

INTERMEDIATE REALM

on ios



HANDS-ON CHALLENGES

Intermediate Realm on iOS

Marin Todorov

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

Challenges: Migrations Part 1	5
-------------------------------------	---

Table of Contents: Extended

Challenges: Migrations Part 1	5
Challenge A: Fix dates broken during app version 1.0 usage	5
Challenge B: Add new property to assign emojis to exams	7


5 Challenges: Migrations Part 1

By Marin Todorov

Challenge A: Fix dates broken during app version 1.0 usage

If you have a look at the list of exams that the app currently shows you will notice that some of the exams are due ... more than a century ago?

3 EXAMS THIS SEMESTER
Math 101(multiple choice) Jun 8, 2017 · pending
French Literature Jun 1, 2017 · pending
Biology(multiple choice) Jun 14, 1900 · completed



This certainly can't be true :) For the purpose of this challenge pretend this is the result of a bug in the code of Exams v.1.0 (which set certain due dates to the year 1900) and this bug has been fixed in Exams v.1.2.

The code change will ensure that no new broken dates are created for new Exam objects, but the issue is with existing exams.

Migration is precisely the place to take care of existing invalid data – in this

challenge you will change all dates that are way too much in the past to today's date. This way the user will have a chance to notice that an exam is due today and figure out they need to re-adjust the due date manually.

In your **Exams.swift** file scroll to the `migrate(_:,fileSchemaVersion:)` method.

Since this bug will be present only in Realm files with schema version of 1 you will need to place the migration code inside the `if fileSchemaVersion < 2` block.

Insert this directly under `if fileSchemaVersion < 2 {`:

```
let thePast = Date(timeIntervalSince1970: 0)
```

This is the date you will compare all exam dates to in order to find out broken dates. It's safe to assume you wouldn't have had any real exams before 1970 (that you would have tracked on your iPhone).

Inside the closure of your `enumerate` call and **at the top** of the `if let newObject = newObject` block add (still **inside** the `if` block though):

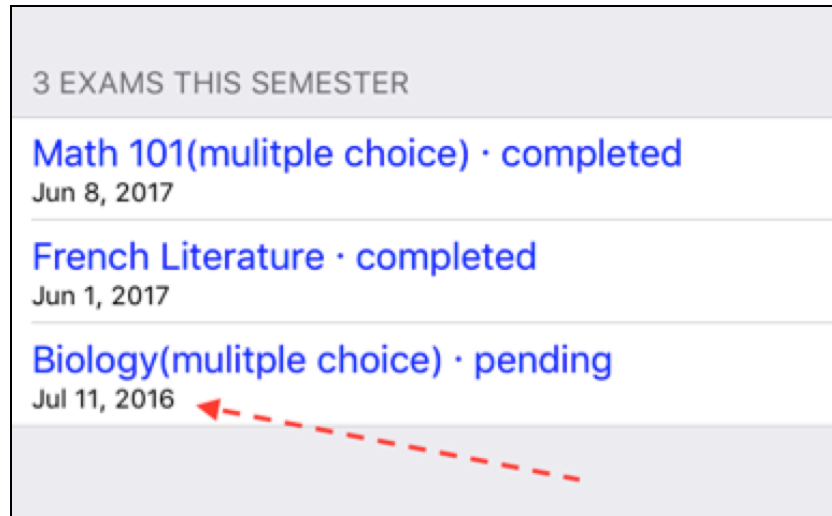
```
if let date = newObject["date"] as? NSDate,
   date.compare(thePast) == .orderedAscending {
    newObject["date"] = now
}
```

You cast the value of the "date" MigrationObject key to a Date and if it's even further in the past than thePast you set its value to today's date.

This is a quick and easy solution to the problem and leaves the initiative to the user to notice incorrect dates (of the exams due today) and fix them manually.

Note: To test the migration again, just delete the app from your iPhone Simulator and run the project again. This will trigger the migration block another time and you can test your code.

The exam list looks better now:



In a real life project you could think of a more interactive approach to fixing the broken data – for example you can gather all the Exam objects with incorrect data in a list and once the app has started ask the user to fix them one by one. Or even suggest possible due dates based on other information the app might have.

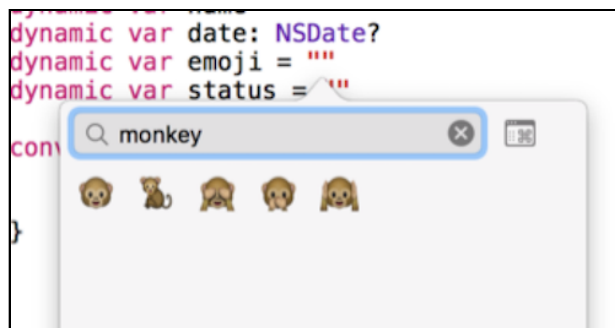
Challenge B: Add new property to assign emojis to exams

In this quick challenge you will add one more property to the Exam class to store an emoji for each of the exams.

First of all open **Exam.swift** and add the new property:

```
dynamic var emoji = "🐵"
```

To insert the monkey face (or another emoji of your choice) press together Ctrl+Cmd+Space and that will bring a searchable list of emojis to choose from:



Since the property has a default value all new Exam objects will have an emoji set by default. However existing Exam object will not get this default value (since that

might not be what you'd like to do).

To provide a default emoji for existing Exam objects you will need to add some code to your migration.

Open **Exams.swift** and find the piece of code, which you just added in the previous challenge. Right under that code (but still inside the if let block) add:

```
newObject["emoji"] = "👨🏻"
```

Next let's make some use of that new property!

Open **ExamsViewController.swift** and find `tableView(_: , cellForRowAtIndexPath:)`. Adjust the code that sets `textLabel` to make use of the new emoji property:

Note: Again – just delete the app to re-trigger the migration block you're testing.

Run the app now and the list perfectly reflects how I feel about any upcoming exams:

