

# NETWORKING WITH URLSession



HANDS-ON CHALLENGES

## Networking with URLSession

Audrey Tam

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 #9: ATS

By Audrey Tam

App Transport Security prevents a WebKit web view from loading non-https URLs, so it's the easiest way to see the effects of ATS exceptions.

1. Open, build and run the Browser app in **Challenge Starter/WKBrowser\_Starter**
2. Tap **objc.io**: this loads fine; it got A+ on [SSL Server Test](#).
3. Tap any of the lower 3 buttons: they won't load, because they're all http-only - they don't use SSL.

## Exception to allow insecure HTTP loads from your site

The scenario is: you want to build an app for the Quality Coding site, so either it has to start using SSL, or your app must make it an ATS exception.

1. Open **Info.plist** and add **App Transport Security Settings\Exception Domains\qualitycoding.org**.
2. Change the type of qualitycoding.org to dictionary, then add `NSExceptionAllowsInsecureHTTPLoads:YES` and `NSIncludesSubdomains:YES`.
3. Check that **App Transport Security Settings\Allow Arbitrary Loads** is NO.
4. **Change the simulator** to a different iPhone size. I find this is a reliable way to make Xcode process Info.plist during build, so it actually changes the ATS settings.
5. Build and run. Tap **Quality Coding**. It should load.
6. Tap **mokacoding** or **PBS**: they won't load.

You'll still have to justify the exception for App Store review, but it should be much easier than trying to justify Allow Arbitrary Loads: YES.

## Exception to protect your site

Alternatively, your app might need to allow arbitrary loads from as-yet-unknown servers, perhaps for displaying ads, or subscribing to mail servers. But you want to keep ATS protection for your own back-end server — we'll pretend that Quality Coding is your back-end server.

1. Change **App Transport Security Settings\Allow Arbitrary Loads** to YES.
2. In the **Exception Domain**, change `NSExceptionAllowsInsecureHTTPLoads` to NO.
3. Change the simulator to a different iPhone size.
4. Build and run. Tap **mokacoding** or **PBS**: they should load.
5. Tap **Quality Coding**: it won't load.

## SafariServices alternative

A SafariServices Safari view controller is not subject to ATS.

1. Open, build and run the Browser app in **Challenge Starter/SafariBrowser**.
2. Tap any button.
3. The **Safari** browser view loads, and shows a progress bar while the site loads.
4. Look at **Info.plist**: no ATS settings at all!

Apple recommends **SafariServices** for apps that need a general purpose browser, with all the features users expect from Safari, including login autofill.

Use **WebKit** to present specific domains, especially if you need to customize the presentation.