

.....
.....
**iOS DESIGN
PATTERNS**
.....
.....



HANDS-ON CHALLENGES

iOS Design Patterns

Joshua Greene

Copyright ©2017 Razeware LLC.

Notice of Rights

All rights reserved. No part of this book or corresponding materials (such as text, images, or source code) may be reproduced or distributed by any means without prior written permission of the copyright owner.

Notice of Liability

This challenge and all corresponding materials (such as source code) are provided on an "as is" basis, without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose, and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in action of contract, tort or otherwise, arising from, out of or in connection with the software or the use of other dealing in the software.

Trademarks

All trademarks and registered trademarks appearing in this book are the property of their own respective owners.

Table of Contents: Overview

| | |
|---|---|
| Composition over Inheritance - Challenge..... | 5 |
|---|---|

Table of Contents: Extended

| | |
|---|---|
| Composition over Inheritance - Challenge..... | 5 |
| Challenge..... | 5 |
| Hint..... | 6 |
| Challenge Solution | 7 |
| Über challenge | 9 |

Composition over Inheritance - Challenge

By Joshua Greene

The app is really coming along- you can even send quote requests! However, you still can't *view* the quotes requests you've sent. You need to implement this! ;]

Challenge

Add a new method to **NetworkClient** for getting quote requests. Use this as a starting point:

```
public func getQuotes(success _success: @escaping ([QuoteRequest]) -> Void,
                    failure _failure: @escaping (NetworkError) -> Void)
{
    let success: ([QuoteRequest]) -> Void = { quoteRequests in
        DispatchQueue.main.async { _success(quoteRequests) }
    }
    let failure: (NetworkError) -> Void = { error in
        DispatchQueue.main.async { _failure(error) }
    }

    // ... write this...!
}
```

The endpoint for getting quote requests uses the **GET** HTTP verb and is created like this:

```
baseUrl.appendingPathComponent("quotes")
```

This endpoint returns a JSON array of **QuoteRequest** objects.

This endpoint requires authentication, so you'll need to make sure you set the authentication headers on the request from the AuthToken. See **sendQuoteRequest(for:, success:, failure:)** for an example.

Lastly, update **QuoteRequestsViewController** to use it.

Hint

Start by copying the code from **sendQuoteRequest(for:, success:, failure:)** into **getQuotes(success:, failure:)** and modifying it.

Instead casting **jsonObject** as dictionary, however, you'll cast this as an array of dictionaries, which will have type `[[AnyHashable: Any]]`.

You can then use the class constructor `array(from jsonArray:) -> [QuoteRequest]` on `QuoteRequest` to convert the JSON array into an array of `QuoteRequest` instances.

Challenge Solution

Add the following to **NetworkClient.swift**:

```
public func getQuotes(success _success: @escaping ([QuoteRequest]) ->
Void,
                    failure _failure: @escaping (NetworkError) -> Void)
{
    let success: ([QuoteRequest]) -> Void = { quoteRequests in
        DispatchQueue.main.async { _success(quoteRequests) }
    }
    let failure: (NetworkError) -> Void = { error in
        DispatchQueue.main.async { _failure(error) }
    }

    authClient.requestAuthToken(self, success: { [weak self] (token, _) in

        guard let strongSelf = self else { return }
        let url = strongSelf.baseURL.appendingPathComponent("quotes")
        var request = URLRequest(url: url)
        token.setAuthenticationHeaders(on: &request)

        let task = strongSelf.session.dataTask(with: request) { (data,
response, error) in

            guard let httpResponse = response as? HTTPURLResponse,
httpResponse.statusCode.isSuccessHTTPCode,
            let data = data,
            let jsonObject = try? JSONSerialization.jsonObject(with: data),
            let jsonArray = jsonObject as? [[AnyHashable: Any]] else {
                if let error = error {
                    let networkError = NetworkError(error: error)
                    print("Send quote failed: \(networkError)")
                    failure(networkError)
                } else {
                    let networkError = NetworkError(response: response)
                    guard !networkError.isAuthError else {
                        print("Get quotes failed: not authenticated... invalidating
token and retrying...")
                        strongSelf.authClient.invalidateAuthToken()
                        strongSelf.getQuotes(success: success, failure: failure)
                        return
                    }
                    guard networkError != .notFound else {
                        success([])
                        return
                    }
                    print("Get quotes failed: \(networkError)")
                    failure(networkError)
                }
            }
            return
        }
        let array = QuoteRequest.array(from: jsonArray)
        success(array)
    }
}
```

```

        task.resume()
    }, userCancelled: {
        print("Send quote failed: user cancelled")
        failure(.userCancelled)
    })
}

```

Navigate to **QuoteRequestsViewController.swift** and add these properties to this class:

```

// MARK: - Instance Properties
internal let networkClient = NetworkClient.shared
internal var quoteRequests: [QuoteRequest] = []

internal let dateFormatter: DateFormatter = {
    let dateFormatter = DateFormatter()
    dateFormatter.dateStyle = .short
    dateFormatter.timeStyle = .short
    return dateFormatter
}()

```

Replace `loadQuoteRequests()` with this:

```

internal func loadQuoteRequests() {
    tableView.refreshControl?.beginRefreshing()

    networkClient.getQuotes(success: { [weak self] quotes in
        guard let strongSelf = self else { return }
        strongSelf.quoteRequests = quotes
        strongSelf.tableView.reloadData()
        strongSelf.tableView.refreshControl?.endRefreshing()
    }, failure: { error in
        print("Load quote requests failed: \(error)")
    })
}

```

Replace `tableView(_, numberOfRowsInSection:)` with this:

```

public func tableView(_ tableView: UITableView,
                     numberOfRowsInSection section: Int) -> Int {
    return quoteRequests.count
}

```

Replace `tableView(_, cellForRowAt:)` -> `UITableViewCell` with this:

```

public func tableView(_ tableView: UITableView,
                     cellForRowAt indexPath: IndexPath) ->
UITableViewCell {
    let identifier = "QuoteRequestCell"
    let cell = tableView.dequeueReusableCell(withIdentifier: identifier)
    as! QuoteRequestTableViewCell
}

```



```
let quoteRequest = self.quoteRequests[indexPath.row]

cell.productTitleLabel?.text =
    quoteRequest.product.title
cell.requestedDateLabel?.text =
    dateFormatter.string(from: quoteRequest.promised)
cell.promiseDateLabel?.text =
    dateFormatter.string(from: quoteRequest.created)

return cell
}
```

Build and Run, create a new "Quote Request," and verify you can now see your previously submitted quote requests.

Über challenge

This Über challenge is pack full of action! There's two challenges this time: an "easy" and "hard" challenge.

The "easy" challenge is:

Wouldn't it be nice if there was a "Your Quotes" option that took you to your previous quote requests on the MenuTableViewController scene within Main.storyboard? Yes, it sure would! Make it so. ;]

Hint: you can use a **Storyboard Reference** added to **Main.storyboard** to accomplish this. Storyboard references rock! ;]

The "hard" challenge is:

Remember the MVVM design pattern? Oh yeah, it's still relevant!

Instead of using QuoteRequest directly within QuoteRequestsViewController, create a new QuoteRequestViewModel, and update QuoteRequestsViewController to use this instead.

If you get stuck, check out the completed challenge in the resources for this video.