

页面缓存技术方案

页面缓存可给用户带来极佳的使用体验，加载数据快，使其不再去等待网络请求。页面缓存特别适用于网络环境变差的情况。

一、页面缓存类型

1. 图片类型

每张图片对应一个URL地址，可以设置缓存时间。如果缓存图片未过期则只使用缓存图片，否则去请求网络，然后替换本地缓存图片供下次取用。

优点：减少重复下载图片而导致的流量浪费。

缺点：1) 缓存图片越积越多，需要手动清除；2) 不适用于图片替换而对应URL不变的情况；

iOS项目使用的图片缓存三方组件是SDWebImage，默认缓存时间是7天，可定制。

2. 内容基本不变的网络请求

如小区内的苑幢列表、房间号列表、关于我们、功能介绍等。这些页面数据基本上不会变化，请求一次就够了。可以适当把缓存时间延长，类似图片的缓存一样，每次只使用缓存。

优点：即使无网络，访问过的页面也可以再次查看；减少重复请求带来的流量消耗；

缺点：在缓存有效期内，内容不会及时更新。

3. 内容有变化的网络请求

如各模块首页、订单列表页等。在这些页面可以先加载本地缓存，再在请求成功以后默默刷新页面数据，可以给用户带来秒入的体验。

优点：即使无网络，访问过的页面也可以再次查看。

缺点：增加了一些内存占用。（但整个应用的网络请求接口数是固定的）

二、待确认

需要产品部决策的有几点：

1. 网络请求先加载缓存时的动画显示方式需不需要有变化；
2. 哪些页面不使用缓存、哪些页面只使用缓存、哪些页面先显示缓存再刷新数据；
3. 各页面数据的缓存有效时间需要确定。
4. 用户在“设置”页面删除缓存时，需不需要删除页面数据缓存。（注：当前删除了图片缓存）

三、实现方案

(iOS客户端)

1. 使用三方组件自带缓存

在iOS现有项目中使用的网络请求组件是YTKNetwork，其中的YTKRequest类支持对网络请求进行缓存，有固定的一套缓存方式。如果使用该缓存机制的话，当在缓存有效期内时只加载缓存；当缓存过期后，才会重新请求网络。现有条件下满足不了如第一节中描述的第三种页面缓存类型。所以需要YTKNetwork中的原代码进行修改。

缺点：需要对三方组件原码进行修改，不利于三方组件的版本管理等工作。

2.自定义一套缓存机制

在现有项目中使用的网络请求都对YTKRequest进行了继承，可在此基础上对返回后的数据进行缓存处理。借鉴YTKRequest的缓存机制，对其进行修改，以满足自身的需求。工程Demo以幸福绿漫为蓝本，工程地址：<http://git.uama.com.cn:8888/Research/LvManPreResearchDemo.git>，打包后的ipa地址：<svn://121.40.102.80/sc-common/技术分享/iOS技术>，可直接用幸福绿漫的账号进行登录。技术预研页面地址：我的->技术预研。

优点：对三方组件无影响。自定义性强，可自由添加、修改满足需要。

四、建议

建议自定义一套缓存机制。

五、注意事项

缓存有风险！使用需谨慎。

可能会发生的情况有：

- 1.后台返回数据导致崩溃。但此时数据已经被缓存到客户端，即使后台把返回数据更正，由于客户端是先加载缓存的，所以即使重启也会一直崩溃。
- 2.客户端升级导致的崩溃。比如同样的数据在老版本可以正常解析，但升级到新版本之后，客户端的处理方式发生变化，导致崩溃。

可避免的方式：每次启动APP，立即检测是否是正常启动，如果不是正常启动（例如上次崩溃），可以先把APP内的所有缓存予以清除。