

NETWORKING WITH URLSession



PART 11: ARCHITECTURE

SO MUCH CODE!

```
let configuration = URLSessionConfiguration.default
--
var request = URLRequest(url: URL(string: "http://localhost:3000/posts/")!)
let task = defaultSession.dataTask(with: request) { data, response, error in
    func urlSession(_ session: URLSession, dataTask: URLSessionDataTask,
        didReceive response: URLResponse, completionHandler: @escaping
        do {
            let array = try JSONSerialization.jsonObject(with: data, options: [])
                as? [JSONDictionary]
            for dictionary in array! {
                if let author = dictionary["author"] as? String,
                    let title = dictionary["title"] as? String,
                    let id = dictionary["id"] as? Int {
                        posts.append(Post(id: id, author: author, title: title))
                    }
            }
        } catch {
            print(error.localizedDescription)
        }
    }
}
```

CREATING REQUESTS

```
public enum PostRouter {  
  
    // Possible requests  
    case getAll  
    case get(Int)  
    case create([String: Any])  
    case update(Int, [String: Any])  
    case delete(Int)  
    . . .  
}
```



INITFROMJSON

```
extension Post {  
  init?(dictionary: JSONDictionary) {  
    guard let author = dictionary["author"] as? String,  
          let title = dictionary["title"] as? String,  
          let id = dictionary["id"] as? Int else { return nil }  
  
    self.init(id: id, author: author, title: title)  
  }  
}
```

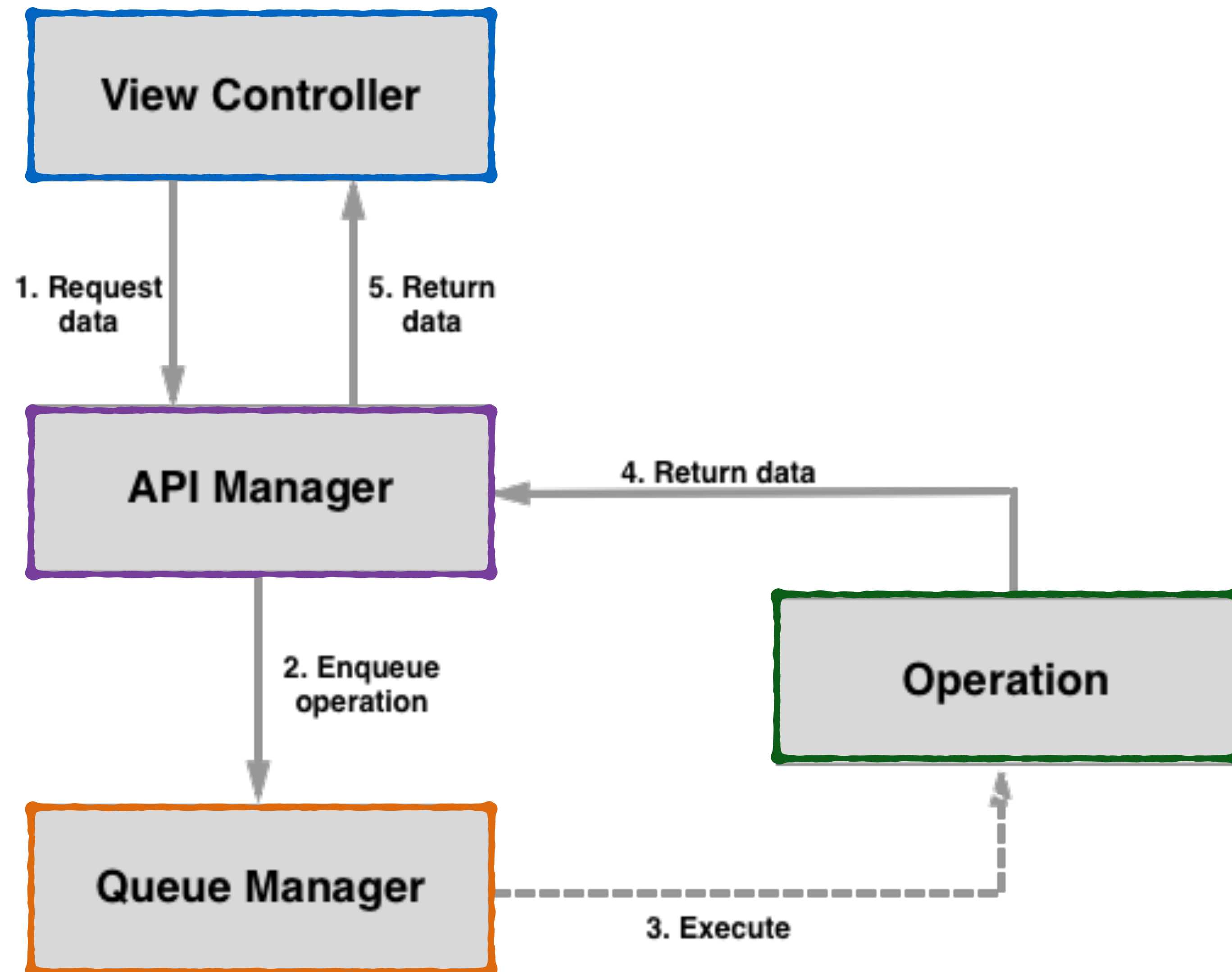


URLSESSION IN DATA OBJECT

```
extension Restaurant {  
    private let urlComponents: URLComponents  
    private let session: URLSession  
    static func restaurants(matching query: String, completion: ([Restaurant]) -> Void) {  
        var searchURLComponents = urlComponents  
        searchURLComponents.path = "/search"  
        searchURLComponents.queryItems = [URLQueryItem(name: "q", value: query)]  
        let searchURL = searchURLComponents.url!  
        session.dataTask(url: searchURL, completion: { (_, _, data, _)  
            var restaurants: [Restaurant] = []  
            // .. normal JSONSerialization of data  
            completion(restaurants)  
        }).resume()  
    }  
}
```



API MANAGER



MICRO-SERVICES

View Controller

Service Layer

UserServices

CommonServices

FriendServices

SecurityServices

SessionManager



DEMO



CHALLENGE TIME!

