

Saving Data in iOS

Hands-On Challenges

Saving Data in iOS Hands-On Challenges

Copyright © 2014 Razeware LLC.

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.

This book 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 or other dealings in the software.

All trademarks and registered trademarks appearing in this book are the property of their respective owners.

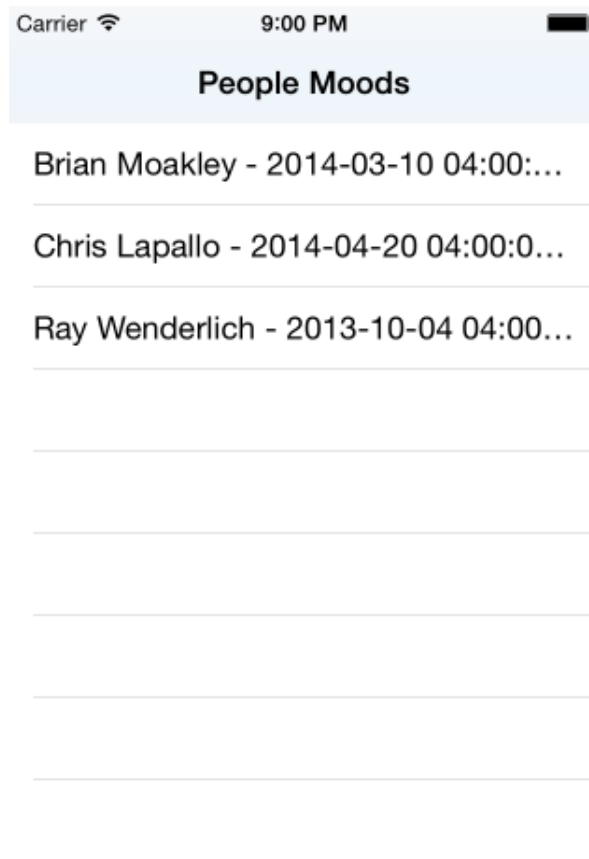


Working with JSON

While XML is a readily used way to transmit data over the web, JSON is another format that accomplishes the same thing albeit in a different syntax. Thankfully, working with JSON in iOS is relatively painless affair.

Challenge

If you took the challenge on reading and writing XML, you will be familiar with the Mood app. This app provides moods for various people. Tapping on the name provides the mood details for that person. This data is all stored in XML, then parsed using either NSXMLParser or RaptureXML.



In this challenge, you will instead read the data as JSON. The first part of the challenge is to convert the actual XML into JSON and include with the app bundle.



Once you convert your XML to JSON, make sure to validate it by using this online tool:

<http://jsonlint.com/>

If you need a quick refresher on how convert XML to JSON, feel free to head over here:

<http://www.xml.com/pub/a/2006/05/31/converting-between-xml-and-json.html>

Open the first sample project and you'll notice an empty tableview. This table view is supposed to be populated with names, and when one of the names are tapped, the app will display the "mood" of each person and the reason for the actual mood.

Once the data has been converted into JSON, parse the JSON to create the actual RWTMood objects for the app to function.

Your challenge is to parse the JSON, create an array of RWTMood objects, then inside of **MoodListingViewController.m**, assign the mood objects to the moodList property.

Build and run, and you will see all of the moods.



Uber Haxx0r Challenge: Saving XML

While the mood reading app is nice, it currently is just a read-only app. It's time to add some writing capabilities to it.

The first time the app runs, it should read from your converted json and create all the RWTMood objects like the previous challenge.

When the user adds a new entry and presses the save button, everything should be written to a JSON file in the document directory.

The next time the app runs, the app should read that new JSON file and load in the new data. Good luck!

