**Steps to integrating SPM**

<https://www.raywenderlich.com/1993018-an-introduction-to-swift-package-manager>

Based on an article written by: By [Joannis Orlandos](https://www.raywenderlich.com/u/joannis)

**Overview of SPM**

Apple’s Swift Package Manager (SPM) lets you manage your project dependencies with ease, facilitating importing of libraries into your application.

**Packages**

Packages are simply repositories that contain one or more libraries and executables. ***Package.swift*** is a manifest file to manage your applications dependencies.

A target is the specification of a single module and its dependencies. You can compile each target that contains a main.swift to an executable. You might have seen modules in Xcode before. Your app itself is an executable module, and it has dependencies that, in turn, are also modules.

With SPM, you can create packages from Terminal. SPM uses the directory name as your project name when setting up your application.

**Steps**

Mkdir Website

Cd Websiteswift package init --type=executable

// swift-tools-version: 5.6

// The swift-tools-version declares the minimum version of Swift required to build this package.

import PackageDescription

let package = Package(

name: "Website",

dependencies: [

// Dependencies declare other packages that this package depends on.

// .package(url: /\* package url \*/, from: “1.0.0"),

**.package(url: "https://github.com/raywenderlich/spm-tutorial.git", .branch("finished"))**,

],

targets: [

// Targets are the basic building blocks of a package. A target can define a module or a test suite.

// Targets can depend on other targets in this package, and on products in packages this package depends on.

.executableTarget(

name: "Website",

dependencies: [**"WebsiteBuilder"**]),

.testTarget(

name: "WebsiteTests",

dependencies: ["Website"]),

]

)

**Editing Dependency Properties**

The following code specifies any version of the dependency, starting at 1.1.3 and less than 2.0.0:

.package(url: "https://github.com/...git", from: "1.1.3")

If you want specify a minimum and maximum version for one dependency, you can use a range:

.package(url: "https://github.com/...git", "1.1.0"..."1.2.1")

Use the following if you’re interested in an exact version of a dependency:

.package(url: "https://github.com/...git", .exact("1.2.3"))

All of these version-specific variations also support beta-versioning. For example:

.package(url: "https://github.com/...git", from: "1.1.3-beta.4")

You can also lock the dependency to a specific branch in git. This is useful if a feature or fix is not yet released:

.package(url: "https://github.com/...git", .branch("bugfix/issue-121"))

Finally, you can specify a commit by its hash:

.package(url: "https://github.com/...git",

.revision(“04136e97a73b826528dd077c3ebab07d9f8f48e2"))

# From terminal run

swift package generate-xcodeproj  
open ./Website.xcodeproj

"https://github.com/raywenderlich/spm-tutorial.git", .exact(“1.0.0"))

swift package update