*//*

*// NewsApi.swift*

*// iCatchUpF*

*//*

*// Created by Diego Salas Noain on 7/4/19.*

*// Copyright © 2019 final. All rights reserved.*

*//*

**import** Foundation

**import** Alamofire

**import** os

*// typealias UsersResponse = [Source]*

**class** NewsApi {

**static** **let** baseUrlString = "https://newsapi.org"

**static** **let** sourcesUrlString = "\(baseUrlString)/v2/sources"

**static** **let** topHeadlinesUrlString = "\(baseUrlString)/v2/top-headlines"

**static** **let** everythingUrlString = "\(baseUrlString)/v2/everything"

**static** **private** **func** get<T: Decodable>(

from urlString: String,

parameters: [String : String],

responseType: T.Type,

responseHandler: **@escaping** ((T) -> Void),

errorHandler: **@escaping** ((Error) -> Void)) {

**guard** **let** url = URL(string: urlString) **else** {

**let** message = "Error on URL"

os\_log("%@", message)

**return**

}

Alamofire.request(url, parameters: parameters).validate().responseJSON(

completionHandler: { response **in**

**switch** response.result {

**case** .success( **\_**):

**do** {

**let** decoder = JSONDecoder()

**if** **let** data = response.data {

**let** dataResponse = **try** decoder.decode(responseType, from: data)

responseHandler(dataResponse)

print("GOOD")

}

} **catch** {

print("errorcito")

errorHandler(error)

}

**case** .failure(**let** error):

errorHandler(error)

}

})

}

**static** **func** getSources(responseHandler: **@escaping** ((SourcesResponse) -> Void),

errorHandler: **@escaping** ((Error) -> Void)) {

**let** parameters = ["apiKey" : apiKey, "language" : "en"]

**self**.get(from: sourcesUrlString, parameters: parameters,

responseType: SourcesResponse.**self**,

responseHandler: responseHandler, errorHandler: errorHandler)

}

**static** **func** getTopHeadlines(responseHandler: **@escaping** ((ArticlesResponse) -> Void),

errorHandler: **@escaping** ((Error) -> Void)) {

**let** parameters = ["apiKey" : apiKey, "country" : "us"]

**self**.get(from: topHeadlinesUrlString, parameters: parameters,

responseType: ArticlesResponse.**self**,

responseHandler: responseHandler,

errorHandler: errorHandler)

}

**static** **func** getEverything(fromSources sources: String?,

responseHandler: **@escaping** ((ArticlesResponse) -> Void),

errorHandler: **@escaping** ((Error) -> Void)) {

**var** parameters = ["apiKey" : apiKey]

**if** **let** sources = sources {

parameters["sources"] = sources

}

**self**.get(from: everythingUrlString, parameters: parameters,

responseType: ArticlesResponse.**self**,

responseHandler: responseHandler, errorHandler: errorHandler)

}

**static** **var** apiKey: String {

**return** Bundle.main.object(forInfoDictionaryKey: "NewsApiKey") **as**! String

}

}