Developing iOS Apps with Liferay Screens

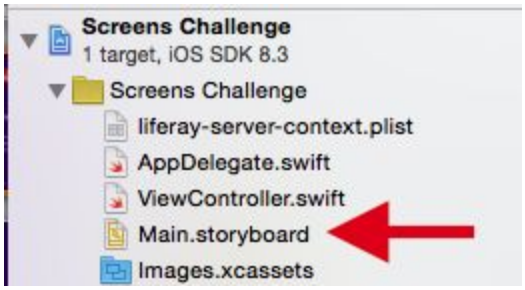
Logging into Liferay Portal

Before you start, you have to create your user account using this page:

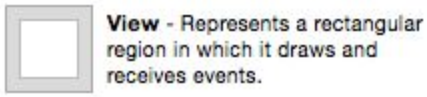
https://screensdemo.liferay.com/web/screens-challenge/signup

In this section, you will use your first screenlet, the LoginScreenlet, which will allow you to sign users into the portal.

1. Run your project (**Product -> Run** or Cmd + R) to see your app running in the simulator (a blank app will appear in the simulator)
2. In Xcode, click on your Main.storyboard file. This represents the screen-flow used in the app.



3. Select the left screen by clicking it.
4. At the bottom of the right pane, browse the list to find the element called UIView (at the very bottom of the list)



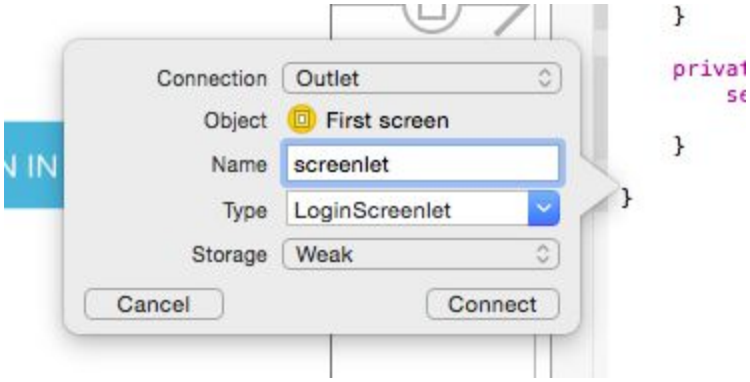
5. Drag that element from the list and drop it into the left storyboard's screen
6. Adjust size and position to center it in the middle of the screen (be careful, since both your screen and the new view are white, you can get confused)
7. If you're a pixel-perfect guy, you can go to the fifth icon in the right sidebar  and type the following positions:
    a. X: 0
    b. Y: 20
    c. Width: 320
    d. Height: 548
8. Now go to the third icon in the right sidebar  and type the following
    a. Class: **LoginScreenlet**
    b. Module: LiferayScreens (should be filled automatically)
9. Xcode will begin the automatic building. If everything goes well, you'll see the login screenlet showing in the left screen
10. Run the project (Cmd + R) and see your brand new screenlet in the simulator
    a. Try to sign in using wrong user and password: you will see how the screenlet shows a default error message
    b. If you type correct credentials the screenlet will silently sign you into the portal

Listen screenlet's events

In this section, you will learn how you can be notified when anything interesting happens inside the screenlet. For instance, you can get your code called when the login is successfully completed.

11. If we want to perform an action when the login operation is successfully completed, we need first to add a delegate (a kind of listener) to the screenlet. When anything interesting happens inside the screenlet, it will notify to the delegate object right away.
12. Go to Main.storyboard, select the main screen (or view controller), find the fifth button starting from the right, in the top toolbar (two circles joined)  , and click on it. The screen will be splitted and the source code will appear at the bottom or on the right.

13. Click on the login screenlet in the screen, hold your Control key, drag and drop it into the code just when a gray label appears saying “Insert Outlet”
14. A small dialog will appear, just type “screenlet” in the **Name** field. This will create a variable called “screenlet” in your view controller, and it will be automatically connected with your UI object.



15. Now code the following at the end of view controller’s **viewDidLoad** method:

```
self.screenlet.delegate = self;
```

This makes your view controller the screenlet’s delegate, so when anything happens inside the screenlet, the view controller will be notified.

16. To make your view controller valid, you have to implement a protocol (like an interface in Java). Do this typing the bold part in the class declaration

```
class ViewController: UIViewController, LoginScreenletDelegate {
```

17. And finally, you have to add to the view controller the protocol’s methods that will be invoked. By now, we only need to know when the login operation is successful:

```
func screenlet(screenlet: BaseScreenlet,
       onLoginResponseUserAttributes attributes: [String:AnyObject]) {
            println("User attributes -> \(attributes)")

    self.goToNextScreen()
}
```

18. If you run the app again (Cmd + R), and type right credentials, you’ll see how a new blank screen appears when the login is completed. Also notice the user attributes returned by the Liferay Portal are printed in the Xcode’s console

Display a Dynamic Form from Liferay

In this section, you will use a dynamic form screenlet to allow the user to send some data to the portal

1. In Xcode, open your Main.storyboard, go to the second screen and add a new UIView on it (using the components list at the bottom of the right sidebar)
2. Place it at full screen or use the following coordinates (fifth icon in the right sidebar ):
    a. X: 0
    b. Y: 20
    c. Width: 320
    d. Height: 548

3. Now go to the third icon in the right sidebar  and type the following
    a. Class: **DDLFormScreenlet**
    b. Module: LiferayScreens (should be filled automatically)
4. Go to the fourth icon in the right sidebar  and type the following values (these values come from the portal instance):
    a. Structure Id: 22354
    b. Record Set Id: 22356
5. That's it! Just run the app (Cmd + R), type right credentials, and you'll see how the form will appear ready to be filled in.
    a. Leave some fields empty, and press Submit button. You'll see how validation fails because of required fields.
    b. Fill all values and press Submit button. You'll see a success message.
6. Now go back to the portal, and check the just added record in the Dynamic Data List

https://screensdemo.liferay.com/web/screens-challenge

Theming

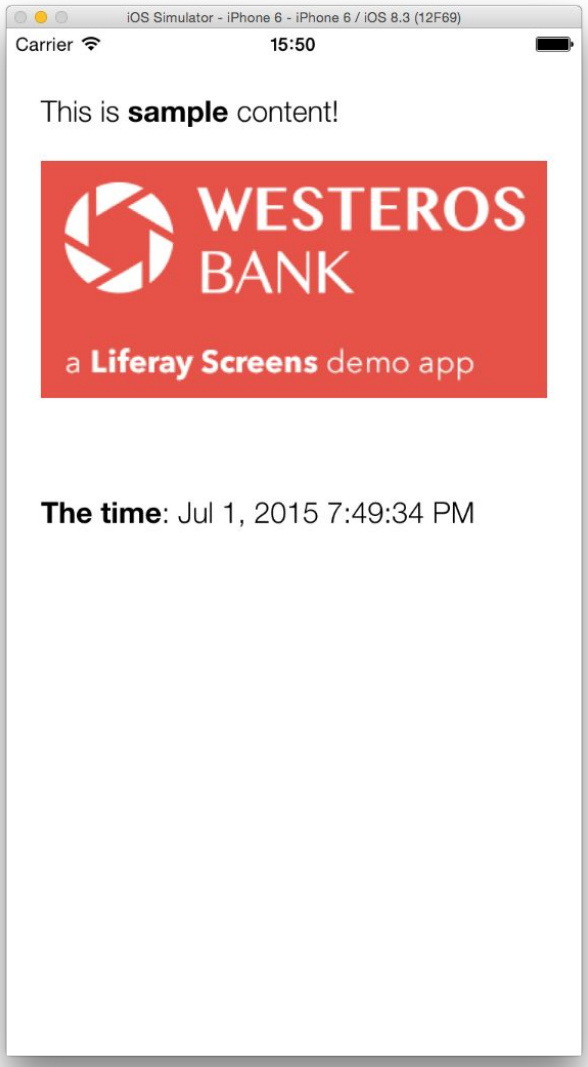
In this section, you finally will learn how to modify the look & feel of your screenlet and meet your app visual requirements. For this, we use the concept of "theme": a library of classes and layouts with a visual representation of the screenlets. You can change one screenlet layout and the look & feel will be changed, without changing the behavior.

1. In Xcode, open your Main.storyboard, select the first screen, and then select the LoginScreenlet.
2. In the properties pane (fourth icon in the right sidebar ), look for "Theme Name" field and type demo there.
3. When the project gets compiled, you'll see the new look and feel of your screenlet (Note: Xcode is buggy and sometimes it doesn't show the preview. In such case, you'll see a blank box instead of the preview)
4. Repeat the same step for the DDLFormScreenlet.
5. Run the app (Cmd + R), type right credentials, and you'll see how the form will appear with the new theme applied.

Post-Event Follow-up Activity

In this section, you will use the skills learned above to change the app to display web content from Liferay's web content management system. Don't worry, the content is already created, all you have to do is insert the screenlet and go!

1. In Xcode, open your Main.storyboard, select the second screen (the one with the DDLFormScreenlet).
2. Now go to the third icon in the right sidebar  and change the type of screenlet by selecting the following
    a. Class: **WebContentDisplayScreenlet**
    b. Module: LiferayScreens (should be filled automatically)
3. Go to the fourth icon in the right sidebar  and type the following values (these values come from the portal instance):
    a. Group Id: 22339
    b. Article Id: 41713
4. That's it! Just run the app (Cmd + R), type right credentials, and you should see the example content:



5. Take a screenshot and post it to the Screens forum thread created for this event. Include your comments, questions, and other feedback! The first 10 people to do this will receive a small gift from Liferay! Note: in order to post, you will need to register for a free account at liferay.com if you do not yet have one.