

---

# Arise Documentation

*Release 2.2*

**Arise.io**

August 02, 2013



# CONTENTS

<b>1</b>	<b>Setup your first A/B test</b>	<b>3</b>
1.1	Overview . . . . .	3
1.2	Getting Started . . . . .	3
<b>2</b>	<b>Arise iOS SDK</b>	<b>5</b>
2.1	Installation steps . . . . .	5
2.2	Full code example . . . . .	6
2.3	Notes . . . . .	7
<b>3</b>	<b>Arise Android SDK</b>	<b>9</b>
3.1	Installation steps . . . . .	9
3.2	Full code example . . . . .	10
3.3	Notes . . . . .	11
<b>4</b>	<b>Indices and tables</b>	<b>13</b>



Contents:



# SETUP YOUR FIRST A/B TEST

## 1.1 Overview

### 1.1.1 What is A/B testing?

Testing is a way of showing users different variations of your app, and then measuring how each variation affects your goals. For example, if you're interested in having your users make more in-app purchases, you might test the phrase "buy more coins now!" versus "save time by buying coins". Arise's A/B testing tools can tell you which one will make you more money. You can also use Arise to test new in app-features, evaluate performance and roll out changes without resubmitting your app.

### 1.1.2 What is a conversion?

Tests are only good when they can measure some metric and show you which path is performing better. To help out our system, that metric should be a "conversion". Good examples of goals are things like an in-app purchase, a new account created, or even a tap on an advertisement. We can then calculate the [conversion rate](#) (Number of conversion/ Number of views).

## 1.2 Getting Started

### 1.2.1 1. Create an account

If you do not have an account yet, register on the [dashboard](#). The dashboard is your tool to configure your experiment variation.

### 1.2.2 2. Configure your variation

The current system supports only one experiment and two variations.

#### Variation values

Variation values will be transmitted to your app. You can experiment:

- call to actions
- headline, product descriptions (words, style, number of words)

- in-app purchases (prices, multiple contents)
- forms (types of fields, length, layout, error handling)
- layout and design (position and grouping of content)
- images

You need to set different values for each variation. We will consider the variation A as the default variation (control).

### Distribution

You can set the distribution of your variations. The total of all the variations must always be 100%. You can for example set 70% on the variation A (control) and 30% on the variation B (test). Once your app is deployed, you will be able to view the conversion rate for each variation.

### 1.2.3 3. Install the Arise SDK in your app

We currently support iOS and Android. Please read the following documentation to install the Arise SDK in your app:

- [\*Arise iOS SDK\*](#)
- [\*Arise Android SDK\*](#)



# ARISE IOS SDK

This documentation will help you to install the Arise iOS client in your application.

## 2.1 Installation steps

### 2.1.1 1. Add the Arise library to your project

First, download the [Arise SDK for iOS](#). Unzip it and drag it inside your project's Frameworks folder. Ensure that you select 'Copy items into the destination group's folder' when the dialogue box appears.

### 2.1.2 2. Add dependencies

In XCode:

- select your project in the project navigator
- in the project settings select your target
- click on the Build Phases tab
- Open the Link With Libraries collapsible panel
- Click on '+'
- Search for libsqlite3.dylib and click on the Add button.

### 2.1.3 3. Initialize the framework

Add the following line in your AppDelegate.h:

```
#import <Arise/Arise.h>
```

Add the following line under application:didFinishLaunchingWithOptions of your AppDelegate.m file to initialize the framework:

```
[Arise initializeWithKey:@"9c51b5e8f06ebd26728f29954365098f052c68c8"];
```

Replace the value of the key by your own key. You can find it in your dashboard.

## 2.1.4 4. Get the experiment value

You can request a variation value using the `getVariation` method in one of your `.m` file:

```
[ABTest getVariation:^(NSString *experimentValue){
    upgradeButton.title = experimentValue;
}];
```

Do not forget to import the Arise SDK in your header file:

```
#import <Arise/Arise.h>
```

## 2.1.5 5. Record events

Now that you have setup your application for testing, you will need to record views and conversion events. Record a view:

```
[ABTest recordView];
```

Record a conversion:

```
[ABTest recordConversion];
```

## 2.2 Full code example

```
#import "ViewController.h"

@interface ViewController ()

@end

@implementation ViewController

- (void)viewDidLoad
{
    [super viewDidLoad];

    // Get and setup the variation
    [ABTest getVariation:^(NSString *value){
        // Use the variation value to customize our application
        // ...

        // For example :
        // Change the title of the purchase button
        purchaseButton.title = value;
    }];
}

- (void)onLoadPurchasePage
{
    // the user is viewing the item purchase page
    // record a view event
    [ABTest recordView];
}
```

```
- (IBAction)onPurchase:(id)sender
{
    // the user has bought the item
    // record a conversion event
    [ABTest recordConversion];
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}
```

## 2.3 Notes

The Arise iOS SDK supports iOS 6 and later.



# ARISE ANDROID SDK

This documentation will help you to install the Arise Android client in your application.

## 3.1 Installation steps

### 3.1.1 1. Add the Arise library to your project

First, download the [Arise library jar file](#) and drag it inside your project's /libs/ folder.

### 3.1.2 2. Add permissions

In your project, right click on AndroidManifest.xml Click on Open With/Android Common XML Editor. Add after the `<uses-sdk .... />` tag:

```
<uses-permission android:name="android.permission.INTERNET"></uses-permission>
```

### 3.1.3 3. Initialize the framework

In the onCreate function of your main activity, you need to initialize the framework:

```
// Initialize the arise library
String authKey = "9c51b5e8f06ebd26728f29954365098f052c68c8";
Arise.initialize(getApplicationContext(), authKey);
```

Replace the value of authKey by your own key. You can find in your dashboard.

### 3.1.4 4. Get the experiment value

When you plan to run the experiment, you will need to call the `getVariationWithListener` to get the experiment data:

```
// Get and setup the variation
ABTest.getVariationWithListener(new VariationListener() {
    @Override
    public void onVariationAvailable(String value) {
        final String buyMessage = value;
    }
});
```

### 3.1.5 5. Record events

Now that you have setup your application for testing, you will need to record views and conversion events. Record a view:

```
ABTest.recordView();
```

Record a conversion:

```
ABTest.recordConversion();
```

## 3.2 Full code example

```
package com.example.shoestore;

import io.arise.ABTest;
import io.arise.Arise;
import io.arise.VariationListener;
import android.os.Bundle;
import android.app.Activity;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Initialize the arise library
        String authKey = "9c51b5e8f06ebd26728f29954365098f052c68c8";
        Arise.initialize(getApplicationContext(), authKey);

        // Get and setup the variation
        ABTest.getVariationWithListener(new VariationListener() {
            @Override
            public void onVariationAvailable(String value) {
                // Change the button label
                final String buyMessage = value;
                // Use the buyMessage to customize our application
                // ...
            }
        });

    }

    private void onLoadPurchasePage() {
        // the user is viewing the item purchase page
        // record a view event
        ABTest.recordView();
    }

    private void onPurchaseCompleted() {
        // the user has bought the item
        // record a conversion event
        ABTest.recordConversion();
    }
}
```

```
}  
}
```

## 3.3 Notes

The Arise Android SDK supports Android 2.3.3 (API level 10) and later.





# INDICES AND TABLES

- *genindex*
- *modindex*
- *search*