RSSReader
Lesson 7

Description

Add the individual RSS feed URLs as user defined runtime attributes for each view controller, and use the attribute to retrieve different data within each view controller.

Learning Outcomes

- Distinguish differences between class definitions and instances.
- Practice creating user defined runtime attributes using Interface Builder `IBInspectable` properties.
- Describe how storyboards assign user defined runtime attributes to `IBInspectable` controller properties.
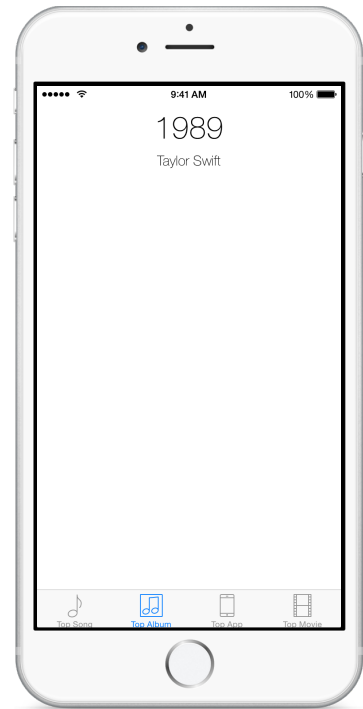
Vocabulary

RSS feed	URL	property
user defined runtime attribute	inspectable attribute	`@IBInspectable`

Materials

- **RSSReader Lesson 7** Xcode project
- RSS feed URLs text file (**feedurls.txt**)
- Internet connectivity to the **ax.itunes.apple.com** domain

Opening

If each controller will perform an identical operation, but each will use a different RSS feed url, how can we use different RSS feeds with each controller?

Agenda

- Discuss how the songs, albums, apps and movies RSS feeds all have a similar data structure, but that each controller instance will need to use a different RSS feed URL.

- Within the `TopMediaController` interface, change the `titleText` property to `feedURL`.

  ```
  @IBInspectable var feedURL: String!
  ```

- Using Interface Builder, select a view controller, open the Identity Inspector (⌥⌘3), and delete the old `titleText` user defined runtime attribute. Repeat the attribute removal for each view controller.

- Using Interface Builder, select the Top Song view controller, open the Attributes Inspector (⌥⌘4), and change the Feed URL attribute value to **http://ax.itunes.apple.com/WebObjects/MZStoreServices.woa/ws/RSS/topsongs/limit=1/json**.

- Update the inspectable Feed URL attribute for the remaining view controllers, using the appropriate url provided in the **feedurls.txt** file.

- Update the implementation of `viewDidLoad` to use the `feedURL` property.

  ```
  func viewDidLoad() {
      super.viewDidLoad()
      let request = NSURLRequest(URL: NSURL(string: feedURL)!)
      ...
  ```

- Discuss how each controller instance will rely on the particular value of its `feedURL` property to retrieve the RSS data.

- Run the app (⌘R), interact with each tab, and observe how each tab displays the respective top song, album, app and movie data.

Closing

Use a web browser to examine the actual RSS feed data. Do you see a way we might retrieve an image for the song, album, app and movie?

Modifications and Extensions

- Increase the robustness of the controller by using optional binding to check that the `feedURL` exists before using it to instantiate an `NSURLRequest` object. Display an appropriate alert view if a problem occurs.

Resources

Apple RSS Feeds http://www.apple.com/rss/

The Swift Programming Language: Properties https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/Properties.html

Interface Builder Help: Adding User Defined Runtime Attributes https://developer.apple.com/library/mac/recipes/xcode_help-interface_builder/Chapters/AddUserDefinedRuntimeAttributes.html

Creating a Custom View that Renders in Interface Builder https://developer.apple.com/library/ios/recipes/xcode_help-IB_objects_media/Chapters/CreatingaLiveViewofaCustomObject.html

The Swift Programming Language: Attributes https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/Attributes.html

NSURL Class Reference https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSURL_Class/

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