

HIGHER SCHOOL OF ECONOMICS UNIVERSITY

HOMEWORK #5

dedicated to Research Seminar
«Development of applications for the Apple iOS platform»

MOSCOW 2022

Theme and Objective

Theme	Working with the Web
Objective	Mastering fetching data from Web and displaying it

Task description

You should create a third NewsListViewController, which appears when news button (see previous homework) is tapped. The controller must display tableview with cells. Each cell is responsible for the news article (Image, Title and Description). When you tap on a cell, another NewsViewController appears and you can see full image, title and full text.

Task requirements

During this task:

- You are not allowed to use React, SwiftUI, ObjC-C or anything except vanilla Swift.
- You cannot use StoryBoard to layout the app.
- You are allowed to use any architecture for this app.
- You can use UIView extension provided to you before.

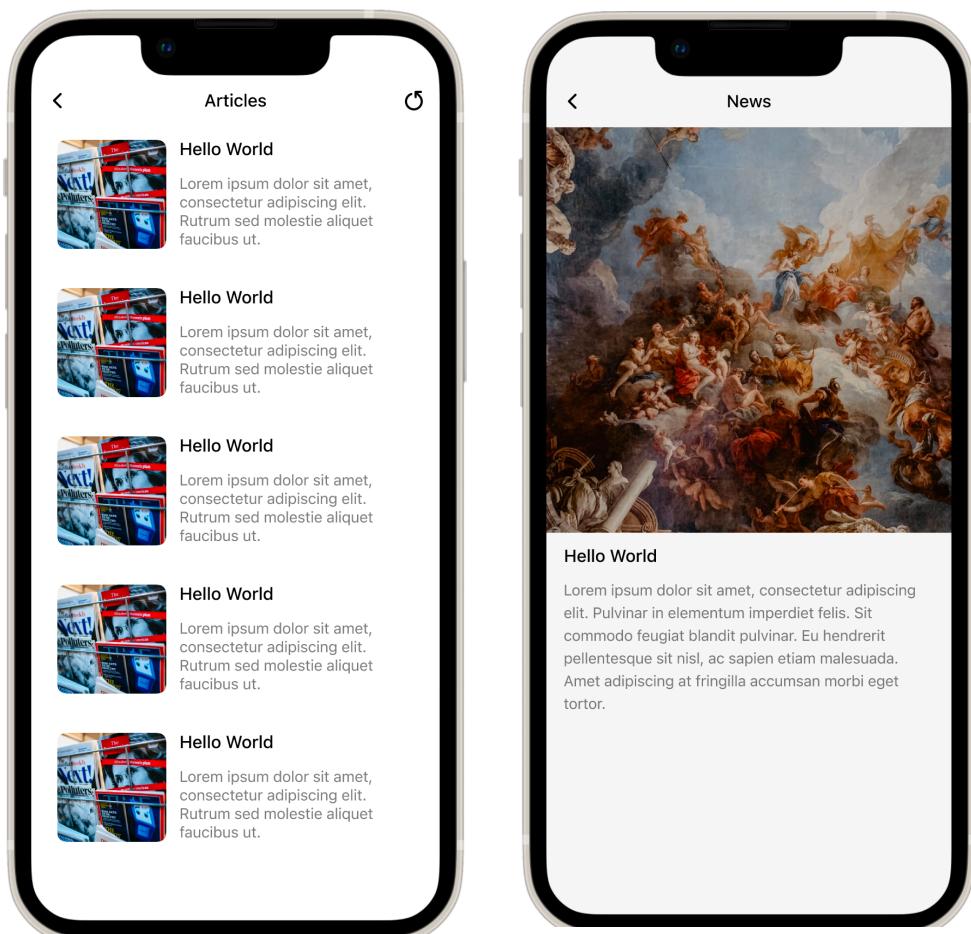
Grading

Grade	Task
0	The task was not submitted or it contains nothing
1	The task has nothing to do with the assignment.
2	NewsListViewController is created
3	NewsListViewController appears when you press the corresponding button
4	NewsListViewController displays cells
5	Each cells contains imageView, titleLabel and descriptionLabel and represents corresponding information of the news article
6	NewsListViewController contains button to return to previous controller
7	NewsListViewController contains button which refreshes dataSource (fetches news articles again)

HOMEWORK #5

8	Cells with valid information (article details) are displayed when NewsListViewController is present
9	Images of the news are cashed in the memory.
10	When data is loading from the web, shimmer orskeleton is shown on the cell

Visualisation



Tutorial

Point 2

Before we start we want to use some news api to fetch data. The good option one can find the following website: <https://newsapi.org/docs/get-started#top-headlines>. To use it we will need the api key which can be obtained after registration.

Now, when preparation work is done we may continue with coding.

We need to create another View controller as in previous HW.

```
import UIKit

final class NewsListViewController: UIViewController {
    private var tableView = UITableView(frame: .zero, style: .plain)

    override func viewDidLoad() {
        super.viewDidLoad()

        setupUI()
    }

    private func setupUI() {
        view.backgroundColor = .systemBackground
    }
}
```

Point 3

Also we need to add target to our button and attach the action which pushes newly created ViewController. (Just like in HW4 except for the navigation controller function)

```
//Inside WelcomeViewController
let newsButton = makeMenuButton(title: "📰")
newsButton.addTarget(self, action: #selector(newsButtonPressed), for: .touchUpInside)
//...

@objc
private func newsButtonPressed() {
    let newsListController = NewsListViewController()
    navigationController?.pushViewController(newsListController, animated: true)
}
```

Point 4

As in previous HW we should conform ViewController to required protocols, implement functions and set UI of the tableView.

HOMEWORK #5

```
import UIKit

final class NewsListViewController: UIViewController {
    private var tableView = UITableView(frame: .zero, style: .plain)
    private var isLoading = false
    private var newsViewModels = []()

    override func viewDidLoad() {
        super.viewDidLoad()

        setupUI()
    }

    private func setupUI() {
        view.backgroundColor = .systemBackground
        configureTableView()
    }

    private func configureTableView() {
        setTableViewUI()
        setTableViewDelegate()
        setTableViewCell()
    }

    private func setTableViewDelegate() {
        tableView.delegate = self
        tableView.dataSource = self
    }

    private func setTableViewUI() {
        view.addSubview(tableView)
        tableView.backgroundColor = .clear
        tableView.separatorStyle = UITableViewCell.SeparatorStyle.none
        tableView.rowHeight = 120
        tableView.pinLeft(to: view)
        tableView.pinTop(to: view.safeAreaLayoutGuide.topAnchor)
        tableView.pinRight(to: view)
        tableView.pinBottom(to: view)
    }

    private func setTableViewCell() {
        tableView.register(NewsCell.self, forCellReuseIdentifier: NewsCell.reuseIdentifier)
    }

    @objc
    private func goBack() {
        self.navigationController?.popViewController(animated: true)
    }
}

extension NewsListViewController: UITableViewDataSource {
    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        if isLoading {
            return newsViewModels.count
        }
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        if isLoading {
            let viewModel = newsViewModels[indexPath.row]
            if let newsCell = tableView.dequeueReusableCell(withIdentifier: NewsCell.reuseIdentifier, for: indexPath) as? NewsCell {
                newsCell.configure(with: viewModel)
                return newsCell
            }
        }
        return UITableViewCell()
    }
}

extension NewsListViewController: UITableViewDelegate {
    func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) { }
}
```

Point 5

Now we should create a custom NewsCell as we did before.

We use hardcoded values up to now in order to test whether NewsListViewController displays cells. A bit later we will off public function for cell configuration.

(«landscape» is the image I added to assets manually to test whether Image is displayed in the desirable way).

Also you may see that we use vanilla constraints here. UIView extension is useful and time-saving, but we shouldn't forget how to use vanilla tools, I used it intentionally.

```
import UIKit

final class NewsCell: UITableViewCell {
    static let reuseIdentifier = "NewsCell"
    private let newsImageView = UIImageView()
    private let newsTitleLabel = UILabel()
    private let newsDescriptionLabel = UILabel()

    // MARK: - Init
    override init(style: UITableViewCell.CellStyle, reuseIdentifier: String?) {
        super.init(style: style, reuseIdentifier: reuseIdentifier)

        self.selectionStyle = .none
        setupView()
    }

    @available (*, unavailable)
    required init?(coder: NSCoder) {
        fatalError("init(coder:) has not been implemented")
    }

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

HOMEWORK #5

```
private func setupView() {
    setupImageView()
    setupTitleLabel()
    setupDescriptionLabel()
}

private func setupImageView() {
    newsImageView.image = UIImage(named: "landscape")
    newsImageView.layer.cornerRadius = 8
    newsImageView.layer.cornerCurve = .continuous
    newsImageView.clipsToBounds = true
    newsImageView.contentMode = .scaleAspectFill
    newsImageView.backgroundColor = .secondarySystemBackground

    contentView.addSubview(newsImageView)
    newsImageView.topAnchor.constraint(equalTo: contentView.topAnchor, constant: 12).isActive
    = true
    newsImageView.leftAnchor.constraint(equalTo: contentView.leftAnchor, constant:
16).isActive = true
    newsImageView.bottomAnchor.constraint(equalTo: contentView.bottomAnchor, constant:
-12).isActive = true
    newsImageView.pinWidth(to: newsImageView.heightAnchor)
}

private func setupTitleLabel() {
    newsTitleLabel.text = "Hello"
    newsTitleLabel.font = .systemFont(ofSize: 16, weight: .medium)
    newsTitleLabel.textColor = .label
    newsTitleLabel.numberOfLines = 1

    contentView.addSubview(newsTitleLabel)
    newsTitleLabel.translatesAutoresizingMaskIntoConstraints = false
    newsTitleLabel.heightAnchor.constraint(equalToConstant:
newsTitleLabel.font.lineHeight).isActive = true
    newsTitleLabel.leadingAnchor.constraint(equalTo: newsImageView.trailingAnchor, constant:
12).isActive = true
    newsTitleLabel.topAnchor.constraint(equalTo: contentView.topAnchor, constant: 12).isActive
= true
    newsTitleLabel.trailingAnchor.constraint(equalTo: contentView.trailingAnchor, constant:
-12).isActive = true
}

private func setupDescriptionLabel() {
    newsDescriptionLabel.text = "World"
    newsDescriptionLabel.font = .systemFont(ofSize: 14, weight: .regular)
    newsDescriptionLabel.textColor = .secondaryLabel
    newsDescriptionLabel.numberOfLines = 0

    contentView.addSubview(newsDescriptionLabel)
    newsDescriptionLabel.translatesAutoresizingMaskIntoConstraints = false
    newsDescriptionLabel.leadingAnchor.constraint(equalTo: newsImageView.trailingAnchor,
constant: 12).isActive = true
    newsDescriptionLabel.topAnchor.constraint(equalTo: newsTitleLabel.bottomAnchor, constant:
12).isActive = true
    newsDescriptionLabel.trailingAnchor.constraint(equalTo: contentView.trailingAnchor,
constant: -16).isActive = true
    newsDescriptionLabel.bottomAnchor.constraint(lessThanOrEqualTo: contentView.bottomAnchor,
constant: -12).isActive = true
}
```

Now we need to implement a model for the news, viewModel which will be preparing and transforming data and function to fetch the news.

```

import UIKit

final class NewsViewModel {
    let title: String
    let description: String
    let imageURL: URL?
    var imageData: Data? = nil

    init(title: String, description: String, imageURL: URL?) {
        self.title = title
        self.description = description
        self.imageURL = imageURL
    }
}

//inside NewsListViewController

private func fetchNews() {
    ApiService.shared.getTopStories { [weak self] result in
        switch result {
        case .success(let articles):
            self?.newsViewModels = articles.compactMap{
                NewsViewModel(
                    title: $0.title,
                    description: $0.description ?? "No description",
                    imageURL: URL(string: $0.urlToImage ?? ""))
            }

            DispatchQueue.main.async {
                self?.isLoading = false
                self?.tableView.reloadData()
            }
        case .failure(let error):
            print(error)
        }
    }
}

```

Important: during implementation of the functions above we have used DispatchQueue. Concurrency topic is highly recommended for reading for better understanding:

- 1) <https://habr.com/ru/post/320152/>
- 2) <https://www.hackingwithswift.com/read/9/4/back-to-the-main-thread-dispatchqueuemain> .

Point 6

We have created new controller which contains tableView and displays cell. Nice! But we cannot return to our WelcomeViewController unfortunately.

In order to fix this lets implement a couple of functions.

HOMWORK #5

One more thing.

When we tap on the cell nothing happens, but we want to present new NewsViewController with some detailed information. Let's fix that.

```
import UIKit

final class NewsViewController: UIViewController {
    private var imageView = UIImageView()
    private var titleLabel = UILabel()
    private var descriptionLabel = UILabel()

    override func viewDidLoad() {
        super.viewDidLoad()

        setupUI()
    }

    // MARK: - Private methods
    private func setupUI() {
        view.backgroundColor = .systemBackground
        setupNavbar()
        setImageView()
        setTitleLabel()
        setDescriptionLabel()
    }

    private func setupNavbar() {
        navigationItem.title = "News"
        navigationItem.leftBarButtonItem = UIBarButtonItem(
            image: UIImage(systemName: "chevron.left"),
            style: .plain,
            target: self,
            action: #selector(goBack)
        )
        navigationItem.leftBarButtonItem?.tintColor = .label
    }

    private func setImageView() {
        imageView.image = UIImage(named: "landscape")
        imageView.clipsToBounds = true
        imageView.contentMode = .scaleAspectFill
        view.addSubview(imageView)
        imageView.pin(to: view, [.left: 0, .right: 0])
        imageView.pinTop(to: view.safeAreaLayoutGuide.topAnchor)
        imageView.pinHeight(to: imageView.widthAnchor, 1)
    }
}
```

HOMEWORK #5

```
private func setTitleLabel() {
    titleLabel.text = "Hello"
    titleLabel.font = .systemFont(ofSize: 16, weight: .medium)
    titleLabel.numberOfLines = 0
    titleLabel.textColor = .label

    view.addSubview(titleLabel)

    titleLabel.pinTop(to: imageView.bottomAnchor, 12)
    titleLabel.pin(to: view, [.left: 16, .right: 16])
    titleLabel.heightAnchor.constraint(greaterThanOrEqualToConstant: 16).isActive = true
}

private func setDescriptionLabel() {
    descriptionLabel.text = «World»
    descriptionLabel.font = .systemFont(ofSize: 14, weight: .regular)
    descriptionLabel.numberOfLines = 0
    descriptionLabel.textColor = .secondaryLabel

    view.addSubview(descriptionLabel)
    descriptionLabel.pin(to: view, [.left: 16, .right: 16])
    descriptionLabel.pinTop(to: titleLabel.bottomAnchor, 8)
}

// MARK: - Objc functions
@objc
func goBack() {
    self.navigationController?.popViewController(animated: true)
}
}
```

```
//Add to NewsListViewController

extension NewsListViewController: UITableViewDelegate {
    func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
        if !isLoading {
            let newsVC = NewsViewController()
            navigationController?.pushViewController(newsVC, animated: true)
        }
    }
}
```

Point 7

To implement this functionality, we need to add new button to navigationController and implement objc function.

Point 8

Everything is functioning nicely, but we do not display valid information about news article when we present NewsViewController. We should erase hardcoded values in it and make a public configure function from newsViewModel.

```
import UIKit

final class NewsViewController: UIViewController {
    private var imageView = UIImageView()
    private var titleLabel = UILabel()
    private var descriptionLabel = UILabel()

    override func viewDidLoad() {
        super.viewDidLoad()

        setupUI()
    }

    // MARK: - Private methods
    private func setupUI() {
        view.backgroundColor = .systemBackground
        setupNavbar()
        setImageView()
        setTitleLabel()
        setDescriptionLabel()
    }

    private func setupNavbar() {
        navigationItem.title = "News"
        navigationItem.leftBarButtonItem = UIBarButtonItem(
            image: UIImage(systemName: "chevron.left"),
            style: .plain,
            target: self,
            action: #selector(goBack)
        )
        navigationItem.leftBarButtonItem?.tintColor = .label
    }

    private func setImageView() {
        imageView.clipsToBounds = true
        imageView.contentMode = .scaleAspectFill
        view.addSubview(imageView)
        imageView.pin(to: view, [.left: 0, .right: 0])
        imageView.pinTop(to: view.safeAreaLayoutGuide.topAnchor)
        imageView.pinHeight(to: imageView.widthAnchor, 1)
    }

    private func setTitleLabel() {
        titleLabel.font = .systemFont(ofSize: 16, weight: .medium)
        titleLabel.numberOfLines = 0
        titleLabel.textColor = .label

        view.addSubview(titleLabel)

        titleLabel.pinTop(to: imageView.bottomAnchor, 12)
        titleLabel.pin(to: view, [.left: 16, .right: 16])
        titleLabel.heightAnchor.constraint(greaterThanOrEqualToConstant: 16).isActive = true
    }

    private func setDescriptionLabel() {
        descriptionLabel.font = .systemFont(ofSize: 14, weight: .regular)
        descriptionLabel.numberOfLines = 0
        descriptionLabel.textColor = .secondaryLabel

        view.addSubview(descriptionLabel)
        descriptionLabel.pin(to: view, [.left: 16, .right: 16])
        descriptionLabel.pinTop(to: titleLabel.bottomAnchor, 8)
    }
}
```

HOMWORK #5

```
// MARK: - Public Methods
public func configure(with viewModel: NewsViewModel) {
    titleLabel.text = viewModel.title
    descriptionLabel.text = viewModel.description
    if let data = viewModel.imageData {
        imageView.image = UIImage(data: data)
    }
    else if let url = viewModel.imageURL {
        URLSession.shared.dataTask(with: url) { [weak self] data, _, error in
            guard let data = data else {
                return
            }
            viewModel.imageData = data
            DispatchQueue.main.async {
                self?.imageView.image = UIImage(data: data)
            }
        }.resume()
    }
}

// MARK: - Objc functions
@objc
func goBack() {
    self.navigationController?.popViewController(animated: true)
}
```

Also we need to add configuration of the NewsViewController to NewsListViewController.

```
extension NewsListViewController: UITableViewDelegate {
    func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
        if !isLoading {
            let newsVC = NewsViewController()
            newsVC.configure(with: newsViewModels[indexPath.row])
            navigationController?.pushViewController(newsVC, animated: true)
        }
    }
}
```

Info

If your project contains [interesting solutions](#), exceptional code style, good presentation, unique references or well designed clean code ([patterns](#), paradigms) up to two bonus points might be added. This means the maximum grade a student can receive per homework is 12. If interesting solution repeats in different projects bonus points won't be added. It is pointless to try and prove to us that your solution is interesting. There won't be any appeals regarding bonus points.

If the task is overdue, for each day [after deadline](#) your maximum grade goes down by 10%. Bonus points might not be added for projects submitted late. The maximum penalty is 50%. Outdated submission can get you up to 5 points, but if you already submitted and received a grade you can submit points you didn't manage to do in time. Therefor you can make your grades better.

The grade will be decreased for code style violation and bad code. If you stumble across any questions feel free to contact teachers or assistants! Our goal is to help you develop skills required to create iOS Apps.

Conclusion

We have mastered working with the Web UITableView and UITableViewCells, also learned how switch between threads and make asynchronous tasks. As a results we created a nice looking app which fetches real US news and presents it.

2:57

< Articles



Biden knocks Truss economic pl...

U.S. President Joe Biden on Saturday criticized British Prime Minister Liz Truss's original economic plan as a...



Jeffrey Dahmer Victim's Mom Sl...

Jeffery Dahmer-inspired Halloween costumes have already gotten a ton of backlash, but more than anything th...



Herschel Walker Flashes Seemi...

Herschel Walker has been trolled on social media after pulling out a seemingly bogus police badge after...



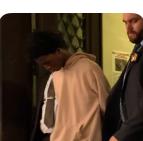
Xi Jinping's expected coronatio...

Chinese leader Xi Jinping on Sunday vowed to steer China through grave challenges toward national rejuvenati...



Stockton, Calif., police arrest su...

STOCKTON, Calif. (AP) — A man suspected of killing six men and wounding a woman in a series of sho...



NYC Crime: 18-year-old arreste...

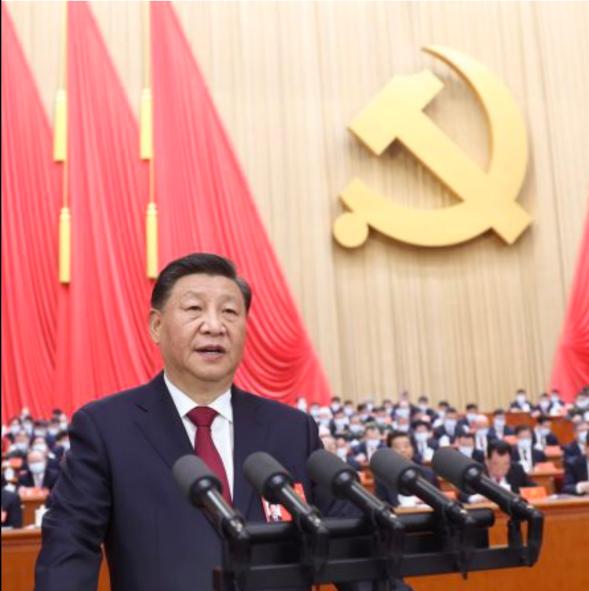
An 18-year-old was arrested and charged in connection with the fatal shooting death of a teen on the sub...



Padres eliminates Dodgers, adva...

8:01

< News



China's Xi opens Party Congress with speech tackling Taiwan, Hong Kong and zero-Covid - CNN

Chinese leader Xi Jinping on Sunday vowed to steer China through grave challenges toward national rejuvenation, advancing a nationalistic vision that has put it on a collision path with the West.