

Branch: master ▾

Find file

Copy path

[carolinafever](#) / [carolina\\_fever\\_0](#) / DashboardViewController.swift

calebwkang comments and clean up

fb15e3d on Sep 6, 2019

[1 contributor](#)

Raw

Blame

History



193 lines (123 sloc) 6.38 KB

```
1 //
2 // FirstViewController.swift
3 // carolina_fever_0
4 //
5 // Created by Caleb Kang on 7/25/19.
6 // Copyright © 2019 Caleb Kang. All rights reserved.
7 //
8
9 import UIKit
10 import Parse
11
12 class DashboardViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {
13
14     var log = [Game]()
15
16     @IBOutlet var points: UILabel!
17     @IBOutlet var rank: UILabel!
18
19     @IBOutlet var loading: UIActivityIndicatorView!
20
21     @IBOutlet var logLabel: UILabel!
22     @IBOutlet var gamesLog: UITableView!
23
24
25
26     override func viewDidLoad() {
27         super.viewDidLoad()
28
29         loading.startAnimating()
30         loading.hidesWhenStopped = true
31         UIApplication.shared.beginIgnoringInteractionEvents()
32
33         gamesLog.isHidden = true
34
35
36         if let user = PFUser.current() {
37             if let username = user.username {
38
39                 let gamesQuery = PFQuery(className: "Game")
40                 gamesQuery.whereKey("student", equalTo: username)
41
42                 gamesQuery.findObjectsInBackground(block: { (objects: [PFObject]?, error: Error?) in
43
44                     var points = 0
45
46
47                     if let games = objects {
48                         print(games)
49
50                         var i = 0;
51
```

```

52         while i < games.count {
53             let game = games[i]
54
55             do {
56                 try game.fetchIfNeeded()
57                 self.log.append(Game(game["student"] as! NSString, game["date"] as! NSDate, game["descrip
58
59                 let point = (game["points"] as! NSNumber)
60
61                 points += Int(exactly: point)!
62
63             } catch {
64                 print("couldnt fetch game object")
65             }
66
67             i += 1
68         }
69
70         DispatchQueue.main.async {
71             self.points.text = String(points) + "pts"
72
73             if let rank_bool = user["isTop150"] as? Bool {
74                 if rank_bool {
75                     self.rank.text = "Top 150!"
76                 } else {
77                     self.rank.text = "Not Top 150"
78                 }
79             }
80         }
81
82         self.gamesLog.reloadData()
83         self.gamesLog.isHidden = false
84
85         UIApplication.shared.endIgnoringInteractionEvents()
86         self.loading.stopAnimating()
87
88     }
89 })
90
91
92     }
93 }
94
95 configLine()
96
97
98 }
99
100 override func viewWillAppear(_ animated: Bool) {
101     let username = PFUser.current()?.username!
102     if let bar = navigationController?.navigationBar {
103         bar.titleTextAttributes = [NSAttributedString.Key.foregroundColor: UIColor.white]
104         bar.topItem?.title = "Welcome, \(username)"
105     }
106
107
108 }
109
110
111 // MARK: TableView Methods
112
113 func tableView(_ tableView: UITableView, numberOfRowsInSectionSection section: Int) -> Int {
114     return log.count+1
115 }
116
117 func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {

```

```
118
119     if log.count == 0 {
120
121         /*log hasnt been queried yet so return empty cell*/
122
123         let cell = UITableViewCell()
124
125         return cell
126
127
128     } else if indexPath.row < log.count {
129
130         /*fill all cells with Content view, except for last row*/
131         if let cell = tableView.dequeueReusableCell(withIdentifier: "content") as? ContentTableViewCell {
132
133             let game = log[indexPath.row]
134
135             let date = game.date as Date
136             let formatter = DateFormatter()
137             formatter.dateStyle = DateFormatter.Style.short
138
139             var date_str = formatter.string(from: date) as NSString
140             date_str = date_str.substring(to: date_str.length-3) as NSString
141
142
143             cell.date.text = String(date_str)
144             cell.gameString.text = String(game.getDescription())
145             if let integer = game.getPoints() as? Int {
146                 cell.points.text = String(integer)
147             }
148
149
150             return cell
151
152         }
153
154     } else if indexPath.row == log.count {
155
156         /*fill the last cell/row with missingPoints cell*/
157         if let cell = tableView.dequeueReusableCell(withIdentifier: "missingPoints") as? MissingPointsTableViewCell {
158             return cell
159         }
160     }
161
162     return UITableViewCell()
163
164
165
166 }
167
168 func tableView(_ tableView: UITableView, heightForRowAt indexPath: IndexPath) -> CGFloat {
169     return 80
170 }
171
172 @IBAction func missingPointsPressed(_ sender: UIButton) {
173     self.navigationController?.navigationBar.topItem?.title = "Dashboard"
174     performSegue(withIdentifier: "toCorrections", sender: self)
175 }
176
177 /*create the line separatng Log label from tableview*/
178 func configLine() {
179     let bottomBorder = CAShapeLayer()
180     let bottomPath = UIBezierPath()
181     bottomPath.move(to: CGPoint(x: 0, y: logLabel.frame.height))
182     bottomPath.addLine(to: CGPoint(x: logLabel.frame.width, y: logLabel.frame.height))
183     bottomBorder.path = bottomPath.cgPath
```

```
184         bottomBorder.strokeColor = UIColor.white.cgColor
185         bottomBorder.lineWidth = 1.0
186         bottomBorder.fillColor = UIColor.white.cgColor
187         logLabel.layer.addSublayer(bottomBorder)
188     }
189
190
191
192 }
193
```