

**Handheld Business Watcher News App -> A neat-designed iOS Newsfeed application which contains richful most-recent news about business stocks, built from scratch in UIKit, incorporated with newapi.org APIs to a custom UI and cache images and CGRect.**

**Note: Xcode project renaming rule: change info.plist File in Build Settings .**

### **Developing Languages, Tools, and Techniques Needed:**

Xcode 13.4.1 iOS 10-15.5 (iOS 16 causes Assertion Errors)

**Best testing simulator version to avoid errors: iPhone12 Pro Max iOS 14.4**

News API <https://newsapi.org/docs/get-started#search>

SwiftUI <https://developer.apple.com/xcode/swiftui/>

SafariServices Documentation <https://developer.apple.com/documentation/safariservices>

Swift 5 <https://developer.apple.com/swift/>

### **Prerequisites & Setups:**

Obtain an API url from newsapi.org for the most-recent TechCrunch news.

Create a new object to call APIs APICaller.swift:

Handle potential errors with completion method, resume task:

```
public func getTopStories(completion: @escaping{
...
        if let error = error{
            completion(.failure(error))
        }
        else if let data = data {
            do {
                let result = try JSONDecoder().decode(String.self,
from: data)
            }
            catch {
                completion(.failure(error))
            }
        }
...
task.resume()
```

Create some API responses models:

```
struct APIResponse: Codable {
    let articles: [Article]
}
struct Article: Codable {
    let source: Source
    let title: String
    let description: String?
    let url: String?
    let urlToImage: String?
    let publishedAt: String
}
```

```
struct Source: Codable {  
    let name: String }
```

ERROR: Title not showing in Xcode simulator. DEBUGGING: In Main.storyboard,  
Editor-> Embed in -> Navigation Controller.

### **Synchronous Developing Notes:**

#### Table View:

```
func tableView (_ tableView: UITableView, numberOfRowsInSection:  
section: Int)-> Int {  
    return 0  
}
```

```
func tableView (_ tableView: UITableView, cellForRowAt indexPath:  
IndexPath) -> UITableViewCell {  
    let cell = tableView.dequeueReusableCell(  
        withIdentifier: "cell",  
        for: indexPath  
    )  
    cell.textLabel?.text = "Something"  
    return cell  
}
```

```
func tableView(_ tableView: UITableView, didSelectRowAt indexPath:  
IndexPath) {  
    tableView.deselectRow(at: indexPath, animated: true)  
}
```

Also create a frame for table view:

```
override func viewDidLoadSubviews() {  
    super.viewDidLoadSubviews()  
    tableView.frame = view.bounds  
}
```

Create a new Cocoa Touch file NewsTableViewCell.swift:

```
class NewsTableViewCell: UITableViewCell {  
    static let identifier = "NewsTableViewCell"  
  
    override init(style: UITableViewCell.CellStyle, reuseIdentifier:  
String?){  
        super.init(style: style, reuseIdentifier: reuseIdentifier)  
    }  
    required init?(coder: NSCoder){  
        fatalError()  
    }  
    override func layoutSubviews(){
```

```

        super.layoutSubviews()
    }
    override func prepareForReuse() {
        super.prepareForReuse()
    }

```

Configure the cell with viewModel:

Create viewModel first:

```

class NewsTableViewCellViewModel {
    let title: String
    let subtitle: String
    let imageURL: URL?
    let imageData: Data? = nil
    init(
        title: String,
        subtitle: String,
        imageURL: URL?
    ) {
        self.title = title
        self.subtitle = subtitle
        self.imageURL = imageURL
    }
}

```

Then add viewModel configuration:

```

func configure(with viewModel: NewsTableViewCellViewModel) {
    newsTitleLabel.text = viewModel.title
    subtitleLabel.text = viewModel.subtitle
}

```

Also configure the images:

```

if let data = viewModel.imageData {
    newsImageView.image = UIImage(data: data)
}

```

Now the top 20 news headlines fetched and loaded successfully:

**news headlines fetched.PNG**

Fatal Error:

Error Message: [Assert] UINavigationController decoded as unlocked for UINavigationController, or navigationBar delegate set up incorrectly. Inconsistent configuration may cause problems.

navigationController=<UINavigationController: 0x7f7e5a00d400>,  
 navigationBar=<UINavigationController: 0x7f7e5d80ae60; frame = (0 47; 0 50);  
 opaque = NO; autoresize = W; layer = <CALayer: 0x600002ab6f80>>  
 delegate=0x7f7e5a00d400

DEBUGGING: iOS 16 incompatibility error. Try downloading Simulator packages <= iOS 15.5.

Fetch images by downloading the urls:

```

//fetch
URLSession.shared.dataTask(with: url) { [weak self] data, __, error in

```

```

        guard let data = data, error == nil else {
            return
        }
        viewModel.imageData = data
        DispatchQueue.main.async {
            self?.newsImageView.image = UIImage(data: data)
        }
    }.resume()

```

Now news headlines, details and images along are properly aligned:

**news details and images are properly aligned.PNG**

Import Safari Service:

```
import SafariServices
```

Make Safari service presented:

```

func tableView(_ tableView: UITableView, didSelectRowAt indexPath:
IndexPath) {
    tableView.deselectRow(at: indexPath, animated: true)
    let article = articles[indexPath.row]
    guard let url = URL(string: article.url ?? "") else {
        return
    }
    let vc = SFSafariViewController(url: url)
    present(vc, animated: true)
}

```

Now all the visible news are clickable and will redirect us to its safari viewing page:

**redirected to safari viewer.PNG**