EasyBrowser
Lesson 4

Description

Set the controller as the text field delegate, and display the web page for the URL entered into the text field.

Learning Outcomes

- Apply the concepts of delegates and delegation, and practice connecting an interface `delegate` property to a view controller.
- Relate the use of protocols to delegation, and practice declaring the adoption of a protocol within a class.
- Discover how to implement controller actions connected to a text field.

Vocabulary

Document Outline	outlet connection	delegate
delegation	protocol	protocol adoption
`UITextFieldDelegate`	`UITextField`	

Materials

- **EasyBrowser Lesson 4** Xcode project
- **Delegates and Delegation** presentation
- **Protocols** presentation

Opening

How can we get the web view to open the URL that the user types in the text field?

Agenda

- Using Interface Builder and the Document Outline (⌘), Control-drag from the Round Style Text Field to the View Controller to set the View Controller as the text field `delegate`.
- Present the concept of delegates and delegation.
- Using the Xcode Documentation and API Reference (⇧⌘0), explore the `UITextFieldDelegate` protocol reference. Observe how the documentation states that the methods are optional.
- Discuss the `UITextFieldDelegate` protocol method `textFieldShouldReturn:`.
- Present the concept of protocols.
- In the view controller class, declare the adoption of the `UITextFieldDelegate` protocol.

```
class ViewController: UIViewController, UITextFieldDelegate {
```

- In the view controller implementation, delete the old `openPage:` method, and add an implementation of `textFieldShouldReturn:`.

```
func textFieldShouldReturn(textField: UITextField) -> Bool {
   textField.resignFirstResponder()
   if let url = NSURL(string: textField.text) {
      webView.loadRequest(NSURLRequest(URL: url))
   }
   return true
}
```

- Explain the will/did/should naming convention of iOS event handlers, and how "should" methods tend to return a `Bool`, which controls whether or not the event should complete.
- Discuss how the `UITextField` argument passed to `textFieldShouldReturn:` will refer to the text field in the interface.
- Explain how the `resignFirstResponder` method causes the keyboard to disappear.
- Run the app (⌘R), enter a valid http url (e.g., http://developer.apple.com), tap the **Go** key, and observe the web view load the content.

Closing

What happens if the `textFieldShouldReturn:` method returns `false`?

Modifications And Extensions

- One limitation of the text field input is that it requires the user to enter the http:// prefix. Another limitation is that there is no notification to the user when an invalid URL is entered. Enhance the behavior of the app to allow the user to type a URL without the http:// prefix, and to display an alert when an invalid URL is entered.

Resources

Start Developing iOS Apps Today: Using Design Patterns https://developer.apple.com/library/ios/referencelibrary/GettingStarted/RoadMapiOS/DesignPatterns.html

Cocoa Application Competencies: Delegation http://developer.apple.com/library/ios/documentation/general/conceptual/DevPedia-CocoaCore/Delegation.html

Cocoa Application Competencies: Protocol http://developer.apple.com/library/ios/documentation/General/Conceptual/DevPedia-CocoaCore/Protocol.html

Setting an Object's Delegate https://developer.apple.com/library/mac/recipes/xcode_help-IB_objects_media/Chapters/set_object_delegate.html

UIKit User Interface Catalog: Text Fields https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/UIKitUICatalog/UITextField.html

UITextField Class Reference https://developer.apple.com/library/prerelease/ios/documentation/UIKit/Reference/UITextField_Class/index.html

UITextFieldDelegate Protocol Reference https://developer.apple.com/library/prerelease/ios/documentation/UIKit/Reference/UITextFieldDelegate_Protocol/index.html

NSURL Class Reference http://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/Classes/NSURL_Class/Reference/Reference.html

NSURLRequest Class Reference http://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSURLRequest_Class/Reference/Reference.html

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

Cocoa Application Competencies for iOS: Responder Object http://developer.apple.com/library/ios/documentation/general/conceptual/Devpedia-CocoaApp/Responder.html