

Runner Import C+ Builder Team Library IN SYNC Jonny B Filter Find all mess 1. Auth Regis 2. Auth Logir 3. Add User + No Environment Examples (0) POST http://localhost:3005/v1/account/login Params Send Save Authorization Headers (1) Body Pre-request Script Tests Cookies Code form-data x-www-form-urlencoded raw binary JSON (application/json) { "email": "j@j.com", "password": "123456" } Status: 200 OK Time: 842 ms Size: 413 B Body Cookies Headers (6) Tests Pretty Raw Preview JSON Save Response { "user": "j@j.com", "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9 .eyJpZCI6IjU5Mjc2ODAxODdlODBlYmY1ODY2MGFlMyIsImlhCI6MTUwMDMyNDMzNSwiZXhwIjoxNTA4MTAwMzM1fQ" }

```
6 // Copyright © 2017 Jonny B. All rights reserved.
7 //
8
9 import Foundation
10
11 typealias CompletionHandler = (_ Success: Bool) -> ()
12
13 // URL Constants
14 let BASE_URL = "https://chattychatjb.herokuapp.com/v1/"
15 let URL_REGISTER = "\(BASE_URL)account/register"
16 let URL_LOGIN = "\(BASE_URL)account/login" ←
17
18 // Segues
19 let TO_LOGIN = "toLogin"
20 let TO_CREATE_ACCOUNT = "toCreateAccount"
21 let UNWIND = "unwindToChannel"
22
23 // User Defaults
24 let TOKEN_KEY = "token"
25 let LOGGED_IN_KEY = "loggedIn"
26 let USER_EMAIL = "userEmail"
27
28 // Headers
29 let HEADER = [
30     "Content-Type": "application/json; charset=utf-8"
31 ]
32
33
```

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help

Smack | Build Smack: **Succeeded** | Today at 1:12 PM 32

Smack > Smack > Services > AuthService.swift > M loginUser(email:password:completion:)

```
func loginUser(email: String, password: String, completion: @escaping CompletionHandler) {  
    let lowerCaseEmail = email.lowercased()  
  
    let body: [String: Any] = [  
        "email": lowerCaseEmail,  
        "password": password  
    ]  
  
    Alamofire.request(URL_LOGIN, method: .post, parameters: body,  
                      encoding: JSONEncoding.default, headers: HEADER).responseJSON {  
        (response) in  
  
        if response.result.error == nil {  
            if let json = response.result.value as? Dictionary<String,  
                Any> {  
                if let email = json["user"] as? String {  
                    self.userEmail = email  
                }  
                if let token = json["token"] as? String {  
                    self.authToken = token  
                }  
            }  
            self.isLoggedIn = true  
            completion(true)  
        }  
    }  
}
```

The screenshot shows the Xcode interface with the Smack project open. The left sidebar displays the project structure, including the Smack module with its subfolders Services, Utilities, Model, View, Controller, and Supporting Files. The CreateAccountVC.swift file is selected in the list.

```
20     super.viewDidLoad()
21 }
22 }
23
24 @IBAction func createAccntPressed(_ sender: Any) {
25     guard let email = emailTxt.text , emailTxt.text != "" else { return
26         }
27     guard let pass = passTxt.text , passTxt.text != "" else { return }
28
29     AuthService.instance.registerUser(email: email, password: pass)
30     { (success) in
31         if success {
32             AuthService.instance.loginUser(email: email, password:
33                 pass, completion: { (success) in
34                 if success {
35                     print("logged in user!",
36                         AuthService.instance.authToken)
37                 }
38             }
39         }
40     }
41
42 @IBAction func pickAvatarPressed(_ sender: Any) {
43
44 @IBAction func pickBGColorPressed(_ sender: Any) {
45 }
```

A red box highlights the code block starting at line 31, which handles the successful registration and subsequent login. This block prints "logged in user!" and the authentication token to the console.

The screenshot shows the Xcode interface with a project named "Smack" running on an iPhone 6s. The file "AuthService.swift" is open in the editor. A red arrow points to the line "import SwiftyJSON".

```
1 //  
2 //  AuthService.swift  
3 //  Smack  
4 //  
5 //  Created by Jonny B on 7/17/17.  
6 //  Copyright © 2017 Jonny B. All rights reserved.  
7 //  
8  
9 import Foundation  
10 import Alamofire  
11 import SwiftyJSON| ←  
12  
13 class AuthService {  
14  
15     static let instance = AuthService()  
16  
17     let defaults = UserDefaults.standard  
18  
19     var isLoggedIn : Bool {  
20         get {  
21             return defaults.bool(forKey: LOGGED_IN_KEY)  
22         }  
23         set {  
24             defaults.set(newValue, forKey: LOGGED_IN_KEY)  
25         }  
26     }  
27 }
```

0. WORK WITH MOYA

For Legacy Swift support, take a look at the [swift2 branch](#)

[中文介绍](#)

Why is the typical JSON handling in Swift NOT good?

Swift is very strict about types. But although explicit typing is good for saving us from mistakes, it becomes painful when dealing with JSON and other areas that are, by nature, implicit about types.

Take the Twitter API for example. Say we want to retrieve a user's "name" value of some tweet in Swift (according to Twitter's API https://dev.twitter.com/docs/api/1.1/get/statuses/home_timeline).

The code would look like this:

```
if let statusesArray = try? JSONSerialization.jsonObject(with: data, options: .allowFragments) as? [[String: Any]]  
    let user = statusesArray[0]["user"] as? [String: Any],  
    let username = user["name"] as? String {  
        // Finally we got the username  
    }
```

It's not good.

Even if we use optional chaining, it would be messy:

```
if let JSONObject = try JSONSerialization.jsonObject(with: data, options: .allowFragments) as? [[String: Any]],  
    let username = (JSONObject[0]["user"] as? [String: Any])?["name"] as? String {  
    // There's our username
```



Usage

Initialization

```
import SwiftyJSON

let json = JSON(data: dataFromNetworking)

let json = JSON(jsonObject)

if let dataFromString = jsonString.data(using: .utf8, allowLossyConversion: false) {
    let json = JSON(data: dataFromString)
}
```

Subscript

```
//Getting a double from a JSON Array
let name = json[0].double

//Getting an array of string from a JSON Array
let arrayNames = json["users"].arrayValue.map({$0["name"].stringValue})

//Getting a string from a JSON Dictionary
let name = json["name"].stringValue
```

The screenshot shows the Xcode interface with the file `AuthService.swift` open. The code is written in Swift and handles user login. A red box highlights a section of code that uses the `SwiftyJSON` library to parse the JSON response.

```
        encoding: JSONEncoding.default, headers: HEADER).responseJSON {  
    (response) in  
  
        if response.result.error == nil {  
            if let json = response.result.value as?  
                Dictionary<String, Any> {  
                    if let email = json["user"] as? String {  
                        self.userEmail = email  
                    }  
                    if let token = json["token"] as? String {  
                        self.authToken = token  
                    }  
                }  
        }  
  
        // Using SwiftyJSON  
        guard let data = response.data else { return }  
        let json = JSON(data: data)  
        self.userEmail = json["user"].stringValue  
        self.authToken = json["token"].stringValue  
  
        self.isLoggedIn = true  
        completion(true)  
    } else {  
        completion(false)  
    }  
}
```