Prepare native for DronaHQ integration

Android:

- Register a custom scheme for your application in AndroidManifest

```
<activity
    android:name=".NewActivity"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.BROWSABLE" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:scheme="customscheme" />
    </intent-filter>
</activity>
```

- Receive the invocation parameters in the activity

```

      //Use uid & nonce to make HTTP call to DronaHQ users API
   }
  } @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        if(Intent.ACTION_VIEW.equals(i.getAction()))
        {
            Uri uri = i.getData();
            String dhqNonce = uri.getQueryParameter("nonce");
            String dhqUid = uri.getQueryParameter("uid");

            //Use uid & nonce to make HTTP call to DronaHQ users API
        }
    }
```

- On app launch, invoke DronaHQ container app for user authentication. If your app is successfully invoked, it means the authentication was successful.

```
void openDronaHQApp()
{
    Intent intent = new Intent();
    intent.setAction(Intent.ACTION_VIEW);
    intent.addCategory(Intent.CATEGORY_DEFAULT);
    intent.addCategory(Intent.CATEGORY_BROWSABLE);

    //Pass your custom scheme, so that it can be invoked
    //by DronaHQ post user authentication
    intent.setData(Uri.parse("dhq://?scheme=customscheme"));

    startActivity(intent);
}
```

iOS 8 & below

- Register a custom URL scheme for your app in the plist file
- Receive the invocation parameters from DronaHQ. Add below method in your AppDelegate.m

```

  //Use nonce & uid to make a http call to DronaHQ users API

  return YES;
}
```

```
//Helper method for parsing query strings.
-(NSDictionary *)parseQueryString:(NSString *)query {
  NSMutableDictionary *dict = [[NSMutableDictionary alloc] initWithCapacity:6];
  NSArray *pairs = [query componentsSeparatedByString:@”&”];


  for (NSString *pair in pairs) {
      NSArray *elements = [pair componentsSeparatedByString:@”=”];
      NSString *key = [[elements objectAtIndex:0] stringByReplacingPercentEscapesUsingEncoding:NSUTF8StringEncoding];
      NSString *val = [[elements objectAtIndex:1] stringByReplacingPercentEscapesUsingEncoding:NSUTF8StringEncoding];


      [dict setObject:val forKey:key];
  }
  return dict;
}-(BOOL)application:(UIApplication *)application openURL:(NSURL
*)url sourceApplication:(NSString *)sourceApplication
annotation:(id)annotation {
    NSString *strQuery = [url query];
    NSDictionary *dictQuery = [Utilities parseQueryString:strQuery];
    NSString *strNonce = [dictQuery objectForKey:@"nonce"];
    NSString *strUid = [dictQuery objectForKey:@"uid"];

    //Use nonce & uid to make a http call to DronaHQ users API


    return YES;
}

//Helper method for parsing query strings.
-(NSDictionary *)parseQueryString:(NSString *)query {
    NSMutableDictionary *dict = [[NSMutableDictionary alloc]
initWithCapacity:6];
    NSArray *pairs = [query componentsSeparatedByString:@"&"];


    for (NSString *pair in pairs) {
        NSArray *elements = [pair componentsSeparatedByString:@"="];
        NSString *key = [[elements objectAtIndex:0] stringByReplacingPercentEscapesUsingE
        NSString *val = [[elements objectAtIndex:1] stringByReplacingPercentEscapesUsingE

        [dict setObject:val forKey:key];
    }
    return dict;
}
```

- On app launch, invoke DronaHQ container app for user authentication. If your app is successfully invoked, it means the authentication was successful.

```
-(void)openDronaHQApp {
    NSURL *url = [[NSURL alloc] initWithString:@"dhq://?scheme=nativeApp"];
    bool canOpenApp = [[UIApplication sharedApplication] canOpenURL:url];
    if (canOpenApp) {
        [[UIApplication sharedApplication] openURL:url];
    } else {
        //DronaHQ app not installed
        //show user message
    }
}
```

iOS 9

On iOS 9 & above, Apple deprecated few methods which are required to invoke the native app. Due to which the above methods will not work, alternative is to use Universal Links instead.

DronaHQ will handle the server side part of hosting the universal link, and preparing the apple-app-site-association file.

Prepare native app

- Enable ’Associated Domains’ in your app identifier on developers.apple.com
  - Click on Certificates, Identifiers & Profiles
  - Click on Identifiers.
  - Select your App ID (create it first if required),

    - Click Edit and enable the Associated Domains entitlement
- Enable ’Associated Domains’ in your Xcode project
    - In Xcode, select your App’s target,
    - Click Capabilities
    - toggle Associated Domains to On.
- Next Add an entry for the domain, prefixed with ‘applinks:’
    - s0.deeplink.dronahq.com
    - s1.deeplink.dronahq.com
    - s2.deeplink.dronahq.com

Handle invocation from Universal Link

```


    //Use nonce & uid to make a http call to DronaHQ users API

  }

  return YES;

}-(BOOL)application:(UIApplication *)application continueUserActivity:(NSUserActivity
*)userActivity restorationHandler:(void (^)(NSArray * _Nullable))restorationHandler
{

    DLog(@"User Activity - %@", userActivity);

    if ([userActivity.activityType isEqualToString:NSUserActivityTypeBrowsingWeb])
{

        NSURL *url = userActivity.webpageURL;

        NSString *strQuery = [url query];

        NSDictionary *dictQuery = [Utilities parseQueryString:strQuery];

        NSString *strNonce = [dictQuery objectForKey:@"nonce"];

        NSString *strUid = [dictQuery objectForKey:@"uid"];


        //Use nonce & uid to make a http call to DronaHQ users API

    }

    return YES;

}
```

Invoke DronaHQ app

On iOS 9 & above, the apps need to whitelist which they can invoke using `openUrl` method.

- Add LSApplicationQueriesSchemes as array in your .plist file
- Add Item0 as string with value ‘dhq’

Reference:

1. Prepare native app for universal link invocation: https://blog.branch.io/how-to-setup-universal-links-to-deep-link-on-apple-ios-9
2. Handle universal links: https://github.com/vineetchoudhary/iOS-Universal-Links

Windows 10 UWP:

- Register a custom scheme for your application in package.appxmanifest

```
<Extensions>
    <uap:Extension Category="windows.protocol">
      <uap:Protocol Name="customscheme">
        <uap:Logo>Assets\AppIcon.png</uap:Logo>
        <uap:DisplayName>DronaHQ</uap:DisplayName>
       </uap:Protocol>
     </uap:Extension>
</Extensions>
```

- Receive the invocation parameters from DronaHQ. Add below method in your App

```
protected override void OnActivated(IActivatedEventArgs args)
  {
      if (args.Kind == ActivationKind.Protocol)
      {
         ProtocolActivatedEventArgs eventArgs = args as
ProtocolActivatedEventArgs;
         // TODO: Handle URI activation
         // The received URI is eventArgs.Uri.AbsoluteUri
      }
  }
```

- On app launch, invoke DronaHQ container app for user authentication. If your app is successfully invoked, it means the authentication was successful.

```
private async void openDronaHQApp()
    {
        Uri launchUri = new System.Uri(@"dhq://?scheme=customscheme")
        bool result = await Windows.System.Launcher.LaunchUriAsync(launchUri);
    }
```

Reference:

1. Microsoft Windows Handle URI activation https://docs.microsoft.com/en-us/windows/uwp/launch-resume/handle-uri-activation