



FREE eBook

LEARNING alamofire

Free unaffiliated eBook created from
Stack Overflow contributors.

#alamofire

Table of Contents

About	1
Chapter 1: Getting started with alamofire	2
Remarks.....	2
Examples.....	2
Installation or Setup.....	2
Installation.....	2
CocoaPods.....	2
Carthage.....	2
Manually.....	3
Chapter 2: Component Libraries	5
Introduction.....	5
Examples.....	5
Image Response Serializers.....	5
Credits	6

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [alamofire](#)

It is an unofficial and free alamofire ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official alamofire.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with alamofire

Remarks

This section provides an overview of what alamofire is, and why a developer might want to use it.

It should also mention any large subjects within alamofire, and link out to the related topics. Since the Documentation for alamofire is new, you may need to create initial versions of those related topics.

Examples

Installation or Setup

Installation

CocoaPods

[CocoaPods](#) is a dependency manager for Cocoa projects. You can install it with the following command:

```
$ gem install cocoapods
```

CocoaPods 1.1.0+ is required to build Alamofire 4.0.0+.

To integrate Alamofire into your Xcode project using CocoaPods, specify it in your `Podfile`:

```
source 'https://github.com/CocoaPods/Specs.git'
platform :ios, '10.0'
use_frameworks!

target '<Your Target Name>' do
  pod 'Alamofire', '~> 4.0'
end
```

Then, run the following command:

```
$ pod install
```

Carthage

[Carthage](#) is a decentralized dependency manager that builds your dependencies and provides you with binary frameworks.

You can install Carthage with [Homebrew](#) using the following command:

```
$ brew update
$ brew install carthage
```

To integrate Alamofire into your Xcode project using Carthage, specify it in your `Cartfile`:

```
github "Alamofire/Alamofire" ~> 4.0
```

Run `carthage update` to build the framework and drag the built `Alamofire.framework` into your Xcode project.

Manually

If you prefer not to use either of the aforementioned dependency managers, you can integrate Alamofire into your project manually.

Embedded Framework

- Open up Terminal, `cd` into your top-level project directory, and run the following command "if" your project is not initialized as a git repository:

```
$ git init
```

- Add Alamofire as a git [submodule](#) by running the following command:

```
$ git submodule add https://github.com/Alamofire/Alamofire.git
```

- Open the new `Alamofire` folder, and drag the `Alamofire.xcodeproj` into the Project Navigator of your application's Xcode project.

It should appear nested underneath your application's blue project icon. Whether it is above or below all the other Xcode groups does not matter.

- Select the `Alamofire.xcodeproj` in the Project Navigator and verify the deployment target matches that of your application target.
- Next, select your application project in the Project Navigator (blue project icon) to navigate to the target configuration window and select the application target under the "Targets" heading in the sidebar.
- In the tab bar at the top of that window, open the "General" panel.
- Click on the + button under the "Embedded Binaries" section.
- You will see two different `Alamofire.xcodeproj` folders each with two different versions of the `Alamofire.framework` nested inside a `Products` folder.

It does not matter which `Products` folder you choose from, but it does matter whether you choose the top or bottom `Alamofire.framework`.

- Select the top `Alamofire.framework` for iOS and the bottom one for OS X.

You can verify which one you selected by inspecting the build log for your project. The build target for `Alamofire` will be listed as either `Alamofire iOS`, `Alamofire macOS`, `Alamofire tvOS` **OR** `Alamofire watchOS`.

- And that's it!

The `Alamofire.framework` is automatically added as a target dependency, linked framework and embedded framework in a copy files build phase which is all you need to build on the simulator and a device.

Read [Getting started with alamofire online](https://riptutorial.com/alamofire/topic/7521/getting-started-with-alamofire): <https://riptutorial.com/alamofire/topic/7521/getting-started-with-alamofire>

Chapter 2: Component Libraries

Introduction

In order to keep Alamofire focused specifically on core networking implementations, additional component libraries have been created by the Alamofire Software Foundation to bring additional functionality to the Alamofire ecosystem. - AlamofireImage - AlamofireNetworkActivityIndicator

Examples

Image Response Serializers

```
import AlamofireImage

Alamofire.request("https://httpbin.org/image/png").responseImage { response in
    debugPrint(response)

    print(response.request)
    print(response.response)
    debugPrint(response.result)

    if let image = response.result.value {
        print("image downloaded: \(image)")
    }
}
```

Read Component Libraries online: <https://riptutorial.com/alamofire/topic/8316/component-libraries>

Credits

S. No	Chapters	Contributors
1	Getting started with alamofire	Ashish Kakkad , Community
2	Component Libraries	Ekta Padaliya