

## Simple Delegation

### 1. As a user, I want to view a website

- Create a new project called **MySafari**
- *Add this project as a repository to your GitHub account*
- Add a `UIWebView` to the Storyboard, using Interface Builder
- Connect the `UIWebView` to an `IBOutlet` in the `ViewController.m` file. Name the `IBOutlet` variable **webView**
- Add the `<UIWebViewDelegate>` protocol to your `@interface`
- Set the delegate for your web view to the View Controller

### 2. As a user, I want to enter a website URL so that I can view the web page

- Add a `UITextField` to the Storyboard
- Connect that text field to an `IBOutlet` in your `ViewController.m` and name it **urlTextField**

### 3. As a user, I want to use a URL friendly keyboard when typing

hint: You can do this in Storyboard

### 4. As a user, I want to use a GO button on the keyboard to navigate to that URL in the webView

- In your Storyboard, set the keyboard to display a GO button
- Add the `<UITextFieldDelegate>` after your `@interface`, this stating your intention to implement the `UITextFieldDelegate` protocol
- Implement `UITextFieldDelegate's` `textFieldShouldReturn:` protocol method in the `ViewController.m` file. In the body of this method, retrieve the value of **urlTextField** and create a `NSURL` using that value
- In the `textFieldShouldReturn:` method body in the `ViewController.m` file, add code to load the `NSURL` into **webView**

### 5. As a user, I want to show a spinner when my web page is loading

- Implement the `networkActivityIndicator` to show general network activity
- Show the spinner when the web page is loading
- Hide the spinner when the web page is not loading

## 6. As a user, I want to navigate back to a previous web page I visited

- Add a `UIView` to `ViewController`'s view using the Storyboard (We will refer to this as the **Button View**)
- Resize the `UIWebView`, making room for the new `UIView` beneath the `UIWebView`
- The new `UIView` should take up the width of the `UIWebView` (this `UIView` does NOT need to be hooked up to an `IBOutlet` variable)
- Add a `UIButton` to the **Button View** and set the text of the button to be, "<"
- Hook up the `UIButton` to an `IBAction` in the `ViewController.m` file
- Name the `IBAction` method **`onBackButtonPressed:`**
- In the **`onBackButtonPressed:`** method body in the `ViewController.m` file, add code to call the `goBack` method on **`webView`**
  - hint: Check out the `UIWebView`'s `goBack` method in the Apple Doc's

## 7. As a user, I want to navigate forward to a previous web page I visited

- Add a `UIButton` to the **Button View**
- Hook up the `UIButton` to an `IBAction` in the `ViewController.m` file
- Name the `IBAction` method **`onForwardButtonPressed:`**
- In the **`onForwardButtonPressed:`** method body in the `ViewController.m` file, add code to call the `goForward` method on the web view.

## 8. As a user, I want to stop a web page from loading

- Add a `UIButton` to the **Button View**
- Hook up the `UIButton` to an `IBAction` in the `ViewController.m` file
- Name the `IBAction` method **`onStopLoadingButtonPressed:`**
- In the **`onStopLoadingButtonPressed:`** method body in the `ViewController.m` file, add code to stop the web view from loading

## 9. As a user, I want to reload a web page

- Add a `UIButton` to the **Button View**
- Hook up the `UIButton` to an `IBAction` in the `ViewController.m` file
- Name the `IBAction` method **`onReloadButtonPressed:`**
- In the **`onReloadButtonPressed:`** method body in the `ViewController.m` file, add code to reload the contents of the web view

## 10. As a user, I want to manage the state of the back and forward buttons

- If the user is in a state such that the back button will take the user back a page, enable it. Otherwise, disable it.
- If the user is in a state such that the forward button will take the user forward a page, enable it. Otherwise, disable it.

## 11. As a user, I want to be able to type in urls without the "http://" prefix

If the value of `urlTextField` does not begin with "http://", use `stringWithFormat` to create a new `NSString`

## 12. As a user, I want to see a teaser for new features

- Add a `UIButton` to the **Button View** and make its text, "+"
- When the + button is pressed, display a `UIAlertController` displaying the string, "Coming soon!"

## 13. As a user, I want to have the address bar set to the current URL

- The `UITextField` should update to display the current URL every time the user goes to a new web page
- The `UITextField` should have a clear button associated with it to quickly clear out an existing URL instead of having to repeatedly hit the delete button
- Add some placeholder text to the `UITextField` when it's empty. This should indicate when there is no webpage or URL in the `UITextField`, that a URL can be typed here

## 14. As a user, I want to see the current web site's title in the navigation bar

Add a label somewhere and display the current web page's title (this is **NOT** the URL)

## 15. As a user, I want to concentrate on my web content

- Detect when the `UIWebView` scrolls
- When scrolling down, alter location and opacity of the `UITextField` to hide it or show it based on scroll direction