BoxPopup

BoxPopup is an example app that shows how to use the Box iOS API. It is useful to also consult the Box iOS API Documentation for more detailed questions. This document describes some of the general techniques used in the example app.

Note that every application that accesses Box.net must have a unique Box API key. To retrieve this key, go to: https://www.box.net/developers/services and click "Create New Application" After that is completed, an API key will be assigned to your application, and this API key must be place in the BoxRESTAPIFactory.m. If you do not do this, the example app will not run.

Overview

The initial view of the BoxPopup app allows the user to login, select a folder, and select an action. The large run button then runs the action. There are three main pieces of code that are of interest to developers:

BoxLoginController.m -> doLoginAction and BoxRegisterViewController.m -> doRegisterAction, which shows how to do login and register users

BoxFolderChooserTableViewController.m -> getFoldersAsync, which shows how to retrieve the list of folders and files from the Box.net servers

BoxPopupMainViewController.m -> Action Methods, which shows how to use the various actions

Login and Registration

To login, first, layout a .xib file in interface builder that contains a UIWebView. This webview should be hidden to begin with – the login builder will show it when appropriate. Then, create a login builder like:

```
_loginBuilder = [[BoxLoginBuilder alloc] initWithWebview:_webView delegate:self];
```

and start the login process via:

```
        [_loginBuilder startLoginProcess];
```

when the login process is complete, the login builder will call the methods:

```
-(void)loginCompletedWithUser:(BoxUser *)user;
-(void)loginFailedWithError:(BoxLoginBuilderResponseType)response;
```

on its delegate as appropriate. Note that if your view controller implements startActivityIndicator and stopActivityIndicator, this can be used to indicate when network activity is taking place behind the scenes. See BoxLoginController.m -> doLoginAction.

To register, create a registration operation via a call like:

```
_registerOperation = [[BoxRegisterOperation alloc]
initForLogin:userName password:password delegate:self];
```

and execute it either by calling `start` directly, or by adding it to an operation queue. When the registration process completes, the operation will call:

```
- (void)operation:(BoxOperation *)op didCompleteForPath:(NSString
  *)path response:(BoxOperationResponse)response;
```

on its delegate with an appropriate response code. Note that the registration operation object will contain a property user that contains the user information. See BoxRegisterViewController.m -> doRegisterAction.

Retrieving files and folders

To retrieve the information about a folder, perform a call like:

```
BoxFolder * folderModel = [BoxFolderXMLBuilder
folderForId:folderIdToDownload token:ticket
responsePointer:&responseType basePathOrNil:nil];
```

Note that this will retrieve all of the children of the folder, both files and folders. Also, remember that this call is blocking for the duration of the network operation and so should be done on a background thread. See BoxFolderChooserTableViewController.m -> getFoldersAsync.

Other operations

Other operations are performed similarly to the registration operation. Operations are created and added to an operation queue. When they are finished, they send a operation:didCompleteForPath:response: message to their delegate. See BoxPopupMainViewController.m -> Action Methods for detailed examples.