EasyBrowser
Lesson 5

Description

Bind the forward and back buttons directly to the web view `goBack` and `goForward` methods.

Learning Outcomes

- Discover how some interface components can be connected to one another directly, without the controller as an intermediary.
- Analyze and consider usability issues, and assess how a web view might be integrated in an app to serve particular use cases.

Vocabulary

`UIWebView`	Connections Inspector	selector

Materials

- **EasyBrowser Lesson 5** Xcode project

Opening

What happens when we start navigating web links within the web view, and we wish to go forward and backward through our history?

Agenda

- Run the app (⌘R), open some web content, and click on a few links.
- Discuss how the web view only displays content, and provides no navigation interface itself.
- Using the Xcode Documentation and API Reference (⇧⌘0), explore the `UIWebView` class reference.
- Using Interface Builder, select the web view and open the Connections Inspector (⌥⌘6). Observe the Received Actions panel.
- Discuss how the web view is able to receive the actions `goBack` and `goForward`.
- Discuss one approach to enabling the back and forward buttons: adding connections from the toolbar buttons to controller actions, which can imperatively tell the web view to move forward or back through its history.
- Using Interface Builder, select one of the toolbar buttons, and open the Connections Inspector (⌥⌘6).
- Discuss how the toolbar button has only one available action binding, called `selector`, which represents the name of a method to call when the button is tapped.
- Using Interface Builder, Control-drag from the Rewind button to the web view to bind the selector to the `goBack` action.
- Using Interface Builder, Control-drag from the Fast Forward button to the web view to bind the selector to the `goForward` action.
- Discuss how, when the Rewind and Forward buttons are tapped, each button will call the `UIWebView` methods `goBack` or `goForward`, respectively.
- Run (⌘R) the app, view some web content, navigate through a few links, tap the back button, and tap the forward button.

Closing

What are the functional differences between our "easy browser" and Safari? What usability issues do you see in our app, and how might you fix them? What features would you add to the app? How might you incorporate web views into your own app?

Modifications and Extensions

- Investigate the `UIWebViewDelegate` protocol reference and the `UIWebView` class reference. Enhance the app, such that the back and forward buttons are enabled or disabled depending on the state of the web view.
- Use the `NSUserDefaults` API to load the last visited page when the app starts.

Resources

UIWebView Class Reference http://developer.apple.com/library/ios/documentation/uikit/reference/UIWebView_Class/Reference/Reference.html

Managing a User Interface Object's Connections https://developer.apple.com/library/ios/recipes/xcode_help-IB_connections/chapters/Connections.html

Cocoa Core Competencies: Selector http://developer.apple.com/library/ios/documentation/general/conceptual/devpedia-cocoacore/Selector.html