

对AFNetworking 3.x 与YYCache的二次封装,一句代码搞定数据请求与缓存,告别FMDB!

📌 66 commits	🌿 1 branch	📦 3 releases	👤 1 contributor
Branch: master ▾	New pull request	Create new file	Upload files
		Find file	Clone or download ▾
👤 jkpang update read.md Latest commit ef323c6 2 days ago			
📁 PPNetworkHelper.xcodeproj	修复在无网络进行下载时,会触发下载成功回调的Bug.	2 days ago	
📁 PPNetworkHelper	修复在无网络进行下载时,会触发下载成功回调的Bug.	2 days ago	
📁 PPNetworkHelperTests	update	19 days ago	
📄 LICENSE	update	19 days ago	
📄 PPNetworkHelper.podspec	修复在无网络进行下载时,会触发下载成功回调的Bug.	2 days ago	
📄 README.md	update read.md	2 days ago	
📄 network.gif	Revert "优化文件结构"	12 days ago	

📖 README.md

PPNetworkHelper

对AFNetworking 3.x 与YYCache的二次封装,封装常见的GET、POST、文件上传/下载、网络状态监测的功能、方法接口简洁明了,并结合YYCache实现对网络数据的缓存,简单易用,不用再写FMDB那烦人的SQL语句,一句代码搞定网络数据的请求与缓存.

iPhone 6 - iPhone 6 / iOS 9.3 (13E230)

运营商 上午1:15

网络数据

```
{
  "subjects": [
    {
      "rating": {
        "stars": "50",
        "average": 9.6,
        "min": 0,
        "max": 10
      },
      "id": "1292052",
      "original_title": "The Shawshank Redemption",
      "collect count": 949058.
    }
  ]
}
```

缓存数据

缓存关闭 ☐

开始下载

Requirements 要求

- iOS7+
- Xcode 7+

Installation 安装

1.手动安装:

下载DEMO后,将子文件夹PPNetworkHelper拖入到项目中，导入头文件PPNetworkHelper.h开始使用

2.CocoaPods安装:

first pod 'PPNetworkHelper', '~> 0.1.2'

then pod install或pod install --no-repo-update

Usage 使用方法

1. 无自动缓存(GET与POST请求用法相同)

1.1 无缓存

```
[PPNetworkHelper GET:url parameters:nil success:^(id responseObject) {
    //请求成功
} failure:^(NSError *error) {
    //请求失败
}];
```

1.2 无缓存,手动缓存

```
[PPNetworkHelper GET:url parameters:nil success:^(id responseObject) {
    //请求成功
    //手动缓存
    [PPNetworkCache saveHttpCache:responseObject forKey:url];
} failure:^(NSError *error) {
    //请求失败
}];
```

2. 自动缓存(GET与POST请求用法相同)

```
[PPNetworkHelper GET:url parameters:nil responseCache:^(id responseCache) {
    //加载缓存数据
} success:^(id responseObject) {
    //请求成功
} failure:^(NSError *error) {
    //请求失败
}];
```

3.图片上传(也可以上传其他文件)

```
[PPNetworkHelper uploadWithURL:url
    parameters:@{@"参数":@"参数"}
    images:@[@"UIImage数组"]
    name:@"文件对应服务器上的字段"
    fileName:@"文件名称"
    mimeType:@"图片的类型,png,jpeg"
    progress:^(NSProgress *progress) {
        //上传进度,如果要配合UI进度条显示,必须在主线程更新UI
        NSLog(@"上传进度:%.2f%%",100.0 * progress.completedUnitCount/progress.totalUnitCount);
    } success:^(id responseObject) {
        //上传成功
    } failure:^(NSError *error) {
        //上传失败
    }];
```

4.文件下载

```
NSURLSessionTask *task = [PPNetworkHelper downloadWithURL:url fileDir:@"下载至沙盒中的制定文件夹(默认为Download)" pr
//下载进度,如果要配合UI进度条显示,必须在主线程更新UI
NSLog(@"下载进度:%.2f%",100.0 * progress.completedUnitCount/progress.totalUnitCount);
} success:^(NSString *filePath) {
//下载成功
} failure:(NSError *error) {
//下载失败
}];

//暂停下载,暂不支持断点下载
[task suspend];
//开始下载
[task resume];
```

5.网络状态监测

```
//开始监测网络状态,在判断网络状态之前调用,建议在APPDelegate.m中的didFinishLaunchingWithOptions方法中调用
[PPNetworkHelper startMonitoringNetwork];

//实时监测网络状态的变化,只要网络状态一改变,此block就会回调
[PPNetworkHelper checkNetworkStatusWithBlock:^(PPNetworkStatus status) {
switch (status) {
case PPNetworkStatusUnknown: //未知网络
break;
case PPNetworkStatusNotReachable: //无网络
break;
case PPNetworkStatusReachableViaWWAN://手机网络
break;
case PPNetworkStatusReachableViaWiFi://WIFI
break;
}
}];

//网络状态一次性判断,返回值YES:有网络,NO:没有网络
BOOL networkStatus = [PPNetworkHelper currentNetworkStatus];
```

6. 网络缓存

6.1 获取缓存总大小

```
NSInteger totalBytes = [PPNetworkCache getAllHttpCacheSize];
NSLog(@"网络缓存大小cache = %.2fMB",totalBytes/1024/1024.f);
```

6.2 删除所有缓存

```
[PPNetworkCache removeAllHttpCache];
```

以上就是对AFN3.x结合YYCache的简单封装,全部是类方法调用,使用简单,麻麻再也不用担心我一句一句地写SQLite啦~~~欢迎各路大神的批评指正以及建议.

你的star是我持续更新的动力!

CocoaPods更新日志

- 2016.09.06(tag:0.1.2)--修复在无网络进行下载时,会触发下载成功回调的Bug.
- 2016.09.05(tag:0.1.1)--多个请求的情况下采取一个共享的AFHTTPSessionManager;
- 2016.08.26(tag:0.1.0)--初始化到CocoaPods;

联系方式:

- Weibo : @CoderPang
- Email : jkpang@outlook.com
- QQ : 2406552315

许可证

PPNetworkHelper 使用 MIT 许可证，详情见 LICENSE 文件。

