Networking with NSURLSession

Hands-on Challenges

Networking with NSURLSession Hands-On Challenges

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.