

DBNetworking

简介

DBNetworking是基于AFNetworking开发的网络请求框架。DBNetworking集成依赖AFNetworking(3.1.0)、DBProgressHUD，提供HTTPS请求的相关方法，并可以自动监控网络状态的变化并弹出相应的提示。在网络请求时有提供默认的加载中提示。

DBNetworking中关于GET/POST网络请求的封装主要代码放在DBHTTPSSessionManager中。利用DBHTTPSSessionManager可以实现各种网络请求。但是，目前项目中的请求基本都是POST请求，所以在DBHTTPSSessionManager的基础上提供了 `DBPoster` 专门用来发送POST请求，DBPoster中目前提供了三种样式的POST请求。

关于网络状态改变的提示，目前默认在网络状态发生改变时进行提示。

将DBNetworking加入到项目中

建议使用CocoaPods对项目中的第三方框架进行管理。

- 在Podfile文件中添加 `pod 'DBNetworking'`；
- 使用 `pod install` 命令将DBNetworking加入到项目中；
- 在使用的时候将其引入或者在PCH文件中将其引入；

DBNetworking的使用

初始化

DBNetworking中的网络请求是HTTPS，需要配置相应的证书。在使用前需要将证书加入项目中。（如果改项目的证书暂时没有，也可以使用。）

在使用DBNetworking发送网络请求之前要对网络请求的管理者进行初始化操作，建议在AppDelegate中进行初始化。例如：

```
DBNetWorkingManager *manager=[DBNetWorkingManager sharedManager];
manager.db_BaseURLString=@"http://taofei.me";
manager.db_certificateString=@"donkey.dabay.cn";
```

DBNetWorkingManager使用sharedManager方法创建后作为单例存在，然后设置项目的基地址和HTTPS请求相关的证书名称，并且将证书加入到项目中。

DBHTTPSSessionManager

关于GET/POST网络请求的封装主要代码放在DBHTTPSSessionManager中；

发送POST请求，默认带有加载中提示的HUD

```

/**
    DBNetworking HTTPS请求 默认带有HUD

    @param urlString 网络请求的URL地址字符串
    @param parameters 网络请求的参数
    @param successBlock 网络请求成功的回调
    @param failedBlock 网络请求失败的回调
    */
+ (void)db_postWithURLString:(NSString *)URLString Parameters:(NSDictionary
*)parameters succeed:(SuccessBlock)successBlock failure:
(FailedBlock)failedBlock;

```

发送POST请求，加载中提示的HUD可以进行配置是否显示

```

/**
    DBNetworking HTTPS请求 默认为POST请求

    @param urlString 网络请求的URL地址字符串
    @param parameters 网络请求的参数
    @param isWithHUD 是否带有HUD提示
    @param view HUD显示在View上
    @param successBlock 网络请求成功的回调
    @param failedBlock 网络请求失败的回调
    */
+ (void)db_postRequestWithURLString:(NSString*)URLString Parameters:
(NSDictionary *)parameters isWithHUD:(BOOL)isWithHUD inView:(UIView *)view
succeed:(SuccessBlock)successBlock failure:(FailedBlock)failedBlock;

```

发送POST或者GET请求，加载中提示的HUD可以进行配置是否显示

```

/**
    DBNetworking HTTPS请求 可以选择请求方式：GET,POST

    @param urlString 网络请求的URL地址字符串
    @param method 网络请求的方式：GET/POST
    @param parameters 网络请求的参数
    @param isWithHUD 是否带有HUD提示
    @param view HUD显示在View上
    @param successBlock 网络请求成功的回调
    @param failedBlock 网络请求失败的回调
    */
+ (void)db_requestWithURLString:(NSString *)URLString httpsMethod:
(DB_HTTPSMETHOD)method parameters:(NSDictionary *)parameters isWithHUD:
(BOOL)isWithHUD inView:(UIView *)view succeed:(SuccessBlock)successBlock
failure:(FailedBlock)failedBlock;

```

DBPoster

DBPoster是专门发送POST请求的类，在项目中如果POST请求使用的频率比较高，使用DBPoster发送POST请求可以简化代码。DBPoster中提供了一些实用的方法。

网络请求默认带有HUD提示加载中

```
/**
    DBNetworking--发送POST请求--带有默认的HUD提示

    @param urlString 网络请求的URL地址字符串
    @param parameters 网络请求的参数
    @param successBlock 网络请求成功的回调
    @param failedBlock 网络请求失败的回调
    */
+ (void)db_urlString:(NSString *)urlString parameters:(NSDictionary
*)parameters succeed:(SuccessBlock)successBlock failure:
(FailedBlock)failedBlock;
```

网络请求的同时不显示HUD提示加载中

```
/**
    DBNetworking--发送POST请求--没有默认的HUD提示

    @param urlString 网络请求的URL地址字符串
    @param parameters 网络请求的参数
    @param successBlock 网络请求成功的回调
    @param failedBlock 网络请求失败的回调
    */
+ (void)db_withoutHUDWithURLString:(NSString *)urlString parameters:
(NSDictionary *)parameters succeed:(SuccessBlock)successBlock failure:
(FailedBlock)failedBlock;
```

网络请求的同时加载HUD到特定的View上

```
/**
    DBNetworking--发送POST请求--指定HUD显示在View上

    @param view HUD显示在View上
    @param urlString 网络请求的URL地址字符串
    @param parameters 网络请求的参数
    @param successBlock 网络请求成功的回调
    @param failedBlock 网络请求失败的回调
    */
```

DBPoster的使用

```
[DBPoster db_withHUDInView:self.view urlString:@"index/banner.json?"
parameters:nil succeed:^(NSDictionary *responseDict) {

    NSLog(@"请求成功");
} failure:^(NSError *error) {
    NSLog(@"请求失败");
}];
```