Xcoders

Using Python In Swift

Xcoders October 2022

PythonKit

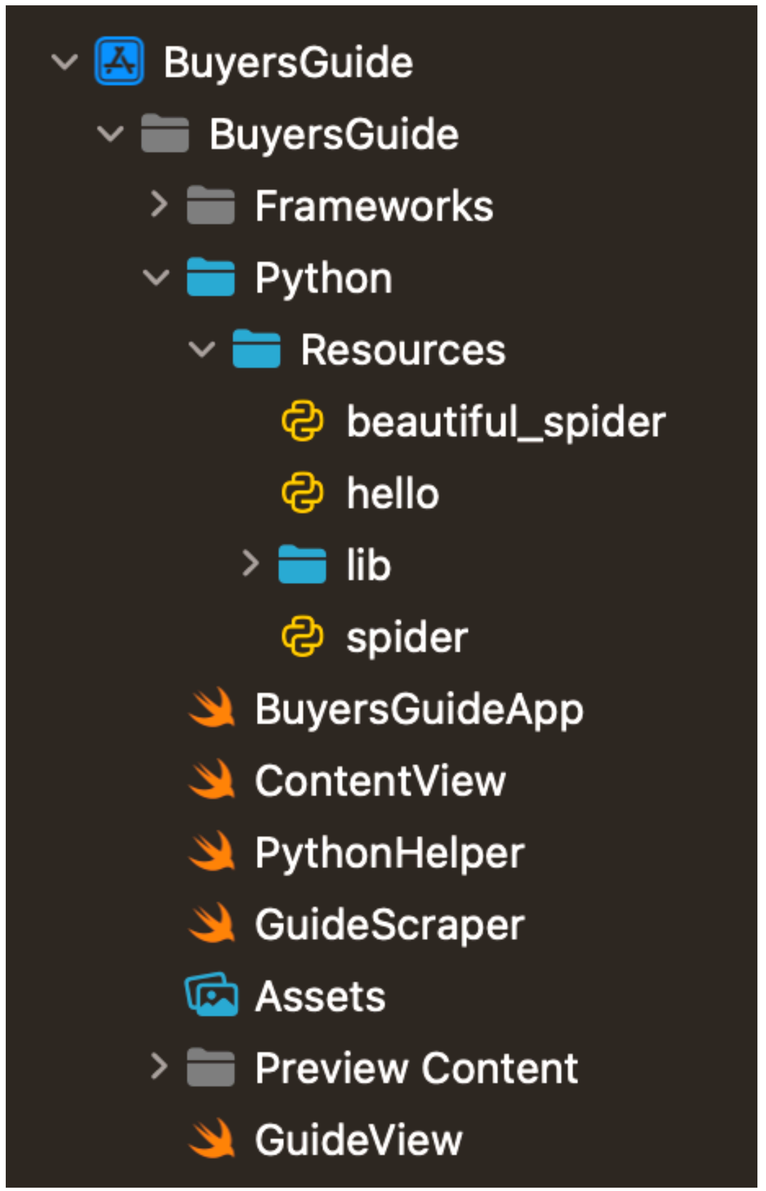
Integrating Python and Swift

- Library for calling Python code from Swift
  - Available via Swift Package Manager
- Originally based on code from the Swift for Tensorflow project by Google
- Forked and made easy to use by other Swift apps

Writing Python

Preparing Python for Swift

- Write Python code in whatever environment you prefer
  - Visual Studio Code, PyCharm, Emacs… whatever you like!
  - Xcode is an available, but sub-par option.
- Include scripts as a resource file
  - If you have many scripts, a folder resource works best



Calling Python

Using Python code from Swift

- Get a path to your scripts folder with path(forResource:ofType:)
- Set PYTHONPATH environment variable
- Use Python.import() to import your script
- Call functions in Python like any Swift function

```
enum PythonHelper {
  static func initialize() throws {
    guard let stdlibURL = Bundle.main.url(
      forResource: "Python/Resources",
      withExtension: "")
    else { throw PythonHelperError.cannotFindStandardLibrary }
    setenv("PYTHONHOME", stdlibURL.path, 1)
    setenv("PYTHONPATH", stdlibURL.path, 1)

    Py_Initialize()
  }
}

struct GuideScraper {
  static func fetchGuides() -> [Guide] {
    let spider = Python.import("beautiful_spider")
    return spider.parse_guides().compactMap(Guide.init(_:))
  }
}
```

Translating

Converting Python objects to Swift types

- Every returned object is PythonObject
- Some Swift types have built-in inits from PythonObject
  - String, Int, Dictionary, Array, etc.
- Getting elements from PythonObject uses subscripting

```
struct Guide: Identifiable {
  let name: String
  let section: String
  let releases: [Release]
  var id: String { name }

  init?(_ pythonObject: PythonObject) {
    guard let pythonName = String(pythonObject["name"])
    else { return nil }
    name = pythonName

    guard let pythonSection = String(pythonObject["section"])
    else { return nil }
    section = pythonSection

    guard let pythonReleases =
      [PythonObject](pythonObject["releases"])
    else { return nil }
    releases = pythonReleases.compactMap(Release.init(_:))
  }
}
```

Demo