Build Cross Platform Video Chat Apps With Xamarin
Vidyo.io Webinar Series

The webcast will begin shortly. Please stand by.
Build Cross Platform Video Chat Apps With Xamarin
Vidyo.io Webinar Series

Phillip Futernik
Senior Software Engineer
November 15, 2017
Upcoming Webinar

Vidyo.io Getting Started Webinar Series:
Topic: Recording Your Video Chat

December 14 @ 2:00pm ET (11:00am PT)

Presented by:
Sachin Hegde

We will email you an invitation next week!
Poll Time!

How do you prefer to learn a new API?
Getting Started Series:
Build Cross Platform Video Chat Apps
With Xamarin

Philip Futernik
Senior Software Engineer
November 16, 2017
Xamarin Overview
System for Cross-Platform Development

Cross Platform
- Share application logic
- Average 75% code shared

Targets
- Build for multiple OSs
- Primarily used for mobile

IDE
- Visual Studio (Windows)
- Visual Studio for Mac

C#
iOS
Android
Windows
macOS

© Copyright 2017 Vidyo Inc., confidential, proprietary and patent pending information
Xamarin Overview

App Types

**Xamarin.Android**
- XML based UI builder
- AndroidManifest.xml:
  - assign permissions, Android versions, etc
- Activity life cycle methods:
  - OnCreate, OnStart, OnResume, etc

**Xamarin.iOS**
- Storyboard UI builder
- Entitlements.plist, Info.plist, app delegate
- View Controller life cycle methods:
  - ViewDidLoad, ViewWillAppear, ViewDidAppear, etc

**Xamarin.Forms**
- Allows devs to easily create native UI layouts that can be shared across iOS and Android
- Includes more than 40 controls and layouts, which are mapped to native controls at runtime
- Life cycle methods:
  - OnStart, OnSleep, OnResume
Which flavor of Xamarin to use?

Either way, you’ll get fully native apps with shared business logic

**Xamarin.Forms**
- Apps that require little platform specific functionality
- Code sharing is more important than custom UI
- Developers comfortable with XAML

**Xamarin.iOS / Xamarin.Android**
- Apps with interactions that require native behavior
- Apps that use many platform specific APIs
- Apps where custom UI is more important than code sharing
Get started with free vidyo.io account

- Go to https://vidyo.io
- Create a free account
- Download SDKs
  - iOS and Android for Xamarin development
- Start building!
Xamarin + Vidyo.io Important Notes

• Vidyo.io C# bindings:
  - Native C library
    - Android: jar + .so
    - iOS: dylib
  - C# source files

• To render video, Vidyo lib needs a handle to native control
  - Xamarin.iOS / Xamarin.Android : exposed in the UI control
  - Xamarin.Forms : create custom renderer for each platform
## Code Walkthrough

### Include vidyo.io SDK

<table>
<thead>
<tr>
<th>Xamarin.iOS</th>
<th>Xamarin.Android</th>
</tr>
</thead>
<tbody>
<tr>
<td>VidyoClient-iOSSDK:</td>
<td>VidyoClient-AndroidSDK:</td>
</tr>
<tr>
<td>• C# source files</td>
<td>• C# source files</td>
</tr>
<tr>
<td>• libVidyoClient.dylib</td>
<td>• libVidyoClient.so</td>
</tr>
<tr>
<td></td>
<td>➢ Build Action: Android Native Library</td>
</tr>
<tr>
<td></td>
<td>• vidyoclient.jar</td>
</tr>
<tr>
<td></td>
<td>➢ Build Action: Android Java Library</td>
</tr>
</tbody>
</table>

### Initialize vidyo.io

<table>
<thead>
<tr>
<th>Xamarin.iOS</th>
<th>Xamarin.Android</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectorPKG.Initialize();</td>
<td>ConnectorPKG.SetApplicationUIContext(this); ConnectorPKG.Initialize();</td>
</tr>
</tbody>
</table>
Code Walkthrough (cont.)

Create VidyoConnector object

```
Xamarin.iOS
// videoView is of type UIView
Connector
vc = new Connector(videoView.Handle, Connector.CONNECTORVIEWSTYLE.CONNECTORVIEWSTYLEDEFAULT, 15, "warning info@VidyoConnector info@VidyoClient", ",", 0);

Xamarin.Android
// videoView is of type FrameLayout
Connector
vc = new Connector(videoView.Handle, Connector.CONNECTORVIEWSTYLE.CONNECTORVIEWSTYLEDEFAULT, 15, "warning info@VidyoConnector info@VidyoClient", ",", 0);
```

Render video

```
Xamarin.iOS / Xamarin.Android
vc. ShowViewAt(videoView.Handle, 0, 0, videoViewWidth, videoViewHeight);
```
Connect to video chat

```csharp
vc.Connect("prod.vidyo.io", // host
generatedToken, // token
"Philip Futernik", // display name
"PhilipRoom", // resource ID
this); // need to inherit Connector.IMuddy interface

// Implementation of Connector.IMuddy interface:

public void OnSuccess()
{
    Console.WriteLine("OnSuccess");
}

public void OnDisconnected(Connector.ConnectorDisconnectReason reason)
{
    Console.WriteLine("OnDisconnected: " + reason);
}

public void OnFailure(Connector.ConnectorFailReason reason)
{
    Console.WriteLine("OnFailure: " + reason);
}
```
Code Walkthrough (cont.)

Disconnect from video chat

```csharp
Xamarin.iOS / Xamarin.Android
vc.Disconnect();
```

Cycle Camera

```csharp
Xamarin.iOS / Xamarin.Android
vc.CycleCamera();
```
Demo
Finding Help

Available Resources

• How-To Videos
  - https://vidyo.io/how-to-videos/

• Ready to deploy samples
  - https://developer.vidyo.io/packages
  - https://github.com/vidyo

• Get Help
  - https://support.vidyo.io
  - https://stackoverflow.com/ (Use the “vidyo” tag when asking question)
Please follow us on Twitter

Vidyo.io @vidyo_io
Thank you!

Philip Futernik
pfuternik@vidyo.com
Twitter – philnbass

https://vidyo.io
Twitter - @Vidyo_io
LinkedIn – vidyo-io
https://github.com/vidyo