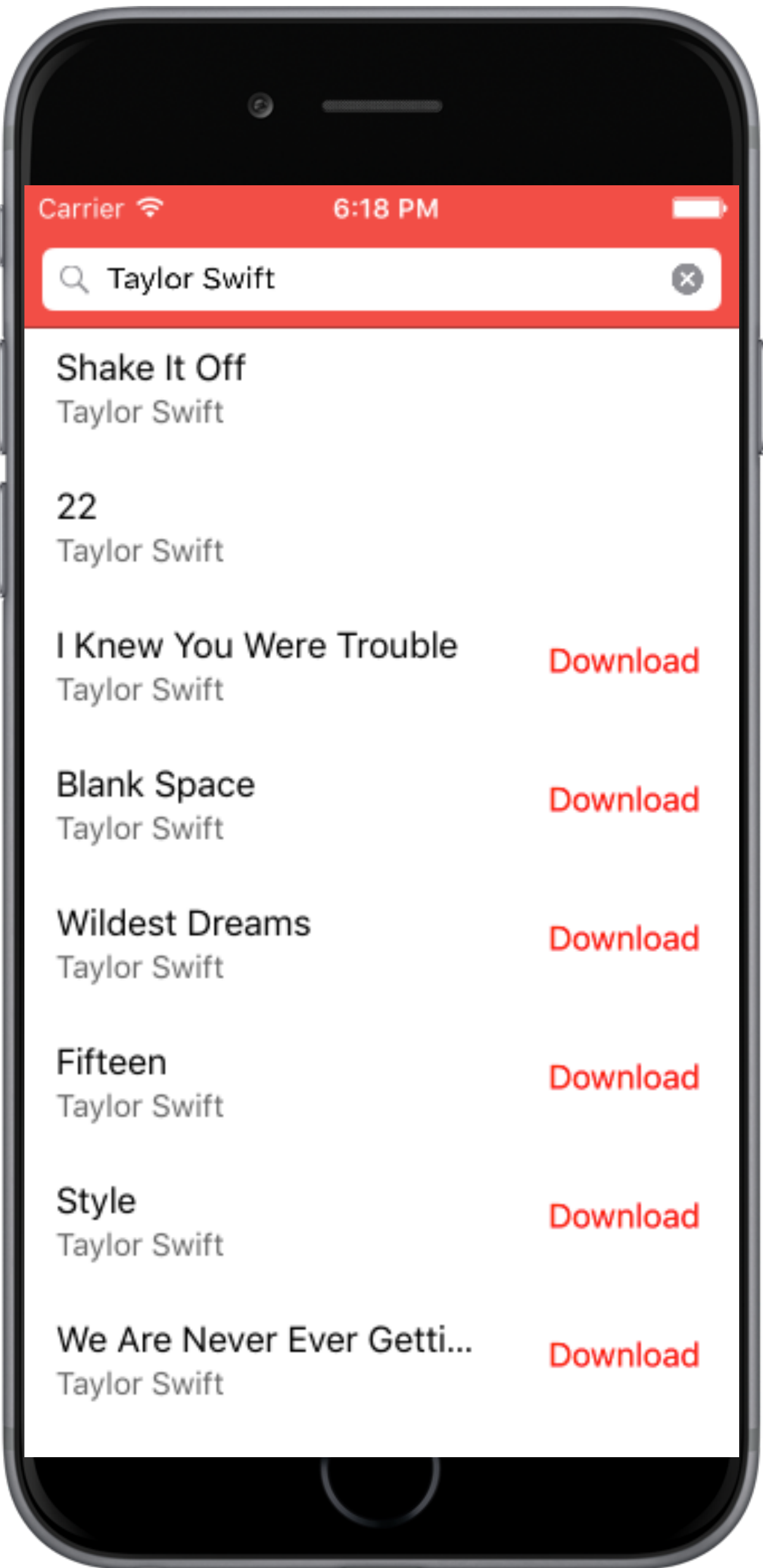


NETWORKING WITH URLSession



PART 6: DOWN/UPLOAD TASK



URLSESSIONDOWNLOADTASK

```
func downloadTask(with: URL,  
  completionHandler: @escaping  
    (URL?, URLResponse?, Error?) -> Void)  
func downloadTask(with: URLRequest,  
  completionHandler: @escaping  
    (URL?, URLResponse?, Error?) -> Void)
```

```
func downloadTask(with: URL)  
func downloadTask(with: URLRequest)
```

```
func cancel(byProducingResumeData:  
  @escaping (Data?) -> Void)
```

```
func downloadTask(withResumeData: Data,  
  completionHandler: @escaping (URL?,  
    URLResponse?, Error?) -> Void)  
func downloadTask(withResumeData: Data)
```

Server-side errors as NSError:

```
401 userAuthenticationRequired  
403 noPermissionsToReadFile  
407 userAuthenticationRequired  
Other FileDoesNotExist
```

URLSESSIONUPLOADTASK

```
func uploadTask(with: URLRequest, from: Data?,  
    completionHandler: @escaping (Data?, URLResponse?, Error?) -> Void)  
func uploadTask(with: URLRequest, from: Data)
```

```
func uploadTask(with: URLRequest, fromFile: URL,  
    completionHandler: @escaping (Data?, URLResponse?, Error?) -> Void)  
func uploadTask(with: URLRequest, fromFile: URL)
```

```
func uploadTask(withStreamedRequest: URLRequest)
```

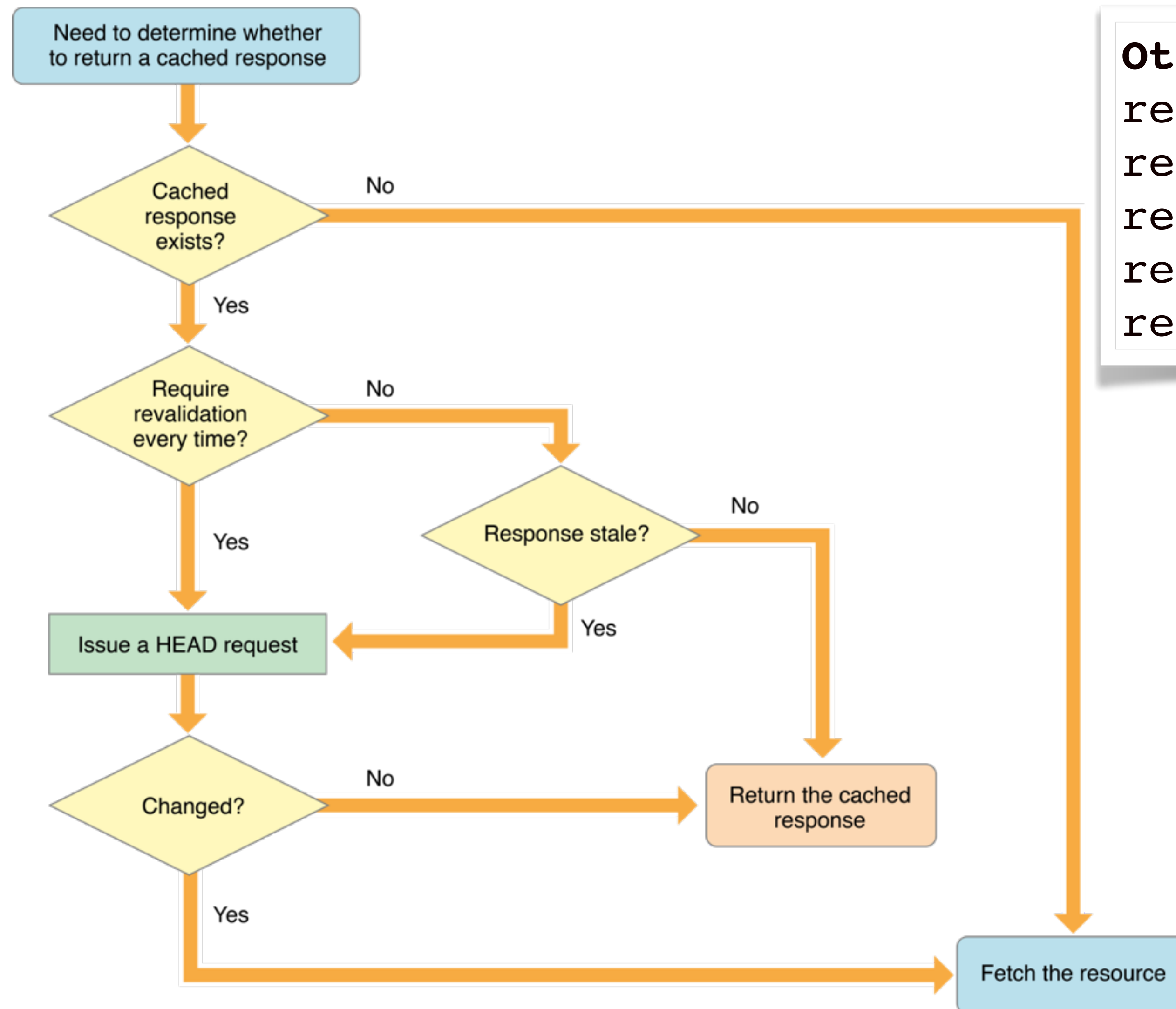


VARIOUS LEVELS OF PRIORITY HINTS

- ▶ URLSessionConfiguration
 - ▶ networkServiceType hint to OS: default, voip, video, background, voice
- ▶ URLSessionTask
 - ▶ priority hint to server: 0.0 (lowest) to 1.0 (highest), default is 0.5
- ▶ URLRequest
 - ▶ networkServiceType: overrides configuration's value
- ▶ URLSession's delegateQueue
 - ▶ qualityOfService: userInteractive, userInitiated, utility, background,



NSURLRequest.USEPROTOCOLCACHEPOLICY



Other cache policy values:

`reloadIgnoringLocalCacheData`
`reloadIgnoringLocalAndRemoteCacheData`
`reloadRevalidatingCacheData`
`returnCacheDataElseLoad`
`returnCacheDataDontLoad`

DEMO



CHALLENGE TIME!

```
let urlString = "http://localhost:3000/posts/1"
```

```
let dictionaryMod = ["author": "an old friend", "title": "an old title"]  
as? [JSONDictionary]
```

```
putRequest.httpBody = putData
```



```
public enum PostRouter {  
  
    // Possible requests  
    case getAll  
    case get(Int)  
    case create([String: Any])  
    case update(Int, [String: Any])  
    case delete(Int)  
    . . .  
}
```