Critter Crush iOS APP Documentation

Useful documentation:

CritterCrush API Calls:
See Elyas's: CritterCrush API Documentation - Google Docs

Alamofire and AlamofireImage documentation:
https://github.com/Alamofire/AlamofireImage#documentation

General credits:

UIKit Apprentice (Second Edition): Beginning IOS Development with Swift, raywenderlich Tutorial Team, Farook, F., 9781950325474, 2021

segue:
https://stackoverflow.com/questions/30991551/segue-from-modal-view-to-tab-bar-view-controller-and-not-lose-tab-bar/30992088#30992088
Passing Information Through Segue:
https://stackoverflow.com/questions/26207846/pass-data-through-segue

style:
https://stackoverflow.com/questions/39624675/add-shadow-on-uiview-using-swift-3
https://www.hackingwithswift.com/example-code/calayer/how-to-round-the-corners-of-a-uiview
https://developer.apple.com/documentation/quartzcore/calayer/1410896-maskstobounds
https://www.appcoda.com/rounded-corners-uiview/
Add Rounded Corner for Buttons:
https://stackoverflow.com/questions/1509547/giving-uiview-rounded-corners

Figma UI Icons:
iOS 16 UI Kit for Figma
Dummy Login & Register (Discord Design)e
Bottom Navigation Bar for iOS UI

Testing API Calls:
Postman

Adding Row Height to Table View:
https://stackoverflow.com/questions/25632394/swift-uitableview-set-rowheight
Remove DS_Store files from github
https://stackoverflow.com/questions/107701/how-can-i-remove-ds-store-files-from-a-git-repository
https://medium.com/swift-india/uialertcontroller-in-swift-22f3c5b1dd68 (change style of action to destructive)

iOS Files and Classes

SigninViewController:

Uses API Call: (hostname)/api/users/register
Parameters: username, password, email

Used Following links to set the labels and segue of logIn
https://www.youtube.com/watch?v=-cYLkXKNDJA

LoginViewController:

Used Following links to set the labels and segue of logIn
https://www.youtube.com/watch?v=dbW8UMnrOj0
https://www.youtube.com/watch?v=-cYLkXKNDJA

Uses API call: (hostname)/api/users/login
Parameters: username, password

KeychainHelper:

Code is entirely copied from: https://swiftsenpai.com/development/persist-data-using-keychain/
MIT License
Copyright (c) 2023 Lee Kah Seng
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so.

Used to securely retrieve and store sensitive data like passwords or authentication tokens locally.
Use example (save login token locally upon successful login):
```
KeychainHelper.standard.save(token, service: "com.crittercrush.authToken", account: "authToken")
```

selectSpeciesController:

Used tutorial from UIKit Apprentice (Second Edition): Beginning IOS Development with Swift, raywenderlich Tutorial Team, Farook, F., 9781950325474, 2021 chapter 25, pages: 624-626

AddReportController:

AddReport style and functionality code is very similar to the tutorial in
UIKit Apprentice (Second Edition): Beginning IOS Development with Swift, raywenderlich Tutorial Team, Farook, F., 9781950325474, 2021 chapter: 25

Image Picker:
UIKit Apprentice (Second Edition): Beginning IOS Development with Swift, raywenderlich Tutorial Team, Farook, F., 9781950325474, 2021 chapter: 30

Predict Species button icon:
https://www.vectorstock.com/royalty-free-vector/find-bugs-icon-vector-35774965

Generating random names (for image):
func randomName(length: Int)

Takes user set length of string to randomly generate image names for upload
https://stackoverflow.com/questions/26845307/generate-random-alphanumeric-string-in-swift

Submitting reports:
Completion handler for API/multipart image uploads using Alamofire
Every image upload function we use here has

completion: @escaping (Result<Data, Error>) -> Void)

It requires us to unwrap the result before using the data, in case there is an error from the API call.
ios - How to set up completion handlers in API Call - Swift 5 - Stack Overflow

func upImage():
API: (hostName)/api/reports
Headers: authentication token

Running image recognition on attached image:
func predict(), func upImage
Attaching image to alamofireimage upload multipart data:
https://www.letsbuildthatapp.com/videos/5302
API: "\(hostName)/api/critterdetect"
Parameters: attached image

Func showPredictAlert, showSubmitAlert:
UIAlertController (Also used for other screens):
UIKit Apprentice (Second Edition): Beginning IOS Development with Swift, raywenderlich Tutorial Team, Farook, F., 9781950325474, 2021 chapter: 22, page 550
https://stackoverflow.com/questions/25511945/swift-alert-view-with-ok-and-cancel-which-button-tapped

AddressUploadViewController:

Address geocoding (address to lat/lon):
https://stackoverflow.com/questions/42279252/convert-address-to-coordinates-swift

Used tutorial from UIKit Apprentice (Second Edition): Beginning IOS Development with Swift, raywenderlich Tutorial Team, Farook, F., 9781950325474, 2021 chapter: 22,23

SingleReportViewController:

Retrieve image using Alamofireimage:
API: "\(hostName)/api/reports/image/\(imgName)"

Retrieve username:

API: "\(hostName)/api/users/userprofile"
parameters : UserID

Delete report:
func delReport():
API: "\(hostName)/api/reports/\(reportID)"
Headers: auth token

Server:

Singleton object file to store server name as a global variable.
Change to your localhost by modifying:
let localhost = Server(hostname:"ipaddress")

ProfileViewController:

Display details of currently logged in user.

Hide navbar: refresh button makes navbar unnecessary and persistent
Code and concept from:
ios - How to hide a navigation bar from first ViewController in Swift? - Stack Overflow

Retrieve username:
API: "\(hostName)/api/users/userprofile"
Parameter: login userid

Display count of reports submitted by user (by species):
API: "\(hostName)/api/reports/count"
Parameter: userid, speciesid

Retrieve User total Score:
API: "\(hostName)/api/reports/count"
Parameter: userid

SettingsViewController:

Retrieve user profile data:
API: "\(hostName)/api/users/userprofile"
Header: authToken

Delete user account
API: "\(hostName)/api/users/userprofile"
Header: authToken

EditEmailViewController:

Change email:
API: "\(hostName)/api/users/userprofile"
Header: authToken

EditPasswordViewController:

Change password:
API: "\(hostName)/api/users/userprofile"
Header: authToken

MapViewController:

Map Pin:
https://makeapppie.com/2018/02/20/use-markers-instead-of-pins-for-map-annotations/

PopUpScreen when user clicks pins:
UIKit Apprentice (Second Edition): Beginning IOS Development with Swift, raywenderlich Tutorial Team, Farook, F., 9781950325474, 2021 chapter: 25

Image alamofire request:
https://stackoverflow.com/questions/28636622/how-to-load-image-in-swift-using-alamofire

Remove pins:
https://stackoverflow.com/questions/54705756/how-to-remove-my-custom-annotation-from-mapview

Map:
https://developer.apple.com/documentation/mapkit/mkmapviewdelegate/1452393-mapview
UIKit Apprentice (Second Edition): Beginning IOS Development with Swift, raywenderlich Tutorial Team, Farook, F., 9781950325474, 2021 chapter: 25

Load report image: "http://\(hostName)/api/reports/image/\(imgName)"
Get batch of reports (set to 20):

```
batch.mapApiReport(mapNo: 20)
```

Uses BatchReport

BatchReport:

Class used to only store functions that require api call, to simplify code on complex files like MapViewController.

func getReports():
For testing report api calls (uses func apiReport)

Get reports based on userID, species:
func apiReport
API: "http://\(hostname)/api/reports"
Parameter: userid, speciesid
(used in DetailSubmission)

Get specified number of reports (based on most recently submitted report):
func mapApiReport
API: "http://\(hostname)/api/reports"
Parameter: number of reports

ParseBatch:

Structs and class to convert data from API Json call into swift
Swift Codable object. Added MKAnnotation fields like coordinate and title, for Map display.

Parse JSON:
Automatically formatted into Swift from our API call's return JSON using:
https://app.quicktype.io/#l=go

Parsing nested object help:
https://stackoverflow.com/questions/54821142/parsing-nested-json-arrays-in-swift

TabViewController:

tabbar:
https://www.youtube.com/watch?v=cb7izCNfKqg

SpeciesViewController:

A UITable. Click on each cell to go to the species information of selected species.
Hand drawn custom assets for icon display.

SpeciesDetailViewController:

Species Scroll View:
horizontal gallery code:
https://github.com/zhiyao92/Horizontal-Sliding-Image

Images and information summarized from:
Spotted Lanternfly - NYS Dept. of Environmental Conservation
Asian Longhorned Beetle (ALB) - NYS Dept. of Environmental Conservation
Emerald Ash Borer (EAB) - NYS Dept. of Environmental Conservation
Spongy Moth - NYS Dept. of Environmental Conservation

SpeciesCell:

Table cell to display species (used in ProfileView and SpeciesView)

Species:

Custom object class to store information for our bug species.

```
var name: String //common name
  var science: String //scientific name
  var id: Int //our id system, same as speciesId in database
  var color: UIColor //color chosen from icons to be used for map annotation
  var desc: String //summary of species description from NYSDEC
  var risk: String //summary of risk description from NYSDEC
  var link: URL //NYSDEC link info is credited to, see below
  var threat: Int //threat level for report score calculation
```

Risk and description data summarized from:
Spotted Lanternfly - NYS Dept. of Environmental Conservation
Asian Longhorned Beetle (ALB) - NYS Dept. of Environmental Conservation
Emerald Ash Borer (EAB) - NYS Dept. of Environmental Conservation
Spongy Moth - NYS Dept. of Environmental Conservation

Our bugs were created as instances of Species and stored in a global variable array to be used in every screen that needs it.

```
//global variables of Species
let SLF = Species(name: "Spotted lanternfly", science: "Lycorma delicatula", id: 1, threat: 2)
let ALB = Species(name: "Asian longhorned beetle", science: "Anoplophora glabripennis", id: 2, threat: 3)
let EAB = Species(name: "Emerald ash borer", science:"Agrilus planipennis", id: 3, threat: 3)
let SPM = Species(name: "Spongy moth", science:"Lymantria dispar", id: 4, threat: 1)
let speciesList = [SLF,ALB,EAB,SPM]
```

The threat levels were decided based on their risk. 1 means naturalized (but still invasive in large amounts), 2 means invasive, 3 means “this species is now rare because it should have been eradicated in NYS, and poses an immediate threat”.

Report Scores are calculated server side after image recognition call:

$$(100 * \text{threat}) + (\text{numberSpecimens} * 10 * \text{threat})$$

DetailSubmissionViewController:

Load images `"http://\(hostName)/api/reports/image/\(imgName)"`
Grab reports based on species id and login user id:

    batch.apiReport

Displays image data from batch reports using UICollectionView

ReportCell:

Collection view cell used in DetailSubmissionView

AppDelegate:

Default swift doc, no changes

SceneDelegate:

Log in automatically to bypass login screen if login auth token stored locally exists:
API: "\(localhost)/api/users/userprofile"
Headers: authToken

OFFLINE DATA

Now deprecated, classes created for an “offline” version of our map before we transitioned to using our database.
Qns_test_2, test_slf: csv mock report data generated using python (pandas) and linking inaturalist’s bug images.
Individual Report: Codable “JSON” object rough draft
Submission: works like our ParseBatch, store report data, except it’s for offline CSV data not JSON.
TestData: has a function for reading CSV files and converting to an array of Submission objects.