Getting Started With Mobile SDK For Windows 8.1

@salesforcedocs

Last updated: March 6, 2015

© Copyright 2000–2015 salesforce.com, inc. All rights reserved. Salesforce is a registered trademark of salesforce.com, inc., as are other names and marks. Other marks appearing herein may be trademarks of their respective owners.

CONTENTS

GETTING STARTED WITH MOBILE SDK

Let's get started creating custom mobile apps! This is your starting place if you've already signed up for Force.com and have created a Salesforce Connected App.

The next steps are:

1. Make sure you have the required software.
2. Install Mobile SDK on page 1.
3. Do any of the following:
   - Create a new Mobile SDK native project for Windows 8.1.
   - Add Mobile SDK to an existing Windows 8.1 native project.

IN THIS SECTION:

Prerequisites

Before you begin using Mobile SDK for Windows, make sure you've installed the following software.

- Microsoft Visual Studio 2013 with Update 3 or newer.
- Visual Studio SDK (http://msdn.microsoft.com/en-us/library/bb166441.aspx).
- SQLite for Visual Studio (http://www.sqlite.org/2014/sqlite-winrt81-3080600.vsix). This package is for Microsoft Windows Phone and Store 8.1 apps.
- SQLite for Microsoft Windows Phone 8.1 (http://visualstudiogallery.msdn.microsoft.com/1d04f82f-2fe9-4727-a2f9-a2db127ddc9a).
- Visual Studio template for Mobile SDK: SalesforceUniversalApplicationTemplate.zip at https://github.com/forcedotcom/SalesforceMobileSDK-Windows/blob/unstable/template/SalesforceUniversalApplicationTemplate.zip. Copy this archive to the `<user>\Documents\Visual Studio 2013\Templates\ProjectTemplates` directory.

Install Mobile SDK

If you've installed the required software, installing Mobile SDK for Windows is just a matter of adding the Mobile SDK project template to Visual Studio.

1. Download the Visual Studio template ZIP file from GitHub at http://bit.ly/1ymHU8F
2. Copy the template ZIP file to the `<user>\Documents\Visual Studio 2013\Templates\ProjectTemplates` directory.

When you click **File** > **New** > **Project...** in Visual Studio, the Mobile SDK template appears under Visual C# templates as **Salesforce Application (Universal Apps)**. When you create your project, Visual Studio retrieves all missing Mobile SDK modules from their NuGet repositories.

Create a Mobile SDK Native Project For Windows 8.1

For quickest results, use the Mobile SDK template and NuGet packages to create your new projects.

Before creating a new project, be sure that you've installed the prerequisite software described in Prerequisites on page 1.

1. In Visual Studio, click **File** > **New** > **Project**.
2. In the left pane, choose **Visual C#** .
3. In the right pane, select the **Salesforce Application (Universal Apps)** template.
4. If you don't want to creat a solution file, deselect `Create directory for solution`.
5. Set Project Name and Solution Name, if applicable.
6. To use source control for your project, select `Add to source control`.
7. Click OK.
8. (Optional) For Connected App Settings, overwrite the default Client ID and Callback URL with the client ID and callback URL from your connected app.
9. (Optional) Select the OAuth scopes that your app requires. All apps require at least the `api` scope.
10. Click **Finish**.

    Important: If you use the default connected app values during development, be sure to change to your own connected app settings before you publish your app.

11. If you configured your project to use source control, select a source control provider.
12. Open the Solution Explorer. Right-click the solution and click **Enable NuGet Package Restore**. Click **Yes** when you're asked if you want to configure this solution to download and restore missing NuGet packages during build.
13. In the **Build** menu, click **Configuration Manager**.
14. For each active solution configuration, reset the Platform values for listed projects to the appropriate processor type. Do not leave any projects set to **Any CPU**.

Your new solution contains desktop and phone projects, as well as NuGet packages for Salesforce Mobile SDK and other dependencies.

Add Mobile SDK to an Existing Windows 8.1 Project

If you have an existing generic Windows 8.1 project, follow these steps to make it Mobile SDK-enabled.

1. In Visual Studio, open the existing Windows 8.1 project.
2. Open the Solution Explorer and right-click the solution.
3. Select **Enable NuGet Package Restore**, then click **Yes** when asked if you want to configure NuGet to download and restore missing NuGet packages during build.

4. Right-click the main project and select **Manage NuGet Packages**.
5. Click **Online** and select **All**.
6. Select one or more of the following packages:
   - `SalesforceSDKCore`
   - `SalesforceSmartStore`
   - `SalesforceSmartSync`

   Note: For each package you select, Visual Studio downloads any dependent packages.
7. Configure `app.xaml` and `app.xaml.cs` to derive from `SalesforceApplication`.
   a. In `App.xaml.cs`, change the derivation of your class to the following:

   ```
   public sealed partial class App : SalesforceApplication
   ```

   b. In `App.xaml`, add the following namespace attribute:

   ```
   xmlns:app="using:Salesforce.SDK.App"
   ```

   c. In `App.xaml`, set your app block to `SalesforceApplication`, using the namespace specified in the attribute you just set:

   ```
   <app:SalesforceApplication>
   ...
    xmlns:app="using:Salesforce.SDK.App"
   ...
   </app:SalesforceApplication>
   ```
8. In `app.xaml.cs`, override `SetRootApplicationPage()` to return the type of your root page.
9. In `app.xaml.cs`, override `InitializeConfig()`. Follow the examples from the sample apps in GitHub.
10. Create a subclass of `SalesforceConfig`. In this class, define your bootstrap configuration, including `ClientId`, `CallbackUrl`, and `Scopes`. For an example, see the configuration in the sample apps.
11. For Phone applications, reference `Salesforce.SDK.Core` and `Salesforce.SDK.Phone`.
12. For Windows applications, reference `Salesforce.SDK.Core` and `Salesforce.SDK.Store`.
13. When you create an application, use `NativeMainPage` instead of `Page`. The `NativeMainPage` class helps to maintain the user's authentication state.

    Note: If you can't use `NativeMainPage` for any reason, replicate the few things that `NativeMainPage` does in your own `Page` implementation.
14. Clean and build all projects.

SAMPLE APPS IN MOBILE SDK FOR WINDOWS 8.1

Mobile SDK for Microsoft Windows 8.1 provides sample apps that demonstrate major Mobile SDKfeatures.

RestExplorer
Demonstrates the OAuth and REST API functions of Salesforce Mobile SDK. Mobile SDK provides Store and Phone versions of this app.

NativeSmartStoreSample
Demonstrates registering, populating, querying, and manipulating data in a local SmartStore database.

SmartSyncExplorer
Demonstrates the power of the native SmartSync library for offline productivity.

Salesforce1.Container
Provides the SalesforceA app wrapped in a Mobile SDK container.