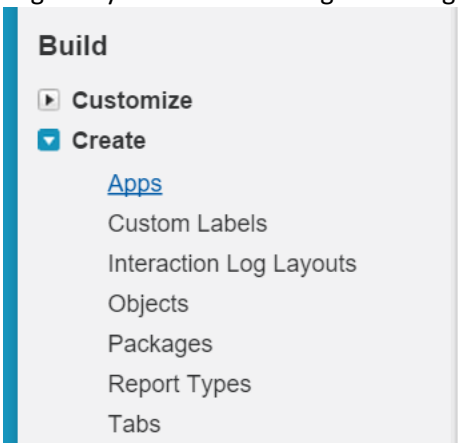
Note that on OSX most commands may need to be run as an administrator.

Install npm, Cordova, Forcedroid/Forceios (depending on OS):

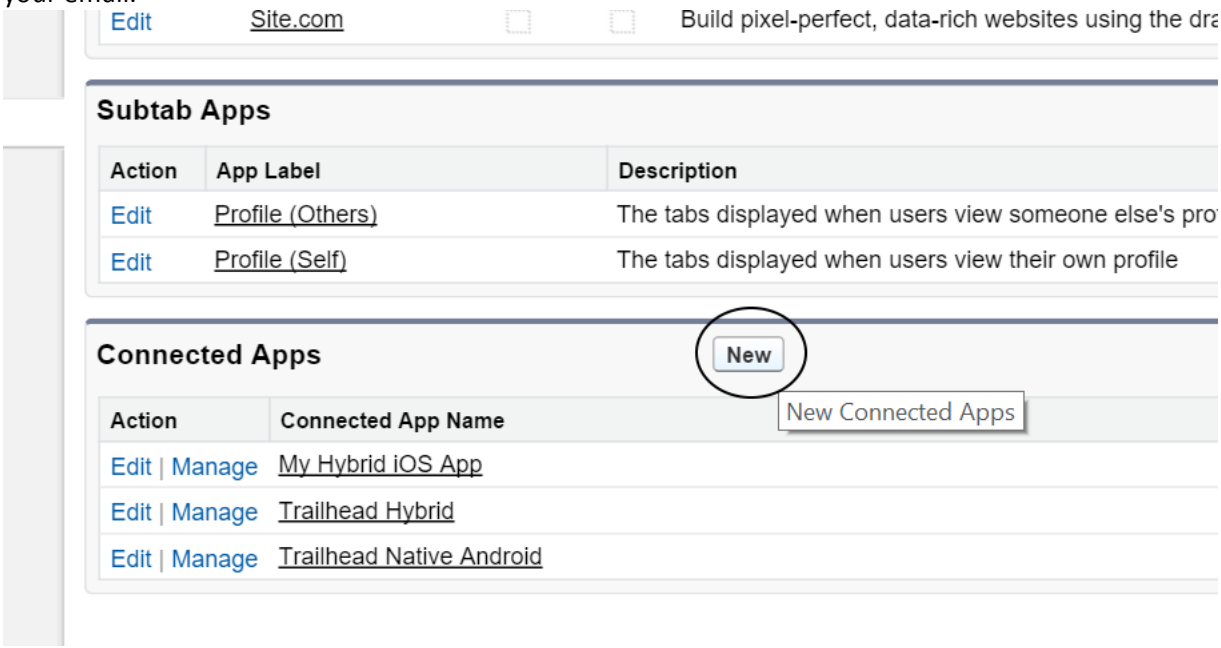
1. If you do not have Node.js installed already, download it: https://nodejs.org/en/
2. Node comes with npm, a package manager, automatically. The -g command installs packages globally on your machine. Open Command Prompt or another shell.
3. Use npm to install cordova: *npm install -g cordova*
4. Use npm to install forcedroid or forceios: *npm install -g forcedroid* OR *npm install -g forceios*

Set up Connected App and REST Endpoint in Org:

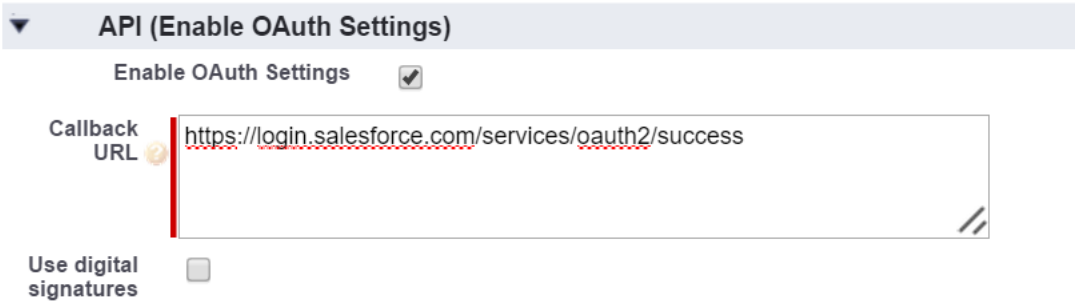
1. Log into your Salesforce org and navigate to Create > Apps.



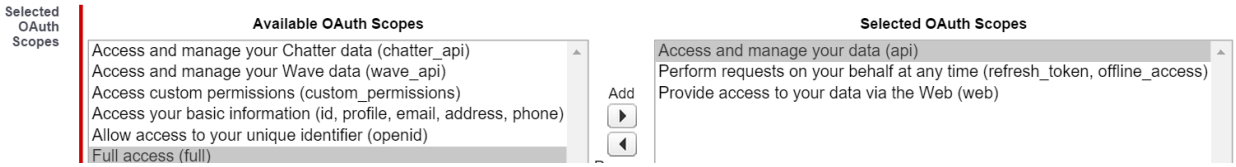
2. In the Connected Apps section towards the bottom, click 'New' and enter your app's Name and your email.



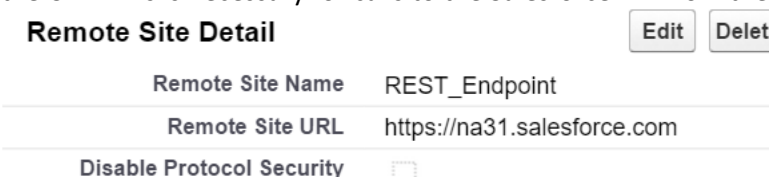
3. Check the Enable OAuth Settings box in the API section. You can enter any callback URL you like; a sample one is 'https://login.salesforce.com/services/oauth2/success'.



4. Select 'Access and manage your data (api)', 'Provide access to your data via the Web (web)', and 'Perform requests on your behalf at any time (refresh_token, offline_access)' as the app's OAuth Scopes and save the app.

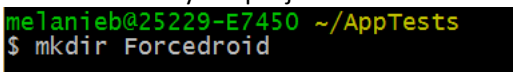


5. Find 'Remote Sites' in the Setup menu, and create a new one. Name it something like 'REST_API_Endpoint' and save your Salesforce instance (e.g. https://na31.salesforce.com/) as the URL. This is necessary for calls to the Salesforce API from the Mobile SDK to work.



Using Forcedroid:

1. Create directory for project.



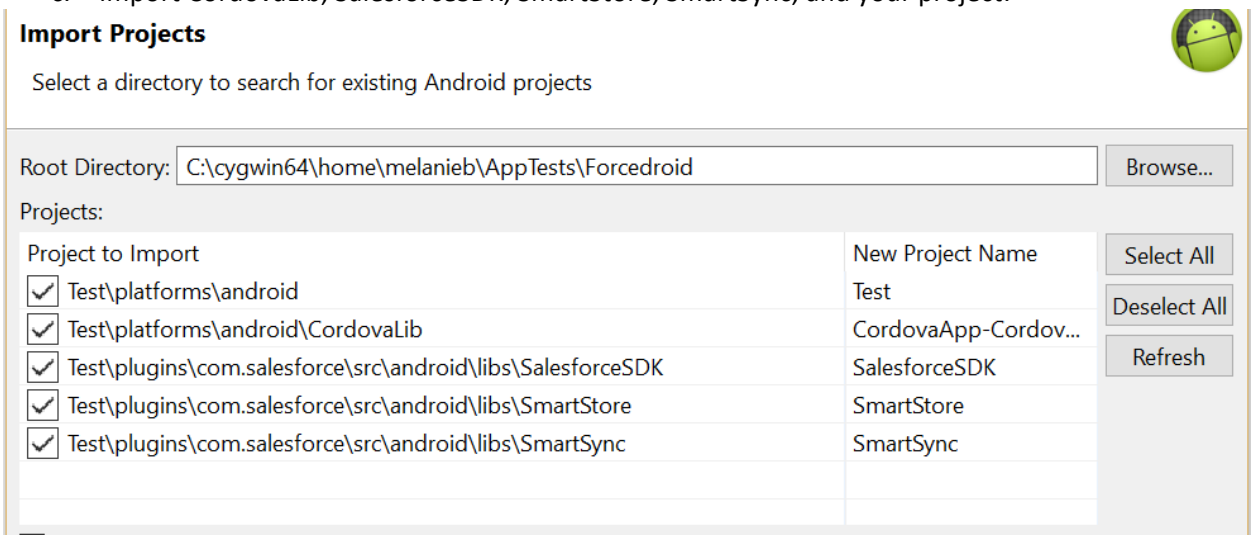
(note that image uses a bash shell for Windows instead of the standard CMD)

2. Run *forcedroid create* in Command Prompt and follow the prompts. When setup is finished (forcedroid will tell you it is done), press *ctrl+c* to exit setup.
NOTE: DO NOT RUN CORDOVA BUILD! It will not work, and it will break settings forcedroid

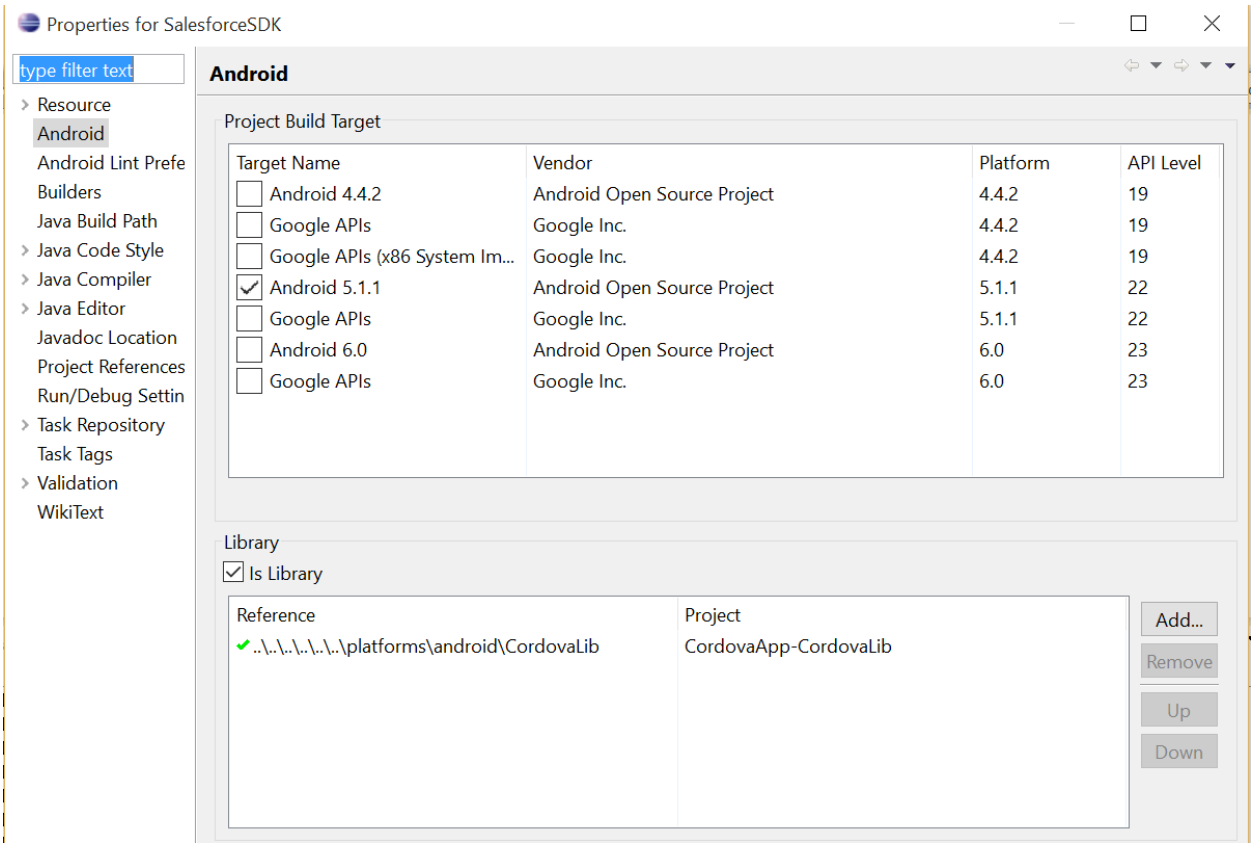
needs for Eclipse.

```
melanieb@25229-E7450 ~/AppTests
$ forcedroid create
Enter your application type (native, hybrid_remote, or hybrid_local): hybrid_loc
al
Enter your application name: Test
Enter the target directory of your app: Forcedroid
Enter the package name for your app (com.mycompany.my_app): com.test.test

<snip>

Your application project is ready in Forcedroid.

To use your new application in Eclipse, do the following:
   - Go to File -> Import...
   - Choose Android -> Existing Android Code into Workspace, and click Next >
   - For the root directory, browse to the Forcedroid folder
   - Pick the following projects: Test/platforms/android, Test/platforms/android
/CordovaLib, Test/plugins/com.salesforce/android/libs/SmartSync, Test/plugins/co
```

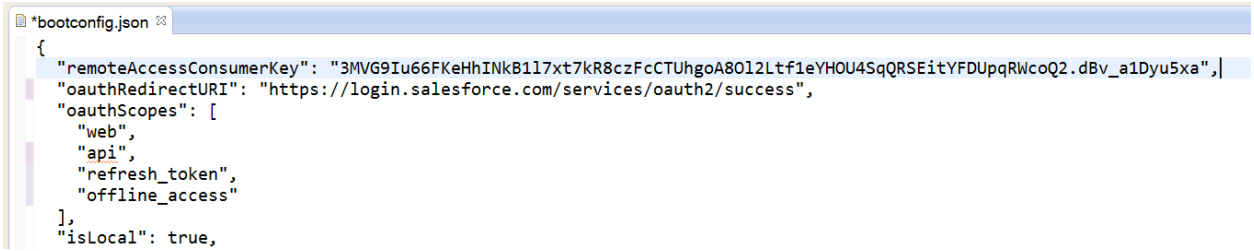
3. Open Eclipse.
4. Import projects by right-clicking the project list on the left or by using File > Import:
    a. Android > Existing Projects into Workspace
    b. Select directory created in step 1 as root for project.
    c. Import CordovaLib, SalesforceSDK, SmartStore, SmartSync, and your project.



5. For each project, right-click and open Properties. Under the Android tab, set the desired build target (recommended 5.1.1), and set the following dependencies:
    a. CordovaApp-CordovaLib: check isLibrary, no dependencies
    b. SalesforceSDK: check isLibrary, dependent on CordovaLib
    c. SmartStore: check isLibrary, dependent on SalesforceSDK
    d. SmartSync: check isLibrary, dependent on SmartStore
    e. Your Project: not a library, dependent on all others



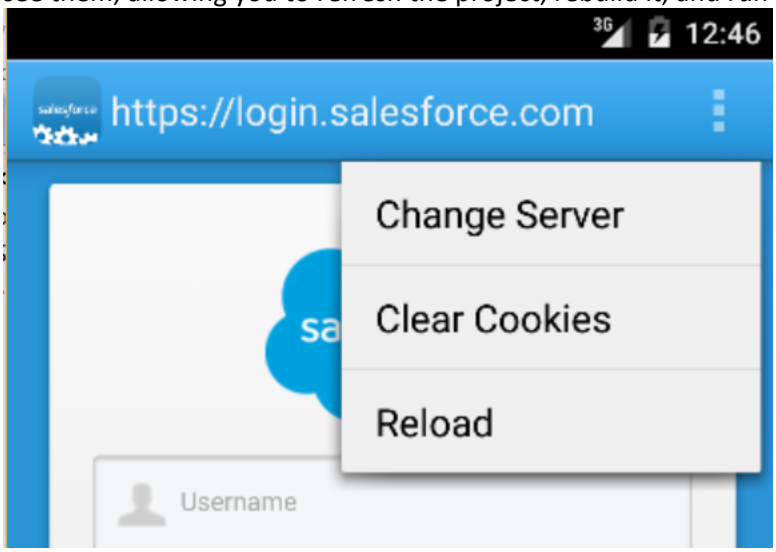
6. If 'Build Automatically' is off (recommended), build in the same order.
7. Open Your Project > www > bootconfig.json and enter the connected app you set up earlier's Consumer Key, OAuth Redirect URI, and OAuth Scopes. If you want to specify a different index or error page, this is the place for that as well.



8. At this point, you should be able to right click your project and select Run As > Android Application. If you need to set up Android virtual devices, one of the buttons in the row underneath the toolbar should be the Android Device Manager, from which you can do so.

9. You should see the Salesforce login screen. If you need to change from the Login server to another, the ... button in the top right will let you do so. You can now use any preferred editor to modify the html, js, or css files in the 'www' folder at the top level of your project (not inside the platform directory, but above that). Using *cordova prepare* in Command Prompt at the top level of the project directory will move all modified assets where they need to be for Eclipse to

see them, allowing you to refresh the project, rebuild it, and run it to see changes.



Troubleshooting:

- If when using *forcedroid create* you get an error similar to 'Plugin doesn't support this project's cordova-android version. cordova-android: 3.6.4, failed version requirement >=4.0.0-dev', this is because the Salesforce plugin does not use the most recent version of Cordova. Install an older version of Cordova:
    - *npm uninstall -g cordova && npm install -g cordova@4.3.0*
- If when building a project, you get an error about not finding the '@drawable/sf__icon' resource, this means your project does not link the SalesforceSDK project correctly. Check that you imported projects into Eclipse from the correct directory. For example, if your directory structure looked like 'Demonstration > Demonstration Proj > www ...' be sure you imported from Demonstration and not Demonstration Proj. If this is correct, check that the dependency references point to the correct locations, as in step 5 of the **Using Forcedroid** section.
- If when uploading your project to the Android Emulator you get timeout errors, try increasing the timeout value in Window -> Preferences -> Android -> DDMS -> ADB Connection Timeout (ms).

Notes:
Because Cordova is platform-agnostic, the html, js, and css files can be used in any Cordova project. At the time of this writing, it is unknown if building for Android on OSX is possible, but if not, similar steps as above are likely used to set up a forceios project, and the same html, js, and css files can be placed into that project to create an iOS app.

Other Resources:

- Cordova documentation: http://cordova.apache.org/docs/en/5.0.0/
- Salesforce SmartStore and SmartSync sample app descriptions: https://developer.salesforce.com/docs/atlas.en-us.mobile_sdk.meta/mobile_sdk/samples_intro.htm

- Github repository for above samples: https://github.com/forcedotcom/SalesforceMobileSDK-Samples
- Salesforce Mobile SDK documentation (but beware: much of this is incorrect or outdated): https://developer.salesforce.com/docs/atlas.en-us.mobile_sdk.meta/mobile_sdk/