AFNetworking is a delightful networking library for iOS and Mac OS X. It's built on top of the [Foundation URL Loading System](http://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/URLLoadingSystem/URLLoadingSystem.html), extending the powerful high-level networking abstractions built into Cocoa. It has a modular architecture with well-designed, feature-rich APIs that are a joy to use.

Perhaps the most important feature of all, however, is the amazing community of developers who use and contribute to AFNetworking every day. AFNetworking powers some of the most popular and critically-acclaimed apps on the iPhone, iPad, and Mac.

Choose AFNetworking for your next project, or migrate over your existing projects—you'll be happy you did!

**How To Get Started**

* [Download AFNetworking](https://github.com/AFNetworking/AFNetworking/archive/master.zip) and try out the included Mac and iPhone example apps
* Read the ["Getting Started" guide](https://github.com/AFNetworking/AFNetworking/wiki/Getting-Started-with-AFNetworking), [FAQ](https://github.com/AFNetworking/AFNetworking/wiki/AFNetworking-FAQ), or [other articles on the Wiki](https://github.com/AFNetworking/AFNetworking/wiki)
* Check out the [documentation](http://cocoadocs.org/docsets/AFNetworking/) for a comprehensive look at all of the APIs available in AFNetworking
* Read the [AFNetworking 3.0 Migration Guide](https://github.com/AFNetworking/AFNetworking/wiki/AFNetworking-3.0-Migration-Guide) for an overview of the architectural changes from 2.0.

**Communication**

* If you **need help**, use [Stack Overflow](http://stackoverflow.com/questions/tagged/afnetworking). (Tag 'afnetworking')
* If you'd like to **ask a general question**, use [Stack Overflow](http://stackoverflow.com/questions/tagged/afnetworking).
* If you **found a bug**, *and can provide steps to reliably reproduce it*, open an issue.
* If you **have a feature request**, open an issue.
* If you **want to contribute**, submit a pull request.

**Installation**

AFNetworking supports multiple methods for installing the library in a project.

**Installation with CocoaPods**

[CocoaPods](http://cocoapods.org) is a dependency manager for Objective-C, which automates and simplifies the process of using 3rd-party libraries like AFNetworking in your projects. See the ["Getting Started" guide for more information](https://github.com/AFNetworking/AFNetworking/wiki/Getting-Started-with-AFNetworking). You can install it with the following command:

$ gem install cocoapods

CocoaPods 0.39.0+ is required to build AFNetworking 3.0.0+.

**Podfile**

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

source 'https://github.com/CocoaPods/Specs.git' platform :ios, '8.0' target 'TargetName' do pod 'AFNetworking', '~> 3.0' end

Then, run the following command:

$ pod install

**Installation with Carthage**

[Carthage](https://github.com/Carthage/Carthage) is a decentralized dependency manager that builds your dependencies and provides you with binary frameworks.

You can install Carthage with [Homebrew](http://brew.sh/) using the following command:

$ brew update $ brew install carthage

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

github "AFNetworking/AFNetworking" ~> 3.0

Run carthage to build the framework and drag the built AFNetworking.framework into your Xcode project.

**Requirements**

| **AFNetworking Version** | **Minimum iOS Target** | **Minimum OS X Target** | **Minimum watchOS Target** | **Minimum tvOS Target** | **Notes** |
| --- | --- | --- | --- | --- | --- |
| 3.x | iOS 7 | OS X 10.9 | watchOS 2.0 | tvOS 9.0 | Xcode 7+ is required. NSURLConnectionOperation support has been removed. |
| 2.6 -> 2.6.3 | iOS 7 | OS X 10.9 | watchOS 2.0 | n/a | Xcode 7+ is required. |
| 2.0 -> 2.5.4 | iOS 6 | OS X 10.8 | n/a | n/a | Xcode 5+ is required. NSURLSession subspec requires iOS 7 or OS X 10.9. |
| 1.x | iOS 5 | Mac OS X 10.7 | n/a | n/a |  |
| 0.10.x | iOS 4 | Mac OS X 10.6 | n/a | n/a |  |

(OS X projects must support [64-bit with modern Cocoa runtime](https://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/ObjCRuntimeGuide/Articles/ocrtVersionsPlatforms.html)).

Programming in Swift? Try [Alamofire](https://github.com/Alamofire/Alamofire) for a more conventional set of APIs.

**Architecture**

**NSURLSession**

* AFURLSessionManager
* AFHTTPSessionManager

**Serialization**

* <AFURLRequestSerialization>
  + AFHTTPRequestSerializer
  + AFJSONRequestSerializer
  + AFPropertyListRequestSerializer
* <AFURLResponseSerialization>
  + AFHTTPResponseSerializer
  + AFJSONResponseSerializer
  + AFXMLParserResponseSerializer
  + AFXMLDocumentResponseSerializer *(Mac OS X)*
  + AFPropertyListResponseSerializer
  + AFImageResponseSerializer
  + AFCompoundResponseSerializer

**Additional Functionality**

* AFSecurityPolicy
* AFNetworkReachabilityManager