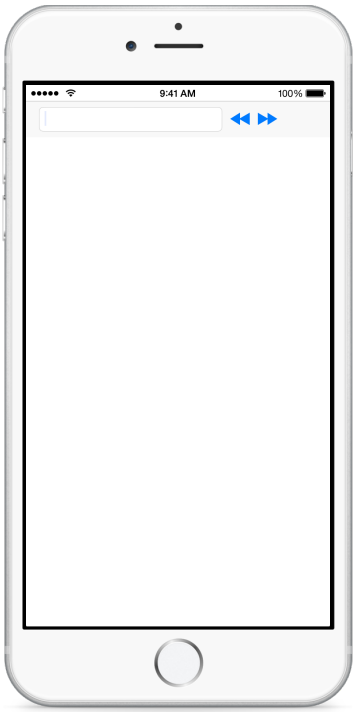
EasyBrowser
Lesson 3

Description

Add a toolbar, text field and two toolbar buttons to the interface.



Learning Outcomes

- Practice using Interface Builder to construct a user interface.
- Discover the toolbar, text field and toolbar button item interface elements.
- Discover how iOS supports different keyboard types to afford particular kinds of input.

Vocabulary

toolbar	text field	bar button item

Materials

- **EasyBrowser Lesson 3** Xcode project

Opening

How can we build an interface that enables the user to type a Web URL to view?

Agenda

- Using Interface Builder, delete the existing button.

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License, by Yong Bakos.

- Using the Object Library (⌥⌘L), add a Toolbar to the top of the main view.
- Control-drag from the toolbar to the View in the Document Outline to add leading, trailing, and top edge constraints relative to the View.
- Drag the top edge of the web view to the bottom edge of the toolbar, and Control-drag from the web view to the toolbar to create a new vertical space constraint.
- Using the Object Library (⌥⌘L), add a Text Field to the toolbar, placing it to the left of the default **Item** button.
- Using the Object Library (⌥⌘L), add a Bar Button Item to the right of the default **Item** button.
- Select each bar button item, and use the Attributes Inspector (⌥⌘4) to modify the **Identifier** attribute. Choose **Rewind** for the left button, and **Fast Forward** for the right button.
- Expand the size of the text field by dragging its right edge.
- Run the app (⌘R), and make sure the menu item *Hardware* > *Keyboard* > *Connect Hardware Keyboard* (⇧⌘K) is unchecked in the Simulator. Tap on the text field, and observe the keyboard appear.
- Discuss how the keyboard is a "standard" text keyboard, yet one's intent for the text field is to ease the entering of HTTP URLs.
- Using Interface Builder, select the text field, and use the Attributes Inspector (⌥⌘4) to modify the **Text Field** attributes.
- Select **Appears while editing** for the **Clear Button** attribute, **URL** for the **Keyboard Type** attribute, and **Go** for the **Return Key** attribute.
- Run the app (⌘R), tap on the text field, and observe how the keyboard includes the **.com** key and a **Go** key instead of the standard **Return** key. Begin typing in the text field, and observe the clear (⊗) button appear within the text field.

Closing

What other keyboard types are there, and what are they best suited for?

Modifications and Extensions

- Investigate the Toolbars documentation in the UIKit User Interface Catalog, and notice how iOS toolbars are usually meant to be placed at the bottom of the interface. Make a design decision to justify why the toolbar in the app is placed at the top, or assert that it should be placed at the bottom.
- Observe how the toolbar is always visible. Investigate how a `UIWebView` contains a `UIScrollView`, and use its `scrollView` property to display the toolbar when the `UIScrollView` is at the top, and to otherwise hide the toolbar.

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License, by Yong Bakos.

Resources

UIKIt User Interface Catalog: Toolbars https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/UIKitUICatalog/UIToolbar.html

iOS Human Interface Guidelines https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/

Text Programming Guide for iOS: Managing the Keyboard http://developer.apple.com/library/ios/documentation/StringsTextFonts/Conceptual/TextAndWebiPhoneOS/KeyboardManagement/KeyboardManagement.html

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License, by Yong Bakos.