

# Rewarded Video Integration for iOS

Get up and running with the Rewarded Video Ad Unit in 5 easy steps:

1. **Implement the Rewarded Video Delegate**
2. **Initialize the Rewarded Video Unit**
3. **Show a Video Ad to Your Users**
4. **Reward the User**
5. **Verify Your Integration**

## Before You Start

Make sure you have correctly integrated the Supersonic SDK as well as any additional Ad Network Adapters into your application. Integration is outlined here ([/ios/getting-started-with-supersonic-ios-sdk/](#)).

## Step 1. Implement the Rewarded Video Delegate


The Supersonic SDK fires several events to inform you of ad availability and completions so you'll know when to reward your users.

To receive these events, register to the delegate **setRVAdapterDelegate:rvDelegate** :

### OBJECTIVE C

**OBJECTIVE-C**

```
[[Supersonic sharedInstance] setRVDelegate:YOUR_REWARDED_VI
```


**SWIFT**

```
let YOUR_REWARDED_VIDEO_DELEGATE:RVDelegate = RVDelegate()
```



**Note:** Please do not assume the callbacks are always running on the main thread. Any UI interaction or updates resulting from Supersonic callbacks need to be passed to the main thread before executing.

The SDK will notify your delegate of all possible events listed below:

**OBJECTIVE-C**

```
@interface RVDelegate : NSObject <SupersonicRVDelegate>
@end
@implementation RVDelegate
/**
 * Invoked when initialization of RewardedVideo ad unit has f
 */
- (void)supersonicRVInitSuccess{}
/**
 * Invoked when RewardedVideo initialization process has fail
 * contains the reason for the failure.
 */
- (void)supersonicRVInitFailedWithError:(NSError *)error{}
/**
 * Invoked when there is a change in the ad availability stat
 * @param - hasAvailableAds - value will change to YES when r
 */
- (void)supersonicRVAdAvailabilityChanged:(BOOL)hasAvailable
/**
 * Invoked when the user completed the video and should be re
 * If using server-to-server callbacks you may ignore these c
 * for the callback from the Supersonic server.
 * @param - placementInfo - SupersonicPlacementInfo - an obje
 */
- (void)supersonicRVAdRewarded:(SupersonicPlacementInfo*)pla
```

```

/**
 * Invoked when an Ad failed to display.
 * @param - error - NSError which contains the reason for the
 * failure. The error contains error.code and error.message.
 */
- (void)supersonicRVAdFailedWithError:(NSError *)error{}
/**
 * Invoked when the RewardedVideo ad view has opened.
 */
- (void)supersonicRVAdOpened{}
/**
 * Invoked when the user is about to return to the applicati
 * RewardedVideo ad.
 */
- (void)supersonicRVAdClosed{}
/**
 * Note: the events below are not available for all supporte
 * Ad Networks.
 * Check which events are available per Ad Network you choose
 * your build.
 * We recommend only using events which register to ALL Ad Ne
 * include in your build.
 */
/**
 * Available for: AdColony, Vungle, AppLovin, UnityAds
 * Invoked when the video ad starts playing.
 */
- (void)supersonicRVAdStarted{}
/**
 * Available for: AdColony, Flurry, Vungle, AppLovin, UnityAc
 * Invoked when the video ad finishes playing.
 */
- (void)supersonicRVAdEnded{}
@end

```

## SWIFT

```

class RVDelegate: NSObject, SupersonicRVDelegate {
/**

```

```

* Invoked when initialization of RewardedVideo ad unit has f
*/
func supersonicRVInitSuccess(){ }
/**
* Invoked when RewardedVideo initialization process has fail
* contains the reason for the failure.
*/
func supersonicRVInitFailedWithError(error:NSError){ }
/**
* Invoked when there is a change in the ad availability stat
* @param - hasAvailableAds - value will change to YES when r
*/
func supersonicRVAdAvailabilityChanged(hasAvailableAds:Bool)
/**
* Invoked when the user completed the video and should be re
* If using server-to-server callbacks you may ignore these e
* for the callback from the Supersonic server.
* @param - placementInfo - SupersonicPlacementInfo - an obje
*/
func supersonicRVAdRewarded(placementInfo:SupersonicPlacemer
/**
* Invoked when an Ad failed to display.
* @param - error - NSError which contains the reason for the
* failure. The error contains error.code and error.message.
*/
func supersonicRVAdFailedWithError(error:NSError){ }
/**
* Invoked when the RewardedVideo ad view has opened.
*/
func supersonicRVAdOpened(){ }
/**
* Invoked when the user is about to return to the applicati
* RewardedVideo ad.
*/
func supersonicRVAdClosed(){ }
/**
* Note: the events below are not available for all supporte
* Ad Networks.
* Check which events are available per Ad Network you choose

```

```

* your build.
* We recommend only using events which register to ALL Ad Networks
* include in your build.
*/
/**
* Available for: AdColony, Vungle, AppLovin, UnityAds
* Invoked when the video ad starts playing.
*/
func supersonicRVAdStarted(){ }
/**
* Available for: AdColony, Flurry, Vungle, AppLovin, UnityAds
* Invoked when the video ad finishes playing.
*/
func supersonicRVAdEnded(){ }

```

**Note:** Supersonic provides an error code mechanism to help you understand errors you may run into during integration or live production. See the complete guide here (</ios/supersonic-sdk-error-codes/>).

## Step 2. Initialize the Rewarded Video Unit

Once the Rewarded Video Unit is initialized, Supersonic's SDK will automatically check all available Ad Networks for Video Ads throughout the life-cycle of the app. You should initialize Rewarded Video as early as possible to allow time for all Ad Networks to prepare Video Ad content. We recommend doing so on app launch.

Initialize the Rewarded Video Ad Unit:

### OBJECTIVE-C

```

[[Supersonic sharedInstance] setRVDelegate:YOUR_REWARDED_VIDEO_DELEGATE];
[[Supersonic sharedInstance] initRVWithAppKey:APP_KEY withUserIdentifier:USER_IDENTIFIER];

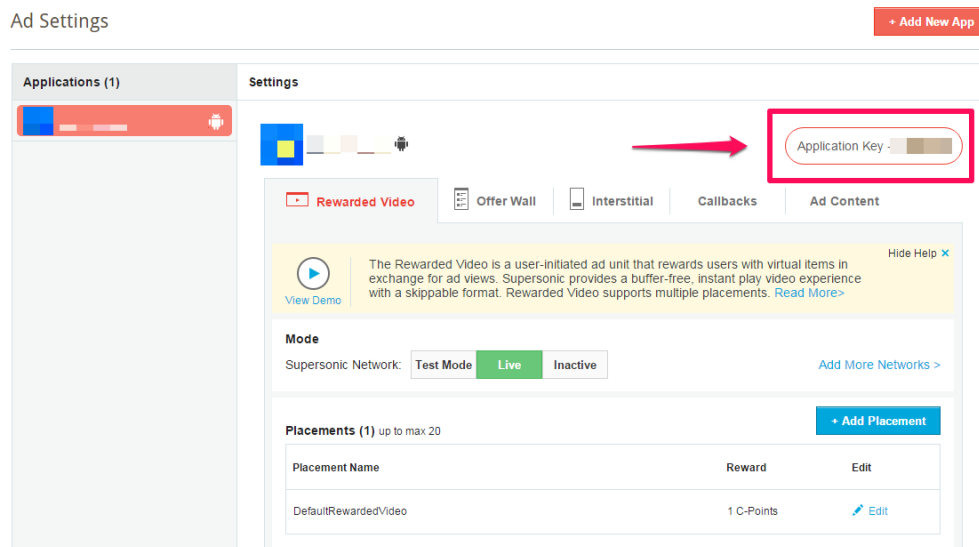
```

### SWIFT

```
Supersonic.sharedInstance().setRVDelegate(YOUR_REWARDED_VIDEO)
Supersonic.sharedInstance().initRVWithAppKey(APP_KEY, withUs
```

### Note:

1. **APP\_KEY** is the unique ID of your Application in your Supersonic account.



(<http://developers.ironsrc.com/wp-content/uploads/2016/01/Supersonic-Application-Key.png>)

2. **USER\_ID** is the unique ID of your end user. We support NSString from 1 to 64 characters. Common practice is to use the Apple Advertising ID (IDFA). More information on User IDs can be found here (</general/app-monetization/monetization-configurations/application-user-id/>).
3. Supersonic supports **Network Change Status**, which enables the SDK to change the availability according to network modifications, i.e. in the case of no network connection, the availability will turn to FALSE. The default of this function is **False**; in the case you'd like to utilize it, you can activate it in the Init with the following string:

```
[Supersonic sharedInstance] setShouldTrackReachability:Y
```

## Step 3. Show a Video Ad to Your Users


By correctly implementing the Rewarded Video Delegate and its functions, you can receive the availability status through the **supersonicRVAdAvailabilityChanged**. You will then be notified with the delegate function below upon ad availability change:

### OBJECTIVE-C

```
supersonicRVAdAvailabilityChanged:(BOOL)hasAvailableAds;
```

### SWIFT

```
func supersonicRVAdAvailabilityChanged(hasAvailableAds:Bool)
```



Alternatively, you can also request ad availability directly by calling:

### OBJECTIVE-C

```
[[Supersonic sharedInstance] isAdAvailable];
```


### SWIFT

```
Supersonic.sharedInstance().isAdAvailable()
```

Once an Ad Network has an available video, you are ready to serve this video ad to your user. This is the ideal moment to insert a trigger to encourage your users to watch the video ad. By calling the **showRV** method on your **Supersonic** instance, you can show a Video Ad to your users:

### OBJECTIVE-C

```
[[Supersonic sharedInstance] showRVWithViewController:viewCo
```



### SWIFT

```
Supersonic.sharedInstance().showRVWithViewController:viewCo
```



With Supersonic's Ad Placements, you can customize and optimize the Rewarded Video experience. This tool enables you to present videos to your users from different placements depending on the reward. You can use the below function to define the exact Placement you'd like to show an ad from. Navigate to the Ad Placement (/general/app-monetization/monetization-configurations/ad-placements-setting/) document for more details.

## OBJECTIVE-C

```
[[Supersonic sharedInstance] showRVWithPlacementName:placementName];
```

```
SupersonicPlacementInfo * pInfo = [[Supersonic sharedInstance]
                                     pInfo];
if(pInfo != NULL)
{
    NSString * rewardName = [pInfo rewardName];
    NSNumber * rewardAmount = [pInfo rewardAmount];
}
```

## SWIFT

```
Supersonic.sharedInstance().showRVWithPlacementName(placementName)
```

```
if let pInfo:SupersonicPlacementInfo = Supersonic.sharedInstance()
    pInfo {
    let rewardAmount:NSNumber = pInfo.rewardAmount
    let rewardName:NSString = pInfo.rewardName
}
```

To get details about the specific Reward associated with each Ad Placement, you can call the following:

## OBJECTIVE-C

```
[[Supersonic sharedInstance] getRVPlacementInfo:@"PLACEMENT_NAME"];
```

## SWIFT



**SWIFT**

```
Supersonic.sharedInstance().getRVPlacementInfo("PLACEMENT_NA
```

In addition to Supersonic's Ad Placements, you can now configure capping and pacing settings for selected placements. Capping and pacing improves the user experience in your app by limiting the amount of ads served within a defined timeframe. Read more about capping and pacing [here](http://developers.supersonic.com/general/app-monetization/ad-placements-setting/#step-2) (<http://developers.supersonic.com/general/app-monetization/ad-placements-setting/#step-2>).

**Note:** If you choose to use the capping and pacing tool for Rewarded Video, we recommend calling the below method to verify if a certain placement has reached its ad limit. This is to ensure you don't portray the Rewarded Video button when the placement has been capped or paced and thus will not serve the Video ad.

```
[[Supersonic sharedInstance] isRewardedVideoPlacementCapped:
```

## Step 4. Reward the User

Each time the user successfully completes a video, the Supersonic SDK will fire the **supersonicAdRewarded:amount** event. Upon reward, you will be notified with the delegate function below.

The Placement object contains both the **Reward Name & Reward Amount** of the Placement as defined in your Supersonic Admin:

**OBJECTIVE-C**

```
(void)supersonicRVAdRewarded:(SupersonicPlacementInfo*)placementInfo {
    NSNumber * rewardAmount = [placementInfo rewardAmount];
    NSString * rewardName = [placementInfo rewardName];
}
```

**SWIFT**

```
func supersonicRVAdRewarded(placementInfo:SupersonicPlacemer
    let rewardAmount:NSNumber = placementInfo.rewardAmc
    let rewardName:NSString = placementInfo.rewardName
}
```

### Note:

1. The default setting in your Supersonic account is to notify you of user completions/rewards via the **supersonicAdRewarded:amount** callback within the client of your app. Additionally, if you would also like to receive notifications to your back-end server, you can turn on server-to-server callbacks.
2. If you turn on server-to-server callbacks, remember not to reward the user more than once for the same completion. We will be firing both the client-side callback and the server-to-server callback, so you will get two notifications for each completion. To utilize server-to-server callbacks, see here (</general/app-monetization/server-to-server-callback-setting/>).

## Step 5. Verify Your Integration

The Supersonic SDK provides an easy way to verify that your Rewarded Video Mediation integration was completed successfully. To verify your Supersonic SDK integration as well as any additional Ad Networks you may have implemented, navigate to this article (</ironsource-mobile-ios/integration-helper-ios/>).

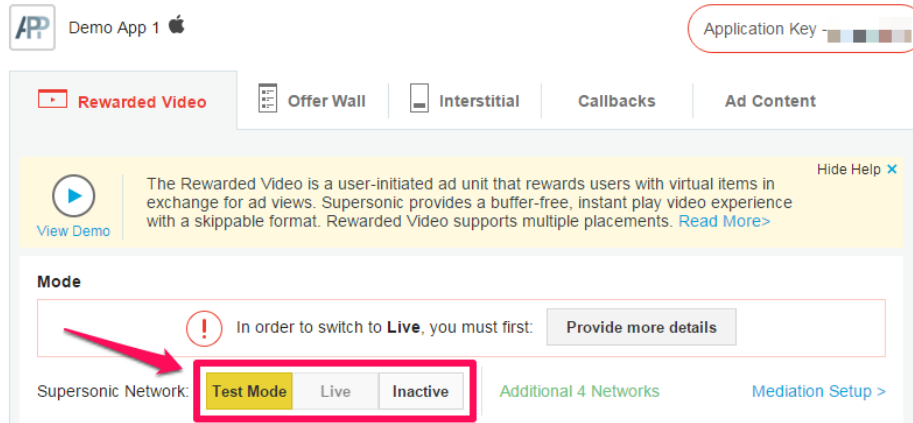
### Done!

You are now all set to deliver Rewarded Video in your application.

### Tip:

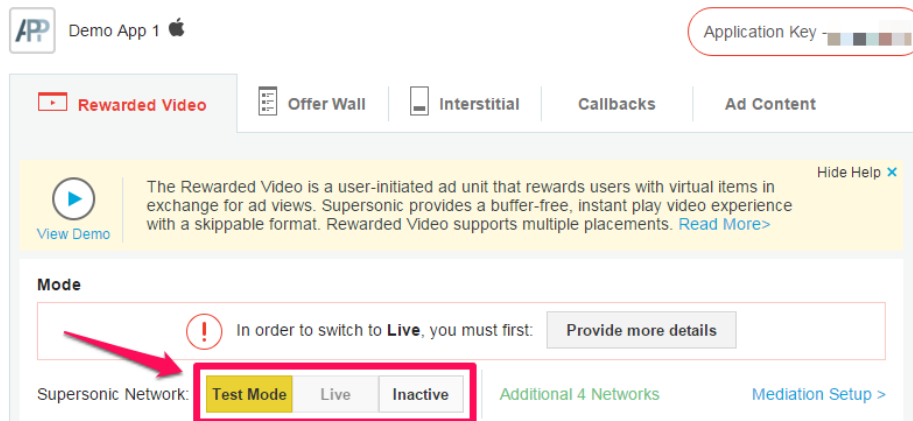
If this is a new integration, by default your application in your Supersonic account will be in '**Test Mode**'. While in Test

Mode the Supersonic SDK will print more logs to the console in order to provide greater visibility into the SDK's processes. In addition, the first ad you will see is a Supersonic test campaign. Turn on **Live** Ad Inventory in the **App Settings** page of your Supersonic console.



(<http://developers.supersonic.com/wp-content/uploads/2016/01/Supersonic-Switch-App-to-Live-Mode-Rewarded-Video-Ad-Unit.png>)

(<http://developers.supersonic.com/wp-content/uploads/2016/01/Supersonic-Switch-App-to-Live-Mode-Rewarded-Video-Ad-Unit.png>)



(<http://developers.ironsrc.com/wp-content/uploads/2015/12/RV-Set-to-Live.png>)

## What's Next?

Follow our integration guides to integrate additional Rewarded Video Ad networks or configure additional Ad Units:

- Mediation  
(/ios/sdk-integration/mediation-networks-ios/)
- Offerwall  
(/ios/offerwall-integration-ios/)
- Interstitial (/ios/sdk-integration/interstitial-integration-ios/)

---

© ironSource LTD. 2016. [Terms of service \(/sonic-general/terms-and-conditions/platform-service-general-terms-conditions/\)](#)