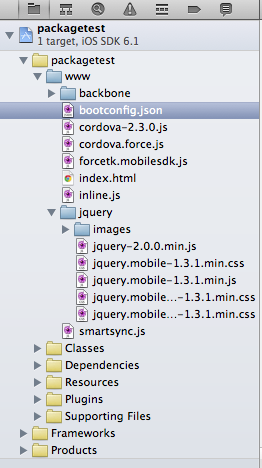
Steps To develop salesforce IOS app in Chrome

1) To create a hybrid iOS mobile application using the Salesforce Mobile SDK

http://www2.developerforce.com/mobile/getting-started/ios#hybrid  
  
Using this app, users can view a list of Account and Contact records from Salesforce.  
The following structure available on the app folder names as "packagetest"



2) By default the start page app is set to index.html, you can change it in "bootconfig.json" file is located on "www" folder.

"startPage": "index\_ios.html",

bootconfig.js

{

"remoteAccessConsumerKey": "Enter your consumer key which is kept from salesforce app",

"oauthRedirectURI": "Enter your callback url",

"oauthScopes": ["web","api"],

"isLocal": true,

"startPage": "index\_ios.html",

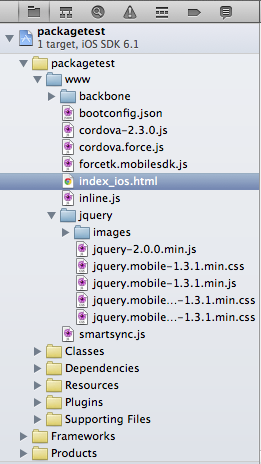
"errorPage": "error.html",

"shouldAuthenticate": true,

"attemptOfflineLoad": false

}

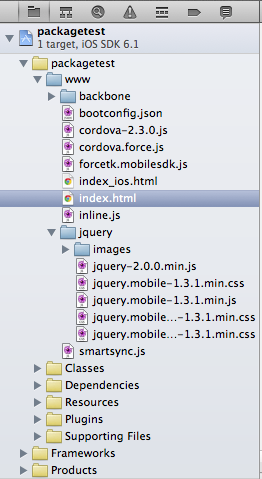
3) To rename the "index.html" to "index\_ios.html"



To run this app in xcode with the current setting, without error and then go to next stpes.

4) Now only we go to the our development browser oriented setting, i.e, we go to run our app in "Chrome browser". we need to create a new connected app on salesforce, following callback url,  
 http://localhost/packagetest/packagetest/www/oauthcallback.html

5) To create new html file named as "index.html" www folder.



6) The required jquery, cordova and forcetk.mobilesdk files are included in index.html ,

<link rel="stylesheet" href="jquery/jquery.mobile-1.3.1.min.css" />

<script src="jquery/jquery-1.6.2.min.js"></script>

<script src="jquery/jquery.mobile-1.0b2.min.js"></script>

<script src="jquery/jquery.popup.js"></script>

<script src="cordova-2.3.0.js"></script>

<script src="cordova.force.js"></script>

<script src="forcetk.mobilesdk.js"></script>

7) Then set the OAuth configuration variables and connect the salesforce api object, here proxyUrl file is the simple php proxy to get html, json and more response

<script>  
 var loginUrl = "https://ap1.salesforce.com/";  
 var remoteAccessConsumerKey = "Enter your consumer key";  
 var oauthRedirectURI = "http://localhost/packagetest/packagetest/www/oauthcallback.html";  
 var proxyUrl = "http://localhost/packagetest/packagetest/www/proxy.php?mode=native';  
 var forcetkClient = new forcetk.Client(remoteAccessConsumerKey, loginUrl, proxyUrl);  
 </script>

8) To connect salesforce login page with the available details using jquery popup window .

9) After the login response function, "sessionCallback" fire after the login, here we assign the salesforce session token,

function sessionCallback(oauthResponse) {  
 if (typeof oauthResponse === 'undefined'|| typeof oauthResponse['access\_token'] === 'undefined') {  
 errorCallback({  
 status: 0,   
 statusText: 'Unauthorized',   
 responseText: 'No OAuth response'  
 });  
 } else {  
 forcetkClient.setSessionToken(oauthResponse.access\_token, null, oauthResponse.instance\_url);  
 var $j = jQuery.noConflict();  
 $j("#sf\_login\_form").toggle(); //Hide login button  
 $j("#sf\_home").toggle(); //Show main content  
 }  
 }

10) Now you can fetch the detalis from salesforce using the REST API,

//Success method for SFDC Contacts Query   
 var onSuccessSfdcContacts = function(response) {  
 var $j = jQuery.noConflict(); $j("#div\_sfdc\_contact\_list").html("")  
 var ul = $j('<ul data-role="listview" data-inset="true" data-theme="a" data-dividertheme="a"></ul>'); $j("#div\_device\_contact\_list").append(ul);  
 ul.append($j('<li data-role="list-divider">Device Contacts: ' + response.records.length + '</li>'));  
 $j.each(response.records, function(i, contact) {  
 var formattedName = contact.Name;  
 if (formattedName) {  
 var newLi = $j("<li><a href='#'>" + (i+1) + " - " + formattedName + "</a></li>");  
 ul.append(newLi);  
 }  
 });  
 $j("#div\_device\_contact\_list").trigger( "create" )  
 }

//Failure method for SFDC Contacts Query   
 var onErrorSfdcContacts = function(error) {  
 console.log(error);  
 alert("Error occured when connect salesforce");  
 };

//SFDC Contacts Query   
forcetkClient.query("select Name from contact", onSuccessSfdcContacts, onErrorSfdcContacts);

Finally you can run the app by the url,  
<http://localhost/packagetest/packagetest/www/>

The required files are available on https://github.com/sirajudeen037/Saleforce

