

Intercept HTTP(s) network requests in iOS

Edit

Manage topics

🔄 69 commits	🌿 2 branches	📦 0 packages	📦 2 releases	👤 1 contributor	📄 MIT
Branch: master ▾	New pull request	Create new file	Upload files	Find file	Clone or download ▾
🔄 yongpeng.zhu update: README Latest commit 7ae8bcd 2 days ago					
📁 Example	update: README	3 days ago			
📁 HttpInterceptor	update: import	4 days ago			
📄 .gitignore	add: gitignore	5 days ago			
📄 BestHttpInterceptor.podspec	update: podspec version 1.0.1	4 days ago			
📄 LICENSE	update	5 days ago			
📄 README.md	update: README	2 days ago			
📁 _Pods.xcodeproj	update	5 days ago			

📖 README.md

HttpInterceptor

build

success

pod

v1.0.1

swift

4.2

license

MIT

platform

ios

项目有在调用苹果私有api, 为了拦截WKWebView发出去的网络请求, 但是方法名已经使用Base64进行了编码, 为了防止苹果静态扫描和检查, 所以存在一定架风险, 请大家合理利用, 最安全还是在Debug下, 开发中使用。

The project is calling apple private api, in order to intercept the WKWebView sent to the network request, but the api signature has been encoded in Base64, in order to prevent apple static scanning and inspection, so there is a certain risk, please reasonable use, the most secure or in the Debug, the development of the use. 🙏🙏

如果HTTPInterceptor对你有帮助, 给楼主一个小星星🌟 是对楼主最好的支持和动力。
如果有疑问, 欢迎来提issue, 你们的提问是对楼主最好的反馈。

If it helps you, plesae give the project a little stars 🌟 is the best support for me. If you have any questions, please submit issue for me. Your issues are the best feedback for me.

About HTTPInterceptor

HTTPInterceptor 是一个iOS网络请求拦截器, 它可以拦截基于 URLSession 和 NSURLConnection 发出的HTTP(s)请求。对于已经使用过URLProtocol的项目来说, 如果 HTTPInterceptor 和其他 URLProtocol Handle的URL一样, 就要看注入的先后顺序。

1. 你可以使用它去拦截一个特定的request, 然后返回一个新的request, 意味着你可以改变它访问的目标地址和参数等等。
2. 你可以使用它去拦截一个特定的request, 然后返回一个新的 URLResponse 和 Data , 意味着你可以返回一个自定义 Response, 比如Mock数据等操作。
3. 你可以使用它去拦截一个特定的request, 然后返回一个 URLSessionTaskMetrics , 意味着你可以得到这个Request创建的请求Task的性能指标, 比如创建Task, DNSLookup, Establish链接, 各项指标, 用于数据上报等。

HTTPInterceptor is an iOS network request interceptor that intercepts HTTP(s) requests made by URLSession and NSURLConnection.

1. You can intercept a specific request and return a new one.
2. It can intercept a specific request and return a specific response and data.
3. It can also intercept a specific request and callback the URLSessionTaskMetrics object.

About WKWebView

HTTPInterceptor 虽然可以拦截WKWebView中的网络请求，但是却无法得到 POST 请求的 HTTPBody 。因为去拦截 WKWebView中的网络请求，HTTPInterceptor 使用了苹果私有API，这是不安全的，也是不应该的，但是如果你是 Debug 下使用它，这就另当别论了。

关于如何得到，WKWebView中 POST 请求的 HTTPBody ，我看了一些文章。例如: WKWebView注入JS脚本，JS脚本Hook所有的 XHR 和 Fetch 网络请求，然后再通过Native和Web通信通道，比如在WK容器中，JS使用 alert 方法， prompt 方法，都可以将JS侧的内容发送给Native，Native解析传来的内容，解析出类名，方法签名，参数等等，在Runtime下创建对象调用去保存，等等。

可以看到的是这样的逻辑实在有些麻烦，而且需要考虑的东西非常多，因为你是 Hook 了JS的请求入口，这是一个block操作，稍有不慎就会给H5带来线上灾难，而且全部是因为要拿一个 HTTPBody ，所以我认为这是非常不值得的，为了程序的鲁棒性也不应该使用这样的实现。

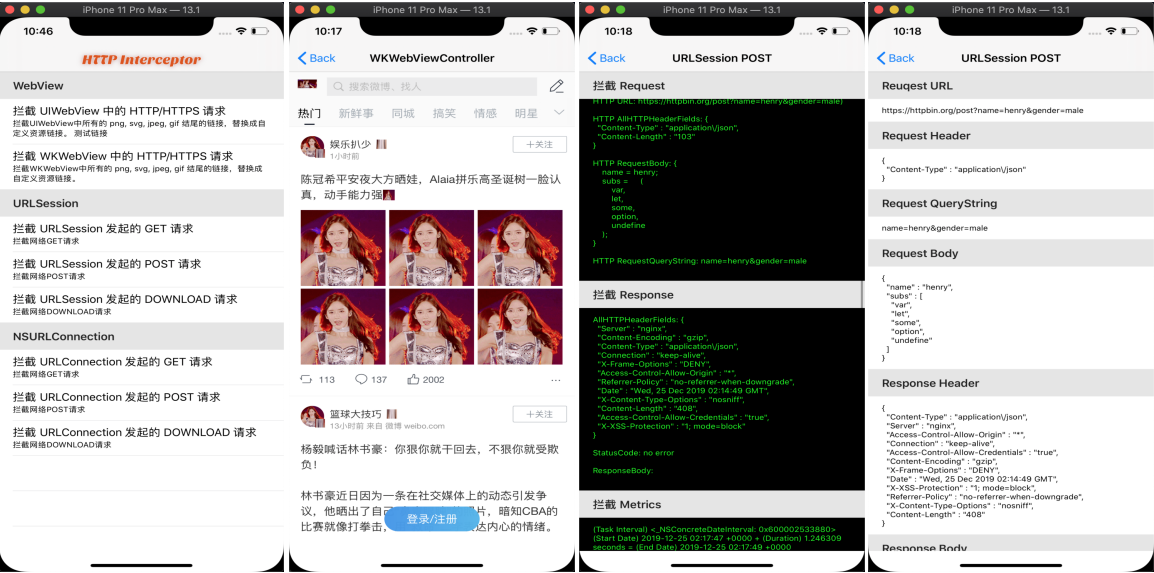
Fetching HTTPBody using HTTPInterceptor is not supported.

About Name

The HttpInterceptor name is taken by someone else, so use BestHttpInterceptor .

Example

To run the example project, clone the repo, and run pod install from the Example directory first.



1.注册拦截条件

```
import UIKit
import WebKit
import BestHttpInterceptor

class WKViewController: UIViewController {

    @IBOutlet weak var webView: WKWebView!

    var interceptor: HttpInterceptor? = nil

    enum PathExtension: String {
        case gif = "gif"
        case png = "png"
        case jpeg = "jpeg"
        case svg = "svg"
        case jpg = "jpg"
    }
```

```

}

override func viewDidLoad() {
    super.viewDidLoad()
    title = "WKWebViewController"

    registerInterceptor()

    let url = URL(string: "https://www.baidu.com/")
    webView.load(URLRequest(url: url!))
}

func registerInterceptor() {
    // 定义拦截条件, 根据URLRequest参数来进行判断, 是否需要拦截此Request, 返回Bool告诉HTTPInterceptor。
    let condition = HttpIntercepCondition(schemeType: .all) { (request) -> Bool in

        // 拦截逻辑, 这里是要拦截host带有ss和timgmb和结尾是 gif, jpeg, png, svg, jpg的Request。
        // 大家可以根据自己的需求来定义这个条件。
        guard let pathExtensionStr = request.url?.pathExtension,
              let host = request.url?.host else {
            return false
        }
        if host.contains("ss") || host.contains("timgmb") {
            return true
        }
        let pathExtension = WKViewController.PathExtension(rawValue: pathExtensionStr)
        switch pathExtension {
        case .gif, .jpeg, .png, .svg, .jpg:
            return true
        case .none:
            return false
        }

    }

    interceptor = HTTPInterceptor(condition: condition, delegate: self)

    // 开始注册
    interceptor?.register()
}

deinit {
    print("WKViewController deinit")
    // 取消注册
    interceptor?.unregister()
}
}

```

2.实现 HTTPInterceptorDelegate 代理

```

extension WKViewController: HTTPInterceptorDelegate {
    // Replace Request URL
    func httpRequest(request: URLRequest) -> URLRequest {
        // 将拦截到的Request, 换成我们新的Request。
        var newRequest = request
        newRequest.url = URL(string: "https://timgsa.baidu.com/timg?image&quality=80&size=b9999_10000&sec=")
        return newRequest
    }
}

```

Features

- ☒ WKWebView, UIWebView, URLSession, URLConnection, all network http(s) requests can be intercepted.
- ☒ Cannot intercept HTTPBody in WKWebView Post request.
- ☒ Custom filter URLRequest.
- ☒ Can intercept and replace URLRequest, URLResponse, Data.

Usage

1. Define intercept condition

```
let condition = HttpInterceptCondition(schemeType: .all) { (request) -> Bool in
    return true
}
```

2.New a interceptor instance

```
let interceptor = HttpInterceptor(condition: condition, delegate: self)
```

3.Start register

```
interceptor.register()
```

4.Cancel register

```
interceptor.unregister()
```

5.Implement HttpInterceptorDelegate delegate, callback to you

```
extension SomeClass: HttpInterceptDelegate {

    // Intercept http request then return a new request
    func httpRequest(request: URLRequest) -> URLRequest {
        return request
    }

    // Intercept http response then return a new response
    func httpRequest(response: URLResponse) -> URLResponse {
        return response
    }

    // Intercept http response data then return a new response data
    func httpRequest(request: URLRequest, data: Data) -> Data {
        return data
    }

    // When the request is complete call this api
    func httpRequest(request: URLRequest, didCompleteWithError error: Error?) {

    }

    func httpRequest(request: URLRequest, didFinishCollecting metrics: URLSessionTaskMetrics) {

    }
}
```

Requirements

iOS 10+

Installation

HttpInterceptor is available through [CocoaPods](#). To install it, simply add the following line to your Podfile:

```
pod 'BestHttpInterceptor'
```

if you want use it with Debug

```
pod 'BestHttpInterceptor', :configurations => ['Debug']
```

Author

zColdWater

License

HttpInterceptor is available under the MIT license. See the LICENSE file for more info.