを起動し直してください。（インターネットに接続できる環境でアプリを起動する必要があります。）



## × Android アプリに Firebase を追加



### アプリの登録

Android パッケージ名: com.sips.firebase.crashlytics.sample.firebasecrashlyticssample、アプリのニックネーム: Firebase Crashlytics Sample



### 設定ファイルのダウンロード



### Firebase SDK の追加



### アプリを実行してインストールを確認



Firebase がアプリに正常に追加されました。

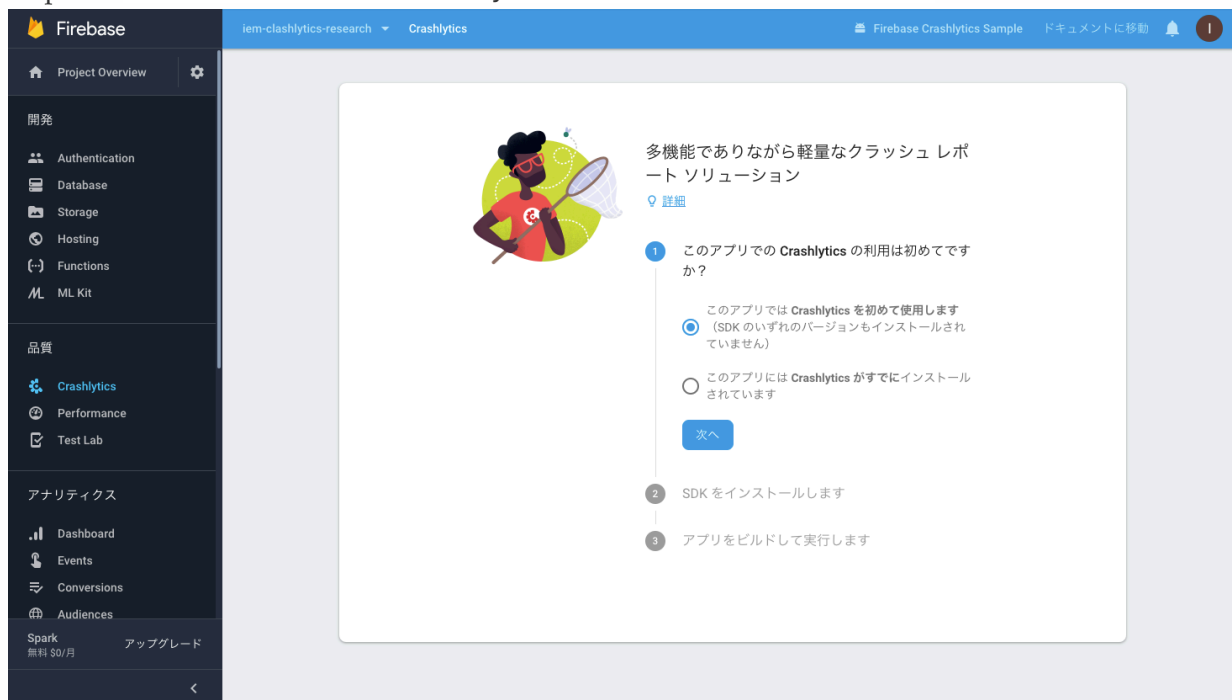
前へ

コンソールに進む

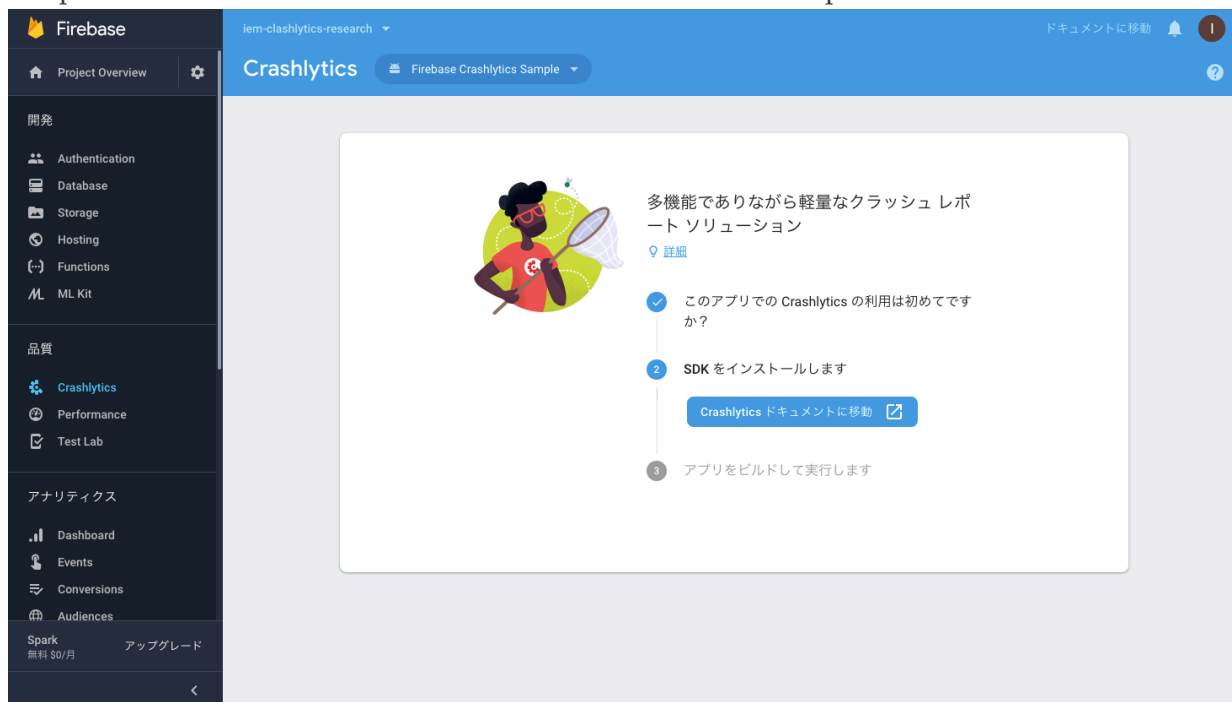
## *Firebase Console 上で Crashlytics の設定する*

当作業は、Firebase Console > 対象プロジェクト(iem-clashlytics-research) > 左メニューのCrashlyticsを開いて行います。

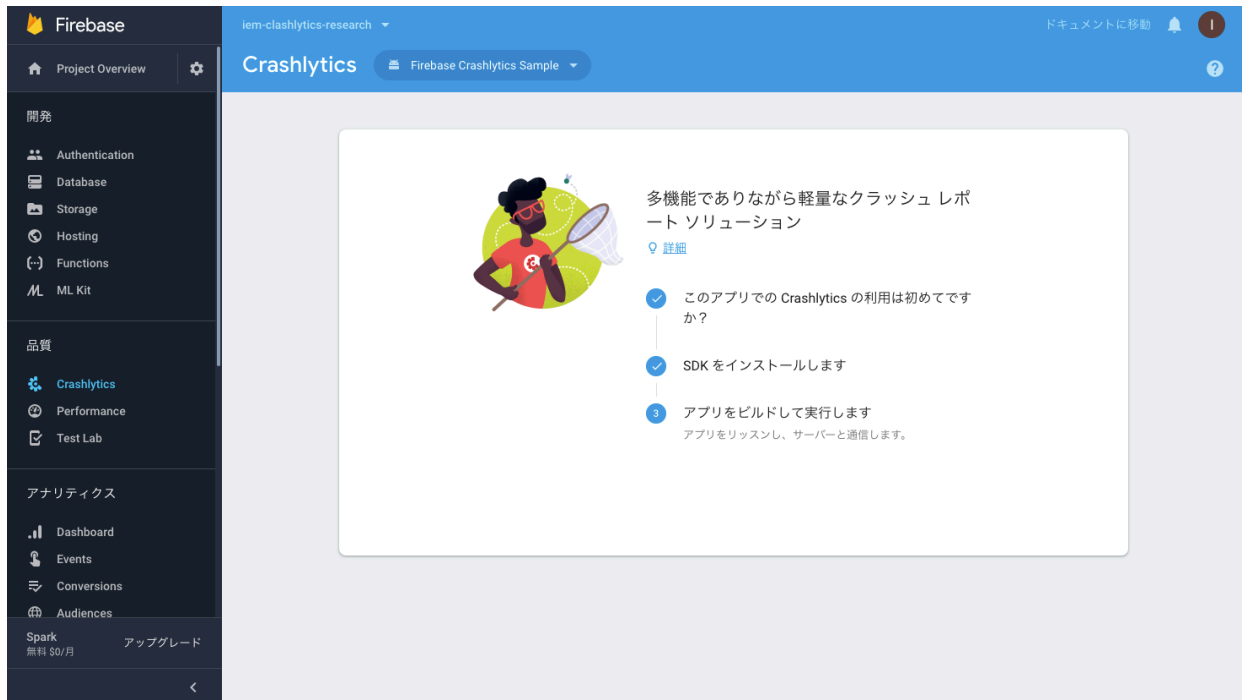
Step 1. このアプリでは Crashlytics を初めて使用しますを選択します。



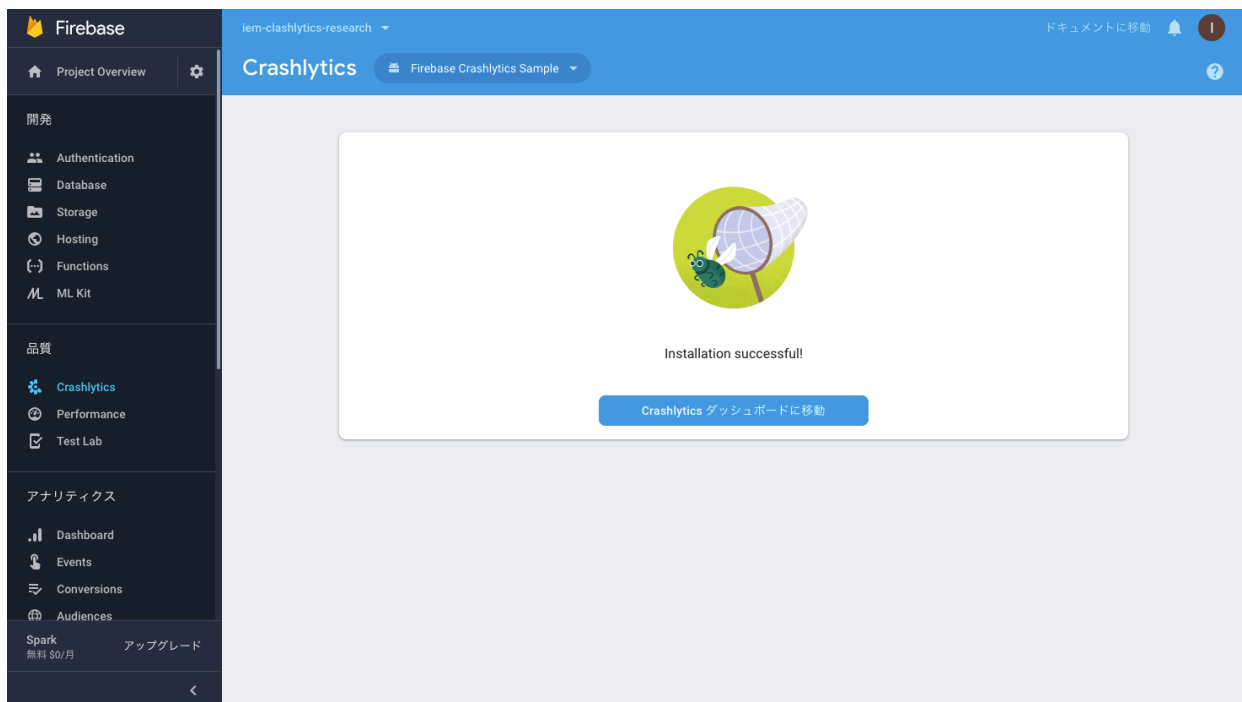
Step 2.すでに、SDKをインストール済みなので、Step 2は、スキップします。



Step 3. Android Studio 上で、アプリを Run します。



アプリが起動すると、以下の画面に切り替わります。切り替わらない場合は、複数回、アプリを起動してください。（インターネットに接続できる環境でアプリを起動する必要があります。）



## Crashレポートが送信されるか確認する

1. 強制的にクラッシュを引き起こすコードをアプリの任意のソースに追加します。

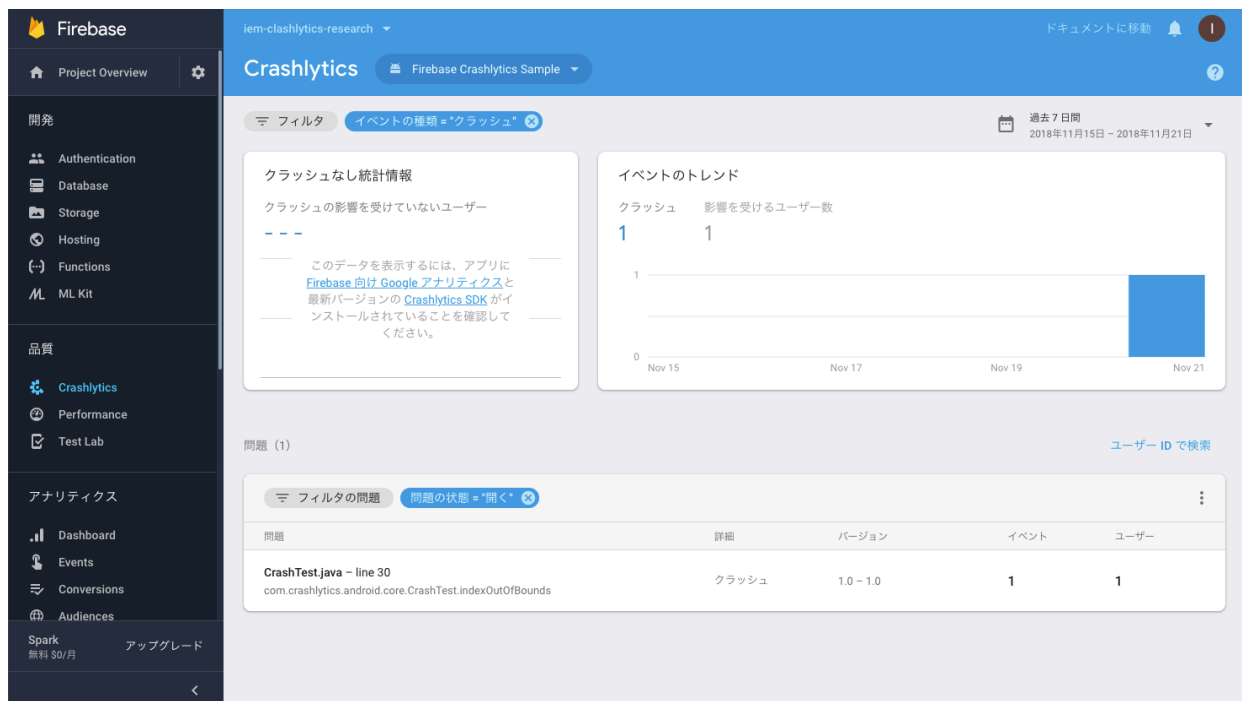
```
1 package
  com.sips.firebase.crashlytics.sample.firebasecrashlyticssample;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.os.Bundle;
5 import android.view.View;
```

```

6
7 + import com.crashlytics.android.Crashlytics;
8
9 public class MainActivity extends AppCompatActivity implements
  View.OnClickListener {
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_main);
15     }
16
17     @Override
18     public void onClick(View v) {
19 +         Crashlytics.getInstance().crash(); // Force a crash
20     }
21
22 }
23

```

2. アプリをRunし、クラッシュを引き起こすコードが実行されるよう操作します。
3. Firebase Console > 対象プロジェクト > Crashlytics > 対象アプリを開いて、Crashレポートがアップロードされていることを確認します。



## iOSアプリにCrashlyticsを導入する

本項では、iOSアプリにCrashlyticsを導入する手順を[サンプルアプリ](#)を通して説明します。

## アプリにCrashlyticsのライブラリを追加する

### ■ CocoaPods

- Podfileに以下を追加します。

```
1 pod 'Firebase/Core'
2 pod 'Fabric', '~> 1.7.11'
3 pod 'Crashlytics', '~> 3.10.7'
```

- XcodeのRun Scriptに以下を追加します。

```
1 ${PODS_ROOT}/Fabric/run
```

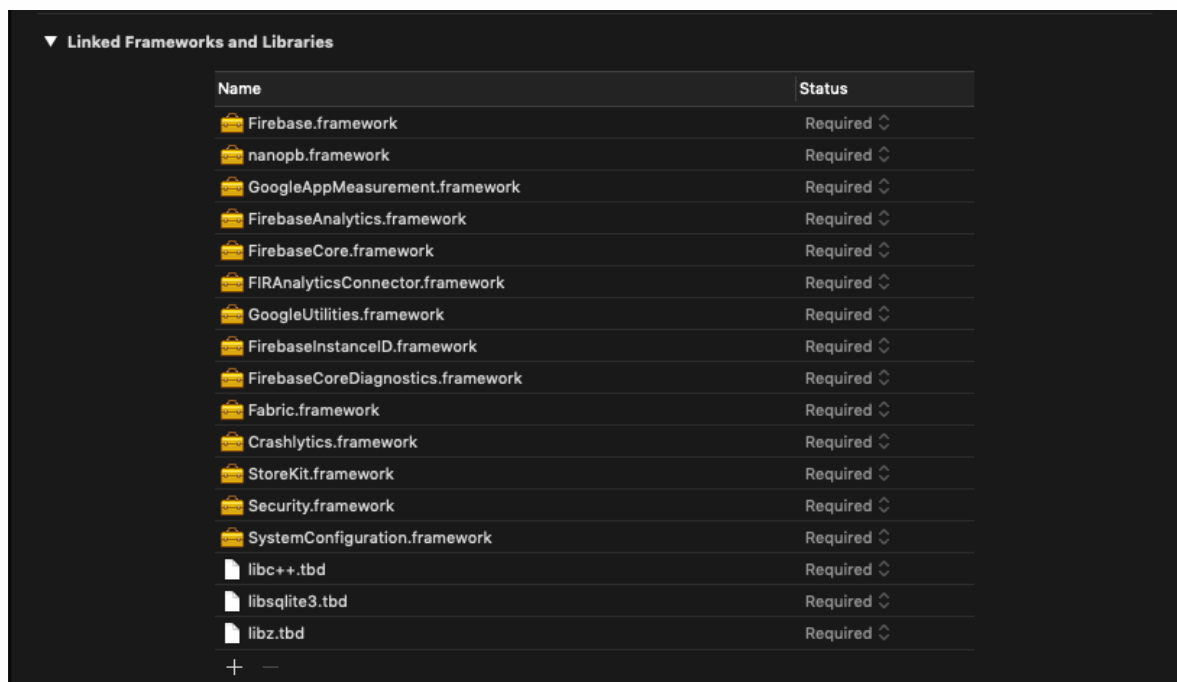
### ■ Carthage

- Cartfileに以下を追加します。

```
1 binary
  "https://dl.google.com/dl/firebase/ios/carthage/FirebaseAnalyticsBinary.json" == 5.13.0
2 binary "https://building42.github.io/Specs/Carthage/iOS/Fabric.json"
3 binary
  "https://building42.github.io/Specs/Carthage/iOS/Crashlytics.json"
```

- `carthage update --platform iOS`で取得した.frameworkをLinked Frameworks and Librariesに追加します。

(`/usr/local/bin/carthage copy-frameworks`のRun Scriptには追加しないようにしてください。)



- XcodeのRun Scriptに以下を追加します。

```
1 | $SRCROOT/Carthage/Build/iOS/Fabric.framework/run
```

- CocoaPods/Carthage、いずれの場合も、追加したRun ScriptのInput Filesに以下を追加します。(Xcode 10 Only)

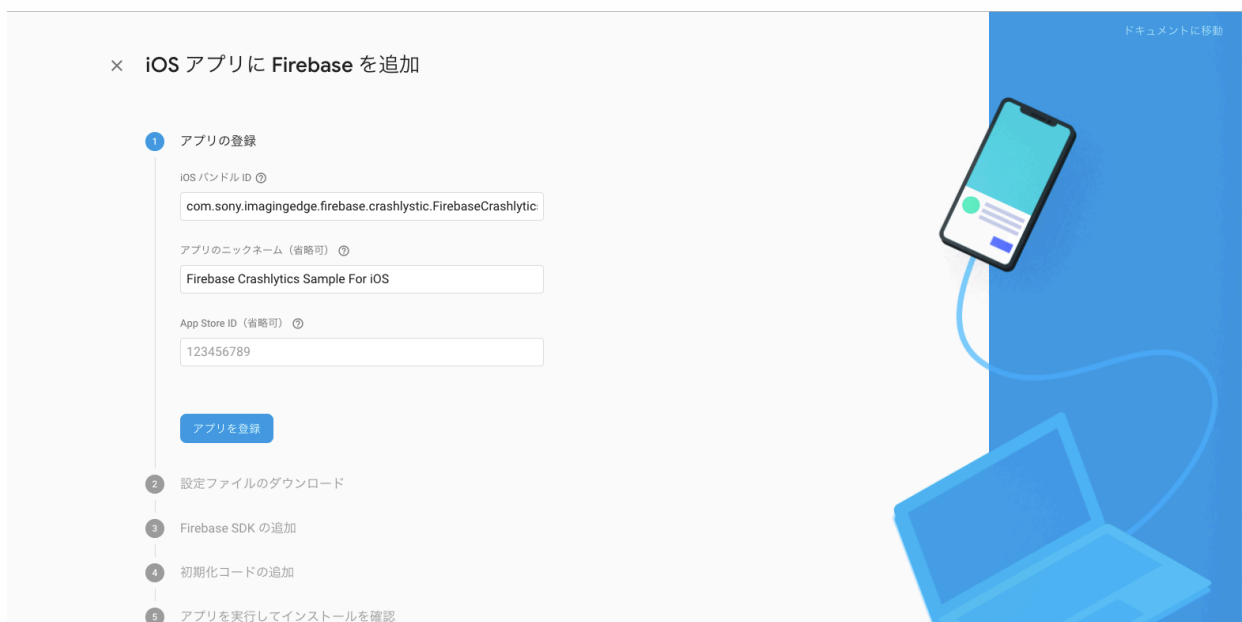
```
1 | $(BUILT_PRODUCTS_DIR)/$(INFOPLIST_PATH)
```

- 対象アプリのXcodeのBuild Settingsを以下のとおり変更します。
  - Debug Information FormatをDWARF with dSYM Fileに変更します。
  - Other Linker Flagsに-ObjCを追加します。

## Firestore Console上に作成したプロジェクトにアプリを登録する

Androidアプリと同様に、Firestore Console > 対象プロジェクト(iem-clashlytics-research) > プロジェクトの設定 > アプリを追加を開いて行います。この作業は、アプリのバンドルIDが同一の場合は、一度行えばよいです。

Step 1. 必要事項を入力してアプリを登録します。



Step 2. google-services-jsonファイルを取得して、アプリのXcodeプロジェクトに配置します。

× iOS アプリに Firebase を追加

1

アプリの登録

iOS バンドル ID: com.sony.imagingedge.firebase.crashlytic.FirebaseCrashlyticsSampleForiOS、アプリのニックネーム: Firebase Crashlytics Sample For iOS

2

設定ファイルのダウンロード

Xcode については下記参照 | [Unity](#) [C++](#)

GoogleService-Info.plist をダウンロード

ダウンロードした GoogleService-Info.plist ファイルを Xcode プロジェクトのルートに移動し、すべてのターゲットに追加します。



前へ [次へ](#)

3

Firebase SDK の追加

4

初期化コードの追加

ドキュメントに移動



Step 3. Firebase SDKは追加済みなので先へ進みます。

1

アプリの登録

iOS バンドル ID: com.sony.imagingedge.firebase.crashlytic.FirebaseCrashlyticsSampleForiOS、アプリのニックネーム: Firebase Crashlytics Sample For iOS

2

設定ファイルのダウンロード

3

Firebase SDK の追加

CocoaPods の手順 | [ZIP をダウンロード](#) [Unity](#) [C++](#)

Google サービスは [CocoaPods](#) を使用し、依存関係のインストールと管理を行います。ターミナルウィンドウを開き、アプリの Xcode プロジェクトの場所に移動してください。

Podfile がない場合は作成します:

```
$ pod init
```

Podfile を開いて以下を追加します:

```
pod 'Firebase/Core'
```

デフォルトでアナリティクスを含める

ファイルを保存し、次のコマンドを実行します。

```
$ pod install
```

これにより、アプリの .xcworkspace ファイルが作成されます。このファイルは、アプリで今後行うあらゆる開発で利用できます。

前へ [次へ](#)

4

初期化コードの追加

5

アプリを実行してインストールを確認

ドキュメントに移動

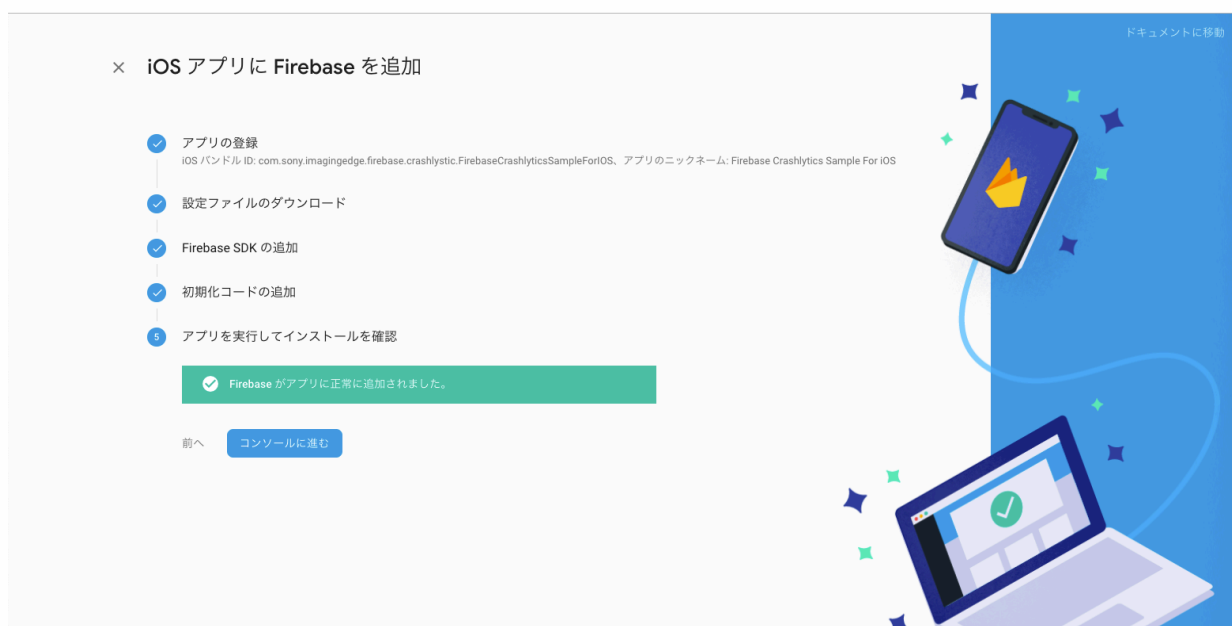


Step 4. 初期化コードをアプリに追加します。



Step 5. Xcode上で、アプリをRunします。

アプリが起動されたら、Waitingが解除され、以下のようなメッセージが表示されます。切り替わらない場合は、複数回、アプリを起動し直してください。（インターネットに接続できる環境でアプリを起動する必要があります。）

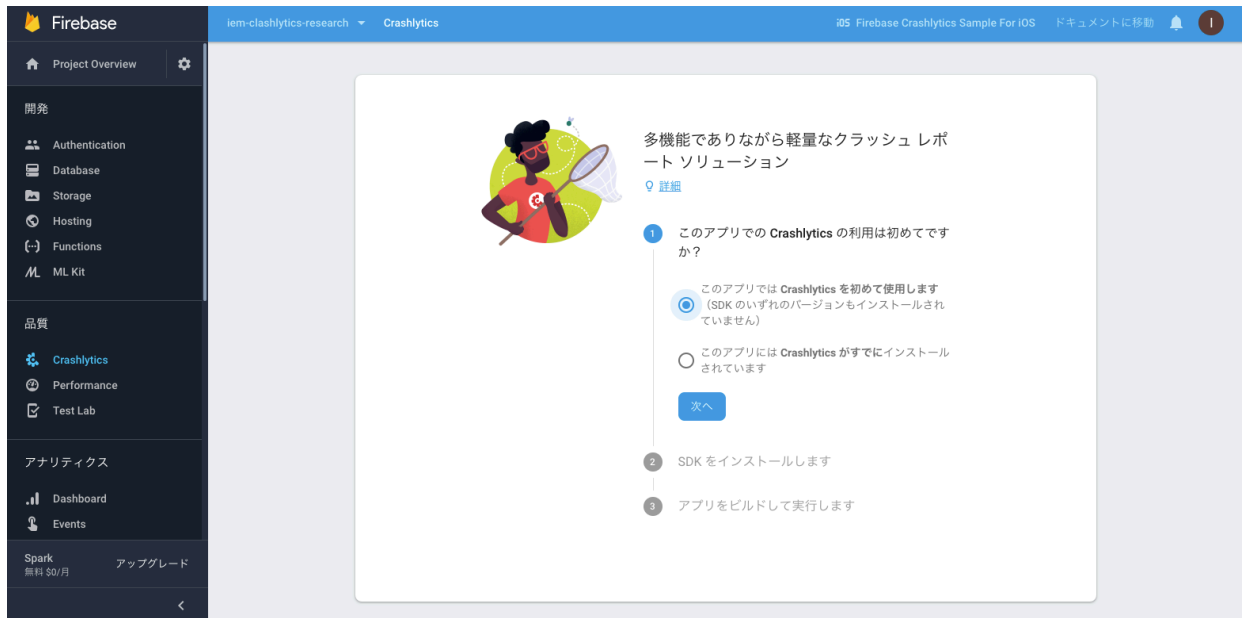


## Firestore Console上でCrashlyticsの設定する

Androidアプリと同様に、当作業は、Firestore Console > 対象プロジェクト(iem-clashlytics-research) > 左メニューのCrashlyticsを開いて行います。

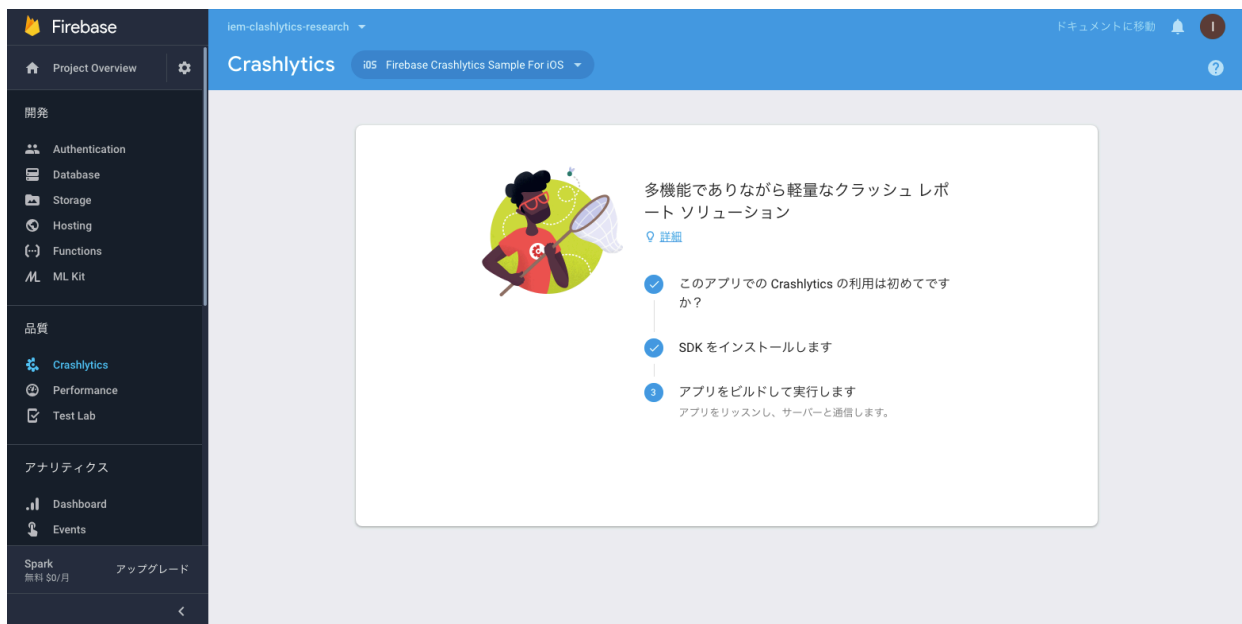
Step 1. このアプリではCrashlyticsを初めて使用しますを選択します。



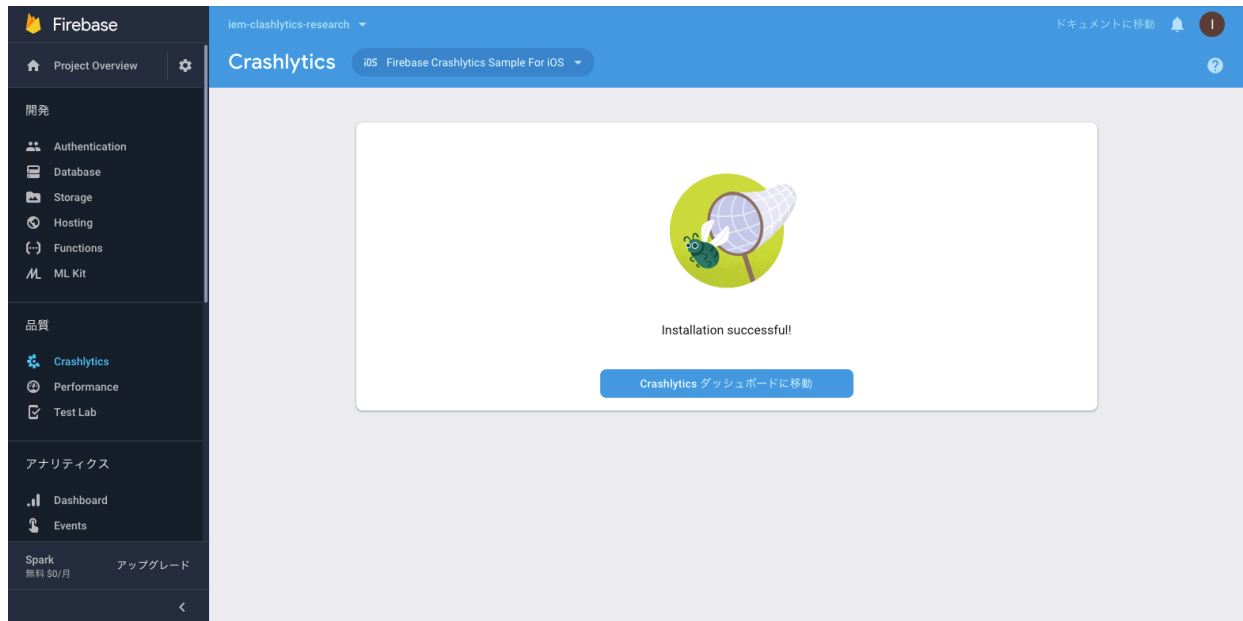


Step 2. すでに、SDKをインストール済みなので、Step 2は、スキップします。

Step 3. Xcode上で、アプリをRunします。



完了すると以下のような画面が表示されます。

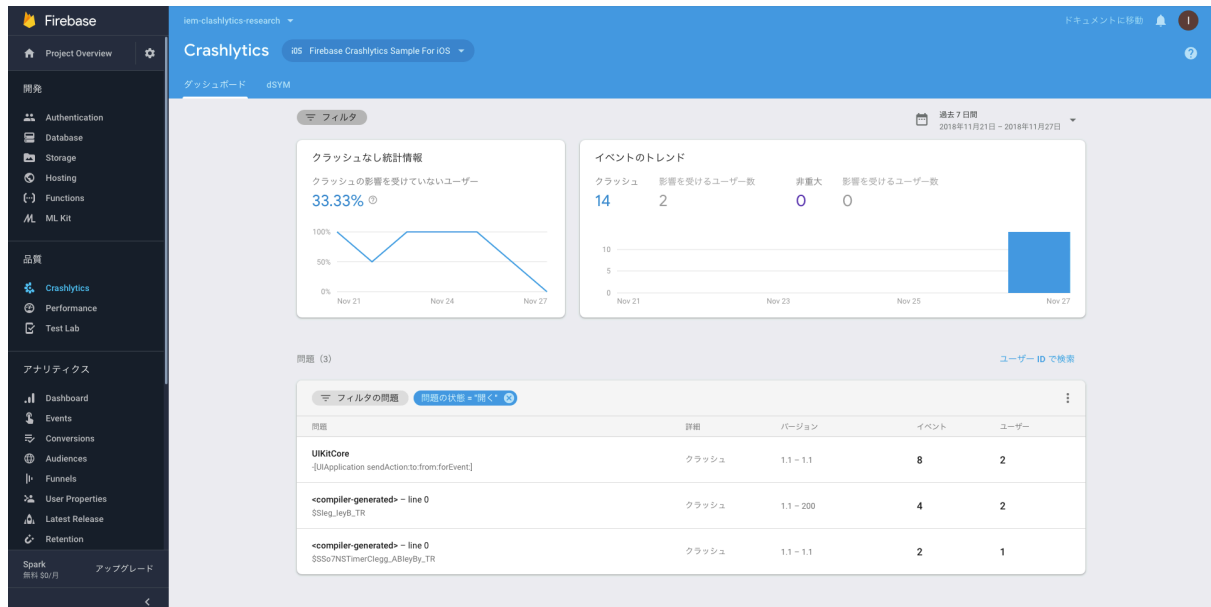


## Crashレポートが送信されるか確認する

1. 強制的にクラッシュを引き起こすコードをアプリの任意のソースに追加します。

```
1  import UIKit
2  + import Crashlytics
3
4  class ViewController: UIViewController {
5
6      override func viewDidLoad() {
7          super.viewDidLoad()
8      }
9
10     @IBAction func touchUpInsideCrashButton(_ sender: UIButton) {
11 +         Crashlytics.sharedInstance().crash()
12     }
13
14 }
```

2. アプリをRunし、クラッシュを引き起こすコードが実行されるよう操作します。
3. Firebase Console > 対象プロジェクト(iem-clashlytics-research) > Clashlytics > 対象アプリを開いて、Crashレポートがアップロードされていることを確認します

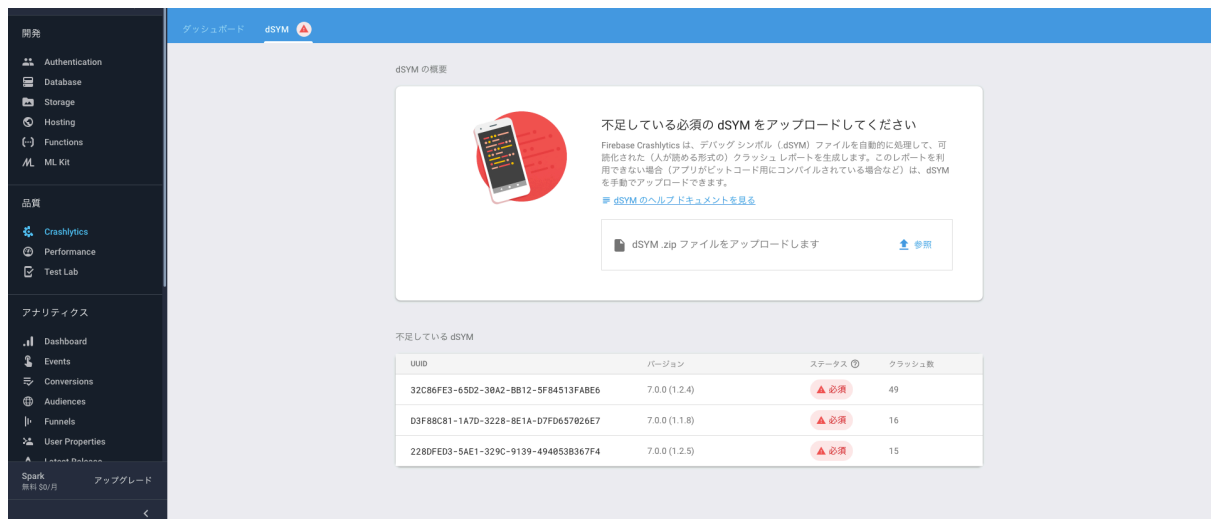


- CrashlyticsのDashboardに、dSYMが不足している旨のメッセージが表示され、クラッシュレポートが表示されない場合は、



別途、以下の画面から、dSYMをアップロードしてください。

(プロキシ環境等によって、アップロードに失敗することがあります)



## References

<https://firebase.google.com/docs/crashlytics/?authuser=1>

<https://github.com/firebase/firebase-ios-sdk>

## Tips & Tricks

- Debug時は、Crashレポートを収集したくない。

<https://qiita.com/YusukeIwaki/items/e767c45edad48302cec8>