**Pull Bears技术文档**

－**关于客户端（iOS）网页缓存建议**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 更新日期 | 更新版本 | 更新人 | 更新内容 | 审批人 |
| 2015/7/22 | 1.0.0 | 胡家驹 | 新建 |  |
|  |  |  |  |  |

# 文档说明

## 1．文档编写目的：

本文档编写的目的是为解决客户端（iOS）网页html缓存遇到的问题及解决方案而编写。

## 2．文档适用范围：

本文档描述的内容适用于开发人员（iOS），也部分适用于本项目－《共时财经》中的技术人员、设计人员。

## 3．文档实现目的作用：

为解决客户端网页缓存提供几点建议。

# 文档内容

## 概述：

目前客户端（共时财经1.0版本）

1，第一级页面（专题、热点文章、P2P…）等采取了从服务端pull数据缓存到本地轻量级嵌入式数据库sqlite中，以便在没有网络的情况下用户可以访问第一级页面（前提用户在有网络的情况预览过第一级），

2，第二级页面（文章详情页面）采取了webview＋tableview的布局实现，在数据层没有采取缓存，基于此分析：

1）优点：

a，后端编辑页面简单，只需编辑当前文章页面，相关阅读部分无需维护。

b，前端布局很清晰，采取表格布局，webview作为表格的头部首先加载，加载完毕后拉取相关阅读数据作为表格内容显示；如无网络则显示hud提示错误页。

2）缺点：

a，前端缓存在系统兼容上有问题，iOS8以下系统提供组件（UIWebView）访问html页面并可执行javascript语句进行简单交互，组件调用loadRequest方法后系统会自动截获URLCache并进行缓存（注意是缓存到内存中，可以通过继承父类实现缓存到disk并已实现），iOS8及以上则系统不再截获该方法（强制调用也无效），参考文档：<http://blog.airsource.co.uk/2014/10/11/nsurlcache-ios8-broken/>

3，解决方案：

a）归档页面：在用户离开文章阅读页面时或加载完毕后，直接归档（NSArchiver）页面，以处理后的url为key，页面需实现Coding，Coping协议，具体参看apple开发文档。

优点：直接缓存页面，无需考虑html和相关阅读细节缓存。无需考虑系统兼容（目前）。不依赖系统缓存策略。

缺点：将大量缓存一些页面的相关的信息，磁盘缓存将爆炸式增长，并且不利于扩展。

综述：不易采取。

b）缓存html内容body ：在用户离开文章阅读页面时或加载完毕后，缓存组件的body内容和相关阅读内容到数据库。

优点：分开缓存，无需考虑系统兼容（目前）。不依赖系统缓存策略。

缺点：页面相关CSS需要特别处理，缓存类型不利于扩展。

综述：不易采取。

c）后台数据＋前端html模版：调整页面结构布局，采取网易新闻文章阅读详细页面架构，，参考：<http://www.cnblogs.com/xukunhenchouchang/p/4543322.html>。

优点：后端需编辑N个页面模版，后续专注编辑页面内容。前端CSS样式等与内容无关项直接读取本地模版。前端直接读取接口json数据获取到html内容数据然后提供WebView组件访问即可无需tableview布局作为折中。无需考虑系统兼容（目前）。不依赖系统缓存策略。

缺点：此布局架构与现行不一致，后台与前端均需要不同程度的调整（调整量需要具体评估），需要一定量的前期开发测试。

综述：推荐采用，利于扩展。