

YTKBaseRequest

start方法 - 即网络请求发起的方法

通知遵守了YTKRequestAccessory代理的对象  
request请求即将开始  
调用YTKNetworkAgent的单例对象中的  
addRequest方法

YTKRequest

主要负责缓存处理 —— 下面的步骤中可以体现出现

start方法

判断是否设置了忽略缓存

忽略缓存的话则会重新请求

对request开始网络请求

先将有关Cache的缓存清除掉，比如一些  
XML、JSON、缓存元数据，缓存数据等置为nil  
调用YTKBaseRequest的start方法，即开始请求

判断当前是否还有下载任务

有下载任务的话就会重新请求

上面两个均无的话，就尝试加载缓存

进入到缓存加载方法

判断缓存时间 —— 这个是要用户自己设置的，  
系统一般设置为-1

当用户自定义了缓存时间

判断能否加载缓存元数据

缓存元数据里主要存的是：1.缓存版本号 2.敏感  
信息 3.编码信息 4.创建时间 5.app版本号

当加载出了缓存元数据，那么就会通过缓存元数  
据来验证缓存是否还有效

根据缓存元数据存的信息进行验证

判断缓存时间是否已经超时

判断缓存元数据的版本号与当前版本号是否相符

判断编码格式

判断App版本号

当前面的判断都匹配了，那么就会开始加载缓存  
数据

根据指定路径加载缓存，并根据响应序列转化成  
对应的数据，比如JSON或者XML

到这一步，响应数据已经从缓存加载了，或者已  
经是请求完成了

判断数据是否要写进缓存

进行requestComplete的回调 —— 这个是用户自  
己实现的回调

通知YTKRequest的代理 请求完成了

回调successBlock

清理所有的Block，防止出现循环引用

加载缓存失败。重新请求

上面执行完之后，要么已经从缓存里取出数据  
了，要么已经重新请求了

判断是否将响应数据写进缓存

通知代理请求完成

执行请求完成Block

在工程文件中要对每一个网络请求实例化，类要  
继承于YTKRequest

addRequest方法

判断请请求序列化是否成功

对request的task进行优先级映射

将request保存到YTKNetworkAgent管理的的字典中，以request的datatask的id作为标识符

开启Task

先判断用户是否用自定义的请求url —— 直接返  
回了NSURLrequest对象，而不是重写url、参数  
等交给YTK去包装，获取请求对应的task赋值给  
request的requestTask属性

有的话，那么直接交给子属性AFN的管理者，让  
管理者去实例化request的dataTask

没有的话，则表示用户新建了一个类继承了  
YTKRequest，并且重写了请求的url、请求方  
法，以及请求序列和响应序列。那么YTK就要先  
将用户自定的url、参数等的包装起来，实例化  
request的dataTask

获取用户自定义的请求方法、请求参数、请求序  
列化方式以Block等，然后根据不同的请求方法  
开始包装sessionTask

最后其实还是交给了AFN的管理者去包装  
sessionTask，只不过YTKNetworkAgent先将用  
户自定义的数据收集起来。

在Agent管理的字典中，通过task的标识符取出  
对应的request

对请求的响应数据进行处理，根据返回的响应的  
序列化的类型来得到对应类型的响应

判断是否出现了错误

成功的话

请求成功的回调

失败的话

请求失败的回调

回调后在字典中移除对应的request，并清空一  
些临时数据，比如请求成功的Block，失败的  
Block的等