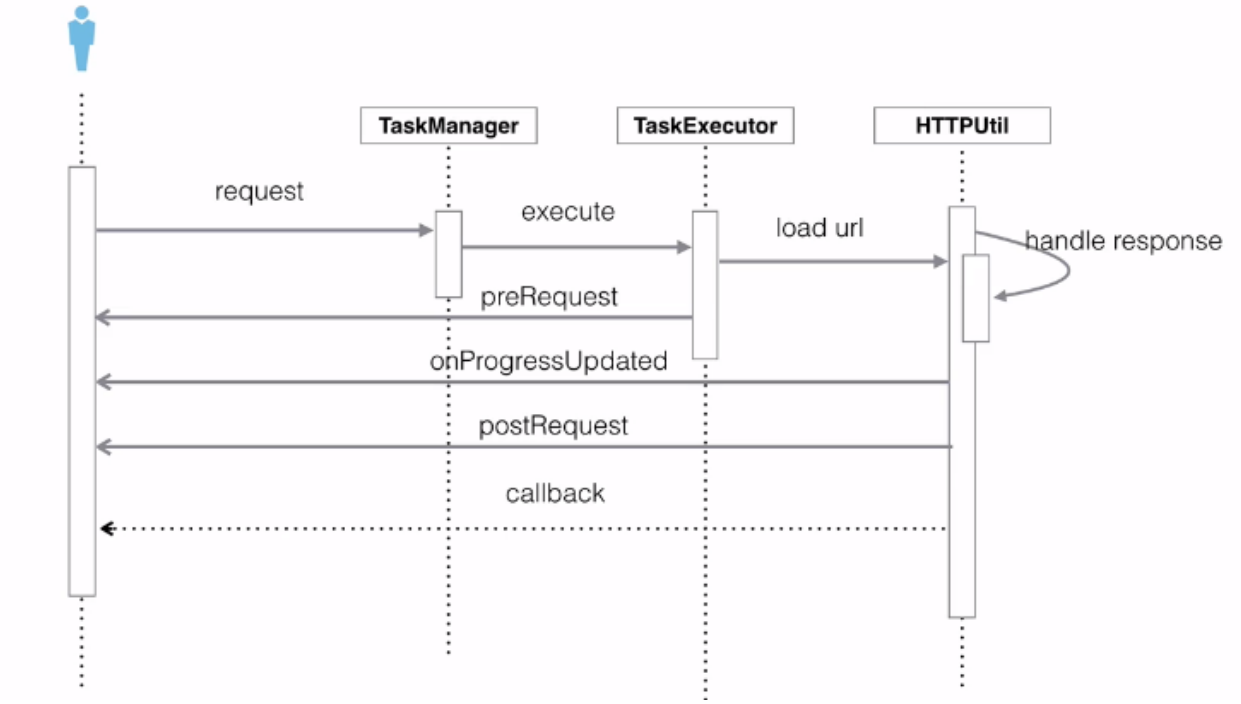
- B 站地址：https://www.bilibili.com/video/BV1Sv411v7kr

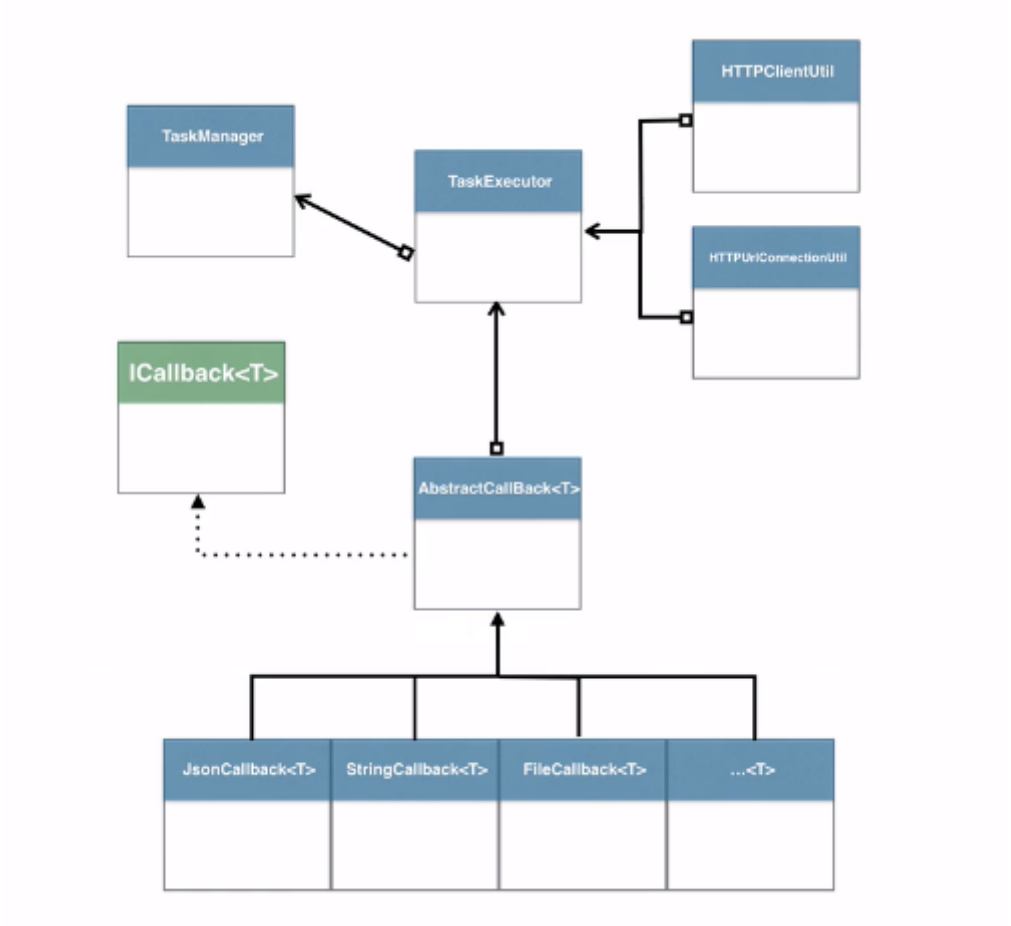
自己动手写 HTTP 框架

包含什么功能

- 支持 `GET/POST/PUT/DELETE`
- 可选 `HttpClient` 和 `HttpUrlConnection`
- 支持 gzip 与 deflate
- 请求错误统一处理
- 预处理服务器数据，直接返回对象
- 取消请求，支持 Activity 生命周期
- TIMEOUT 重试机制
- 多任务同步队列
- 多文件上传下载/进度更新
- 缓存刷新机制



- TaskManager 处理发送的请求，启动一个 TaskExecutor。
- 在执行请求前回调 preRequest 进行预处理。
- 接着如果需要发送请求就使用 HttpUtil 发送请求。
- 如果是上传或者下载，HttpUtil 会回调 onProgressUpdated 返回进度。
- postResult 对请求结果进行处理，比如可以存数据库。
- callback: 可以是请求成功，也可以是请求失败。



- TaskManager 管理所有请求。
- TaskExecutor 在异步线程执行请求，可以是 AsyncTask。 或者其他，并且可以切换 HttpClient 或者 HttpUrlConnection。
- AbstractCallBack<T> 请求结果回调，可扩展，可以是 json、String、File 或者其他。

开始动手

这里不会展示具体的代码，只包含封装库的过程，具体可以直接看 **HttpSample** 这个目录的提交记录

- 1.项目初始化,使用 `HttpUrlConnection` 进行网络请求，实现 `Get` 和 `Post` 的方法，并编写测试用例测试，API 为 **wanandroid** 的开放接口。
- 2.将 Get 和 Post 方法的参数封装成 Request Bean，方便后面扩展，同时将 get 和 post 方法设为 private，使用 execute 方法来调用，同时在 Reqeust 增加枚举类，用来设定请求的方法 `GET POST PUT DELETE`。
- 3.添加 `AsyncTask`，将网络请求放到子线程，同时增加回调接口，在网络请求完成时把成功或失败的接口返回给调用者。
- 4.添加 Json 解析器，在 Callback 中添加 parse 方法，反序列化 json 为实体对象。
- 5.添加一个 `XmlCallback` 用来处理 xml 解析，此时 JsonCallback 和 XmlCallback 中都有读取请求数据的公共逻辑，所以建立 AbstractCallback 把公共代码提取进去，创建模板方法 `bindData(String result)`，交给 `JsonCallback XmlCallback` 或者其他的 Callback 去实现。
- 6.添加 `FileCallback` 下载文件，注意文件保存的路径需要有权限才可以访问。

- 7.增加文件下载进度回调，注意进度回调需要在主线程，需要用到 `publishProgress`，将结果发送到 `AsyncTask` 的 `onProgressUpdate` 方法中，然后再通过 Request 的回调发送给 UI。
- 8.自定义 `AppException` 统一处理请求过程中的异常（ `statusCode` 和 `message` ）。创建 `OnGlobalExceptionListener` 统一处理 `AppException`,并将处理逻辑放在 `BaseActivity` 中。
- 9.添加重试机制，当出现 `InterruptedIOException` 的时候，抛出自定义异常，并定义超时类型，请求方法会根据此异常进行重试请求。
- 10.取消 `http` 请求，可以通过 `AsyncTask` 的 `cancel` 方法，但是这个方法是添加了一种标志，在 `doInBackground` 方法执行完后才会生效，所以我们手动给 `request` 添加一个标志位，在请求的各个阶段来检查标志位的值，从而决定是否取消请求并抛出异常信息。为了取消解析阶段的程序，我们同时也给 `AbstractCallback` 添加了 `isCancelled` 标志位用来判断是否中断程序。
- 11.创建 `RequestManager` 单例类统一缓存请求，执行和取消请求，当 `Activity` 的 `onStop` 方法调用时，可以取消请求。
- 12.增加 `postRequest` 回调方法，该方法运行在子线程，可以在此处更改反序列化后的数据，可以更改数据，也可以插入数据库。
- 13.增加 `preRequest` 方法，该方法运行在子线程，可以在发送 http 请求之前执行，比如判断数据库有数据就直接读取数据库，不从网络拉取。
- 14. `AsyncTask` 3.1 之后是一个同步队列，里面只有一个线程，如果我们要实现异步队列请求，需要自己创建线程池。如果创建了一个 5 个的线程池，假如正在执行的任务有 10 个，5 个正在执行， 5 个在等待，此时需要取消任务，那么根据前面的根据变量值在 `doInBackground` 执行过程中埋点取消的方法就不合适了，我们这里需要把 `RquestTask` 取消掉，因此在 `cancel` 方法中需要获得 `RequestTask` 对象，这里把 `ReqeustTask` 从 `RequestManager` 中移到 `Request` 中，取消任务的时候同时调用 `requestTask.cancel(true)`
- 15.由于 json 过大导致 OOM，使用 `JsonReader` 来解析 json。需要先将 json 下载到 sdcard 上，然后再从 Sdcard 上来解析 json。
- 16.文件上传，使用 `HttpUrlConnection` 和 `Socket` 上传文件，需完善。