Integrating CST into an iOS Project

Introduction

The CST iOS Lib is a container wrapper for a WebView that displays the contents of a HTML5 page that is responsible for rendering the contents and UI elements of a specific CST deployment.

The latest stable version of the CST iOS Lib can be found in the following URL:

http://developer.viacom.com/mdp_download/cst-ios/

Steps to integrate CST into an IOS Project

1. Add the folder CSTLib into your XCode Project.

2. Import CSTView.h to the view controller class that will be responsible for calling the CST View.

```
#import "CSTView.h"
```

3. Setup a button to launch the CST in the host app. Have the button action point to one of the launchCST methods bellow:

3.1. iPhone

```
- (IBAction)launchCST_iPhone:(id)sender{

    NSString *mgidValue = @"MGID value for iPhone App";

    NSString *platformValue = @"iPhone"; // this could also be "iPhone,iPad".

    [CSTView showCSTInView:self.view forMgid:mgidValue forPlatform:platformValue
inFullScreenMode:TRUE];

}
```

3.2. iPad

```
- (IBAction)launchCST_iPad:(id)sender{

    UIView *popUpView = [[[UIView alloc] initWithFrame:CGRectMake(100, 100, 320, 480)] autorelease];

    NSString *mgidValue = @"MGID value for iPad App";

    NSString *platformValue = @"iPad"; // this could also be "iPhone,iPad".

    [CSTView showCSTInView:popUpView forMgid:mgidValue forPlatform:platformValue
inFullScreenMode:FALSE];

    [self.view addSubview:popUpView];

}
```

3.3. Universal

```
- (IBAction)launchCST_Universal:(id)sender{

    NSString *mgidValue_iPhone = @"MGID value for iPhone App";

    NSString *mgidValue_iPad = @"MGID value for iPad App";

    if (NSClassFromString(@"UISplitViewController") != nil && UI_USER_INTERFACE_IDIOM() ==
UIUserInterfaceIdiomPad) {

        UIView *popUpView = [[[UIView alloc] initWithFrame:CGRectMake(100, 100, 320, 480)]
autorelease];

        [CSTView showCSTInView:popUpView forMgid:mgidValue_iPad forPlatform:@"iPad"
inFullScreenMode:FALSE];

        [self.view addSubview:popUpView];

    }

    else {

        [CSTView showCSTInView:self.view forMgid:mgidValue_iPhone forPlatform:@"iPhone"
inFullScreenMode:TRUE];

    }

}
```

4. Inside the folder CSTLib you will find a plist entitled cst_config.plist. This plist has 4 params that configure the UI of the native navigationBar that wraps the CST Web View:

-backgroundImage(height:44px)

-logoImage (240x35px)

-closeButtonImage (40x40px)

-appName: a string with the linkshare name of the app

Each of these fields should point to an image asset present in the project's resource folder. The CST lib already comes with 3 example images that can be used in production, in case the brand your app belongs to is Nickelodeon: close_btn.png, headerBackground.pn and nick_logo.png.

If your app belongs to a different brand or you would like to change the layout of the navigation bar of CST, drag into your project resources folder the appropriate image assets and update the cst_config.plist accordingly with the right image names.

5. Retina Display Support

If you'd like to add support for retina display enable-devices, you would need to provide an extra image (with the dimensions duplicated) for each of the images mentioned above, and include the prefix "@2x" after each image name (For example: MyImage@2x).