

Networking with NSURLSession

Hands-on Challenges

Networking with NSURLSession Hands-On Challenges

Copyright © 2016 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.



Challenge 5: Customizing the Operation Queue

By this point, you should have a pretty good handle on `NSURLSession`. If your app is doing network operations for different scenarios, you'll probably want to create multiple sessions. A common reason for doing this is to assign different quality of service values for the delegate queues.

Your challenge is to create a custom operation queue and set its quality of service to `.UserInitiated`.

For step-by-step instructions, read on.



delegateQueue

In **NetworkClient.swift**, in the init method, replace this code:

```
urlSession = NSURLSession(configuration: configuration)
```

With the following code:

```
let queue = NSOperationQueue()  
queue.qualityOfService = .UserInitiated  
urlSession = NSURLSession(configuration: configuration,  
    delegate: nil, delegateQueue: queue)
```

This code creates a new `NSOperationQueue`, sets the `qualityOfService` property on it, and uses that queue to initialize the `NSURLSession`.

That's it! That's all there is to customizing the delegate queue. If you set a breakpoint in one of the completion handlers, like on this line in `getURL`:

```
self.parseJSON(data, completion: completion)
```

When the breakpoint is hit, open the Debug navigator. The current thread should have a "(QOS: USER_INITIATED)" label on it now.

