

⚠⚠⚠ **WARNING** ⚠⚠⚠ This documentation is out of date during the Alamofire 5 beta process.

- [Features](#)
- [Component Libraries](#)
- [Requirements](#)
- [Migration Guides](#)
- [Communication](#)
- [Installation](#)
- [Usage](#)
  - **Introduction** - [Making Requests](#), [Response Handling](#), [Response Validation](#), [Response Caching](#)
  - **HTTP** - [HTTP Methods](#), [Parameters and Parameter Encoder](#), [HTTP Headers](#), [Authentication](#)
  - **Large Data** - [Downloading Data to a File](#), [Uploading Data to a Server](#)
  - **Tools** - [Statistical Metrics](#), [cURL Command Output](#)
- [Advanced Usage](#)
  - **URL Session** - [Session Manager](#), [Session Delegate](#), [Request](#)
  - **Routing** - [Routing Requests](#), [Adapting and Retrying Requests](#)
  - **Model Objects** - [Custom Response Serialization](#)
  - **Connection** - [Security](#), [Network Reachability](#)
- [Open Radars](#)
- [FAQ](#)
- [Credits](#)
- [Donations](#)
- [License](#)

## Features

---

- Chainable Request / Response Methods
- URL / JSON / plist Parameter Encoding
- Upload File / Data / Stream / MultipartFormData
- Download File using Request or Resume Data
- Authentication with URLLCredential
- HTTP Response Validation
- Upload and Download Progress Closures with Progress
- cURL Command Output
- Dynamically Adapt and Retry Requests
- TLS Certificate and Public Key Pinning
- Network Reachability
- Comprehensive Unit and Integration Test Coverage
- [Complete Documentation](#)

# Component Libraries

---

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](#) - An image library including image response serializers, UIImage and UIImageView extensions, custom image filters, an auto-purging in-memory cache and a priority-based image downloading system.
- [AlamofireNetworkActivityIndicator](#) - Controls the visibility of the network activity indicator on iOS using Alamofire. It contains configurable delay timers to help mitigate flicker and can support URLSession instances not managed by Alamofire.

## Requirements

---

- iOS 10.0+ / macOS 10.12+ / tvOS 10.0+ / watchOS 3.0+
- Xcode 10.2+
- Swift 5+

## Migration Guides

---

- Alamofire 5.0 Migration Guide: To be written!
- [Alamofire 4.0 Migration Guide](#)
- [Alamofire 3.0 Migration Guide](#)
- [Alamofire 2.0 Migration Guide](#)

## Communication

---

- If you **need help with making network requests** using Alamofire, use [Stack Overflow](#) and tag `alamofire`.
- If you need to **find or understand an API**, check [our documentation](#) or [Apple's documentation for URLSession](#), on top of which Alamofire is built.
- If you need **help with an Alamofire feature**, use [our forum on swift.org](#).
- If you'd like to **discuss Alamofire best practices**, use [our forum on swift.org](#).
- If you'd like to **discuss a feature request**, use [our forum on swift.org](#).
- If you **found a bug**, open an issue here on GitHub and follow the guide. The more detail the better!
- If you **want to contribute**, submit a pull request.

## Installation

---

## CocoaPods

[CocoaPods](#) is a dependency manager for Cocoa projects. For usage and installation instructions, visit their website. To integrate Alamofire into your Xcode project using CocoaPods, specify it in your Podfile:

```
pod 'Alamofire', '~> 5.0.0-rc.3'
```

## Carthage

[Carthage](#) is a decentralized dependency manager that builds your dependencies and provides you with binary frameworks. To integrate Alamofire into your Xcode project using Carthage, specify it in your Cartfile:

```
github "Alamofire/Alamofire" "5.0.0-rc.3"
```

## Swift Package Manager

The [Swift Package Manager](#) is a tool for automating the distribution of Swift code and is integrated into the swift compiler. It is in early development, but Alamofire does support its use on supported platforms.

Once you have your Swift package set up, adding Alamofire as a dependency is as easy as adding it to the dependencies value of your Package.swift.

```
dependencies: [  
    .package(url: "https://github.com/Alamofire/Alamofire.git", from: "5.0.0-rc.3")  
]
```

## Manually

If you prefer not to use any 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 macOS.  
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 automagically 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.

## Open Radars

---

The following radars have some effect on the current implementation of Alamofire.

- [rdar://21349340](#) - Compiler throwing warning due to toll-free bridging issue in test case
- [rdar://26870455](#) - Background URL Session Configurations do not work in the simulator
- [rdar://26849668](#) - Some URLProtocol APIs do not properly handle `URLRequest`

## Resolved Radars

---

The following radars have been resolved over time after being filed against the Alamofire project.

- [rdar://26761490](#) - Swift string interpolation causing memory leak with common usage.
  - (Resolved): 9/1/17 in Xcode 9 beta 6.
- [rdar://36082113](#) - `URLSessionTaskMetrics` failing to link on watchOS 3.0+
  - (Resolved): Just add `CFNetwork` to your linked frameworks.

## FAQ

---

### What's the origin of the name Alamofire?

Alamofire is named after the [Alamo Fire flower](#), a hybrid variant of the Bluebonnet, the official state flower of Texas.

## Credits

---

Alamofire is owned and maintained by the [Alamofire Software Foundation](#). You can follow them on Twitter at [@AlamofireSF](#) for project updates and releases.

## Security Disclosure

If you believe you have identified a security vulnerability with Alamofire, you should report it as soon as possible via email to [security@alamofire.org](mailto:security@alamofire.org). Please do not post it to a public issue tracker.

## Donations

---

The [ASF](#) is looking to raise money to officially stay registered as a federal non-profit organization. Registering will allow us members to gain some legal protections and also allow us to put donations to use, tax free. Donating to the ASF will enable us to:

- Pay our yearly legal fees to keep the non-profit in good status
- Pay for our mail servers to help us stay on top of all questions and security issues
- Potentially fund test servers to make it easier for us to test the edge cases
- Potentially fund developers to work on one of our projects full-time

The community adoption of the ASF libraries has been amazing. We are greatly humbled by your enthusiasm around the projects, and want to continue to do everything we can to move the needle forward. With your continued support, the ASF will be able to improve its reach and also provide better legal safety for the core members. If you use any of our libraries for work, see if your employers would be interested in donating. Any amount you can donate today to help us reach our goal would be greatly appreciated.

## License

---

Alamofire is released under the MIT license. [See LICENSE](#) for details.