

BEGINNING

CORE DATA



HANDS-ON CHALLENGES

Beginning Core Data

Luke Parham

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.

Challenge #1: Adding Attributes

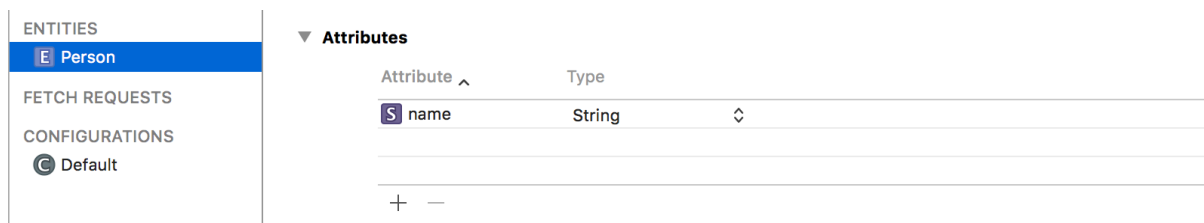
By Luke Parham

At the moment, your Hit List is pretty simple. Knowing people's names is useful, but to really step up your game you'll want to at least add a way to save people's addresses.

Adding an Attribute

To begin, go to the Data Model Editor by clicking on **HitList.xcdatamodeld**.

As you saw in the demo, adding attributes to your entities is easy. First, click on **Person** under the **Entities** section in the panel on the left.



Next, click the plus icon to start adding a new attribute to the Entity. Name it **address** and set its **Type** to String.

And that's it! You've successfully added another attribute to the Person entity. I know it was hard but you've pulled through. ;]

Adding an Attribute

Now that you've successfully added a new attribute, it's time to actually use it.

The first thing you'll need to do is navigate to **ViewController.swift**.

Once you're there, scroll down to the `addName(_:)` method where all the magic

happens.

Currently, the UIAlertController is being displayed with only a name field. Your goal is to add a second field with which the user can specify an address.

Go ahead and replace the guard statement inside of the saveAction with the following:

```
guard let nameTextField = alert.textFields?.first,
      let nameToSave = nameTextField.text,
      let addressTextField = alert.textFields?[1],
      let addressToSave = addressTextField.text else {
    return
}
```

This guard statement assumes you've displayed two text fields, one for the person's name and one for the new address field.

Next, replace

```
alert.addTextField()

alert.textFields?[0].placeholder = "Name"
```

with

```
alert.addTextField()
alert.addTextField()

alert.textFields?[0].placeholder = "Name"
alert.textFields?[1].placeholder = "Address"
```

To actually add the second text field to the alert.

Note: You may be thinking to yourself, `addName(_:)` is kind of a stupid name for this method now that we're adding a more complex Person. You're totally right! Feel free to rename it and fix the connection in the storyboard.

Saving The Full Person

Now that the alert can take a name and an address, you need to save the full person.

First, change the method signature to take in the address.

```
func save(name: String, address: String) {
```

Next, add

```
person.setValue(address, forKeyPath: "address")
```

below

```
person.setValue(name, forKeyPath: "name")
```

to save the new value to the managed object.

Showing the Full Person

The last thing you need to do is show all this new information in the table. All you need to do to accomplish this is replace:

```
cell.textLabel?.text = person.value(forKey: "name") as? String
```

with:

```
let name = person.value(forKeyPath: "name") ?? "Name unknown"
let address = person.value(forKey: "address") ?? "Address unknown"
cell.textLabel?.text = "\(name) -- \(address)"
```

Go ahead and build and run to see the results!

